

# **AI and IoT Meet Mobile Machines: Towards a Smart Working Site**

Zur Erlangung des akademischen Grades eines

**DOKTORS DER INGENIEURWISSENSCHAFTEN  
(Dr.-Ing.)**

von der KIT-Fakultät für Maschinenbau des  
Karlsruher Instituts für Technologie (KIT)

angenommene

**DISSERTATION**

von

**M.Sc. Yusheng Xiang**

aus Fujian, China

Tag der mündlichen Prüfung:

Hauptreferent:

Korreferent:

22.09.2021

Prof. Dr.-Ing. Marcus Geimer

Prof. Dr.-Ing. Tamim Asfour



# Abstract

Infrastructure construction is society's cornerstone and economics' catalyst. Therefore, improving mobile machinery's efficiency and reducing their cost of use have enormous economic benefits in the vast and growing construction market. For this purpose, many methods have been proposed by industry and academia during the past few decades which contribute to even better products. As research in this area becomes more mature, significant optimization of single construction machine is less likely to exist. Therefore, instead of focusing on improving the performance of single construction machinery, I considered a group of construction machinery as a whole system to improve the productivity of the working site. In this thesis, I envision a novel concept *smart working site* to increase productivity through fleet management from multiple aspects and with Artificial Intelligence (AI) and Internet of Things (IoT).

Investigating the famous construction site for the hospital, namely Huoshenshan, where the project was finished at an unprecedented speed in Wuhan during the coronavirus outbreak in 2020, the most impressive distinguishing features can be concluded as a large amount of machines investment and the well-ordered coordination. Inspired by this particular working site, this thesis aims to present the approaches to substitute some human coordinators using AI and IoT and thus make the concept of a smart working site offering high productivity closer to reality.

Firstly, I introduced a novel multi working-machines pathfinding algorithm to solve the path conflicts among machines and prioritize the more critical machines. The proposed algorithm outperforms the State-of-the-Art (SOTA) solution in pathfinding time, whereas it achieves optimal solution. To navigate the optimal path from

the start point to the destination, an accurate localization, and mapping algorithm is indispensable. Therefore, a multi GPS/IMUs Simultaneous Localization And Mapping (SLAM) system based on commodity sensors on account of dynamically changing of the working site and noise of sensors was developed. This SLAM system offers the location and terrain information to supporting the successful path planning. Since some difficult tasks may still be finished by human drivers in the next decade, I endow my AI system the capability to predict the motion of manned machines. Concretely, I introduced combined neural networks to detect the manned machines' working process and validated with experimental data did on a wheel loader. Because the selected combined neural network is more suitable for transfer learning compared to the SOTA solution of the Multivariable Time Series Classification algorithm, my deep learning model has better generalization capability on different working sites and is more robust against the diversity of construction machines. Then, I created a visual monitoring system for the safety of participants without localization equipment. Given that the machines in a closed site can be treated as an L4 automation driving task, I built a mobile machines dataset to be used as a base dataset to train the SOTA deep-learning-based visual algorithm. By taking full advantage of L4 features, I proved my approach is extremely effective. To share all the information mentioned above between the command center and construction machines, I evaluated two major wireless communication systems for working sites, i.e., WLAN-based IEEE 802.11p and cellular network 5G, to achieve the seamless share of the large volume of information. The research about 5G indicates the working site setup, and the research about ad-hoc networks presents the handover strategy.

This thesis contributes to the path making Wuhan's speed a normal speed in the future, by quantitatively evaluating feasibility considering cutting-edge AI and IoT technologies.

**Keywords:** Smart Working Site, Multi Working Machine Pathfinding Algorithm, Multivariable Time Series Classification Algorithm, Mobile Machines Dataset, SLAM, 5G, IEEE 802.11p

# Kurzfassung

Der Bau von Infrastrukturen ist ein Eckpfeiler der Gesellschaft und ein Katalysator der Wirtschaft. Daher haben die Verbesserung der Effizienz mobiler Maschinen und die Senkung ihrer Nutzungskosten enorme wirtschaftliche Vorteile auf dem riesigen und wachsenden Baumarkt. Zu diesem Zweck wurden in den letzten Jahrzehnten viele Methoden von Industrie und Wissenschaft vorgeschlagen, die zu noch besseren und leistungsfähigeren Produkten beitragen. Da die Forschung in diesem Bereich reifer wird, ist es weniger wahrscheinlich, dass eine signifikante Optimierung einer einzelnen Baumaschine vorliegt. Anstatt mich auf die Verbesserung der Leistung einzelner Baumaschinen zu konzentrieren, habe ich eine Gruppe von Baumaschinen als Gesamtsystem betrachtet, um die Produktivität der Baustelle zu verbessern. In dieser Arbeit stelle ich ein neuartiges Konzept *Smart Working Site* vor, um die Produktivität durch Flottenmanagement unter verschiedenen Gesichtspunkten und mit künstlicher Intelligenz (KI) und Internet der Dinge (IoT) zu steigern.

Bei der Untersuchung der berühmten Baustelle des Krankenhauses, in Wuhan nämlich Huoshenshan, auf der die Baustelle während des Ausbruchs des Coronavirus im Jahr 2020 mit beispielloser Geschwindigkeit abgeschlossen wurde, ist uns die große Menge an Maschineninvestitionen und die geordnete Koordination aufgefallen. Inspiriert von dieser speziellen Arbeitsstelle, ist das Ziel dieser Arbeit, die Ansätze vorzustellen, eine "intelligenten Baustelle" um einige menschliche Mitarbeiter durch AI und das IoT zu ersetzen und die Produktivität zu erhöhen, der Realität näher zu bringen.

Zunächst habe ich einen neuartigen Algorithmus für die Routenplanung mehrerer Arbeitsmaschinen eingeführt, um die Kollision des Pfads zwischen Maschinen zu lösen und die kritischeren Maschinen zu priorisieren. Der vorgeschlagene Algorithmus übertrifft die SOTA-Lösung (State-of-the-Art) in der Rechenzeit, während eine optimale Lösung erzielt wird. Um den optimalen Weg vom Startpunkt zum Ziel zu navigieren, ist ein genauer Lokalisierungs- und Zuordnungsalgorithmus unerlässlich. Aus diesem Grund habe ich ein SLAM-System (Simultaneous Localization And Mapping), das auf Warensensoren basiert, da sich die Arbeitsstelle dynamisch ändert und das Rauschen der Sensoren auftritt. Dieses SLAM-System bietet Lokalisierungs- und Geländeinformationen zur Unterstützung der erfolgreichen Routenplanung. Da einige schwierige Aufgaben in den nächsten zehn Jahren möglicherweise noch von Menschen erledigt werden, kann ich auch auf der intelligenten Baustelle nicht auf diese verzichten. Konkret habe ich kombinierte neuronale Netze eingeführt, um den Arbeitsprozess der bemannten Maschinen zu erfassen und mit experimentellen Daten validiert, die mit einem Radlader erstellt wurden. Da das ausgewählte kombinierte neuronale Netzwerk im Vergleich mit SOTA-Lösung zur Klassifizierung multivariabler Zeitreihen besser für das Transferlernen geeignet ist, verfügt unser Deep-Learning-Modell über eine bessere Generalisierungsfähigkeit auf verschiedenen Arbeitsplätzen und ist robuster gegenüber der Vielfalt von Baumaschinen. Anschließend habe ich ein visuelles Überwachungssystem für die Sicherheit der Teilnehmer ohne Lokalisierungs-ausrüstung erstellt. Da die Maschinen an einem geschlossenen Standort als L4 des automatisiertes Fahren behandelt werden können, habe ich einen Datensatz für mobile Maschinen erstellt, der als Basisdatensatz zum Trainieren des auf SOTA Deep Learning basierenden visuellen Algorithmus verwendet werden kann. Durch die volle Nutzung der L4-Funktionen habe ich bewiesen, dass unser Ansatz äußerst effektiv ist. Um alle oben genannten Informationen zwischen der Kommandozentrale und den Baumaschinen auszutauschen, habe ich zwei wichtige drahtlose Kommunikationssysteme für Arbeitsstätten evaluiert, nämlich WLAN-basiertes IEEE 802.11p und Mobilfunknetz 5G, um den nahtlosen Austausch des großen Informationsvolumens zu erreichen. Die Forschung zu 5G zeigt die Einrichtung der Arbeitsstelle an, und die Forschung zu Ad-hoc-Netzwerken präsentiert die Übergabestrategie.

Diese Arbeit trägt dazu bei, dass durch die quantitative Bewertung der Machbarkeit unter Berücksichtigung modernster AIoT-Technologien die Geschwindigkeit der Bauprojekte in Wuhan in Zukunft zu einer normalen Vorgehensweise bringen zu können.

**Schlüsselwörter:** Smart Working Site, Multi Working Machine Pathfinding Algorithm, Multivariable Time Series Classification Algorithm, Mobile Machines Dataset, SLAM, 5G, IEEE 802.11p





# Preface

The dissertation includes some parts of my research during the last three years as a PhD student, research scientist, and visiting scholar at Karlsruhe Institute of Technology, Robert Bosch LLC, and University of California, Berkeley. Here I would like to sincerely thank some people who did contribute to my dissertation and my growth.

First of all, I am very grateful for the guidance from **Prof. Dr. Marcus Geimer** at KIT. During the time when we taught lectures at the university, Prof. Geimer shows me what means responsible and rigorous with action. Also, I appreciate the time we generated novel ideas and wrote academic papers together. It is my pleasure to cooperate with such a great educator. His virtues will affect me for my whole life.

Also, I would like to thank **Dr. Steffen Mutschler** at Bosch Rexroth, who trusts and offers me this opportunity to do my research at a prestigious university and company. During the early phase of my research, he guided me thoroughly and patiently to become familiar with the projects quickly. Without a doubt, he did an outstanding contribution to my personal and professional growth in my PhD early stage. All my “achievements” during my PhD time cannot be accomplished without him.

Another important person is **Dr. Christine Brach** at Bosch Rexroth. After Dr. Steffen Mutschler left our department, she took the responsibility to guide me and spent plenty of time on me, and even she must be extremely busy as a department leader. Her support much accelerates my projects and research. Also, her tolerance for my unconventional viewpoints guarantees the innovation of my

research. Her openness, wisdom, and encouragement broadened my horizon and let me have the freedom to consider problems at a higher level.

**Prof. Dr. Samuel S. Mao** at UC, Berkeley makes me realize the difference between tiny innovation and disruptive innovation. Inspired by the conversations with him, I have a more in-depth insight into Silicon Valley's charm and comprehensively understand the idea of making human life better. As an outstanding professor with more than 46,000 citations and a pioneer to transfer the research results into the industry through high-tech startups, he guides me to continue to break through my limits. An outstanding research in engineering convinces both professors and venture capital partners.

I sincerely thank **Dipl.-Ing. Norman Brix** for his valuable expertise and impressive know-how in both software and hardware. What's more, he is my best German language teacher.

In addition, my special thank gives to the company, **Robert Bosch LLC**, where I gained knowledge and happiness at the same time in the past four years, started from internship. Besides these thanks, I am grateful for **my graduate- and undergraduate students** who supported me with their great enthusiasm and aptitude, and **Karlsruhe House of Young Scientist** for providing me the scholarship to visit abroad.

Finally, I would like to thank my partner, **M.Sc. Tianqing Su**. Her excellent programming skill and in-depth knowledge in the field of computer science help me overcome many technical difficulties. Not to mention, she always encouraged and accompanied me during the time before deadlines and appointments, making these hard times colorful and beautiful. Moreover, I appreciate her patience to listen the content of my "boring" dissertation almost every day.

*Thank you!*

*Yusheng Xiang on Mar. 01, 2021*

# Contents

<b>Abstract</b> . . . . .	<b>i</b>
<b>Kurzfassung</b> . . . . .	<b>iii</b>
<b>Preface</b> . . . . .	<b>vii</b>
<b>Acronyms and symbols</b> . . . . .	<b>xv</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Problem and Goal Statement . . . . .	2
1.2 Applications . . . . .	3
1.3 Contributions . . . . .	4
1.4 Thesis Outline . . . . .	5
<b>2 Background Knowledge of Smart Working Site</b> . . . . .	<b>7</b>
2.1 Concepts and Consensus . . . . .	7
2.2 Current Applications and Challenges . . . . .	10
<b>3 Path Planning for Machines Fleet Management</b> . . . . .	<b>15</b>
3.1 Introduction . . . . .	16
3.2 Related Works . . . . .	19
3.2.1 Building Information Modelling (BIM) . . . . .	19
3.2.2 Path Planning for Construction Machines . . . . .	20
3.2.3 MAPF for Mobile Robotics . . . . .	22
3.3 Problem Statement . . . . .	25
3.4 Model Building . . . . .	25
3.4.1 Multi-Layer Grid Map . . . . .	27

3.4.2	Lower Level Search . . . . .	29
3.4.3	Higher Level Search . . . . .	34
3.5	Experiment on Real Working Sites . . . . .	37
3.6	Experimental Results . . . . .	39
3.7	Advantages of My Methods . . . . .	47
3.8	Conclusion . . . . .	48
<b>4</b>	<b>SLAM for Machines on a Smart Working Site . . . . .</b>	<b>49</b>
4.1	Introduction . . . . .	50
4.2	Problem Statement . . . . .	51
4.3	Goal of This Chapter . . . . .	51
4.4	Related Works . . . . .	52
4.4.1	Sensors . . . . .	52
4.4.2	Localization Technologies . . . . .	54
4.5	Model Building . . . . .	56
4.5.1	Sensor Fusion for Localization . . . . .	60
4.5.2	Sensor Fusion Methods . . . . .	61
4.5.3	Realtime Map Plotter . . . . .	65
4.6	Vehicle Simulation Scenarios in ROS and Gazebo . . . . .	67
4.7	Experiment and Results . . . . .	70
4.7.1	Localization Results . . . . .	70
4.7.2	Plotter results . . . . .	75
4.8	Conclusion . . . . .	78
<b>5</b>	<b>Motion Prediction of Manned Working Machines . . . . .</b>	<b>79</b>
5.1	Introduction . . . . .	80
5.2	Background . . . . .	80
5.2.1	Wheel Loader . . . . .	80
5.2.2	The Future Mobile Machines Drivetrain System . . . . .	81
5.2.3	Working Process Detection Algorithms . . . . .	83
5.3	Problem Statement . . . . .	83
5.4	Why I Use RNN, LSTM? . . . . .	84
5.5	Data for Deep Learning Algorithm . . . . .	85
5.5.1	Data Acquisition and Allocation . . . . .	86
5.5.2	Data Preparation . . . . .	88

---

5.6	Combined Neural Networks . . . . .	89
5.7	Evaluation of the Methods . . . . .	95
5.8	Fast CRDNN . . . . .	97
5.8.1	What is CRDNN? . . . . .	98
5.8.2	Motivation of Fast CRDNN . . . . .	98
5.9	Long Short Term Memory Fully Convolutional Network: a SOTA Solution for TSC Tasks . . . . .	101
5.10	Wireless Human Machine Communication . . . . .	102
5.11	Problem Statement and Brief Description of the Solution . . . . .	103
5.12	Why Transfer-Learning Based Supervised Learning? . . . . .	105
5.13	Connection System Design . . . . .	108
5.13.1	Choice of Wireless Communication Technology . . . . .	108
5.13.2	User Interface of the System . . . . .	110
5.14	Measurement Setup . . . . .	111
5.14.1	The Sliding Windows Labeling Method . . . . .	113
5.15	Comparison Between CRDNN and Other SOTA Time Series Classification Neural Networks . . . . .	115
5.16	Transfer Learning Based CRDNNs . . . . .	117
5.16.1	Training from the Scratch as Benchmark (ND+PD+FS) . . . . .	118
5.16.2	Only Further Train the FCN (ND+FTF) . . . . .	119
5.16.3	Train the Total Part of CRDNN (ND + OTF) . . . . .	120
5.16.4	Evaluation the Benefits of Transfer Learning . . . . .	121
5.17	The Advantages of This System from Engineers' View . . . . .	124
5.17.1	Strong . . . . .	124
5.17.2	Fast . . . . .	124
5.17.3	Easy . . . . .	125
5.18	Conclusion . . . . .	125
<b>6</b>	<b>Visual Monitoring of Working Site . . . . .</b>	<b>127</b>
6.1	Introduction . . . . .	128
6.2	Related Works . . . . .	131
6.2.1	The Well-Known Datasets . . . . .	131
6.2.2	Recent Object Detection Algorithms . . . . .	132
6.2.3	The Previous Contributions on Detecting Mobile Machines	133
6.3	Why I Created the MOMA Dataset? . . . . .	135

6.4	The MOMA Dataset . . . . .	135
6.4.1	Data Acquisition . . . . .	138
6.4.2	Dataset Format . . . . .	140
6.4.3	Manual Annotation . . . . .	141
6.4.4	Dataset Splits . . . . .	146
6.4.5	Data Preprocess . . . . .	146
6.5	Evaluation of the Recent Computer Vision Algorithm Performance on the MOMA Dataset . . . . .	148
6.6	Conclusion . . . . .	158
<b>7</b>	<b>Wireless Communication System . . . . .</b>	<b>159</b>
7.1	Introduction . . . . .	160
7.2	Current Wireless Communication for V2X . . . . .	161
7.2.1	Ad-hoc Networks . . . . .	161
7.2.2	Cellular Networks . . . . .	162
7.3	IEEE 802.11p . . . . .	166
7.3.1	Why I Use the IEEE 802.11p? . . . . .	166
7.3.2	Modelling . . . . .	167
7.3.3	Propagation Model . . . . .	167
7.3.4	CAM' Generation Model . . . . .	168
7.3.5	CSMA/CA and Enhanced DCF Channel Access (EDCA) . . . . .	168
7.3.6	Evaluation of Hidden Node Problem . . . . .	170
7.3.7	Empirical Model for Fast Estimation of Ad-hoc Network Performance . . . . .	174
7.3.8	Validation and Calibration . . . . .	176
7.4	The Fifth-Generation Mobile Networks . . . . .	180
7.4.1	Where Can Working Sites be Benefited from 5G? . . . . .	180
7.4.2	Problem Statement and Goal . . . . .	185
7.4.3	Modelling . . . . .	185
7.4.4	Model Parameters . . . . .	187
7.4.5	Simulation Results . . . . .	190
7.5	Conclusion . . . . .	194
<b>8</b>	<b>Conclusions and Future Directions . . . . .</b>	<b>197</b>
	<b>List of Figures . . . . .</b>	<b>201</b>

**List of Tables . . . . . 211**

**Bibliography . . . . . 213**

**List of Publications . . . . . 249**

    Journal Articles . . . . . 249

    Conference Contributions . . . . . 250

    Supervised Theses . . . . . 252





# Acronyms and symbols

## Acronyms

<b>AEC</b>	Architecture, Engineering, and Construction
<b>AI</b>	Artificial Intelligence
<b>AIFS</b>	Arbitrary Inter-Frame Space
<b>AIoT</b>	Artificial Intelligence of Things
<b>AP</b>	Average Precision
<b>BIM</b>	Building Information Model
<b>BLE</b>	Bluetooth Low Energy
<b>BPSK</b>	Binary Phase Shift Keying
<b>CAM</b>	Cooperative Awareness Message
<b>CBS</b>	Conflict Based Search
<b>CCH</b>	Control Channels
<b>CCTV</b>	Closed-Circuit TeleVision
<b>CEM</b>	Construction Engineering and Management
<b>CL</b>	Confident Learning
<b>CNN</b>	Convolutional Neural Networks

<b>CRDNN</b>	A combination of CNN, RNN, and DNN
<b>CSMA/CA</b>	Carrier Sense Multiple Access with Collision Avoidance
<b>CSP</b>	Constraint Satisfaction Problem
<b>CT</b>	Constraint Tree
<b>DGPS</b>	Differential GPS
<b>DNN</b>	Deep Neural Network
<b>EDCA</b>	Enhanced DCF Channel Access
<b>EIFS</b>	Extended Inter-Frame Spacing
<b>EKF</b>	Extended Kalman Filter
<b>eMBB</b>	enhanced Mobile Broadband
<b>eNodeB</b>	evolved Node B
<b>EPC</b>	Evolved Packet Core
<b>EU</b>	European Union
<b>FCC</b>	Federal Communications Commission
<b>FCN</b>	Fully Convolutional Networks
<b>fMLLR</b>	feature-space Maximum Likelihood Linear Regression
<b>FPS</b>	Frame Per Second
<b>FS</b>	training From Scratch
<b>FTF</b>	Fully connected layers Transfer Learning
<b>GA</b>	Genetic Algorithm
<b>GDOP</b>	Geometric Dilution Of Precision

<b>GNSS</b>	Global Navigation Satellite Systems
<b>GPS</b>	Global Positioning System
<b>HARQ</b>	Hybrid Automatic Repeat Request
<b>HD</b>	High Definition
<b>HetNet</b>	Heterogeneous Network
<b>HMM</b>	Hidden Markov Models
<b>HOG</b>	Histogram of Oriented Gradient
<b>ICT</b>	Information and Communication Technology
<b>IFS</b>	Interframe Spaces
<b>IMU</b>	Inertial Measurement Unit
<b>IoT</b>	Internet of Things
<b>IoU</b>	Intersection over Union
<b>ITS</b>	Intelligent Transportation System
<b>LOS</b>	Line of Sight
<b>LSTM</b>	Long Short-Term Memory
<b>LuT</b>	Lookup-Table
<b>mAP</b>	mean Average Precision
<b>MAPF</b>	Multi-Agents Path Finding
<b>MIMO</b>	Multiple-Input Multiple-Output
<b>mMTC</b>	massive Machine Type Communications
<b>mmWave</b>	millimeter Wave
<b>ND</b>	Newly Gathered Dataset

<b>NFC</b>	Near-Field Communication
<b>NLP</b>	Natural Language Processing
<b>NR</b>	New Radio
<b>NS – 3</b>	Network Simulator 3
<b>OEM</b>	Original Equipment Manufacturer
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>OTF</b>	Overall Transfer Learning
<b>PD</b>	Previous Dataset
<b>PHY</b>	Port Physical (Layer)
<b>PLCP</b>	Physical Layer Convergence Protocol
<b>QoS</b>	Quality-of-Service
<b>QPSK</b>	Quadrature Phase Shift Keying
<b>RFID</b>	Radio-Frequency Identification
<b>RLC</b>	Radio Link Control
<b>RMSE</b>	Root-Mean-Square Error
<b>RNN</b>	Recurrent Neural Networks
<b>ROS</b>	Robot Operating System
<b>RTK</b>	Real Time Kinematic
<b>RTMP</b>	Real Time Messaging Protocol
<b>RTSP</b>	Real Time Streaming Protocol
<b>SAE</b>	Squeeze-And-Excite
<b>SCH</b>	Service Channel

<b>SIFT</b>	Scale-Invariant Feature Transform
<b>SIG</b>	Special Interest Group
<b>SLAM</b>	Simultaneous Localization And Mapping
<b>SOTA</b>	State-Of-The-Art
<b>SVM</b>	Support Vector Machine
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol
<b>TSC</b>	Time Series Classification
<b>UDP</b>	User Datagram Protocol
<b>UE</b>	User Equipment
<b>UI</b>	User Interface
<b>UKF</b>	Unscented Kalman Filter
<b>URDF</b>	Unified Robotic Description Format
<b>URLLC</b>	Ultra Reliable Low Latency Communications
<b>UTM</b>	Universal Transverse Mercator
<b>VTLN</b>	Vocal Tract Length Normalization
<b>WAVE</b>	Wireless Access in Vehicular Environments

## Symbols

### Path Planning for Machines Fleet Management

$C_o$	Intensity
$f$	Estimated cost forward
$f(n)$	Estimated cost from its source through current vertex $n$ to its goal
$g_i$	Real cost from the source to the current vertex $n$ considering the $i_{th}$ weight-grid map
$h_i$	Estimated cost from the current vertex $n$ to the predefined goal based on Manhattan distance
$L$	Total layers of the map
$P_{a_i}$	Priority value of the agent $i$
$r$	Estimated cost backward
$v$	Current point
$W_L$	Weight for the specific layer
$\pi_i$	Single-agent path for $a_i$ at time step $t$

### SLAM for Machines on a Smart Working Site

$A$	Covariance matrix to calculate Mahalanobis distance
$B$	Jacobian matrix of partial derivatives of $f(\bullet)$ with respect to $x$
$D_M$	Mahalanobis distance
$E$	Accumulated number of errors
$f(\bullet)$	Motion model function

$G_{i,j}$	Ground truth map information
$\hat{G}_{i,j}$	Estimated grid map information
$h(\bullet)$	Predicted measurement model
$H$	Jacobian matrix of partial derivatives of $h(\bullet)$ with respect to $x$
$Q$	Process noise of UKF
$r$	Resistance map
$R$	Measurement noise of UKF
$s$	Slope map
$\tilde{S}$	Ground truth x and y position
$\hat{S}$	Vector including estimated x and y position
$v_k$	Measurement noise of EKF
$V$	Jacobian matrix of partial derivatives of $h(\bullet)$ with respect to $v$
$w_k$	Process noise of EKF
$W$	Jacobian matrix of partial derivatives of $f(\bullet)$ with respect to $w$
$\bar{x}$	Selected mean sample point
$\tilde{X}^i$	Predicted position based on the motion model
$\hat{x}_k$	<i>A posteriori</i> estimate of the state at step $k$
$x_{k+1}$	Actual state
$\tilde{x}_{k+1}$	Approximate state
$x_{odom}$	X-coordinate in vehicle's world coordinate frame
$x_{UTM_0}$	X-coordinate of the first reported GPS position in UTM frame
$\mathcal{X}^i$	Final results from UKF

$y^0$	Measurement result with respect to $\check{X}$
$y_{odom}$	Y-coordinate in vehicle's world coordinate frame
$y_{UTM_0}$	Y-coordinate of the first reported GPS position in UTM frame
$z_k$	Measurement vectors
$\tilde{z}_k$	Measurement vectors
$z_{odom}$	Z-coordinate in vehicle's world coordinate frame
$Z_t$	Current measured point
$\bar{Z}_t$	Mean of measured points
$z_{UTM_0}$	Z-coordinate of the first reported GPS position in UTM frame
$\theta$	Vehicle's initial UTM-frame pitch
$\lambda$	Gaussian probability density function
$\psi$	Vehicle's initial UTM-frame yaw
$\phi$	Vehicle's initial UTM-frame roll

## Motion Prediction of Manned Working Machines

$a$	Activation
$b$	Bias unit
$\mathcal{D}_s$	Source domain
$\mathcal{D}_t$	Target domain
$e_0$	Working state traveling
$e_1$	Working state loading
$e_2$	Working state unloading



$m$	Sample number
$p_{bo}$	Pressure inside of bucket
$p_{bu}$	Pressure inside the bucket
$p_{cc}$	pressure inside of closed-circuit drivetrain
$\mathcal{T}_s$	Source task
$\mathcal{T}_t$	Target task
$u_{js}$	Vehicle direction signal on the joystick
$v_{veh}$	Vehicle velocity
$W_k$	Weight of k state
$x^{<t>}$	Input at $t$ step
$y$	Ground truth state
$\hat{y}$	Estimated state
$\Gamma_f$	Forget gate
$\Gamma_o$	Output gate
$\Gamma_u$	Update gate
$\lambda$	Regularization factor
$\sigma()$	activation function, concretely leaky ReLu

## Wireless Communication System

$d$	Distance between transmitter and receiver
$f_c$	Correction factor for IEEE802.11p analytical estimation method
$G_r$	Receive antenna gain

$G_t$	Transmit antenna gain
$L$	System loss
$n$	Path loss distance exponent
$n_S$	Number of nodes inside of the sensing range
$n_T$	Number of nodes inside of the transmission range
$P$	Path loss
$P_r$	Received signal power
$P_t$	Transmit signal power
$\lambda$	Wavelength
$\hat{\Phi}_A$	Performance matrix from the analytical model
$\hat{\Phi}_{A,n}$	Naive estimation of the performance of the ad hoc using the analytical model
$\tilde{\Phi}_S$	Performance matrix from the simulation

# 1 Introduction

Recent progress in Artificial Intelligence (AI) and Internet of Things (IoT) makes me believe that the traditional construction and mining site can be extended with an autonomous system or at least an assistant system, for the purpose of increasing productivity and safety performance, as well as reducing the cost for projects.

In this thesis, I focus on the fleet management logistic problem regarding productivity and safety on smart working sites. Currently, because of a lack of effective cooperation among individual machines, heavy machinery wastes a lot of time waiting for each other, and conflicts exist. Therefore, the total number of affordable construction machines within a working site is limited, and the overall productivity is less satisfying. The basic idea is to solve the moving conflicts inside of the working site so that more machines can be invested in performing tasks simultaneously and thus significantly improve productivity. A persuasive instance to show the benefit of introducing the logistic solution into the working site is the construction site for the famous hospital, namely Huoshenshan, the project in Wuhan during the coronavirus outbreak in 2020. By investing an extraordinary amount of working machines and human cooperators and manually coordinating the machines to avoid conflicts among them, the construction project was finished at an unprecedented speed. Apparently, the economic cost of running such a construction site can be quite expensive due to the salary for experienced workers. Also, since the logistic problem is Non-deterministic Polynomial-time hard (NP-hard), computer algorithms can better perform a series of optimization objects, such as shorter moving distance and realtime performance. In light of that, I try to use AI to replace human decisions in the working site and utilize IoT technology to share the information among the participants seamlessly. Compared to the proposals for an individual machine, which usually maximally increase up

to 50% performance, fleet management solutions show a potential to improve the productivity of a working site several times.

The thesis presents a series of approaches contributing to the machines cooperation strategy, machine motion prediction, site visual monitoring, and site wireless communication. Then, by combining these cornerstone technologies systematically, I demonstrate the blueprint of the future working site benefited from AI and IoT technologies.

## 1.1 Problem and Goal Statement

Apparently, the concept of smart working needs a system solution and thus cannot be solved with only one approach. To make it closer to reality, serious technical difficulties should be overcome. In particular, I focus on the following research questions to achieve AI-based fleet management:

- How can I plan the paths for a fleet of machines so that they can move to their goal quicker and safer?
- How can I predict the next position of machines to avoid the collisions among them? Especially for mobile machines, the heading of the vehicles does not show its moving direction.
- Where exactly are the machines and how can I acquire the working site map through it is dynamically changing?
- How can information be shared among the machines and site managers?
- How can I guarantee the safety of participants without localization equipment, such as workers?

I try to answer the aforementioned questions with cutting-edge AI and IoT technologies, which show the surplus human performance capability and thus have become the technological waves in the second decade of 21th century. Since

the machines inside of a large working site can be diverse due to their various function, a fully autonomous system for all the machines in the working site is still very challenging; thus, I endow my AI and IoT system with the ability to cooperate with the human being.

While the individual ideas to increase both productivity and safety whereas reduce the cost of projects have been intensively tackled from either management's or technical view, the improvement of the entire working site by an isolated technology is limited. I conjecture that it might because a working site is a complicated system and thus need concerted efforts. In this thesis, I first advance the current SOTA technologies for the individual aforementioned research question and then find the appropriate configuration to show the benefit of smart working site as a whole system.

## 1.2 Applications

I highlight two critical scenarios where the proposed concept smart working site shall be adopted.

In **mining site** and **construction site**, there is a high demand for the highly automated and coordinated fleet management from both economic and safety views. Furthermore, since the machines usually operate in a closed area at slower speeds and untrained pedestrians are already kept out of the construction site, it is easier and safer to automate their driving. The unmanned construction and mining site brings enormous benefits regarding safety, productivity, and labor perspective:

- **Safety Benefit:** Since 1900, over 100,000 coal mine accidents have taken places in the US [1]. In China, the number of deaths in coal mining accidents exceeded 2,000 each year from 1993 to 2010, whereas the peak occurred in 2002 with approximately 7,000 deaths [2]. The coal miners are highly exposed to coal dust and toxic gas, which results in a higher rate of ischemic heart disease and workers pneumococcus [3]. The safety problem can be significant eliminated by unmanned construction and mining site.

- **Productivity Benefit:** Currently, because of a lack of cooperation between individual machines, heavy machines waste a lot of time waiting for each other, resulting in low overall productivity. With the smart working site, the whole fleet is managed and dispatched without jams or collisions, and construction productivity is improved holistically. The enhanced productivity does contribute to reducing construction time and has remarkable economic benefits, e.g., to reduce the leasing cost of construction machines and the loan interest paid due to shorter rental contract.
- **Labor Benefit:** Since drivers are asked to relocate to remote locations and know how to drive on difficult terrains on construction sites, there are high training and labor costs for skilled drivers. Unmanned construction and mining sites can further reduce the labor cost.

## 1.3 Contributions

The contributions of this thesis are as follows:

- I proposed a multi working-machines pathfinding algorithm to guide the machine's movement with AI based on graph theory so that the spatial utilization of the working site is better and thus more machines can work simultaneously inside. Compared to the SOTA multi-agents pathfinding solution [4], my algorithm's most extensive feature is that it can quickly replan the path to overcome the emergency on a construction site.
- A SLAM method based on commodity sensors to offer the terrain information of construction machines despite the dynamically changing of the construction site is given.
- I proposed the novel algorithm CRDNN to recognize the working machines' working process and thus give the highly plausible information to predict the motion of working machines. It offers similar prediction performance

while outperforms the SOTA solution [5] in terms of generalization ability, thanks to its faster transfer learning capability.

- A visual monitoring system was created for the participant without localization equipment, e.g., unexpected visitors, to avoid tragedy by comparing the relative location of each participant in the image. The visual monitoring system compensates for the system deficiency in recognizing the participants without a location system, and works as a safety system.
- I demonstrated the first V2X communication system for working machines fleet management based on IEEE 802.11p and 5G worldwide. According to the characteristics of these technologies, I draw the blueprint of the smart working site.

## 1.4 Thesis Outline

This thesis is structured as follows: Chapter 2 shows the current contributions and developments on smart working site. Then, Chapter 3 presents the multi working-machine pathfinding algorithm that is employed. Chapter 4 details the SLAM technologies are used for acquiring the map information and machines' location in the working site. Afterward, Chapter 5 shows the working process recognition through multivariate time series algorithms to predict machines' motion. Followed by Chapter 6, the visual monitoring system is described. Finally, Chapter 7 illustrates the wireless communication system of the working site. Conclusions and outlooks are drawn in Chapter 8.





## **2 Background Knowledge of Smart Working Site**

This chapter discusses some previous contributions in smart working sites with respect to the existing literature. I begin with a brief overview of the novel technologies and ideas on the smart working site here and go deeper at the beginning of the following individual chapters.

### **2.1 Concepts and Consensus**

Smart working site is a novel integrated management and automation model for working sites, which is a high integration of AI, IoT, and traditional construction industry. It takes full advantage of the emerging information technologies such as mobile internet, Artificial Intelligence of Things (AIoT), cloud computing, big data, and focuses on key factors such as people, machines, materials, methods, and the environment to completely change the interaction and working mode on a working site. Obviously, my research also serves the establishment of the smart working site.

In general, construction project management manages construction components, such as workers, materials, and construction machines, aiming to achieve construction objectives quickly and well. Managing a construction site is to make a series of decisions across construction processes utilizing available information and knowledge [6]. The main objective of information management is to support decision-making by ensuring that accurate information is always available at the right time in the right format to the right person [7]. In recent years, there



**Figure 2.1:** An example of a smart construction site illustrated by Komatsu.

have been many studies on how to afford decision-makers with precise, timely, and well-organized information, e.g., exploiting Building Information Modeling (BIM) [8, 9]; by adopting Information and Communication Technologies (ICT) such as Auto-ID [10, 11, 12, 13], and sensing technology [14, 15].

However, due to the complexity and diversity of construction projects and the increasing demand for high-quality engineering, the decisions made by human beings are considered as more and more unreliable with the explosive growth of information, especially compared with the decision made by or with the help of advanced AI technology. Problems are often manifested in product quality defects, overtime, and over budget, caused by insufficient information, cognitive ability, and time. These problems afflict engineers with traditional management techniques before the AIoT era.

Fortunately, thanks to the tremendous progress and profound influence of AI and sensor technology, Construction Engineering and Management (CEM) is experiencing a rapid digital transformation. With more available tools, both academia

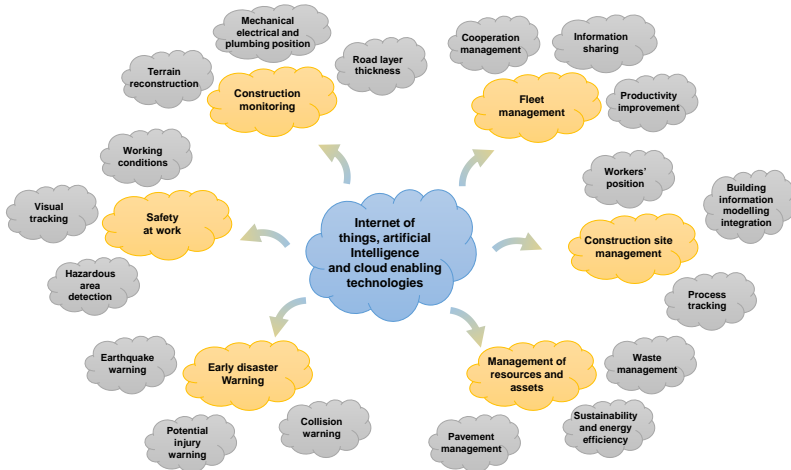
and industry have proposed their novel concepts to realize the construction industry's refined management methods and automation. Obviously, the development of smart construction sites is inseparable from the promotion of information technology. It is noteworthy that most researchers adopt the BIM system as the backbone to build up their contributions. Nowadays, the smart construction site is an integration of comprehensive intelligent systems. The interaction of physical space and cyberspace makes the advantages of smart construction sites be fully demonstrated. From the overall perspective, the management and automation of smart working sites determine a construction project's quality. In addition, compared with traditional construction environments, the smart construction site allows construction quality to be fully supervised. The communication between the staff will become easier and more straightforward, which saves working time, promotes work efficiency, and meanwhile ensures the quality of construction projects. Besides that, smart construction sites have become an indispensable and essential component of safe production. Through various monitoring settings installed at the construction site, as well as a more comprehensive intelligent monitoring and prevention system, it can better make up for various omissions in traditional management work.

The characteristic of CEM can be concluded as uniqueness, labor intensive, high dynamics, complexity, and uncertainty. These literature [16, 17, 18] indicate the basic requirements and the consensus of future working sites are following. First, the problems should be prevented before they actually occur. For instance, the digital twins' concept depicts a cyber-physical system where the digital model offers the simulation results to the physical model that inspection data is collected. Analogously, cloud VR/AR solutions realize more interactions between the cyber and physical worlds. Also, the information shall be shared and make the process more transparent through AIoT and blockchain technology. Consequently, the BIM model can be updated timely whenever the physical model is changed. Moreover, the process is expected to be supervised by using smart robotics, e.g., unmanned aerial vehicles. Last but not least, the AI system shall visualize the most important and concise information to the human decision-makers with data mining tools or substitute the human decisions directly with the advanced-AI

decisions in some cases. Here I found some applications using Natural Language Processing (NLP) to retrieve vital information from the reports for the human decision-maker.

## 2.2 Current Applications and Challenges

Besides the research topics creating improvements in the equipment systems, including increased engine efficiency, reduced greenhouse gas emission, improved electro-hydraulic control, information technology has attracted more and more attention in the last two decades. Especially, the triumvirate of IoT, AI, and cloud technologies offers new opportunities for the development of new applications on smart construction sites [17], shown in Fig. 2.2.



**Figure 2.2:** The critical topics for smart working site summarized by Štefanič [17] and us.

For example, in 2006, Lundeen developed a marker-based pose estimation system for excavators in order to determine the three-dimensional positioning and orientation of the trencher [19]. Later, a novel system was described by Turkan

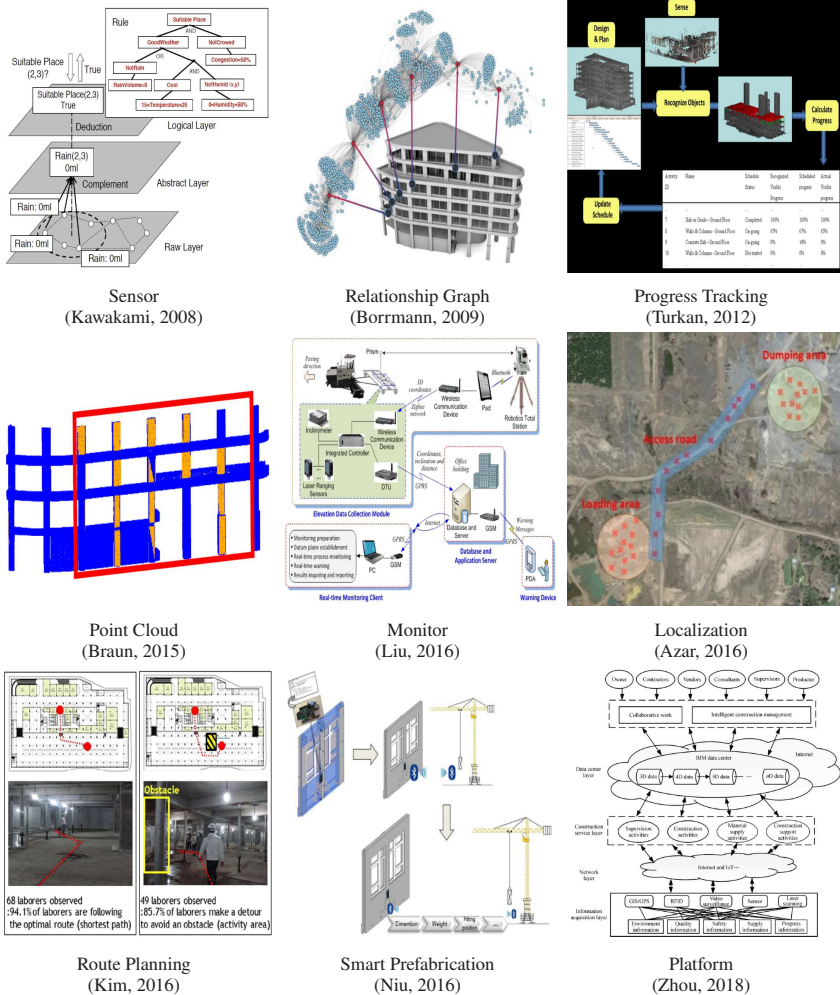
that integrates 3D object recognition technology with schedule information into a combined 4D<sup>1</sup> object recognition system focusing on progress tracking in 2011 [20]. Tested on a comprehensive field database acquired during the construction of the structure of the “Engineering V Building” at the University of Waterloo, this system demonstrates a degree of accuracy for automated structural progress tracking and schedule updating that meets or surpasses manual performance. In 2016, Ren and Wu developed a realtime automated anti-collision system that can warn crane operators about potential collisions and then automatically implement collision-avoidance strategies [21]. One advantage of this system is that it does not require additional devices and can be installed in existing crane controllers. In the same year, another novel system was proposed by Liu that combines inclinometer, laser ranging sensors, and wireless communication technologies to monitor lift-thickness during highway construction [22], and Braun presented a concept for an automated comparison between the actual state of construction and the planned state for the early detection of deviations in the construction process [23]. In this concept, the actual state of the construction site is detected by photogrammetric technology. Concretely, dense point clouds are generated by the fusion of disparity maps created with Semi-Global-Matching (SGM). These are afterward matched against the target state provided by a 4D Building Information Model. Also in 2016, based on the deviation between the optimal route determined by extracting nodes from BIM and the actual route of a laborer collected from the Real-Time Location System (RTLs), Kim proposed an automated hazardous area identification model to improve the efficiency of safety management [24].

As we can see, the contributions about the smart working site demonstrate many benefits and are proved to be promising approaches to improve the current daily working site. However, they are not preferable for commercial solutions until 2020, i.e., most working sites in reality are still quite traditional. I conjecture that it may be because that these contributions do not improve productivity several times. Most of the current information technologies and corresponding management in

---

<sup>1</sup> Besides the x,y,z coordinates denoted by the first three dimension, the 4D model describes the schedule information using the fourth dimension.

## 2 Background Knowledge of Smart Working Site



**Figure 2.3:** Selected works about smart working site [14, 25, 20, 23, 22, 16, 24, 26, 27].

the earthmoving industry address real-time tracking and productivity estimation of the equipments rather than productivity improvement. As a result, these technologies are considered as incomplete solutions for some end customers, e.g.,

construction contractors. For instance, productivity estimation function might be challenging to be accepted by some engineers if there are not corresponding productivity-increasing methods. In fact, the McKinsey Global Institute (MGI) analysis found that the construction industry was among the least digitized industries in the total economy, and the annual productivity growth over the past 20 years was only a third of total economy averages<sup>2</sup>. Hence, improving productivity is a very urgent and important issue in the future.

Notably, the world-famous Huoshenshan construction project in Wuhan demonstrated the possibility to significantly boost construction productivity<sup>3</sup> by means of investing in a large number of machines. In light of that, I explore the approach to increase the number of machines in a working site and solve the cooperation and safety problems among machines and workers utilizing AI and IoT technologies in this thesis.

---

<sup>2</sup> McKinsey: The next normal in construction (2020)

<sup>3</sup> To avoid the misunderstanding, productivity is defined as the efficiency to execute construction projects in this thesis. Although there are no comprehensive analysis of how much faster is the Wuhan's construction site, ultra-rapid is the most well-known feature for this site [28, 29].





# 3 Path Planning for Machines Fleet Management<sup>1</sup>

Multi working-machines pathfinding solution enables more mobile machines simultaneously to work inside of a working site so that the productivity can be expected to increase evolutionary. To date, the potential cooperation conflicts among construction machinery limit the amount of construction machinery investment in a concrete working site. To solve the cooperation problem, civil engineers optimize the working site from a logistic perspective while computer scientists improve pathfinding algorithms' performance on the given benchmark maps. In the practical implementation of a construction site, it is sensible to solve the problem with a hybrid solution; therefore, in my study, I proposed an algorithm based on a cutting-edge multi-pathfinding algorithm to enable the massive number of machines cooperation and offer the advice to modify the unreasonable part of the working site in the meantime. Using the logistic information from BIM, such as unloading and loading point, I added a pathfinding solution for multi machines to improve the whole construction fleet's productivity. In the previous studies described in Section 3.2, the experiments were limited to no more than ten participants, and the computational time to gather the solution was not given; thus, I publish my pseudo-code, my tested map, and benchmark my results. My algorithm's most extensive feature is that it can quickly replan the path to overcome the emergency on a construction site.

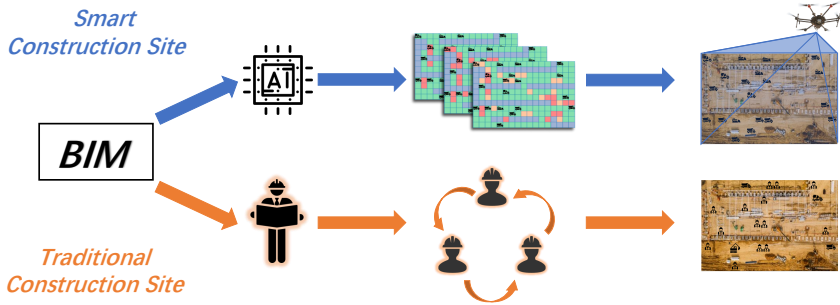
---

<sup>1</sup> All the figures, text, and results of the presented work in this chapter have been published in our publication [30]. My contribution to the paper is summarized as 100% in terms of conception and methodology, 90% of literature review, 90% of code, 60% of results visualization, and 95% of formulation.

## 3.1 Introduction

Although many achievements in construction machines with respect to productivity and safety, humans' pursuit of even higher productivity and better safety never stops. In the past decades, civil engineers and construction-industry-related software engineers introduce the BIM [31] as a powerful tool to increase productivity and safety performance whilst reduce the project cost by means of digital technology. In general, BIM provides the 3D or more than 3D model of the construction projects and even the installation sequence of the components to avoid mistakes during the construction stage. With the maturity of BIM, this software and process are adopted for many large and especially complicated construction projects worldwide [32] since the mistakes during the real construction process cause much more severe consequences than those in virtual engineering. Also, BIM is considered as a lifelong software, contributing to not only the early construction phase [31, 33] but also the time after the construction projects are finished [34]. Despite to the preliminary cost for training corresponding laborers and model building in a computer, BIM is a necessary tool for at least large construction projects becomes a consensus. However, although current BIM software defines the start and end points where the material should be transported, concrete paths guiding the trucks to accomplish the goal are not given. Or more generally, an algorithm that determines the paths of the participants in the working site so that they can move to their destination without collision and hesitation is still developing.

As shown in Fig. 3.1, one motivation to combine these path planning algorithms with BIM can be described as determining the construction machines' travel path so that the machines can be expected to move faster and denser without hesitation. To accomplish higher productivity and better safety in the path planning fashion, some researchers calculate and display efficient paths by a given construction site [35, 36] whereas others optimize the construction site layout considering the high productivity of the path [37, 38]. However, most of the current solutions about path planning on a construction site are mainly focusing on individual units, i.e., the interaction and path conflict of the different machines or other agents



**Figure 3.1:** Overview of the smart construction site concept. I introduce to use artificial intelligence to unify the scheduling of construction machinery on construction sites. Maximize the use of construction machinery on specific construction sites by avoiding conflicts among construction machinery and thus ultimately improve the productivity of the construction sites.

are ignored. Consequently, the working sites' spatial and time utilization is limited. Inspired by warehouse logistics solution [39], where a lot of robotics are working with the commands from path planning algorithms in the meantime and thus achieve considerably higher transport efficiency as a fleet, I envision the Multi-Agents Pathfinding (MAPF) solution can also provide evolutionary to the construction industry. A persuasive instance to show the benefit of introducing MAPF solution into a working site is the construction site for the famous hospital, namely Huoshenshan, in Wuhan during the coronavirus 2019 outbreak. Invested an extraordinary amount of working machines and human cooperators, the construction project was finished at an unprecedented speed, through only manually coordinate to avoid conflicts among machines. Apparently, the economic cost of running such a construction site can be quite expensive due to experienced workers. Also, since the MAPF problem is NP-hard, computer algorithms can surely have a better performance concerning a series of optimization objects, such as shorter moving distance and realtime performance. In light of that, I propose a MAPF solution for working machines as an extension of the BIM system so that more machines can work simultaneously and thereby achieve better holistic productivity.

The aim of this chapter is to extend the current BIM software with AI path-planning algorithms in order that more machines can work simultaneously since the paths are calculated to avoid collisions. Because the start points and endpoints can be given directly in the BIM system, I devote myself to the implementation level, i.e., how exactly the machines should achieve their goal set given by the BIM system. I envision that the case in Wuhan will be a normal case in the future by utilizing AI and IoT [40, 41] technology.

The main contributions of this chapter can be sum up as the following points:

- I introduced a novel MAPF solution to guide a fleet of mobile machines to work simultaneously inside a working site and to give suggestions to optimize the construction site layout.
- The approach I proposed can always provide a feasible solution on weighted maps considering the agents' priority in a predefined short period to deal with emergencies.
- I extended the cutting-edge MAPF solution with a bidirectional searching method for the initial search of the best path, which shall be principally faster.
- My method can be added to BIM software to make up for its lack of path planning.
- I benchmark the performance of my MAPF solution for the mobile machines with respect to the solution found time and cost to reach their goal on the given maps.

The rest of this chapter is organized as follows. Section 3.2 briefly introduces the prerequisite and background knowledge in fields of BIM, path planning methods for construction machines and robotics to understand this chapter quickly. Next, the existing problems are illustrated in Section 3.3. After that, in Section 3.4, I describe the setup of my MAPF approach, including the map to give the start and goal positions, low level for individual pathfinding, and high level for conflicts solving. Then, I show the experiments setup. Followed by Section 3.6, I show my

approach's performance by testing on some maps based on real construction sites. Finally, Section 3.7 summarizes the advantages of my approach, and Section 3.8 gives conclusions and envisions the outlook.

## **3.2 Related Works**

### **3.2.1 Building Information Modelling (BIM)**

BIM is a 3D model-based information management process in the field of Architecture, Engineering, and Construction (AEC) that facilitates efficient design and construction processes and inter-organizational collaboration [42]. There is a lot of BIM-based software: Autodesk Revit Building (Revit), ArchiCAD, Bentley, and SolidWorks [43]. By building the whole project virtually before physical construction begins, construction sequencing is determined, including material ordering, fabrication, and delivery schedules for all building components, etc. Therefore, conflict, interference, and collision are avoided in the early stage, contributing to improved site efficiency and reduced cost [44]. As the function inside BIM increases, such as scheduling, virtual reality [45], and logistic management [42], it has been extended to 4 or more than 4 dimensional model. Nowadays, the research about BIM is prosperous. Combining AI and IoT into the BIM system is considered the next potential boom for BIM systems. Survey papers about that can be found in [46, 47].

In order to automate the whole construction site and compute the optimal path for the heavy machines, logistic information is vital, which demonstrates which materials should be placed in which location at which time in the right quantity. Logistics management in construction involves the strategic storage, handling, transportation and distribution of resources, as well as planning of a building site's layout [48]. Whitlock has proposed a desktop approach to adopt BIM for construction logistics management [42]. Such logistic information, for instance, unloading points, on-site arrangements-logistics layouts, which are generated at

the outset of the pre-construction process, can be used as the input data for the path planning.

### 3.2.2 Path Planning for Construction Machines

On a construction site, there are usually multiple machines working simultaneously in a definite area with given assignments. Therefore, coordinated construction logistics can definitely increase productivity, decrease material usage, and guarantee workers' health and safety [49]. To date, there is plenty of researchers proposed their solution for the logistical problem inside construction sites at diverse levels, i.e., path planning and motion control. In this section, I summarize the previous research about moving paths inside of construction sites.

In the construction industry, the earth-moving sector is among the pioneers in adopting new sensing and information technologies [16], such as bulldozer [50, 51], and grading machines [52]. Given two points A and B on a construction site, the objective is to determine the shortest path from A to B maintaining a safe distance from obstacles. The approach proposed by Kim is a path-planning method for a mobile construction robot to find a continuous collision-free path from the initial position of the construction robot to its target position by improved Bug-based algorithm [53]. The algorithm can work with the disturbance of static and dynamic objects. Obviously, the performance of the approach is based on the accuracy of the sensors. At that time, the methods to acquire site information were still immature. Hence, the spatial model supporting path planning in a partially known and partially unknown environment was brought forward by Lee. Accordingly, the spatial model provides the domain for finding an efficient path on a construction site through the use of an algorithm that combines a shortest path algorithm and a dynamic path-planning algorithm. This approach differs from existing path-planning approaches that assume the construction site is totally known or totally unknown. Thus, problems associated with managing a changing construction environment and ignorance of designated roadway networks are overcome [54]. In the same year, Soltani has compared the performance of different

methods for the path planning inside construction sites [55], such as Dijkstra [56],  $A^*$  [57], and Genetic Algorithm (GA) [58], by evaluating comprehensive multi objects, e.g., site layout representation, distance formulation, hazard zone modeling, and visibility calculations. Also, the author points out the use of Closed-Circuit TeleVision (CCTV) cameras should be considered to enhance site security [55]. Although the simulation results show that the GA has the best performance and the other two algorithms have quite similar results, I conjecture that it might be ascribed to their maps, which are quite easy and do not include difficult obstacles such as bottlenecks. Fairly recently, Song tried to integrate some path planning algorithms into BIM system [35]. The basic idea is to determine the path to transport the materials at the very beginning phase, i.e., the construction site design phase. Also, the study verified the demand for the introduction of path planning by survey questions. The shortcoming of this approach is that the interaction of other participants inside construction sites during the construction project was not taken into account.

So far, the aforementioned studies focus on a path for one machine inside the construction site. In contrast, the following research shows solutions for multi machines working simultaneously on a construction site. An influential study about the path planning on the construction site is from Cheng published in 2012 [59]. The objective of the paper is to provide the  $n$  best and safest paths between two points in a work area while maintaining a safe distance from identified obstacles. Here the approach proposed in the paper uses Dijkstra algorithm and solves the path of different participants in sequence. Due to the limited recognition distance of ultra-wideband sensors, the usage of this approach might not suitable for huge working sites. Also, since the paths of agents are calculated one by one, the computation efficiency is not ideal from today's point of view in 2020. 4 years after Cheng's study, the research from Bohacs shows the difficulties of path planning for a construction site. Also, they use  $A^*$  as basic and develop an algorithm to let limited machines can cooperate without collision [60] within a small map. Concretely, they showed the demo about 3 machines in a  $10 \times 10$  grid map. As the flow chart in their paper shows, the algorithm depends on the

condition statements. As a result of that, it cannot perform with all the dispensable computation effort of the computer.

As the development of information technology, Štefanič provides an overview of emerging smart construction applications in areas such as construction monitoring, construction site management, safety at work, early disaster warning, and resources and assets management [17]. Also, Tumer describes the future construction site utilizing industry 4.0 [61]. Without a doubt, a future working site should be fully benefited from AI [62, 63, 64], automation technology [65], Simultaneous Localization and Mapping (SLAM) [66], and IoT [67, 40, 68, 41]. Therefore, the uncertainty degree of construction sites is reducing, and thus I consider the construction sites as known in my research.

Based on the strict literature review,  $A^*$  is the most developed and latest algorithm to solve construction sites' path tasks. It combines both step-cost calculation from the Dijkstra algorithm and feedback step from the genetic algorithm. However, as the number of machines increases, naive  $A^*$  might not be suitable to solve the MAPF problem due to algorithm complexity. Thus, it is necessary to explore the SOTA solutions in the field of mobile robotics in order to find a more appropriate solution.

### 3.2.3 MAPF for Mobile Robotics

For a single agent, the planning task can be described as finding the lowest cost from the starting point to the targeting point. By using Heuristic Search, e.g.,  $A^*$  Algorithm, such problem can be better solved. However, naive  $A^*$  Algorithm only considers that all the obstacles are static, which is the ideal assumption in the pathfinding problem [69]. In contrast, to solve the MAPF problem, the other participants in the map must be considered, i.e., the dynamic obstacles also affect the optimal path of an individual agent.

MAPF for mobile robotics is both a well-studied and dynamic developing topic. The usage scenarios of MAPF and their corresponding algorithm are diverse, such



as warehouse [39], computer games [70], and autonomous driving in intersection [71]. Until the end of 2020, although the concept of reinforcement learning is attracting more and more attention, current influential research about the shortest path and MAPF is mainly based on graph theory. To date, there is no universal solution for all kinds of pathfinding problems; thus, algorithms with different time and space complexities are proposed [72].

Based on the survey paper from Felner [73], it is known that no algorithm dominates all others in all circumstances. There is a tradeoff between high-quality path solutions and realtime performance. The mainstream MAPF solution can be classified into search-based solvers and rule-based solvers. The former intends to find the best solution or near-optimal solutions, whereas the latter can run much faster, however, produce far away from optimal solutions. Of course, some compromised solutions combined two ideas together, namely, hybrid approaches. Another type of solution, namely reduction-based optimal solvers, focusing on reducing MAPF problems into some problems, such as the Constraint Satisfaction Problem (CSP), with a well-known solution. Since this approach usually only aims at the makespan tasks, I will not go much deeper in this approach. To date, the most influential solutions for MAPF are Conflict Based Search (CBS) and its variants due to their widely used real-world applications.

Sharon proposes Conflict Based Search (CBS), combining both advantages from coupled and decoupled approaches. Although the pathfinding process is strictly single-agent searches, it can guarantee to offer optimal results, unless the variant that deliberately provide a suboptimal solution for the purpose of realtime performance. As the authors introduce, CBS adopts a two-level structure, where the high-level search can be described as a Constraint Tree (CT), including every constraint. Then, the lower level finds the concrete path for each agent individually with the information from the higher level. The brilliance of this design is that the search process is not more exponential in the number of participants but exponential in the amount of conflicts encountered during the pathfinding process.

Since CBS tries to find the optimal solution and thereby causes a relatively longer runtime, in improved CBS algorithm [74], Boyarski summaries two methods

to reduce the runtime. Concretely, it firstly adopts the Meta-Agent CBS (MA-CBS) [4], which merges multi-agents together and handles them as a large agent. Moreover, it uses bypass improvement, which encourages one of the agents to find an alternative path instead of performing a split at the high level. As mentioned by Boyarski, the bypass concept successfully avoid the unnecessary generate the new nodes in the CT [74]. Since the bypass concept only tries to find the solution from the path with the same cost as the one that shall be replaced, the optimality cannot be harmed. In the high-level search, Felner suggests adding heuristics into CBS so that the conflicts are not arbitrarily chosen [75]. After that, Li found the improved heuristics to guide the high-level search [76]. Also, She introduced the CBS with disjoint splitting [77]. The main contribution of CBS with disjoint splitting is the novel terminology of positive constraints forcing the  $a_i$  to be at  $v$  at timestep  $t$ . In this fashion, CBS with disjoint slitting reduces the amount of unnecessarily expanding the CT. In addition, some improvements, such as Lazy CBS, which avoids the behavior that CBS resolves the same conflicts between the same pairs of agents many times owing to lack of connection among subproblems [78]. Hönig proposed an approach called Conflict-Based Search with Optimal Task Assignment (CBS-TA) [79]. The improvement is mainly because it creates the forest on demand. Solving MAPF optimally is proven to be NP-hard, so CBS and all other optimal solvers do not scale up. Alternatively, Barer proposed a suboptimal variant of the CBS algorithm [80] so that the problem can be solved suboptimally but much faster.

To sum up, naive CBS is an optimal pathfinding solution that is based on graph theory. The time consumption of the algorithm mainly depends on the conflicts occurring among the agents since they increase the nodes in the high-level tree. Its performance on bottlenecks and corridors is better, whereas the performance on open space can be worse than enhanced  $A^*$ . Thus, CBS's variants are focusing on reducing the nodes in the CT and therefore let CBS has a higher success rate in general. Note that in this chapter, I use the same terminology as Stern's research to avoid misunderstanding; however, some mathematical descriptions might be adjusted.

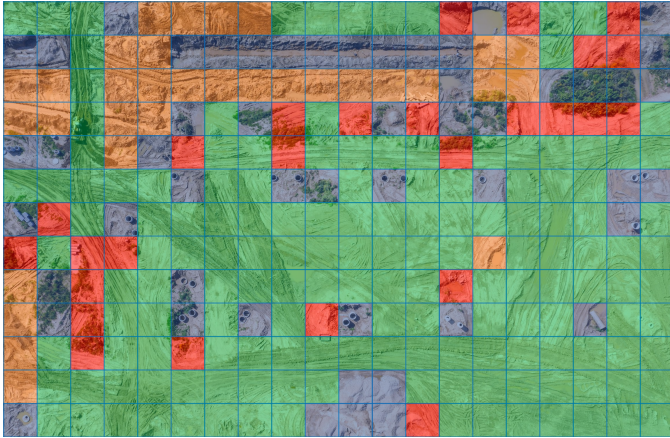
### 3.3 Problem Statement

Although the path planning problem has been attracted engineer's attention, the proposed method is quite tricky to be used in a large construction site with many machines. Theoretically, given a 4 neighborhood movement model and an undirected graph  $G(V, E)$ , the branching factor should be estimated as  $(E/V)^k$  and the search space is  $V^k$  if  $k$  machines should be planned. For 20 agents, considering that a normal working site with  $500m \times 100m$  where  $50 \times 10$  cells are needed, the search space goes to  $(50 \times 10)^{20} = 9.54 \times 10^{53}$ , which is unsolvable within acceptable duration for a real application even if the top CPU in 2020 achieving approaching 200 GFLOPs is used. Thus, using the traditional methods to solve the cooperation task is still challenging. Also, some algorithms are based on replanning the path if agents encounter a head-to-head position. However, these methods require excellent perception capability and limit the movement velocity of agents.

On the other hand, CBS based solution treats all the objects equally. Concretely, each robotics has the same capability, the cell on the ground is assumed as equally challenging to be overcome. However, it does not hold true in a working site; some machines should be assigned a higher priority, and some paths are much easier to pass. For instance, larger machines are usually more challenging to control their velocity. Also, stopping on a slope is much dangerous than on the flat ground. Hence, in my study, I further develop the original CBS so that it can deal with plenty of priority problems in a real working site. Also, the computational time should be much shorter to handle emergence.

### 3.4 Model Building

Nowadays, with the development of SLAM regarding visual recognition, IoT, and satellite technology, the uncertainty degree of construction sites is reducing. Thus I consider the construction sites as known in this chapter. In most instances, the MAPF solution can be evaluated twofold. The first one is sum-of-costs



**Figure 3.2:** A grid map with terrain weight based on a real construction site, drawn by Liu [81]. In this map, the green, orange, and red grids demonstrate the easy, normal, and difficult to pass terrain, separately. The blue cells denote the place where it is considered impossible to pass, such as occupied by the obstacles. On a real construction site, the obstacles can be the place to store construction materials temporarily.

which describes the accumulative cost of all the agents. Such costs can be time consumption, fuel consumption, or some other objective goals. The other way to analyze the performance of MAPF solvers is makespan, indicating the maximal time the last goal has been achieved. Obviously, unlike the robotics in warehouses, working machines perform a relatively long period to do their duty after they have arrived. Thus, makespan is not so vital compared to sum-of-costs in the field of construction or milling machines. Consequently, I adopt sum-of-cost as the evaluation criterion rather than makespan.

For working site MAPF problems, the problem can be described as given a graph,  $G(V, E)$ , and a set of  $k$  agents labeled as  $a_1 \dots a_k$ . Each agent  $a_i$  has a start position  $s_i \in V$  and goal position  $g_i \in V$ . Based on the practical conditions in a working site, I consider vertex conflicts, edge conflicts, and swapping conflicts as unacceptable conflicts, whereas following and cycle conflicts are allowed in my study. Formally, the unacceptable conflicts are described as,

$$\begin{cases} \pi_i[t] = \pi_j[t], \\ \pi_i[t] = \pi_j[t + 1] \cup \pi_i[t + 1] = \pi_j[t] \end{cases} \quad (3.1)$$

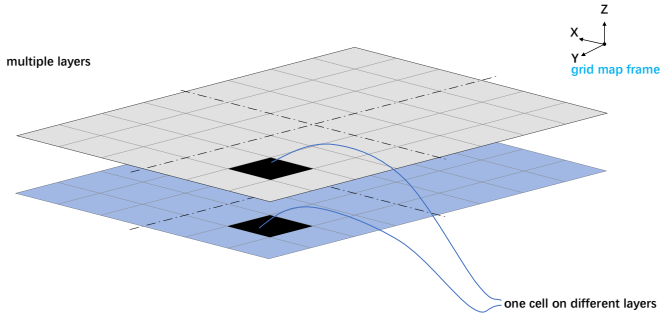
where  $\pi_i$  and  $\pi_j$  are the single-agent path for  $a_i$  and  $a_j$  at time step  $t$ , correspondingly. The first equation shows the vertex conflict whilst the second equation describes the swapping conflict. Apparently, edge conflict is a union of vertex conflict; thus, no additional equation is needed. Intuitively,  $\pi_1[2]$  denotes the location of the first agent at the second time step.

### 3.4.1 Multi-Layer Grid Map

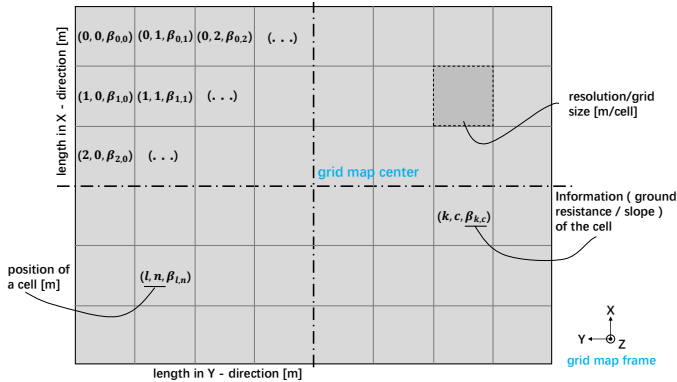
Unlike a standard graph problem which adds the weight on the edges, I add the weight directly on the grids. This is mainly for three reasons. First and foremost, the machines usually occupy a relatively large area and thus should not be modeled as a simple vertex and ignore their geometry. Also, most previous studies in the field of construction machines used the grid-based map. To guarantee compatibility, I tend to use a similar solution since no approach from them obviously outperforms the other one. Last but not least, if weights are applied on edges, it becomes challenging to penalize the waiting process.

Inspired by the research from Fankhauser [82], a map can include many layers to store different types of data information. Obviously, the map information should be saved in BIM system so that the path planning process can be done. In the previous study [66], Xiang developed a realtime map plotter of the construction site according to ground condition, offering multi-layer grid-based maps, which divide the environment into uniform cells. Moreover, maps can also be gathered by Lidar or cameras provided depth information installed on a drone or on the ground.

Fig. 3.3 illustrates the multi-layered grid map concept, where each cell data is stored on the congruent layers. In many construction projects, since resistance and grade of the road are the most of importance information for the construction



**Figure 3.3:** An example of multilayered grid maps. My approach depends on multilayered grid maps to offer data of different types of information to make the best path. Concretely, every grid saves a  $1 \times 3$  matrix, including location information and the corresponding terrain information. The map, which can be visualized in the BIM system, is saved as a  $2 \times m \times n \times 3$  tensor, where  $m$  is the max displacement in the x-direction, while  $n$  is the max displacement in the y-direction. In case a place is unknown, it will be marked as NaN to denote the uncertainty of the regions and be treated as obstacles.



**Figure 3.4:** Detail description of a layer in the grid-based map.

machines, I show a two layers grid map as an example. Concretely in my study, the map is divided into small cells, whose resolution is 10 meters per cell to cover the geometry of the vehicles. In practice, GPS/IMU based Kalman filter algorithms can be used to locate the mobile machine on the construction site. To describe the ground condition of construction sites, I use the value of each cell to represent the information of the ground situation. In Fig. 3.4, a layer that holds the data of a grid-based map is shown. Apparently, although I only demonstrate the map with two layers, it is relatively easy to extend the third layer in case more information should be taken into account for the path planning because I can simply add the weights together.

### 3.4.2 Lower Level Search

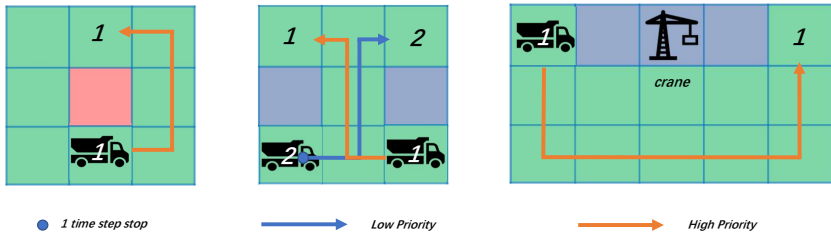
The concept of CBS does not limit the lower-level search algorithm. In addition, since negative weight cannot happen in 3D space, I believe that Dijkstra and  $A^*$  can work well for individual shortest path search. In light of that, I use best-first search<sup>2</sup>. Also, to accelerate my algorithm, I limit the moving direction of an agent to 4 and thus reduce the branching factor to 5, including wait, instead of 9 or more. In order that I can get the optimal solution, I set the heuristics smaller than the real distance since weights are considered. Because grid map is used, I use Manhattan distance as the base of the heuristics to guide the search.

In the lower level, the algorithm searches the best path of individual agent based on the estimated cost of through current vertex to the goal, formally,

$$f_{f,r}(n) = \sum_{i=1}^p W_{L,i} \cdot g_{i,f,r}(n) + h_{f,r}(n) \quad (3.2)$$

where  $f(n)$  is the estimated cost from its source through current vertex  $n$  to its goal,  $g_i$  denotes the real cost from the source to the current vertex  $n$  considering the  $i_{th}$  weight-grid map, and  $h$  denotes the estimated cost from the current vertex

<sup>2</sup> The concept that searches the most likely region first.



**Figure 3.5:** The basic requirements of the path planning algorithm. As shown in the left subfigure, the vehicle should take the lowest-cost path to reach its goal. The middle subfigure shows that the vehicle with lower priority should wait until the vehicle with higher priority pass through if there is no other bypass possibility. Last but not least, the right subfigure indicates that a good path should not be too close to dangerous objects.

$n$  to the predefined goal based on Manhattan distance.  $W_L$  is the weight for the specific layer. The index  $f$  and  $r$  show the estimated cost is forward or backward.

Apparently, planning the best path for machines to reach their goal is a multi-objective task. Generally speaking, the evaluation criterion can be divided into subjective and objective criteria. Obviously, the objective criterion demonstrates the objective criterion of the planned path, especially the terrain which can affect safety and efficiency. As some roads inside the construction site can be built with asphalt, so is considered better road conditions than some road made of sand. Consequently, the cost of passing different routes is different. Besides that, the road slope should be taken into account since waiting on a steep hill is much more dangerous than staying on flat ground. Therefore, I introduce multi-layer to record the individual characteristics of the terrain and plan the best path based on them. Concretely, a construction site map is divided into a series of cells and layers, according to different criteria. The weights of individual cells in one layer are saved as shown in the following matrices.



$$\tilde{W}_{m,n} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix}$$

In contrast, the subjective criterion may not harm the whole system's actual performance; however, it has an impact on people's psychology. For instance, a crane or some other dangerous objects, such as a power station, on a working site should be protected, and the situation that the mobile machines unnecessarily approaching them should be avoided. As we know, even machines did not involve in an accident, getting close to a dangerous object will be stressful for site managers and indicating a potential risk. To address this problem, I add  $g_3$  to penalize the machines for occupying the areas surrounding these special objects, see Eq. 3.3,

$$g_3(n) = \sum_{o=1}^r \frac{C_o}{|(X_n - X_o)| + |(Y_n - Y_o)|} \quad (3.3)$$

where  $[X_o, Y_o]$  is the position of the objects which should try to avoid being approached,  $C_o$  denotes the intensity.

---

### Algorithm 1 Bidirectional A\* Algorithm at Low Level to Speed up the Searching Process

---

**Input:**  $G(v, t)$ ,  $\tilde{s}$ ,  $\tilde{g}$  from predefined map information in yaml, original from visual or Lidar recognition

**Output:**  $Path$ ,  $d_{shortest}$

*Initialisation :*

- 1:  $OpenSet \leftarrow [s, t]$ ;  $ClosedSet \leftarrow [\Phi, \Phi]$   
 $G^R \leftarrow ReverseGraph(G)$
- 2:  $dist[:] \leftarrow \infty$ ,  $dist^R[:] \leftarrow \infty$
- 3:  $dist[s] \leftarrow 0$ ,  $dist^R[t] \leftarrow 0$
- 4:  $[CF, CF^R, proc, proc^R] \leftarrow \Phi$

*LOOP Process*

```
5: while OpenSet[0], OpenSet[1] not empty do
6:    $v_c \leftarrow \text{ExtractMin}(\text{dist})$ , forward otherwise  $\text{dist}^R$ 
7:   OpenSet.remove[0]( $u$ )
8:   if neighbor( $u$ ) = valid then
9:     if  $u$  not in ClosedSet[0] then
10:       $g[0][u]_{tmp} \leftarrow g[0][v] + \text{StepCost}$ 
11:     end if
12:     if neighbor not in OpenSet[0] then
13:       OpenSet.append( $u$ )
14:     else if  $g[0][u]_{tmp} > g[0][u]$  then
15:       continue
16:     end if
17:      $CF[0][u] \leftarrow v$ 
18:      $pi_f \leftarrow |d(v) - d(s)|$ ,  $pi_r \leftarrow |d(t) - d(v)|$ 
19:      $h_f \leftarrow (pi_f - pi_r)/2$ 
20:      $h_r \leftarrow -p_f$ 
21:      $f[0][u] \leftarrow g[0][u] + h_f$ , if forward otherwise  $h_r$ 
22:     end if
23:     if  $u$  in ClosedSet[0] then
24:       break
25:     end if
26:     ClosedSet[0].append( $u$ )
27:     repeat symmetrically for  $v^R$  as for  $v$ , where OpenSet[1] and ClosedSet[1]
       should be used
28:   end while
29:    $distance \leftarrow \infty$ ,  $u_{best} \leftarrow \text{None}$ 
30:   for  $u$  in ClosedSet[0] + ClosedSet[1] do
31:     if  $\text{dist} + \text{dist}^R < distance$  then
32:        $u_{best} \leftarrow u$ 
33:        $distance \leftarrow \text{dist}[u] + \text{dist}^R[u]$ 
34:     end if
35:   end for
```

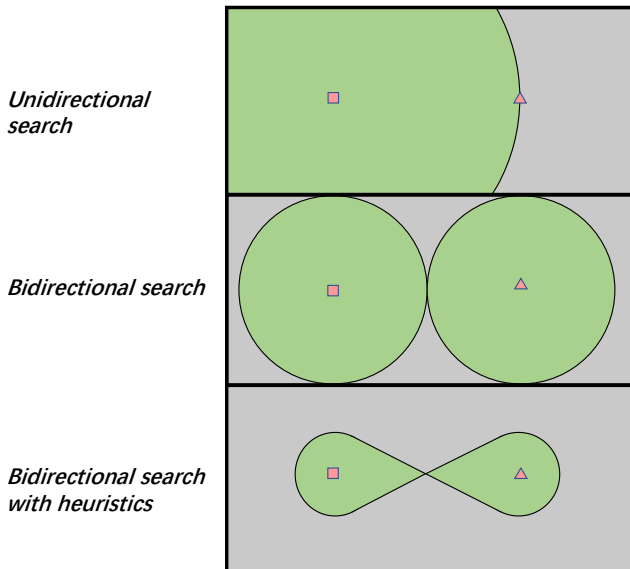
---

```

36:  $last_0, last_1 \leftarrow u_{best}$ 
37: while  $last_0 \neq s$  do
38:   path.append(last)
39:    $last_0 \leftarrow CF[0][last_0]$ 
40:   path.reverse
41: end while
42: while  $last_1 \neq t$  do
43:   path.append(last)
44:    $last_1 \leftarrow CF[1][last_1]$ 
45: end while
46: return path, distance

```

---



**Figure 3.6:** Illustration of the benefit of bidirectional search with heuristics. I use squares and triangles to represent the start point and the goal point separately. Here the grey region denotes the whole region of the map, and the green region shows the region that the algorithm must search before finding the shortest path for agents. Apparently, bidirectional search with heuristics cannot be slower than its antecessors.

To accelerate the low-level searching process, I utilize the bidirectional  $A^*$  algorithm with a path return for the initial search. The following Algorithm 1 shows the algorithm's steps, where CF is the dictionary to save the path sequence. Since bidirectional  $A^*$  is not suitable for dealing with the waiting operation, neighbors are validated if the grids are not occupied by obstacles, i.e., ignoring the conflicts with other agents. The index 0 in the algorithm denotes forward, whereas index 1 means backward.

After the initial solution is proposed, I adopt unidirectional  $A^*$  to solve the conflicts among the agents. Also, the priority of the agents is taken into account here. As a conflict occurs, the algorithm will give the order which agents should avoid other higher priority agents. Unlike some other research using hard priority, which might lead to the algorithm become not complete, I adopt the soft priority, namely penalization function, to guarantee the algorithm to find the feasible solution if the problem is a solvable MAPF problem. Since the  $A^*$  algorithm is well known, I only give the part that involves in the priority of the agents, see Eq. 3.4,

$$g[u]_{tmp} = g[v] + P_{a_i} \cdot \sum_{k=1}^L StepCost \quad (3.4)$$

where  $u$  is the neighbour point and  $v$  is the current point.  $L$  is the total layers of the map, and  $P_{a_i}$  is the priority value of the agent  $i$ .

### 3.4.3 Higher Level Search

Although the papers about CBS and its variants showed the success rate of the algorithms under various specific scenarios, they are performed with a time limit of at least 1 minute. Considering that some machines might not maintain their speed and emergencies may happen, I believe a feasible algorithm for the construction site should give a command to all the participants within 5 seconds even the order can be just wait.

---

**Algorithm 2 High Level Realtime Search**


---

**Input:**  $G(v, t)$ ,  $\tilde{s}$ ,  $\tilde{z}$  from predefined map information in yaml, original from visual or Lidar recognition

**Output:** solution for all agents, total cost

*Initialisation :*

- 1:  $start.constrains \leftarrow \Phi$
- 2:  $start.solution, start.cost \leftarrow bid\_Astar.search()$
- 3:  $allConflicts \leftarrow findConflicts(start.solution)$
- 4: insert start to OPEN

*LOOP Process*

- 5: **while** OPEN not Empty **do**
  - 6:    $P \leftarrow$  the node with lowest solution cost
  - 7:   **if**  $counter > threshold$  **then**
  - 8:      $P.remove(allConflicts)$  (remove some agents)
  - 9:   **end if**
  - 10:   **if**  $Validate(P) = 1$  (no conflicts found) **then**
  - 11:     return  $P.solution, P.cost$
  - 12:   **end if**
  - 13:    $C \leftarrow$  first conflict  $(a_i, a_j, v, t)$  in  $P$
  - 14:    $allConflicts.append(C), counter ++$
  - 15:   **for**  $a_c$  in  $C$  **do**
  - 16:      $ND \leftarrow P$
  - 17:      $ND.constraints.append(a_c, v, t)$
  - 18:      $ND.solution.update(uni\_Astar.search())$
  - 19:      $ND.cost.update(SIC(ND.solution))$
  - 20:     Insert  $ND$  to OPEN
  - 21:   **end for**
  - 22: **end while**
- 

Concretely, I let some machines have priority to move to their goal while the others should wait for a while or find a midway destination in case when the task is too complicated for a realtime response. Or the algorithm suggests to reduce the total amount of machines on site if necessary. Different from the original CBS algorithm, I add four different strategies in case the algorithm cannot find a solution for all agents within 5 seconds. The basic ideas of these acceleration

process are reducing the complexity of the task by solving the problem step by step and described as follow,

- Directly remove the agents causing most conflicts, including initial conflicts and update conflicts, found by line 3 and line 13 in Algorithm 2. Since the computation time of the CBS-based algorithm depends on the conflicts number, reducing the trouble makers can surely speed up the searching process.
- Let the machines have the same moving direction to move first. Obviously, conflicts can be avoided if all the agents move in the same direction.
- Randomly select some machines in independent sub-regions to move first.
- Let the machines having lower estimated cost move first.

The individual path will be compared at the high-level search to find out the conflicts among the agents. I use bidirectional  $A^*$  algorithm with the heuristics proposed by [83] to create the initial path of each agent in order to enhance the realtime performance, and afterward utilize unidirectional  $A^*$  to update the individual path of each agent since only unidirectional  $A^*$  can deal with the waiting process. In the BIM system, during the searching process, the algorithm saves the conflicts position and the corresponding agents. As mentioned, in case the algorithm cannot solve the planning problem due to too many conflicts, the algorithm will remove some agents and then replan the paths of the rest agents. In this fashion, I ensure that the known MAPF problems can be solved in a timely manner. If emergence occurs and the algorithm cannot solve the new task in time, I can easily and quickly locate the trouble maker. Apparently, the optimization direction here is not only to ensure a short calculation time, but also to make as many machines as possible move at the same time.

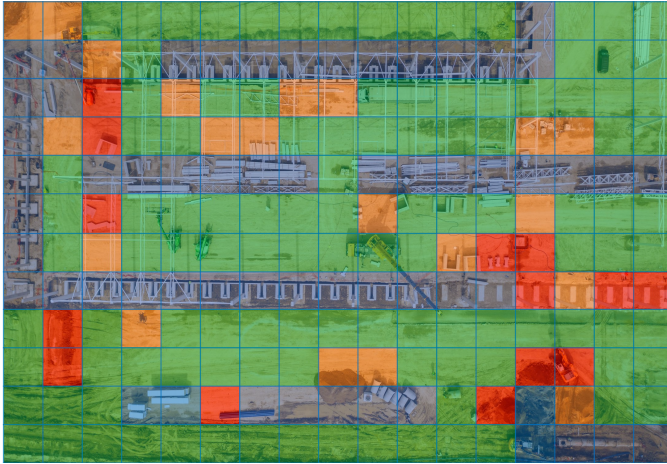
### 3.5 Experiment on Real Working Sites

As a consensus of the research in the field of graph theory, although a conclusion about a specific map cannot guarantee its effectiveness on another map, the closer the map, the closer the effect. In order to show the benefit of the introduction of MAPF in the mobile machines industry, I validated my algorithm on five typical real working sites. Concretely, they are a relatively open field with 20 or 50 agents, an open field with many obstacles, a two-side working site connected by a bridge or narrow corridor, and a typical mining site. Since the map will also be shown in the path results as background, here I only demonstrate how a map will be processed to give the prior information for successfully pathfinding on the first map and third map to avoid repetition. The maps shown in Fig. 3.2 and Fig. 3.7 are on the same site at different times. Obviously, Fig. 3.2 is the earlier stage while Fig. 3.7 shows the later stage as the construction process proceed since more facilities are there. In my experiments, the dimensions of the proposed maps are  $20 \times 13$  and  $17 \times 12$ . For the sake of simplicity, I also assume the velocity of all the machines are constant; however, it is surely easy to achieve the situation that the machines have quite different speed since I can use the fastest speed as a reference and allow the slower agents occupy more than one grid at the same time or vice versa.

Obviously, the faster the project, the faster the construction site changes. This indicates the difficulties of using a pre-calculated path planning for a construction site. In this study, I divided the grids into different regions with respect to whether the road is easy to be passed through, the slope of the road, and whether the place is safe.

**Table 3.1:** Weight table. The weights I use to describe the complicated terrain of construction sites

Layers	Weight
Roughness	[1, 5, 9]
Slope	[1, 5]
Safety	[1, 15]



**Figure 3.7:** Map example drawn by Liu [81]. Another map based on a real construction site which has more narrow corridor.

Here I show the solution finding time on CPU core i7 4720HQ@ 2.6 GHz. Because of its reasonable price at the end of 2020, it is suitable for large-scale commercial use. To reduce the randomness, I did the experiments 50 times and gave the average finding time, and the average number of conflicts occur to analyze the conflicts and thus show the rationality of my optimization. Notice that I rounded the numbers to one decimal place if any.

Before I analyze the results of my experiments, I summarize the basic ideas of the algorithm I used. Similar to the original CBS algorithm, my MAPF algorithm also adopts a two-level search, where the upper level finds the conflicts among the agents and the lower level search the best path for individual agents. The lower level finds the path first and sends the initial proposal to the upper level. Afterward, the upper level will check whether the planned path has a or many conflicts with others. In case there are no conflicts, the center commander, an AI system, agrees to the preliminary proposal to become the final solution of MAPF and all agents are allowed to execute this solution. In other cases, if there have some conflicts, the upper level will find out the conflicts and send this information as constraints to the lower level to avoid the conflicts. To generate

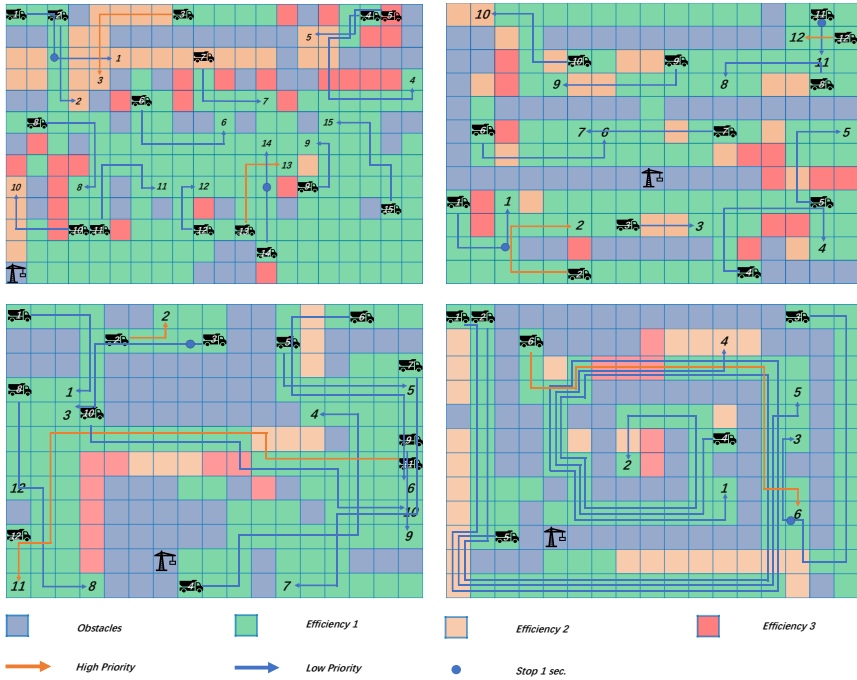


the initial individual path for each machine faster, I use bidirectional search. And then update the individual path if the upper level finds out a conflict with unidirectional  $A^*$  algorithm. The algorithm tries to modify the solution having the lowest cost, which guarantees the solution to be optimal. In the experiments, I do not assume what the participants are, nor do I assume its working process to ensure the generalization of my method.

## 3.6 Experimental Results

In this section, I demonstrate the planned path for each map in Fig. 3.8. As we can see, my algorithm successfully finds out the optimal paths considering the priority of the machines, i.e., the path with the lowest cost considering the main criterion, for all the tested maps. The algorithm commands the machines to drive directly to the goal, find a bypass, or just wait for others first to pass through.

In Tab. 3.3 on page 46, I give the computational time to find out the optimal solution. I divide the searching time into initial search and the following update process. In the first phase, the computational time is no more than 0.1 seconds on the tested maps. In case that the MAPF task is easy, i.e., the counteraction and potential conflicts among the agents are rare, the update process can also be done very fast. As we can see, the total duration to get the optimal solution is within 0.2 seconds for the scenario of agents on map 1. However, in other cases, such as the MAPF tasks on map 2 and map 4, although the period to offer the initial path proposal has no significant difference, the total duration is quite different. Concretely, the tasks on map 2 and map 4 need about 6.8 and 10.4 seconds to be solved. For such tasks, the solution can of course be found inside the BIM system before the machines execute their order saved in the schedule file. In the ideal case, the computational time for finding the solution is not the critical thing. However, in a real application, it is normal that the participants do not act on time when something urgent happens; thus, the ability to replan the path quickly is particularly important rather than let all the machines wait in place. As shown in Fig. 3.11 and Fig. 3.12, the update process is a dominant part of



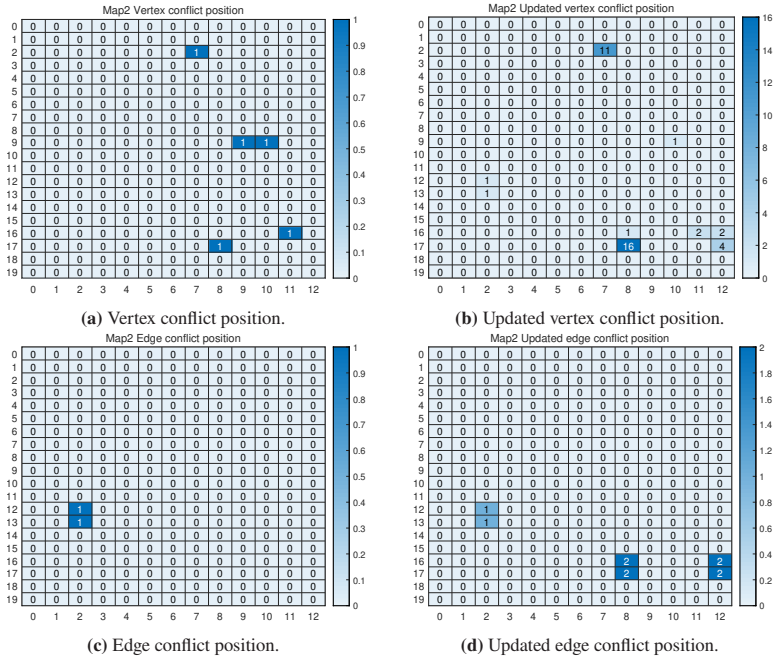
**Figure 3.8:** The planned path for each map. In the clockwise direction, the subfigures demonstrate the final solutions, including the best path for each machine for maps 1, 3, 4, 5. The left bottom point is defined as the original point (0,0), and the horizontal axis is the first axis. The layout of map 2 is the same as map 1; however, the difference is that there are more agents on map 2. Due to its huge amount of information, I give the planned schedule in Tab. 3.2 instead of using figures.

the whole searching process. Also, with the data shown in Tab. 3.3, comparing the duration of the initial search and the following update process, it is easy to conclude that reducing the update process is the main optimization direction to make my algorithm faster.

In this chapter, I demonstrate the optimization mechanism on the MAPF task on map 2 and map 4 to avoid wordy; however, I confirm the conclusions I make are also in line with the other maps I tested. The results on map 2 and map 4 are shown in Tab. 3.3. The optimization depends on the stage of the construction

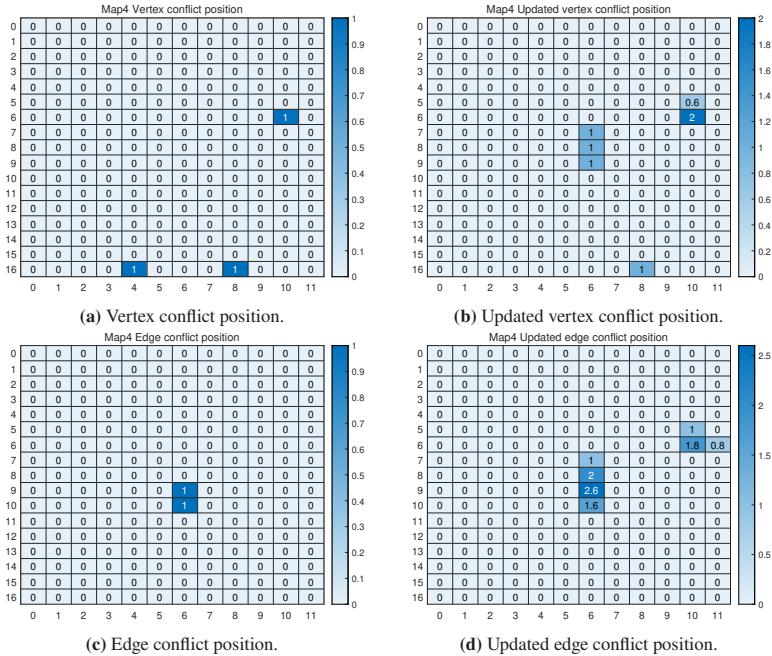
**Table 3.2:** The schedule of MAPF task on map 2. In this task, the algorithm should plan the path for 50 agents

agent name	start	1	2	3	4	5	6	7	8	9	10
agent1	[0, 12]	[1, 12]	[2, 12]	[2, 11]	[2, 10]	[3, 10]	[4, 10]				
agent2	[2, 12]	[2, 11]	[3, 11]	[4, 11]							
agent3	[6, 12]	[7, 12]	[8, 12]	[9, 12]	[10, 12]						
agent4	[17, 12]	[16, 12]	[16, 12]	[16, 11]	[15, 11]	[15, 10]	[15, 9]				
agent5	[18, 12]	[17, 12]	[17, 12]	[16, 12]	[16, 11]	[15, 11]	[15, 10]				
agent6	[14, 11]	[15, 11]	[16, 11]	[17, 11]	[18, 11]						
agent7	[0, 10]	[1, 10]	[2, 10]	[2, 9]							
agent8	[8, 10]	[8, 10]	[9, 10]	[9, 9]	[10, 9]						
agent9	[10, 10]	[9, 10]	[9, 9]	[8, 9]							
agent10	[0, 9]	[1, 9]	[2, 9]	[2, 8]	[2, 7]	[3, 7]					
agent11	[2, 9]	[2, 9]	[2, 8]	[2, 7]	[3, 7]	[4, 7]					
agent12	[6, 9]	[6, 10]	[7, 10]	[8, 10]							
agent13	[12, 9]	[12, 8]	[12, 7]								
agent14	[19, 9]	[19, 8]	[18, 8]	[17, 8]	[16, 8]	[15, 8]	[14, 8]				
agent15	[5, 8]	[6, 8]	[7, 8]	[7, 7]	[7, 6]	[8, 6]	[9, 6]	[10, 6]	[10, 7]	[10, 8]	[11, 8]
agent16	[15, 8]	[16, 8]	[16, 7]	[17, 7]	[17, 8]	[17, 9]					
agent17	[19, 8]	[18, 8]	[17, 8]	[16, 8]	[15, 8]	[14, 8]	[13, 8]				
agent18	[0, 7]	[1, 7]	[2, 7]	[2, 6]	[2, 5]						
agent19	[1, 7]	[2, 7]	[3, 7]	[4, 7]	[4, 6]	[4, 5]	[4, 4]	[3, 4]			
agent20	[6, 7]	[7, 7]	[7, 6]	[7, 5]							
agent21	[7, 7]	[7, 6]	[8, 6]	[9, 6]	[9, 5]						
agent22	[13, 7]	[13, 6]	[14, 6]	[14, 5]							
agent23	[15, 7]	[15, 6]	[15, 5]								
agent24	[17, 7]	[17, 6]	[17, 5]	[16, 5]							
agent25	[4, 6]	[5, 6]	[6, 6]	[6, 5]							
agent26	[6, 6]	[6, 5]	[6, 4]								
agent27	[1, 0]	[2, 0]	[3, 0]	[4, 0]							
agent28	[7, 1]	[8, 1]	[8, 0]								
agent29	[13, 0]	[14, 0]	[15, 0]	[16, 0]	[17, 0]	[18, 0]	[19, 0]				
agent30	[17, 6]	[17, 5]	[18, 5]	[19, 5]	[19, 4]						
agent31	[19, 6]	[19, 5]	[19, 4]	[19, 3]							
agent32	[1, 5]	[0, 5]	[0, 4]								
agent33	[4, 5]	[4, 4]	[4, 3]								
agent34	[7, 5]	[8, 5]	[8, 4]	[8, 3]							
agent35	[8, 5]	[9, 5]	[10, 5]	[10, 4]							
agent36	[13, 5]	[12, 5]	[11, 5]	[11, 4]							
agent37	[17, 5]	[17, 4]	[16, 4]	[16, 3]							
agent38	[6, 4]	[6, 3]	[6, 2]								
agent39	[12, 4]	[12, 3]	[11, 3]								
agent40	[14, 4]	[14, 5]	[13, 5]	[12, 5]	[12, 4]	[12, 3]					
agent41	[19, 4]	[18, 4]	[17, 4]								
agent42	[0, 3]	[0, 2]	[1, 2]	[2, 2]							
agent43	[8, 3]	[8, 2]	[9, 2]	[9, 1]							
agent44	[15, 3]	[15, 2]	[15, 1]	[14, 1]	[13, 1]						
agent45	[18, 3]	[19, 3]	[19, 2]	[19, 1]							
agent46	[0, 2]	[1, 2]	[1, 1]	[2, 1]	[3, 1]	[4, 1]					
agent47	[4, 2]	[4, 1]	[5, 1]	[6, 1]	[6, 0]						
agent48	[9, 2]	[10, 2]	[11, 2]	[12, 2]	[13, 2]						
agent49	[13, 2]	[14, 2]	[15, 2]	[16, 2]	[17, 2]						
agent50	[19, 2]	[18, 2]	[17, 2]	[17, 1]	[16, 1]	[15, 1]					



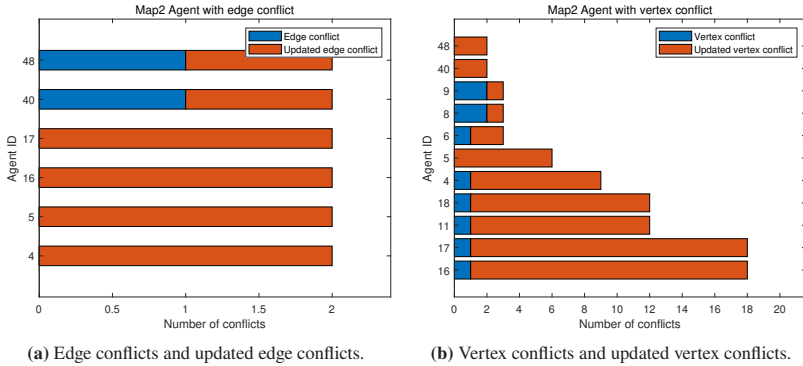
**Figure 3.9:** The place where the agents intend to pass through and the resulting conflicts on the second map. As it shown from the points allocation, initial conflicts have a great relevance to the conflicts occurring during the conflicts avoidance process. Notice that I did the experiment 50 times and the conflicts shown in these figures are the average number of these 50 experiments.

site. In the early state before the site is built up in reality, engineers have more freedom to optimize. To accelerate the computation, two methods are proposed for the early stage. The first idea is to modify the unreasonable part of the construction sites. In this fashion, the throughput of the construction site can be improved. Fig. 3.9 represents the positions where the algorithm commands the machines to pass through but encounters conflicts with other construction machines. As aforementioned, I consider two kinds of conflicts in this study since they are more in line with the construction site, namely edge position conflicts and vertex conflicts, respectively. Fig. 3.9(a) and (c) demonstrate the conflicts found by initial bidirectional search. Since the map is weighted, the best path is usually

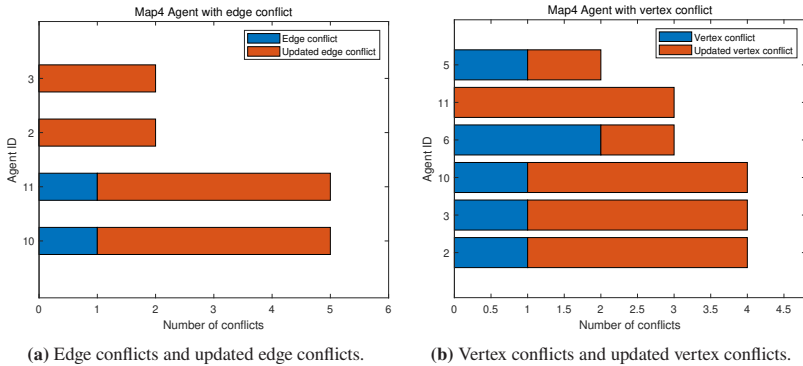


**Figure 3.10:** The place where the agents intend to pass through and the resulting conflicts on the fourth map. As we can see from the points allocation, initial conflicts still have a great relevance to the conflicts occurs during the conflicts update process.

unique; this is partly proved by the fact that the conflicts number found by initial bidirectional search is a multiple of the time I did the experiments. However, the unique best solution increases the possibility of generating conflicts. Taking the MAPF task on map 2 for example, as shown in Fig. 3.9, it is shown that there are three regions that have more conflicts than others. Concretely, they are the region including vertex (16, 8) and (17, 8), the region including vertex (2,7), as well as the region including vertex (17,12) and (16,12). Correspondingly, based on the intended movement of the agents around these positions, the algorithm points out that the vertex (16,9), (17,9), (3,8), and (15,12) shall be modified to have similar characteristics as its surrounding. For instance, the road condition of the vertex (16,9) and (17,9) shall be changed into good condition from bad condition



**Figure 3.11:** Statistics of the conflicts made by corresponding agents on map 2. Here blue histogram denotes the conflicts found by initial bidirectional search, and the orange histogram shows the conflicts solved while updating the solution with unidirectional search.



**Figure 3.12:** Statistics of the conflicts made by corresponding agents on map 4.

since their surroundings have good condition. The computational time is then dramatically reduced to about 0.31 seconds and seems to be the most effective method to reduce the solution finding time. The results are demonstrated in Tab. 3.3.

The other idea is to remove the most troublemaker in the MAPF task. As we know, the capacity of each construction site has a physical upper limit. No matter how excellent the algorithm is, too many participants will eventually lead to a decline

in overall performance. Compared to the first method, which has a potential drawback that there might be some reasons that optimization of the working site is not always feasible, removing a conflicts-causing agent can be used whenever needed. In the case of the MAPF task on map 2, agent 16 is removed according to Fig. 3.11 so that the computational duration reduces to roughly 0.97 seconds. Considering the holistic productivity is only marginally affected since there are still 49 machines that can work well in the working site, and the shorter duration endows the whole system the capability to deal with the emergence on site, this method is recommended if the construction site cannot be modified.

In the BIM system, I consciously made sure that the computational duration was within an acceptable period. However, this cannot guarantee all the potential conflicts for these MAPF tasks were removed. For the example of the MAPF task on map 2, in case that agent 16 did not catch up with the planned schedule and had a two-time step delay, the duration for replanning the solution for the whole fleet went to more than 5 seconds, even the construction site was modified in the early stage. Since I had already optimized the searching process in BIM so that the duration is shorter than 0.5 seconds and the only thing changed here was agent 16, I could quickly draw a conclusion that the expanding computational time was due to agent 16. Afterward, the algorithm checked the path of other agents and found out a new temporary destination for agent 16 to avoid conflicts. Concretely, the new goal for agent 16 shall be the vertex (15,7), and the computational time was then only 0.42 seconds. Notice that agent 16 is not allowed to stop at its original place since it blocks the only way for agent 17 and 14 to reach their goal. In case that machine 16 totally lost its mobility, the algorithm will ask every participant to stop and wait for the human intervention. In contrast, If it is not on time due to external distributions, it will wait for other agents to arrive and then continue to its original target.

In the MAPF task on map 4, I use the same methods to optimize the computation time in order to demonstrate my solution's generalization capability when the terrain is more complex and there are fewer agents. As shown in Fig. 3.10, 2 regions have more conflicts than others, i.e., the region including vertex (8,6) and vertex (9,6), and the region including vertex (6,10). Same as the method, namely

**Table 3.3:** The average algorithm searching duration (s). "None" denote that there is no need to optimize since the response is already quick enough

Map Nr.	Map1	Map2	Map3	Map4	Map5
Upper layer bidirectional A* algorithm duration	0.0302	0.0844	0.0178	0.0400	0.0322
Update algorithm duration	0.1468	6.8192	0.1680	10.4038	1.3689
Theoretical cost before optimization	648.9523	1317.0138	492.3904	435	1066
Total duration by layout optimization	None	0.3058	None	0.1746	None
Total duration by agent optimization	None	0.9726	None	0.1731	None
Total duration emergence	None	0.4200	None	0.1489	None



construction site optimization, used in the previous case, the vertex (5,10) is indicated, which shall be modified from an obstacle to a road with good conditions. In addition, the road condition of the vertex (9,5) and vertex (10,6) shall be changed into good condition from bad condition. After this optimization, the computation time is greatly reduced to roughly 0.17 seconds. For agent optimization, the most conspicuous troublemakers in this task shall be removed, which are agent 10 and agent 11, as shown in Fig. 3.12. Here I removed the agent 10 so that the computation duration reduces to about 0.17 seconds. Although the optimization results of these two methods in the MAPF task on map 4 are almost the same, I recommend layout optimization because only 12 agents were deployed in map 4. If an agent is removed, the overall productivity will be reduced more significantly compared to the previous scenario on map 2. In case that agent 10 did not run perfectly according to the planned schedule and had a delay of one-time step at the beginning, the computation duration for replanning can also exceed 5 seconds, even if the layout of the construction site was optimized with the first method. The algorithm gave a new goal for agent 10, which shall be the vertex (7,4), and the computational time was afterward 0.15 seconds. According to the above results, the algorithm is also proven as effective for the task on map 4.

### **3.7 Advantages of My Methods**

In this study, my approach enables many machines to work simultaneously inside of the working site. Firstly, the method helps civil engineers to arrange the construction site before the site is setup. My approach points out the positions where conflicts occur among the machines and thus indicates the place worthy of being modified. Moreover, my algorithm also helps the engineers to determine the reasonable number of machines in a working site, on the premise of using advanced algorithms. In addition, my algorithm schedules a conflict-free solution for the agents so that the agents can move confidently without hesitation. Last but not least, since the emergencies are inevitable, I design the system to replan

the path solution in a very short period compared to the SOTA MAPF algorithm with only slightly increase the non-optimality of the solution.

## 3.8 Conclusion

In this chapter, I presented an efficient and effective algorithm to calculate the path of a fleet of machines on a construction site. Considering the complicated terrain of a construction site, I endow my algorithm with the ability to handle the weighted maps. By testing my method on five different and diverse maps, my method successfully found the best path for a fleet including participants with different importance. By solving the MAPF problem for a construction site from both algorithmic and construction layout perspectives, I showed the benefits of my hybrid method, especially in reducing the computational time to handle emergencies. Based on my results, modify the unreasonable part is the most efficient fashion to speed up the searching process. Also, removing the agents which cause the most conflicts is always viable and can dramatically reduce the searching time but slightly reduce the whole productivity.

## 4 SLAM for Machines on a Smart Working Site<sup>1</sup>

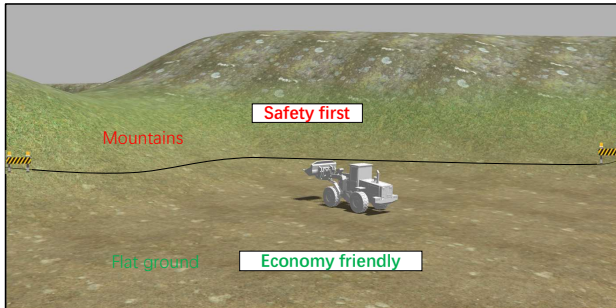
The decision of a reasonable strategy for machines on a working site is not only determined by its intrinsic signals, but also very strongly by environmental information, especially the terrain. Due to the dynamically changing of the construction site and the consequent absence of a High Definition (HD) map, the SLAM offering the terrain information for construction machines is still challenging. Current SLAM technologies proposed for mobile machines are strongly dependent on costly or computationally expensive sensors, such as RTK GPS and stereo cameras, so that commercial use is rare. In this chapter, I proposed an affordable SLAM method to create a multi-layer grid map for the construction site so that the machine can have the environmental information and be optimized and directed accordingly. Concretely, after the machine passes by, I get the local information and record it. Combining with positioning technology, a map of the interesting places of the construction site can be then created. As a result of my research gathered from Gazebo, I showed that a suitable layout is the combination of 1 IMU and 2 differential GPS antennas using the unscented Kalman filter, which keeps the average distance error lower than 2 m and the mapping error lower than 1.3% in the harsh environment. The SLAM technology proposed in the chapter provides the cornerstone to activate the pathfinding solution proposed in the previous chapter.

---

<sup>1</sup> Except some tiny modifications, all the figures, text, and results of the presented work in this chapter have been published in my preprint publication [66]. My contribution to the paper is summarized as 100% in terms of conception and methodology, 90% of literature review, 50% of model building and simulation, 50% of results visualization, and 95% of formulation.

## 4.1 Introduction

The environment also has an essential influence on the performance of a fleet of working machines, i.e., to perform tasks both efficiently and safely, the construction machines shall be conducted by knowing their location and surroundings; thus, I proposed a method that can generate the map information surrounding the mobile machines only with commodity sensors so that provides the possibility to improve the system further. The basic idea of my approach is to generate the map information of the working site based on the vehicle position, rolling resistance, as well as road grade. Concretely, a special recursive least square with forgetting algorithm is used to record the road grade and the rolling resistance in realtime [84, 41]. These information will be saved together with the localization information. Consequently, after the machine passes by, it will record the information about that place. Since the mobile machines are driving repeatably for a special task, the method can be expected to work well even when the map information does not cover most of the working site. Fig. 4.1 illustrates the motivation of my approach.



**Figure 4.1:** Mobile machines perform tasks more efficiently or safer according to their location and surroundings information. The short-term goal of SLAM is to prevent construction machinery from always working in low-efficiency areas for safety reasons, whereas the long-term goal is to increase the productivity of the working site with the help of path planning. The chapter focuses on affordable SLAM technology for construction machines.

## 4.2 Problem Statement

Although the map information can also be obtained from satellite, it is impossible to get the valuable information, such as a HD map, only depending on remote sensing due to a construction site's fast-changing environment. Also, the sensors can be quite noisy. Especially, they will be further exacerbated on a working site. The sensors, such as Global Positioning System (GPS), Inertial Measurement Unit (IMU), and odometry sensor will have higher measurement errors in case of the harsh environment. In addition, different construction machines have different drivetrain system, which makes a predefined motion model difficult. For instance, since mobile machines may work outside the coverage of base stations, the GPS signal can only achieve nearly 10 m accuracy [85] without signal correction. Last but not least, for passenger cars, the longitude error might not have such a negative effect as the latitude error since further measurements can be adopted to avoid the collision. In the case of construction machines, both errors shall be treated equally.

## 4.3 Goal of This Chapter

The goals of this chapter are twofold. The first goal of the chapter is to find out the most suitable sensor arrangements for construction machines. For accurate estimation of the position of machines, rather than only trust the measurement from one sensor, I fused a series of different kinds of sensors with the help of sensor fusion technology, derived from Kalman filter [86], to achieve better accuracy. Afterward, the second goal is to create a map with the environment condition by combining the surface resistance, road grade, and position information in realtime. Thanks to this map, further optimization of operation strategy and path planning can be realized. Although I suggest to measure the surface resistance and road grade by recursive least square with multi forgetting factors, the map-building approach I proposed can also be combined with other methods with other kinds of sensors, such as using ultrasonic proposed by Jung [87].

## 4.4 Related Works

### 4.4.1 Sensors

The combination of several sensing systems so that they can compensate the technical shortcoming of each other is well-known in the field of autonomous systems [88, 89, 90]. Therefore, there are a series of researches focusing on sensor fusion. Here I first summary the commonly used sensors for simultaneous localization and mapping (SLAM). Although I agree the introduction of the HD map can surely increase the accuracy of the localization, I do not consider this technology for the construction machines due to the dynamically changing of the construction site, as mentioned in [91].

#### 4.4.1.1 GPS

Global Navigation Satellite Systems (GNSS) such as GPS, GLONASS, BeiDou, and Galileo rely on at least four satellites to estimate global position at a relatively low cost. Typical standalone GPS average accuracy ranges from few meters to above 20 m [85] due to ionospheric delay, multipath effects, ephemeris & clock errors, and Geometric Dilution of Precision (GDOP). To improve the accuracy, one of the most used techniques is Differential GPS (DGPS), which utilizes measurements from an onboard vehicle GPS unit and a GPS unit on a fixed infrastructure unit with a known location. Here the known fixed infrastructure unit is called reference station, which calculates the local error in the GPS position measurement periodically. The onboard vehicle GPS units then use this correction to adjust their own GPS estimation. According to [92, 93, 94, 95, 96], an average accuracy in the range of 1–2 m can be achieved, mainly depending on the distance between the vehicle and the base station. Another commonly used improvement is Realtime Kinematic (RTK) GPS, which estimates relative position by means of the phase of the carrier signal and can be expected to achieve centimeter-level accuracy. Notice that, both of them depend on a fixed base station with a known position nearby, through the principle of them is quite different. In Oct

2020, when I wrote the content of this chapter, RTK GPS is still an extraordinary expensive approach<sup>2</sup> and usually be used to define the ground truth position of vehicles . Some low-cost RTK GPS sensors, under 1,000 bucks, are designed with much lower receive frequency [97] and thus cause problems as vehicles driving fast. Thus, in most commercial uses, DGPS is preferable for reducing the cost. Therefore, in my research, I conservatively considered the accuracy of DGPS as 2 m, which is consistent with the normal performance of DGPS. Obviously, as the performance, especially the accuracy, of the GPS increases, my mapping approach will also have better performance consequently.

#### **4.4.1.2 IMU**

Inertial Measurement Units (IMUs) are integrated electronic devices that contain accelerometers, magnetometers, and gyroscopes. It can provide raw IMU measurements to calculate attitude, angular rates, linear velocity, and position relative to a global reference frame.

#### **4.4.1.3 Odometry**

Odometry is the most widely used navigation method for positioning; it provides good short-term accuracy, is inexpensive, and allows very high sampling rates. Odometry is based on simple equations, which hold true when wheel revolutions can be translated accurately into linear displacement relative to the floor. The main advantage of odometry is that all localization information comes from the vehicle itself so that this information is always available. Usually, it is the only localization information when other sensors are not able to provide data. Thus, a good odometry based localization system is always necessary, and it is usually the first step to localization [98].

---

<sup>2</sup> For example, the Trimble R10 costs 18,000 US Dollars on Alibaba.

## 4.4.2 Localization Technologies

### 4.4.2.1 Mobile Robotics

A series of researches using sensor fusion to achieve a highly accurate localization has been studied worldwide. The technologies about SLAM can be roughly divided into two parts: indoors and outdoors. For indoor localization, such as domestic robots [99], a GPS system cannot be used. However, the road is relatively flat, and thus only a two-dimensional map is needed. In contrast, when it comes to offroad navigation in rough terrain, the algorithms must be capable to handle three dimensions of the environment. After the success of Kalman filter [86], extended Kalman filter [100], and finally unscented Kalman filter [101, 102], the idea that a mobile robot which executes useful missions should be endowed with navigation ability has become a consensus. However, the selection of combining different sensors is from case to case different. For instance, Bento fused the data from ABS sensors and GPS for outdoor localization, based on extended Kalman filter [103], and Zhang integrated the information from GPS and IMU [104]. Also, Li used a camera instead of GPS to accomplish mean positioning errors of 75 cm [105]. In addition, Wolcott proposed a Visual Localization method within LIDAR Maps for Automated Urban Driving [106]. For the cost purpose, Ward studied the possibility to use radar to localize the vehicle's position and demonstrates that errors go to 27.8 cm laterally and 115.1 cm longitudinally by their approach in worst case [107]. To investigate the use of LiDAR for localization, Hata suggests using LiDAR to detect curbs and road markings to create a feature map of the environment and localize vehicles with the help of RTK-GPS and IMU within the map [108]. Another alternative is using ultrasonic sensors, proposed by Jung [87]. Interestingly, as the development of the IoT, more and more researchers are focusing on SLAM by cooperative localization techniques. The basic idea of this approach is to get crucial information even when the perception capability is affected by adverse weather or obstacles from infrastructure or other vehicles. For example, del Peral-Rosado showed the feasibility of 5G based localization technology [109], and Rohani utilized VANET to enhance GPS accuracy to 3.3 m mean level [110].



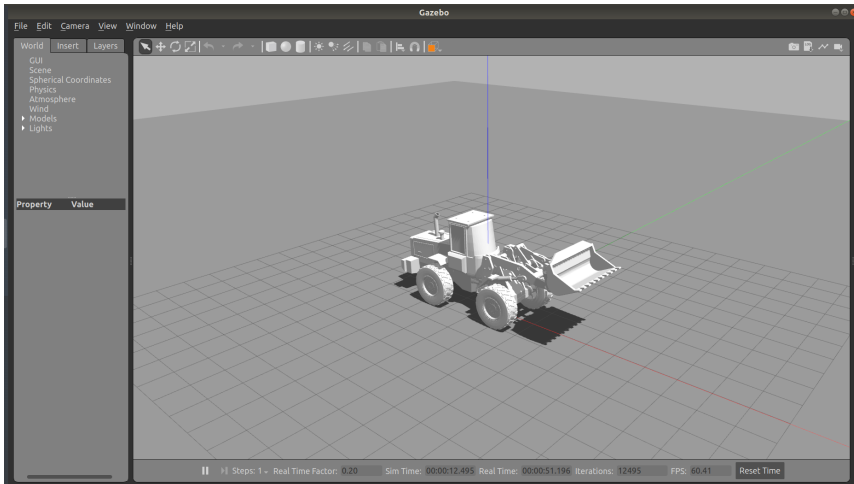
#### 4.4.2.2 Construction Machines

For mobile construction machines, the requirements for localization techniques can be different based on different use cases. Some machines may work in the underground, where the situation is similar to working in a tunnel, whereas others might work on an open-pit mining site. In underground mines, it was proposed to use the laser for extracting the wall positions, and dynamically generate a path from these laser data while considering variable offsets [111, 112]. In contrast, for the open-pit site, an autonomous wheel loader introduced by Gu [113] uses a set of sensors, including GPS and IMU for localization, and LIDAR, radar as well as a camera for obstacles capture and identification, ensuring it perceives surroundings accurately. Moreover, Xiang created a dataset for mobile machines detection from the view of a camera fixed on the ground [64], while Bang proposed a method recognizing the machines from a view of the drone [114]. In addition, the visual SLAM is proposed [115]. Besides that, V2X technology was introduced in the field of construction machines [67, 68]. In 2020, Xiang proposed to use WiFi to achieve the communication between different vehicles by introducing a realtime estimation method with respect to package loss and delay [116]. Afterward, the feasibility of using 5G for machines is also investigated by [62]. To avoid additional costs for the vehicles, smartphones show great potential to be utilized as a solution to complement the flaws of onboard ECU [117, 40].

In summary, similar to general autonomous vehicles, most automated mobile machines fuse information from onboard sensors such as IMU and GPS by using diverse sensor fusion technologies. Furthermore, camera, LIDAR, and radar are used to detect the environment on the construction site, to avoid obstacles, and to instruct the machines where to go. However, owing to the harsh environment and diversity of working sites, LIDAR and radar can be sensitive. In the recent future, wireless communication can also contribute to better localization of vehicles.

## 4.5 Model Building

The wheel loader used in the simulation was modeled in Solidworks and then imported into the Robot Operating System (ROS) to explore the approaches that should be used for mobile machines. Since the first goal of this chapter is to find out suitable sensor arrangements to accurately localize wheel loaders, I fused different arrangements of sensor data. Concretely, I used up to three IMUs and three GPSs in the simulation. Based on the characteristic of GPSs, three GPSs were fixed on the cab of the wheel loader. I then installed two IMU sensors under the front axle and other IMUs under the rear axle, based on the suggestion from Li [118]. Fig. 4.2 illustrates the wheel loader model I used in the Gazebo environment.



**Figure 4.2:** Wheel loader model in Gazebo: once the models had been developed in Solidwork, they were converted to Unified Robotic Description Format (URDF), using a 3rd party URDF conversion tool called “*sw\_urdf\_exporter*”, which allows for conveniently export SW Parts and Assemblies into a URDF file. Gazebo enables us to obtain sensors’ simulation such as IMUs, GPSs, encoders, cameras, and stereo cameras through *gazebo\_plugins*, which can be used to attach into ROS messages and service calling the sensor outputs, i.e., the *gazebo\_plugins* create a complete interface (Topic) between ROS and Gazebo.

As we know, the URDF is an XML file format used in ROS to describe all elements of a vehicle. URDF can specify the kinematic and dynamic properties of a single robot in isolation. To make my vehicle works properly in Gazebo, additional simulation-specific tags concerning the vehicle pose, frictions, inertial elements, and other properties have been added. The transform tree is shown in Fig. 4.3.

Each Link in URDF represents a rigid body. Also, according to the kinematic and dynamic model shown in Fig. 4.3, the wheel loader in the simulation is divided into several parts, including,

1. *base\_link* represents the rear part of the wheel loader, which is also the main coordinate frame of the simulated model in ROS, because most of the sensors are attached to the rear part and considered as child links.
2. *front\_link* represents the front part of the wheel loader, considered as the child link of the *base\_link*, and connected by a revolute joint.
3. *wheel\_link* represents the wheels of the wheel loader. Apparently, the two front wheels are attached to the front part, and the two rear wheels are attached to the rear part.
4. *gps\_link*: the GPS devices, which fixed on the roof of the wheel loader.
5. *imu\_link*: the IMU devices, which two of them fixed under the front part of the wheel loader and other two fixed under the rear part of the wheel loader.

In this project, the wheel loader receives GPS data from an onboard GPS sensor plugin with its latitude and longitude; however, the GPS data provided by the GPS plugin cannot be directly applied to the fusion of the sensor data, so coordinate system conversion for GPS data is required. For the simulation, I set a transform for each GPS that converts the vehicle's world frame coordinates, i.e., the frame with its origin at the vehicle's initial position, to the GPS's UTM coordinates, the same as [119], as

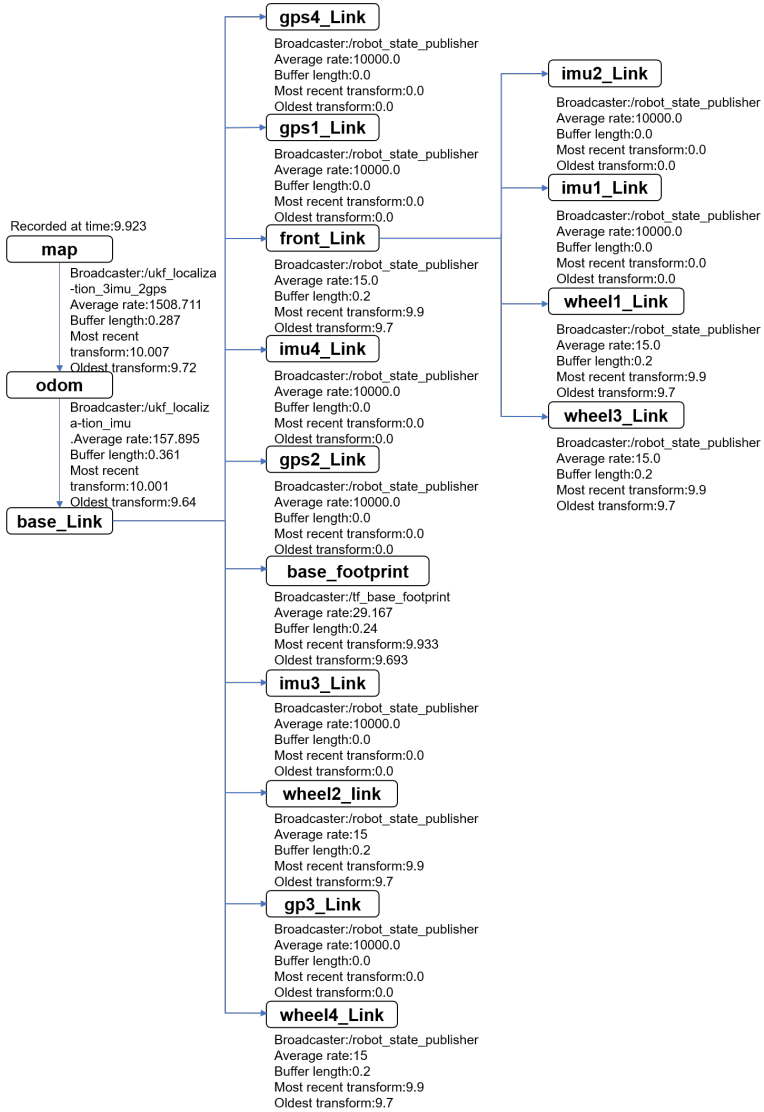


Figure 4.3: The dynamic system simulated by URDF file on ROS.

$$\mathbf{T} = \begin{bmatrix} c\theta c\psi & c\psi s\phi s\theta - c\phi s\psi & c\phi c\psi s\theta + s\theta s\psi & x_{UTM_0} \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -c\psi s\phi + c\phi s\theta s\psi & y_{UTM_0} \\ -s\theta & c\theta s\phi & c\phi s\theta & z_{UTM_0} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

where  $\phi$ ,  $\theta$ ,  $\psi$  denote the vehicle's initial UTM-frame roll, pitch, and yaw.  $c$  and  $s$  denote the cosine and sine functions, respectively.  $x_{UTM_0}$ ,  $y_{UTM_0}$ , and  $z_{UTM_0}$  are the UTM coordinates of the first reported GPS position. After that, the GPS signal is then transformed into the vehicle's world coordinate frame, *odom*, by

$$\begin{bmatrix} x_{odom} \\ y_{odom} \\ z_{odom} \\ 1 \end{bmatrix} = \mathbf{T}^{-1} \begin{bmatrix} x_{UTM_t} \\ y_{UTM_t} \\ z_{UTM_t} \\ 1 \end{bmatrix} \quad (4.2)$$

In my simulation the ROS package “*robot\_localization*” from Moore [119] was used, including a “*navsat\_transform*” node, which provides functions to convert between various coordinate frames and integrate GPS data. It provided a transformation function that allows the conversion between GPS frame, expressed in latitude and longitude, and vehicular coordinate. This process shall be carried out for each GPS independently.

In practical applications, GPS signals can be received infrequently. Yet the localization technology must maintain state estimation even when some of the vehicles' signals are absent. Therefore, the performance of the filters when GPS signals infrequently arrive in the system shall be evaluated. Taking this problem into consideration, I used a ROS node built by Li [118] to filter the collected GPS signals such that GPS data is unavailable for 1 second once every 10 seconds. In case multi GPS sensors are used, the signal failure might not happen at the same time. I aim to observe how the filter and my approach behave with different sensor configurations when some GPS signals go wrong.

### 4.5.1 Sensor Fusion for Localization

For localization of the wheel loader in the simulation environment, I used EKF and UKF node in “*robot\_localization*” [119]. On the one hand, this package has no limitation for the number of sensor inputs, which just in line with my construction machines’ requirements. On the other hand, a concrete motion model is not needed so that this method can easily be used on both excavators and wheel loaders with different drivetrain solutions. In “*robot\_localization*”, the filter’s state will be driven forward by a standard 3D kinematic model derived from Newtonian mechanics to calculate the vehicle’s motion, including position, velocity, and acceleration in three dimensions.

In the correction step, the measurement model integrates sensor data to update the predicted state. GPS provides position information, and wheel encoders provide velocity information for the correction. Moreover, the orientation, velocity, and acceleration information are updated from IMUs via gyroscopes and accelerometers. Furthermore, the process model and measurement model also need to add a noise covariance matrix,  $Q$  and  $R$ , respectively. The noise matrices can generate uncertainty in the system. The process matrix contributes to the overall uncertainty in the algorithm, which adds to the process model. Intuitively, a large value in the  $Q$  matrix means a considerable uncertainty in the process model, causing the system to have greater confidence in the measurement data. In the current implementation, the process noise matrix was set as a diagonal matrix. The state variables, which are directly measured by the sensors, such as  $x$   $y$  position by GPS and orientation by IMUs, were set relatively small. The variables, which were not directly measured, could be updated from the measured data. The measurement covariance matrix  $R$  corresponding to the confidence in the sensor data. Similarly, the greater the noise in the elements of this matrix, the less confidence in the measurement data. The measurement covariances are derived by the sensors noisy.

In addition to the inaccuracy of filters, outliers are also an important source of error. In my simulation, I assume that the measurements have Gaussian distributions. Although sensors follow the normal distribution as setting, improbable,

and extremely noisy measurements can appear due to the high fidelity of ROS. To counteract this problem, I used Mahalanobis distance to detect outliers and thus overcome the consequent adverse effect. After this, the filtered data were used for state correction. Concretely, the Mahalanobis distance is calculated as a product of the from filter processed vector to find out the outliers,

$$D_M(\vec{x}) = \sqrt{(Z_t - \hat{Z}_t)^T A^{-1} (Z_t - \hat{Z}_t)} \quad (4.3)$$

where  $A$  is the covariance matrix.

## 4.5.2 Sensor Fusion Methods

State estimation is one of the most critical issues in many autonomous applications. Having an accurate state estimation, the machines can be effectively navigated in the environment and thereby making optimal decisions for specific purposes. For instance, to reach a target destination, it needs to know its current state, which consists of position, velocity, acceleration, and heading to execute the right maneuvers correctly. Since sensors are susceptible to noise and imperfections introducing uncertainty to the measurements, the filter's goal is to fuse all the available sensor data, as well as the vehicle's own dynamics to obtain a more precise estimation of the vehicle's state. As mentioned, two necessary extensions of the Kalman filter are presented, notably the EKF and UKF.

The filters are modeled to improve the positioning's accuracy by compensating for the disadvantages of the different sensors. As we know, GPS provides relatively accurate positioning, but the signal's availability remains a problem, especially in urban and mountainous environments. This determines that the results of using GPS alone are usually not satisfactory. Also, IMUs use a combination of accelerometers and gyroscopes to measure linear accelerations and angular velocities, respectively. By estimating the position relative to its initial position, the trajectory can be calculated by these information of the wheel loader as the vehicle travels. For sure, there is also a common problem with IMUs, namely the

accumulated errors. To avoid accumulated drift and provide global positioning, the estimated position shall be corrected by using other sensors, e.g. GPS. Without a doubt, a GPS/IMU system is successful in increasing the accuracy beyond standalone GPS or IMU capabilities.

#### 4.5.2.1 Extended Kalman Filter

Although linear Gaussian systems are abundant, most systems, in reality, are non-linear. Also, they often do have Gaussian noise. Wrong assumptions about the system can lead the Kalman filter to diverge and provide estimation with very high errors. Consequently, multiple extensions have been developed to deal with various scenarios encountered in practice. One of the famous variations is the EKF, where it deals with non-linearity by approximating a linear equivalent before performing the required filtering sequence. The idea of the EKF is that if the system is close to linear for short periods, using its linear approximation will then not yield large errors.

Through linearizing the basic equations from Welch [102], the following equations are obtained:

$$x_{k+1} \approx \tilde{x}_{k+1} + B(x_k - \hat{x}_k) + Ww_k \quad (4.4)$$

$$z_k \approx \tilde{z}_k + H(x_k - \hat{x}_k) + Vv_k \quad (4.5)$$

where  $x_{k+1}$  and  $z_k$  are the actual state and measurement vectors,  $\tilde{x}_{k+1}$  and  $\tilde{z}_k$  are the approximate state and measurement vectors.  $\hat{x}_k$  is an *a posteriori* estimate of the state at step  $k$ , random variables  $w_k$  and  $v_k$  represent the process and measurement noise.  $B$  is the Jacobian matrix of partial derivatives of  $f(\bullet)$  with respect to  $x$ ,  $W$  is the Jacobian matrix of partial derivatives of  $f(\bullet)$  with respect to  $w$ ,  $H$  is the Jacobian matrix of partial derivatives of  $h(\bullet)$  with respect to  $x$ ,  $V$  is the Jacobian matrix of partial derivatives of  $h(\bullet)$  with respect to  $v$ .

An essential feature of the EKF is that the Jacobian  $H_k$  in the equation for the Kalman gain  $K_k$  serves to correctly propagate only the relevant component of



the measurement information [102]; the linearization error is always exist as the function is nonlinear. Because increasing the sampling time and reducing the nonlinearity of function are not always viable, error-state EKF is proposed to counteract the adverse effect. The basic idea of error-state EKF is to reduce the distance of the linear approximation from the operating point; instead of linearization of the nominal state, it handles the error state.

### 4.5.2.2 Unscented Kalman Filter

When the state transition and observation models, that is, the predict and update functions  $f$  and  $h$  are highly nonlinear, the EKF can give particularly poor performance. This is because the covariance is propagated through the linearization of the underlying nonlinear model. By contrast, the UKF uses unscented transform instead of linearization in the prediction and correction steps to make the estimation. As the first step, the decomposition of the covariance matrix shall be computed and the sample points shall be carefully selected, for instance, here I selected  $2L+1$  points, described as,

$$\begin{aligned}
 \mathcal{X}^0 &= \bar{x} \\
 \mathcal{X}^i &= \bar{x} + (\sqrt{(L + \lambda)P_x})_i & i = 1, \dots, L \\
 \mathcal{X}^{i+L} &= \bar{x} - (\sqrt{(L + \lambda)P_x})_i & i = 1, \dots, L
 \end{aligned}
 \tag{4.6}$$

where  $\bar{x}$  is the selected mean sample point and  $\lambda$  is set as  $\lambda = 3 - N$  for Gaussian probability density function. After that, the sigma points will be propagated through the nonlinear function as,

$$\tilde{X}^i = f(\mathcal{X}^i) \quad i = 1, \dots, 2L
 \tag{4.7}$$

where  $f(\bullet)$  is the motion model function and  $\tilde{X}^i$  is the predicted position based on the motion model. As the final step of predicted process, the predicted mean and covariance should be calculated.

$$a^i = \begin{cases} \lambda/L + \lambda, & i = 0 \\ \lambda/2(L + \lambda), & \text{otherwise} \end{cases} \quad (4.8)$$

$$\check{X} = \sum_{i=0}^{2N} a^i \check{X}^i \quad (4.9)$$

$$\check{P} = \sum_{i=0}^{2N} a^i (\check{X}^i - \check{X})(\check{X}^i - \check{X})^T + Q \quad (4.10)$$

Here  $Q$  denote the noisy of this process. In the correction step, I firstly calculate the predict measurement with the sigma points,

$$\hat{y}^{(i)} = h(\check{X}^i, 0), i = 0, \dots, 2N \quad (4.11)$$

where  $h(\bullet)$  is the predicted measurement model, considering the process noise:  $\check{L}\check{L}^T = \check{P}$ . Then, I get the mean and covariance of predicted measurements,

$$\mathcal{Y} = \sum_{i=0}^{2N} a^i \mathcal{Y}^i \quad (4.12)$$

$$\check{P}_y = \sum_{i=0}^{2N} a^i (\mathcal{Y}^i - \mathcal{Y})(\mathcal{Y}^i - \mathcal{Y})^T + R \quad (4.13)$$

where  $R$  is the noisy. Based on the previous result, I can compute the cross-covariance and Kalman gain, and then get the corrected covariance and mean.

$$P_{xy} = \sum_{i=0}^{2N} a^i (\check{X}^i - \check{X})(\mathcal{Y}^i - \mathcal{Y})^T \quad (4.14)$$

$$K = P_{xy}P_y^{-1} \quad (4.15)$$

$$\hat{P} = \check{P} - KP_yK^T \quad (4.16)$$

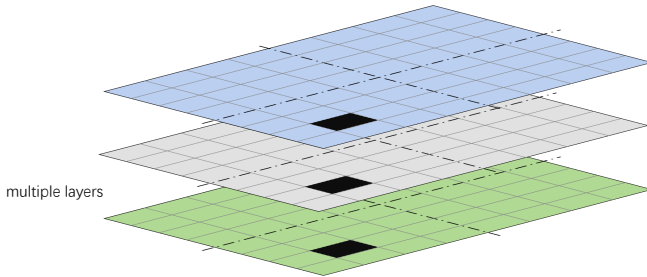
$$\mathcal{X}^i = \check{X} - K(y^0 - \mathcal{Y}^i) \quad (4.17)$$

where  $y^0$  is the measurement result with respect to  $\check{X}$ ,  $\mathcal{X}^i$  is the final results from UKF. As can be seen, UKF does not need Jacobian matrix so that it can achieve a better performance in estimation. Usually, the computation effort can be slightly higher than the other methods, making carefully select sensors configurations meaningful.

### 4.5.3 Realtime Map Plotter

Inspired by the research from Fankhauser [82], a map can include many layers to store different types of data information. Thus, to develop a realtime plotter of the construction site according to ground condition, I use a multi-layer grid-based map, which divides the environment into uniform cells. Fig. 4.4 illustrates the multilayered grid map concept, where each cell data is stored on the congruent layers. In this project, since resistance and grade of the road are the most of importance information for the construction machines, I adopt a two-layer grid map. However, I show the method to save three different information in this chapter. Concretely in this chapter, the map is divided into small cells, whose resolution is 1 m per cell. Notice that I use a much smaller cell than the cell used in the previous chapter. This is on one hand because the measurement did by foot SLAM cannot cover so large areas in one step. On the other hand, a finer grid map describes the terrain information better. Apparently, the map used in the previous chapter can be processed from the finer map providing in this chapter. As I discussed in the previous section, I use GPS/IMU fusion Kalman filter algorithms to locate the mobile machine on the construction site.

As can be seen in Fig. 3.4, to describe the ground condition of construction sites, I use the value of each cell to represent the information of the ground situation. An



**Figure 4.4:** A grid map created by SLAM. My approach uses multilayered grid maps to store data for different types of information. Concretely, every grid saves a  $1 \times 3$  matrix including location information and resistance or grade, depending on which layer it is. A grid with site information will be created after the vehicle passes by. The map is saved as a  $2 \times m \times n \times 3$  tensor, where  $m$  is the max displacement in the x-direction, while  $n$  is the max displacement in the y-direction. In case a grid does not be occupied once by the vehicle, it will be marked as NaN to denote the unknown regions.

exemplar layer that holds the data of a grid-based map has been shown in Fig. 3.4 in the previous chapter. Obviously, although I only demonstrate the approach with two layers, it is relatively easy to extend the third layer in case that uphill or downhill is vital, as the third layer is responsible for recording the heading of the vehicle.

Based on the previous study [84], both resistance and grade of the road can be gathered in realtime. Thus, I assume that the ground resistance and slope are known after the mobile machine passed by. When the mobile machine passes through each grid cell, ground information will be added to the corresponding grid of the two layer-grid-maps. Concretely, the plotting algorithm combines the localization results from the Kalman filter and the grade and resistance information from the recursive least square algorithm. To implement the plotting algorithm in ROS, a *node* was created in ROS that can subscribe to the localization results from the Kalman filter node and gather the current information from ground truth maps with the assumption that resistance and grade of the road can be estimated or measured well [118].

## 4.6 Vehicle Simulation Scenarios in ROS and Gazebo

To test the feasibility of the map plotter, different road conditions on a construction site was first defined. ROS is a mature and flexible framework for robotics programming, providing the required tools to easily access sensors data, process that data, and generate appropriate responses for the robot's actuators. Due to these characteristics, ROS is a perfect tool for many types of research on modern robotics. After all, a mobile machine can be considered as just another type of robotics, so the same types of programs can be used to develop advanced construction machines. In this chapter, a construction site showed in Fig. 4.5 was simulated. According to this real construction site, five different ground resistances in the simulation environment are defined, according to different ground conditions. Moreover, two areas according to different slopes. Concretely, the different regions were distinguished based on road material.

1. **Gravel surface:** Gravel is a loose aggregation of small, variously sized fragments of rock. It has a wide range of applications in the construction industry. Therefore, gravel surface road is very common on the construction site. The rolling resistance coefficient of gravel surface is considered as 0.020.
2. **Sand surface:** Sand is a type of naturally material that is of a loose, granular, fragmented composition, consisting of particulate things such as rock, coral, shells, and so on. The rolling resistance coefficient between the sand surface and mobile machine tires is 0.250.
3. **Dry dirt road:** A dirt road is a type of unpaved road made from the native material of the land surface, which is also very normal on the construction site. The typical rolling resistance coefficient of the dry dirt road is 0.040.
4. **Wet dirt road:** Same as a dry dirt road, the wet dirt road is also a typical road type in the working area. The typical rolling resistance coefficient of the wet dirt road is 0.060.

- 5. Dry concrete surface:** Dry concrete is a normal building material and the typical rolling resistance coefficient of the dry concrete road is 0.008.

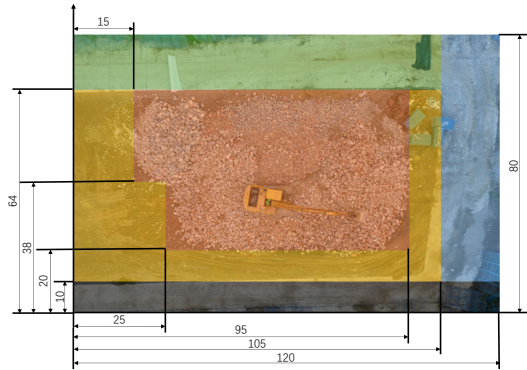
Besides, two different ground slopes were also defined appropriately. Since only a two-layer grid map is used in this research, I do not discriminate uphill or downhill.

- 1. Flat area:** The slope of the ground is near  $0^\circ$ . In the flat area, the mobile machines can move faster to increase working efficiency or reduce the reserved dynamics to let the components work in an more economic region, with only a little concern of safety.
- 2.  $15^\circ$  slope area:** The slope of this area is near  $15^\circ$ . In this area, in contrast, the mobile machines should pay more attention to the safety.

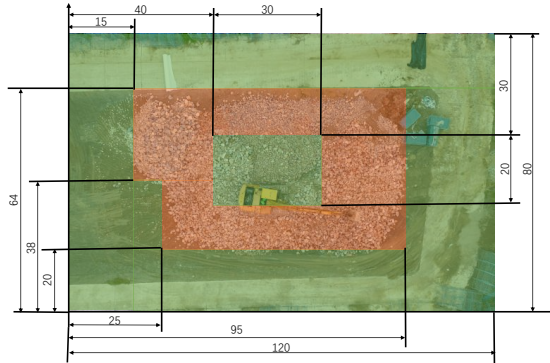
For simulation, the green area was defined as the dry concrete surface in the ground-truth resistance map. The red area represents the gravel surface. The blue area represents the sand surface, the black area, and the yellow area represent dry and wet dirt roads, respectively. With this premise, the ground truth map upon this construction site is drawn. Same as in the ground-truth resistances map, the ground slope map is also defined, where the green area represents the flat area and the red area represents the  $15^\circ$  slope.

In this project, the *plot\_node* [118] was written with Python and OpenCV library to visualize the plotted map. To test the feasibility of a realtime plotter in simulation, a *ground\_truth* node [118] is used in ROS, which provides a ground truth position of the simulated mobile machine in Gazebo. When the mobile machine moves to a certain position, the system uses the ground truth position to determine the rolling resistance coefficient and the road grade, and then uses the Kalman filter filtered position to plot the corresponding information in the grid-based map.

In ROS and Gazebo, the build-in plugins provide many adjustable parameters that can be used to adjust the devices' performance. To get closer to reality, the performance parameters of GPSs and IMUs was set in the simulation according to real GPS and IMU devices. To get the best sensor configuration, different sensor



(a) Example construction site divided into five areas according to different ground resistances.



(b) Example construction site divided into two areas according to different slopes.

**Figure 4.5:** The ground truth map with dimensions. The simulation environment I used in Gazebo was modeled based on a real construction site, and the parameters are selected according to material characteristics. Since simulating a small construction site may cause system error and thus lack plausibility, I augmented this real construction site's dimensions in Gazebo.

configurations for different algorithms were implemented. Each group of sensor

configurations were simulated under the same condition by rosbag, and the results were output simultaneously.

## 4.7 Experiment and Results

### 4.7.1 Localization Results

To explore the most suitable sensor arrangements of the Kalman filter for construction machinery, the results from different sensor configurations with different methods were compared. The concrete sensor arrangements in this project are shown in Tab. 4.1.

Group	Sensor ( 0 = deactivated, 1 = active )							KF
	GPS 1	GPS 2	GPS 3	IMU 1	IMU 2	IMU 3	Encoder	
1	0	0	0	1	0	0	1	EKF
2	1	0	0	1	0	0	1	EKF
3	1	0	0	1	1	0	1	EKF
4	1	0	0	1	1	1	1	EKF
5	1	1	0	1	0	0	1	EKF
6	1	1	0	1	1	0	1	EKF
7	1	1	0	1	1	1	1	EKF
8	1	1	1	1	1	1	1	EKF
9	0	0	0	1	0	0	1	UKF
10	1	0	0	1	0	0	1	UKF
11	1	0	0	1	1	0	1	UKF
12	1	0	0	1	1	1	1	UKF
13	1	1	0	1	0	0	1	UKF
14	1	1	0	1	1	0	1	UKF
15	1	1	0	1	1	1	1	UKF
16	1	1	1	1	1	1	1	UKF

**Table 4.1:** Sensor configuration. In my simulation, I ignore the mirror difference caused by the slightly different installation position of sensors. Therefore, the various configurations are reduced from 64 to 16. Since odometry is robust and necessary for many applications, I do not consider the case without an encoder



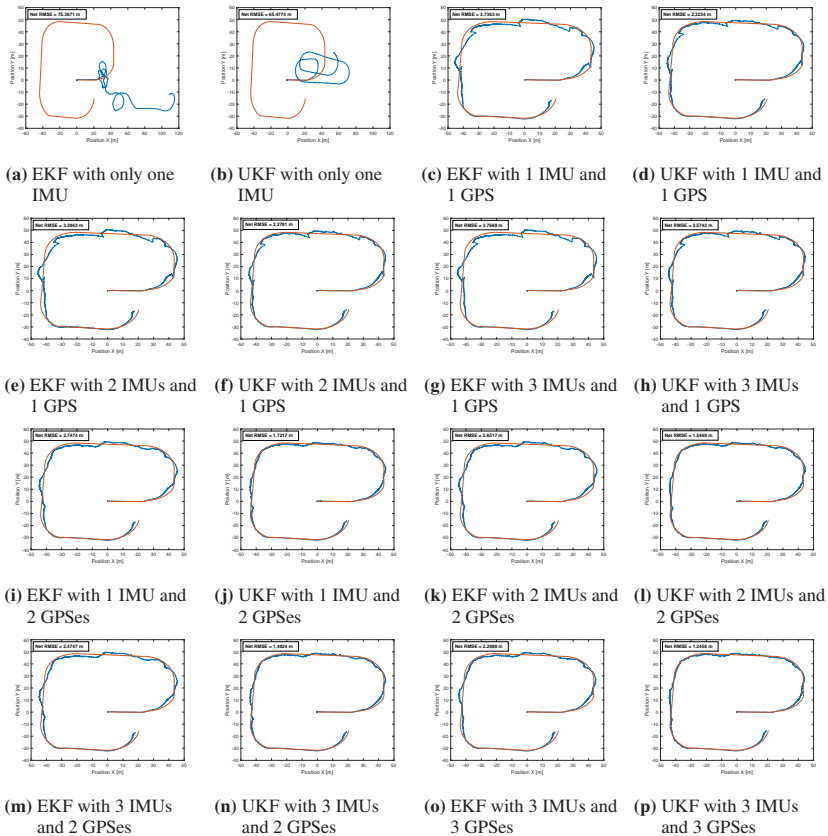
To evaluate the performance of the different variants of sensors and algorithms concerning accurately positioning, I controlled the wheel loader to drive on the previously defined construction site in Gazebo, and recorded the data from sensors at the same time. Afterward, the localization results of the different methods are compared to the ground truth. Here I use the Root-Mean-Square Error (RMSE) as a quantitative indicator of the error to assess the pose estimation results.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\tilde{S}_i - \hat{S}_i)^2}{n}} \quad (4.18)$$

Where  $\hat{S}$  is the vector including estimated x and y position,  $\tilde{S}$  is the denotes ground truth x and y position, n is number of all estimated samples, and the footnote i denotes the  $i^{th}$  time step.

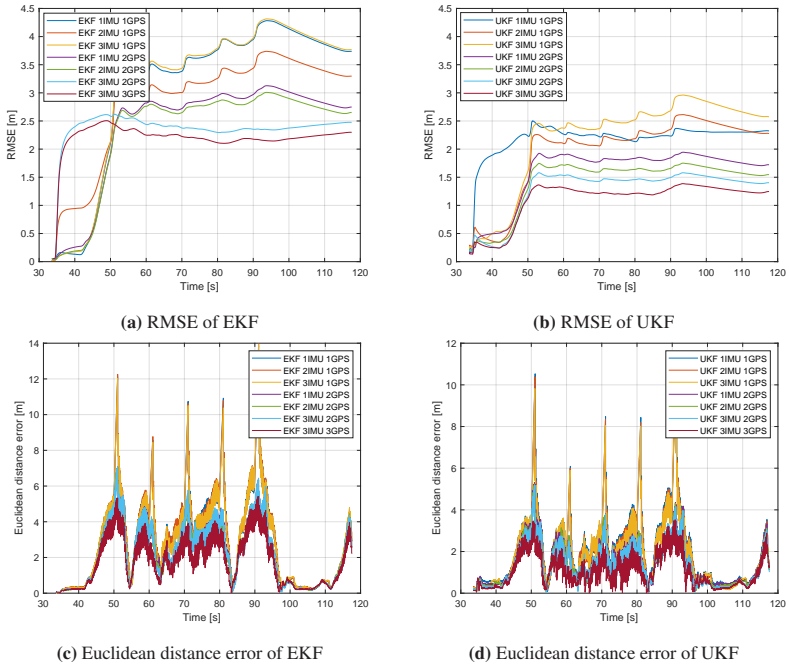
Fig. 4.6 shows the wheel loader estimated trajectories given by different approaches and the ground truth trajectory, where the red lines are the ground truth trajectories that the vehicle passes, and the blue lines are the estimated position of the vehicle. Since only EKF and UKF are capable of handling the nonlinear problem, I compare the results obtained using EKF filtering and the UKF filtering technique with different sensor arrangements. Notice that, odometry sensor is always used though I do not explicitly mention it. Apparently, the UKF performs better than the EKF, which is also in line with the conclusion from most studies. Generally speaking, with GPS fusing in the estimation, the accuracy improves drastically. Also, since the GPS may lose signal every 10 seconds, an additional GPS sensor can surely increase the position accuracy. In contrast, more IMUs can only slightly improve the accuracy of positioning. As can be seen, the IMU + Odometry estimation method yields inferior performance no matter with EKF or UKF. Once the IMU reports an inaccurate heading, the errors will be accumulated, causing the measured position drifts further away from its true position as the wheel loader travels.

Fig. 4.7 shows a comparison of localization error between the different approaches with respect to time. As shown in Fig. 4.7, the RMSE and the euclidean distance



**Figure 4.6:** The IMU + Odometry estimation method yields for both EKF and UKF the RMSE over 70 m. Except for the IMU + Odometry methods, the other results can be divided into four performance levels according to the error scale. The worst level is EKF with 1 GPS, where the RMSE is about 3 ~ 4 m. The second level is EKF with 2 GPSes and EKF with 3 IMUs and 3 GPSes. In these cases, the RMSEs are about 2 ~ 3 m. A better level is UKF with 1 GPS and 1 IMU, where the RMSE of the third level can be achieved about 2 ~ 2.5 m. Finally, my experiment's best class is UKF with 2 GPSes and 1 IMU, and UKF with 3 IMUs and 3 GPSes, which reduce the RMSE to about 1.2 m.

error of each sensor arrangement can be obtained, indicating that the EKF has more significant tracking error than UKF. It happens because the linearization through its Jacobian is an approximation, and the kinematic model of the wheel



**Figure 4.7:** Quantitative evaluation of different methods. Here (a) and (b) show the accumulative error, i.e., RMSE, while (c) and (d) demonstrate the current Euclidean distance error of each sensor arrangement. There are some noticeable instantaneous position changes every ten seconds due to infrequent GPS signal loss. Generally, UKF shows better performance than EKF for both RMSE and euclidean distance error.

loader is highly nonlinear. To get a more intuitive and accurate description of the error for each sensor configuration and method, Tab. 4.2 is created to demonstrate the RMSE in detail.

As aforementioned, the GPS signal will be lost for one second every ten seconds. Thus, there are some noticeable instantaneous position changes every ten seconds. The loss of GPS signals clearly causes these jumps. As can be seen, for sensor configuration with only one GPS, the leap of the localization error as well as the variance values are more sharply. Obviously, with the number of GPS increases, the jumps are diminished and therefore become more acceptable.

**Table 4.2:** Localization errors of each approach

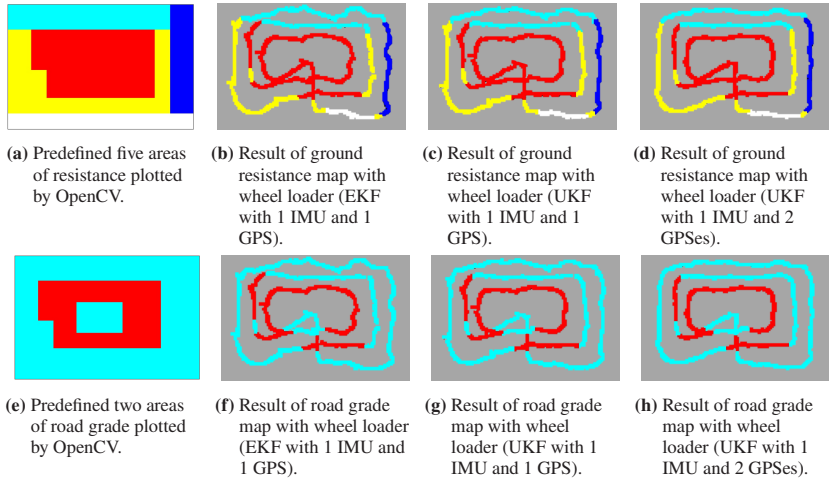
<i>Group</i>	<i>RMSE (m)</i>	<i>Net RMSE (m)</i>
1 (EKF 1 IMU)	(69.122, 30.039)	75.3671
2 (EKF 1 IMU 1 GPS)	(2.472, 2.802)	3.7363
3 (EKF 2 IMUs 1 GPS)	(2.172, 2.480)	3.2963
4 (EKF 3 IMUs 1 GPS)	(2.474, 2.840)	3.7660
5 (EKF 1 IMU 2 GPSes)	(1.830, 2.050)	2.7474
6 (EKF 2 IMUs 2 GPSes)	(1.778, 1.968)	2.6517
7 (EKF 3 IMUs 2 GPSes)	(1.894, 1.593)	2.4747
8 (EKF 3 IMUs 3 GPSes)	(1.794, 1.437)	2.2980
9 (UKF 1 IMU)	(56.586, 32.944)	65.4774
10 (UKF 1 IMU 1 GPS)	(1.654, 1.632)	2.3234
11 (UKF 2 IMUs 1 GPS)	(1.374, 1.817)	2.2781
12 (UKF 3 IMUs 1 GPS)	(1.611, 2.001)	2.5742
13 (UKF 1 IMU 2 GPSes)	(1.093, 1.330)	1.7217
14 (UKF 2 IMUs 2 GPSes)	(0.975, 1.200)	1.5460
15 (UKF 3 IMUs 2 GPSes)	(0.873, 1.098)	1.4024
16 (UKF 3 IMUs 3 GPSes)	(0.826, 0.933)	1.2458

Apparently, the simulation results show that UKF is a better approach using data collected by onboard sensors of the wheel loader in gazebo environments. Intuitively, more sensors represent higher accuracy. However, the results show that an appropriate number of sensors can achieve acceptable accuracy at a lower cost. As can be seen in Tab. 4.2, the RMSE of UKF with 1 IMU 2 GPSes is 1.72 m, and UKF with 3 IMUs and 2 GPSes is 1.40 m. With additional 2 IMUs and 1 GPS, the RMSE is only slightly reduced by 0.3 m. More importantly, the maximal error is strongly diminished by one additional GPS, whereas continually increasing the sensor number does not further reduce the error proportionally. Of course, according to different application scenarios, different sensor configurations shall be chosen. For my application scenario, UKF with 1 IMU and 2 GPSes has sufficient accuracy and a better economy respecting sensor hardware cost and

onboard ECU computational effort. Thus, I suggest using this sensor configuration to locate the mobile machines and then develop the realtime map plotter.

## 4.7.2 Plotter results

As my ultimate goal is to create a map of the current working site in realtime based on localization technology so that corresponding optimization can then be achieved, the ground truth maps and estimated maps with different sensor configurations coupled with various algorithms are shown and compared in Fig. 4.8.



**Figure 4.8:** The ground truth and estimated maps. In the ground resistance map and road grade map plotted by EKF with 1 IMU and 1 GPS, the spikes caused by infrequent GPS are quite obvious. With additional GPS sensor fused in Kalman filter, the spikes improve a lot.

Since I use a two-layer grid map, both rolling friction coefficient and road grade are recorded and used to create the estimated maps. In this chapter, the ground information are saved in corresponding grid after the wheel loader passes by and identify the ground information such as friction and slope through the specific algorithm, e.g. recursive least square. As aforementioned, to locate the mobile

machine's position in a cost-efficient fashion, the configuration of the UKF with 1 IMU and 2 GPSes is adopted. Also, in order to compare the performance of the selected configuration, I draw the results of the EKF with 1 IMU and 1 GPS and the UKF with 1 IMU and 1 GPS as the control groups. The mispredicted points are calculated as described in Eq. 4.19,

$$E_{r,s} = \sum_{i=1,j=1}^{m,n} e_{i,j} = \begin{cases} 1, & \text{if } \neg(\hat{G}_{i,j} = G_{i,j}) \cap \neg(\hat{G}_{i,j} = NaN) \\ 0, & \text{if } (\hat{G}_{i,j} = G_{i,j}) \cap \neg(\hat{G}_{i,j} = NaN) \\ 0, & \text{if } \hat{G}_{i,j} = NaN \end{cases} \quad (4.19)$$

where  $\hat{G}_{i,j}$  denotes the estimated grid map information,  $G_{i,j}$  is the ground truth map information,  $E$  is the accumulated number of errors, the subscript  $r$  and  $s$  denotes resistance and slope map, respectively. Obviously, the goal is to minimize the percent of mispredicted points versus total predicted points, described as Eq. 4.20,

$$\min J_{r,s} = \frac{E_{r,s}}{T} \quad (4.20)$$

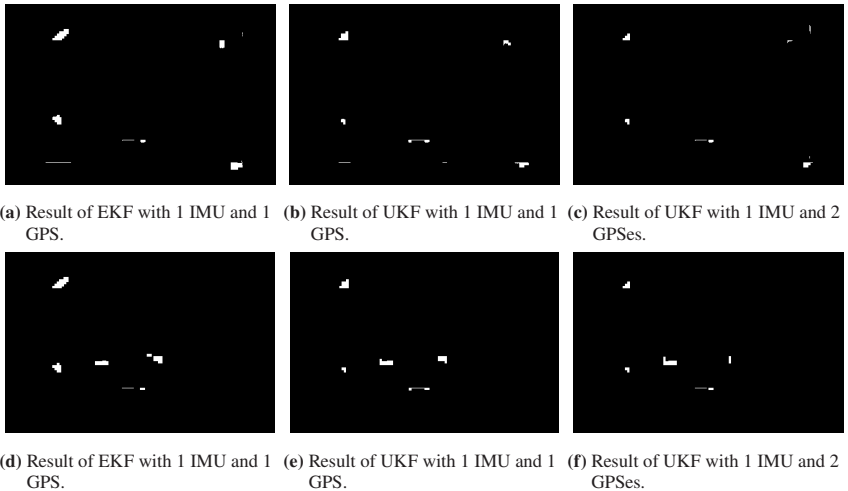
where  $T$  is the total estimated number, and  $J$  is the quantitative criteria to evaluate the accuracy of the plotted map.

The localization errors of each group are shown in Tab. 4.3, where the necessity of the introduction of the second GPS sensor, and the adoption of unscented Kalman filter is shown.

**Table 4.3:** Localization errors of each approach during the plotting

<i>Group</i>	<i>RMSE (m)</i> <i>(x,y)</i>	<i>Net RMSE (m)</i>
1 (EKF 1 IMU 1 GPS)	(2.584, 3.407)	4.4444
2 (UKF 1 IMU 1 GPS)	(2.003, 2.314)	3.0603
3 (UKF 1 IMU 2 GPSes)	(1.336, 1.533)	2.0334

To evaluate the plotted maps' accuracy, Matlab was used to implement an algorithm [118] to compare the predefined area and the plotted path the mobile machine traveled. Fig. 4.9 graphically illustrate the mispredicted grid point with white color, where the mispredicted points are generally distributed in the marginal zone. This is because even the localization technology makes some mistakes; the problem is unlikely to cause an error as long as the vehicle is not at the very edge of different zones, indicating the robustness of this mapping idea of the construction site.



**Figure 4.9:** Difference between ground truth and the estimated map. Here I compare the predefined areas and the plotted path, where the white pixels are the wrong plotted grids. (a),(b),(c) are the Friction map results, and (d),(e),(f) are the road grade map results. UKF shows more accurate positioning capabilities than EKF, and with two GPSes fused in the Kalman filter, the wrong located grid is less than just with one GPS signal.

After calculating the wrong located grids and all the plotted grids and according to the Eq. 4.19 and Eq. 4.20:

1. For EKF with 1 IMU and 1 GPS: an error rate  $J_r = 2.389\%$  of the ground resistances map and an error rate  $J_s = 2.258\%$  of the slope map can be obtained.
2. For UKF with 1 IMU and 1 GPS: an error rate  $J_r = 1.482\%$  of the ground resistances map and an error rate  $J_s = 1.682\%$  of the slope map can be obtained.
3. For UKF with 1 IMU and 2 GPSes: an error rate  $J_r = 0.997\%$  of the ground resistances map and an error rate  $J_s = 1.223\%$  of the slope map can be obtained.

## 4.8 Conclusion

In this chapter, I proposed an approach to creating a multi-layer map of the construction site in realtime so that the environmental information can be taken into account to improve vehicle efficiency and safety further or contribute to the path planning of mobile machines on the construction site. Considering the common phenomenon in reality mentioned by other researchers, such as noisy sensors and infrequent signal loss, the simulation environment was setup in Gazebo with a ROS package based on a real construction site. According to my tests in Gazebo by implementing a series of sensor configurations, I found that the configuration that 1 IMU and 2 GPSes with encoder using UKF has the best for overall performance with respect to accuracy and cost. By comparing the estimated maps drawn by map plotter and the predefined maps, the errors are only 1.0% and 1.2% for road resistant force and grade, separately. Thus, I believe that the developed plotter can be used to save the road condition in realtime within a reasonable error range to offer the terrain and location information to the multi working machines pathfinding algorithm.



## 5 Motion Prediction of Manned Working Machines<sup>1</sup>

The tasks on a working site can be quite diverse. Although a strong tendency to introduce autonomous systems into mobile machines, I think it is still challenging to transfer a whole working site into a fully autonomous working site in the next decade. Thus, I endow AI concept with the capability to cooperate with human drivers regarding road occupation. For instance, wheel loaders even move during working unlike excavators which usually stay there. Also, a wheel loader is a very commonly used machine on working sites worldwide. Obviously, the motion of a wheel loader has a strong relationship with its working process. Thus, in this chapter, I proposed a series of Multivariate Time Series Classification algorithms, namely CRDNNs, which combine Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Dense Neural Networks (DNN), to precisely recognize the working process and thus the motion of machines. Compared to existing algorithms, the CRDNN with bi-directional LSTMs has the best accuracy, and the CRDNN with LSTMs has a comparable performance but much fewer training parameters. Based on my dataset including 119 truck loading cycles, my best neural network shows a 98.2% test accuracy. Afterward, I introduced the transfer learning and human-machine communication system to increase the generalization ability of the selected algorithm.

---

<sup>1</sup> Except some tiny modifications, all the figures, text, and results of the presented work in this chapter have been published in my publications [63, 64]. My contributions to the papers are summarized as 100% in terms of conception and methodology, 90% of literature review, 80% of code, 30% of data collection and labelling, 80% of results visualization, and 95% of formulation, for both of them.

## 5.1 Introduction

The intrinsic sensors in the machines offer robust signals and information about the machines. By assessing the Multivariate Time Series data, the working process of the machines can be acquired. However, the selection of the variables highly depends on the machine system. To date, there are two kinds of mature torque control for hydrostatic drivetrain solution. Unlike the secondary control concept that typically has one or more hydraulic accumulators to build up a constant pressure and controls the output torque by adapting the angle of the hydraulic motor, the primary torque control concept [120] controls the pressure in a closed circuit by changing the angle of the hydraulic pump based on a feedback system but without accumulators. In this chapter, I use the primary-torque-based concept as a reference to show the performance of the working process detection algorithm. In this chapter, I focus on the Y cycles detection algorithms.

## 5.2 Background

### 5.2.1 Wheel Loader

The wheel loader is a typical mobile machine used for moving earth. A typical working process is the so-called Y cycle. Concretely, the machine digs the heap and transfers the soil to a truck. During this process, the machine is usually driving in a trajectory similar to a letter Y. Fig. 5.1 illustrates this cycle.

The Y cycles are the most typical working process of wheel loaders. The performance during the Y cycles has a decisive effect on the holistic performance of the mobile machine. Also, the working process detection is used as a vital criterion to predict the intention of the drivers.

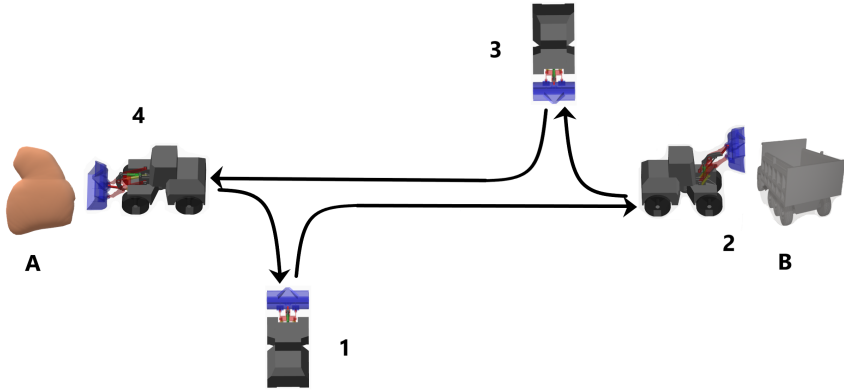


Figure 5.1: A typical truck loading process (Y cycle).

## 5.2.2 The Future Mobile Machines Drivetrain System

Murrenhoff has drawn a rule in [121] to classify the different kinds of control concepts, and concrete details are shown in Fig. 5.2.

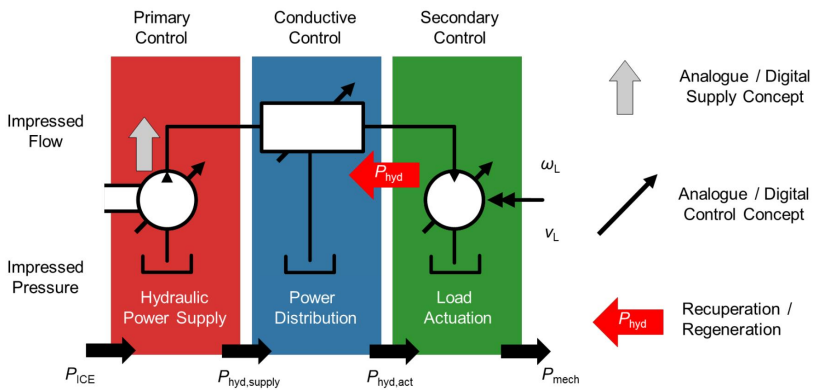


Figure 5.2: Segment of control concepts [121].

Currently, most mobile machines use the flow-based controlled drivetrain, which controls the vehicle speed by the volume flow pass the hydraulic motor and thus the vehicle velocity [122]. The advantages of such a drivetrain solution are due to the decoupling of the engine and the vehicle speed [116]. However, the efficiency of this concept can even lower than 10% [123] in many applications. Thus, many improvements based on these concepts have been drawn [124]. Based on my literature analysis, I find the research focus of the scientists in the field of mobile construction machines goes to the torque based controlled concept [125, 126, 127]. The initial proposal to introduce the torque control concept consists of higher holistic efficiency, flexible system architecture due to modulation, and more suitable for the employment of a hybrid system. Apparently, different control concept leads to different system layout and corresponding internal sensors selection. Since torque-controlled mobile machines may win the competition in the long term, I focus on the technologies that can be used on the torque-based mobile machines in this chapter, especially the primary torque-based control introduced by Bosch Rexroth AG in 2018 [120] for hydrostatic mobile machines. Since the measured variables in the primary torque concept can also be interpreted as secondary control concept, my algorithm can be principally adapted to the secondary controlled mobile machines with some further works.

One significant advantage of primary torque control is its high efficiency due to the successful introduction of central power management. The basic idea of central power management derives from the requirement that the power made available to the system should be precisely the same as the power consumed by the system. Besides, in case of power shortage, power management will cut down the power supply to the devices which have a lower priority [120]. To follow this basic idea, every component will compute the energy it requires first, and then the center power manager gathers the information, compares it with the disposable power of power source, and distributes the power to each requester [120]. On hydrostatic mobile machines, there is no restrain condition between engine rotation speed and vehicle speed. Thus, optimization of engine efficiency is possible. Generally, the engine speed is set to as low as possible considering the requested vehicle dynamics.

### 5.2.3 Working Process Detection Algorithms

Many scholars are interested in utilising machine learning to improve mobile machines regarding efficiency, maintenance, and usability. Especially in the field of working process detection, a series of methods have been drawn. Pohlandt has used two simple neural networks to predict and recognize the desired work process separately on electrical mobile machines [128]. According to his publication, he splits the time series of measured power into many small slip windows to train the simple neural networks. In his research, he found out neural networks might work for some simple cases [128]. Another research is from Brinkschulte who points out that the prediction accuracy with bagged trees may dramatically decrease when the drivers have different driving skills [129]. Besides machine learning algorithm, research by Nilsson introduced a method that combined several individually simple techniques including signal processing, state automation techniques, and parameter estimation algorithms. Based on 159 cycles, the accuracy is 93% [130]. In 2019, Keller made a case study for an excavator to classify the machine functions using decision tree with an accuracy of 99.97% without using slip windows [131]. In addition, Starke shows that Y cycle can be online recognized with hidden Markov Models (HMM) since HMM was widely used within the context of word recognition to deal with the temporal variability of text or speech [132], before 2012 [133]. Also, he pointed out that truck loading is a high variance problem and a simple algorithm should be used owing to the limited of on-board ECU [132].

## 5.3 Problem Statement

Based on my dataset and previous studies, I summarize the problems faced in this research.

First of all, the detection of arbitrary Y cycles is a high variance problem. Y cycles are different from site to site. The distance between heap and truck can be quite different. Moreover, drivers are also different. Some drivers have many

years of driving experience, and thus have become more aggressive. By contrast, some drivers are still novices who correct themselves during some processes. Last but not least, the materials for transport are different. Therefore, a complicated method is needed to handle the high variance.

Another problem is the limitation of the computing capacity of the ECU on mobile machines. Backpropagation consumes more CPU than forward propagation; thus, online learning usually entails the adoption of a shallow neural networks or an efficiency well-known machine learning algorithm, such as Support Vector Machine (SVM). Consequently, less intelligent learning ability is expected. In light of that, I dedicate to adopt an off-line method. Since a simple algorithm might not be really good for dealing with high variance problems, scientists in the fields of Natural Language Processing (NLP) usually use algorithms that combine many technologies. In the case of HMM, Vocal Tract Length Normalization (VTLN) and feature-space Maximum Likelihood Linear Regression (fMLLR) are used before HMMs [134]. Neural networks should also be combined together [135].

## 5.4 Why I Use RNN, LSTM?

The initial idea to use Long Short-Term Memory (LSTM) is inspired by analogy. Recurrent neural network has been proven to be a powerful tool in the fields of NLP in the past years [133]. One of the significant progress is the introduction of LSTM [136, 137]. More details about LSTMs can be found in [138]. In western countries, clauses are used extensively in writing, making sentences extremely and differently long. Splitting the sentence into many words and using a simple DNN with a certain number of input layer units, the translation performance is usually unsatisfying. Intuitively, different lengths of sentences make the selection of numbers of input units difficult. A deeper reason lies in the fact that, the simple neural network does not take the sequence of words appearing in the sentence into consideration. With limited input units, simple neural networks can only detect the current situation based on a specific past period. If the decisive information

occurred a long time ago, the AI must make its decision based on somewhat useless information causing no wonder a detection mistake. To overcome this problem, LSTM uses update and forget gates to make a shortcut for the vital information to help with the current decision. Akin to complicated sentences with clause, Y cycles can have very different lengths due to its transport process or workers of different proficiency. In light of that, LSTM shall solve this principally similar challenge.

Since my goal is using AI algorithm to detect the working process and thus improve the efficiency of mobile machines by regenerating, “future” information can be used to increase the detection accuracy. Generally, earthmover is first be accelerated in reverse direction and then decelerated after digging into a heap. The duration here implies that even though the algorithm does not recognize directly at the time Y cycle begins, it does not harm the regeneration performance as long as it detects the Y cycle slightly before the deceleration process. Therefore, I expect bi-directional LSTM to improve prediction accuracy.

Similar to HMM that may use some additional technologies to improve its performance, LSTMs also have better performance if CNNs and DNNs are cooperating together [139, 140, 141]. The advantages of the combination of CNNs, RNNs, and DNNs, which I call CRDNN in this chapter, are shown in the next sections.

## 5.5 Data for Deep Learning Algorithm

As aforementioned, I used a series of neural networks to recognize Y cycles. For the sake of simplification, it can be concluded that AI is a scientific method for recognizing pattern based on the data with which it has been trained. However, truck loading cycles can be quite different from each other regarding traveling length between heap and truck, driver’s skill level, materials, and the dimension of mobile machines. The training of an end-to-end neural network needs a vast dataset, which is still a cost-challenging task today. Instead, I proposed a multi-step approach to detect truck loading cycles. Instead of predicting the Y cycle directly,

I firstly predict the loading, the traveling, and the unloading processes since much less data is required for training. Furthermore, after neural networks output its prediction, I might use a modification measure to avoid obvious mistakes. In the light of that, I divide my processes into three sub-processes: vehicle travelling, loading, and unloading.

As I mentioned before, I would like to use a more complicated and therefore smarter neural network so that I adopt the off-line learning method to avoid the time-consuming backpropagation.

### **5.5.1 Data Acquisition and Allocation**

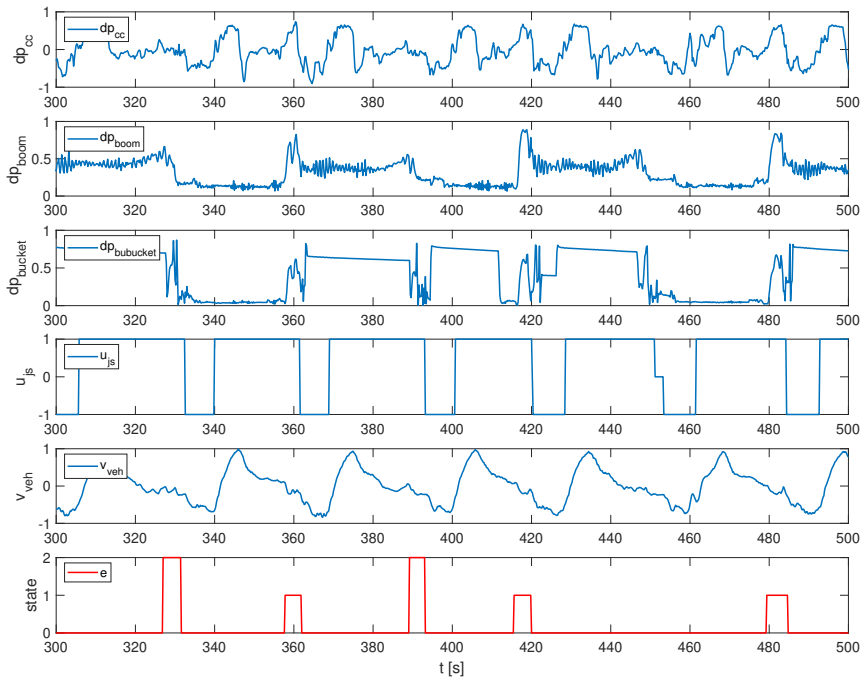
Data is the heart of deep learning. I split the dataset into training and test dataset, 80% and 20%, respectively. Besides, I consciously selected three different drivers and did the measurement in different days. Some measurements were done on a rainy day so that the density of material changes. Moreover, I changed the position of heap and truck to vary the length of the Y cycles. The test drivers were not given the information about what I was going to do so that they would behave the same in their daily operations. In short, I consciously increased the diversity of my dataset and tried to include more challenging cases in my dataset.

The data I fed into the neural networks were selected from the typical sensors on primary torque controlled mobile machines. Concretely, there are the pressure difference inside of the bucket, the vehicle velocity, the vehicle direction signal on joystick, the pressure difference inside of closed-circuit drivetrain, and the pressure difference inside of the boom. The sample rate is 50 Hz so that I would not overload ECU.

Totally, I have created a dataset with 119 Y cycles. 40 of them are gathered from an experienced test engineer, 30 of them are from a development engineer who has aggressive drive behavior, 20 of them are collected when the machine was not well tuned, 29 of them are measured by a senior manager who works many decades in the field of mobile machine. None of the data is collected by a complete layman



since I do not think it makes sense. Notice that I allocated the data collected as the machine with insufficient calibration process into training dataset since it can improve the robustness of my algorithm but not affect the test accuracy. The measurement dataset is labeled as shown in Fig. 5.3. In the real world, the data is often not perfect. That is, some people may mislabel a tiny portion of data. Therefore, I deliberately labeled some windows as travels through it is actually a loading or unloading process to check the robustness of my algorithms.



**Figure 5.3:** Normalized measurement data with label. It is important to highlight that I consciously mislabel the data around 450 s to test the model robustness.

Apparently, the variables pressure inside of bucket and vehicle velocity indicates a very strong seasonally. The pressure inside of closed-circuit implies the behaviors of a loading process. Moreover, the signal on joystick demonstrates the state change. Solely based on these variables, an experienced test engineer can tell

whether the mobile machine is loading or unloading with almost 100% accuracy. Without a doubt, a deep learning model can take over the job to detect Y cycles. However, in the measurement data I have, the Y cycle is not always so regular. For instance, one Y cycle does not always begin with a loading process, followed by an unloading process. The driver might think he has loaded too small amount of load so he comes back after a small reversing process and digs into the heap again. It happens when the driver is not so skilled or is mistaken. Such a case increases the difficulties of detecting the truck loading process by deep learning.

## 5.5.2 Data Preparation

The measurement data were not pre-treated by human observation depending on the dataset before fed into neural networks even I agree a pre-processing can surely increase the accuracy of prediction. The reason for that is I am worried about the pre-processing may exaggerate the performance of neural network since some pre-processing technologies are almost impossible in reality. Therefore, I did use the non-adaptive method to prepare the dataset: only a first-order system is used to smooth my data. After that, the dataset will be split into small slip windows. If the size of time windows is 10 sample times, the events in the past 2 seconds are taken into consideration since my sample frequency for creating slip windows is 5 Hz. Every slip window slips one sample compared to its previous window. Slip windows are used for avoiding the influence of data too long ago.

The measured data is normalized before training since I want to avoid one single variable that has too much influence on each gradient descent step. As a result of that, the cost function's shape changes into a more spherical one rather than a high curvature ellipse one.

Besides, the labeled date is converted into one hot vector to have the same categorical value, as shown in Eq. 5.1,

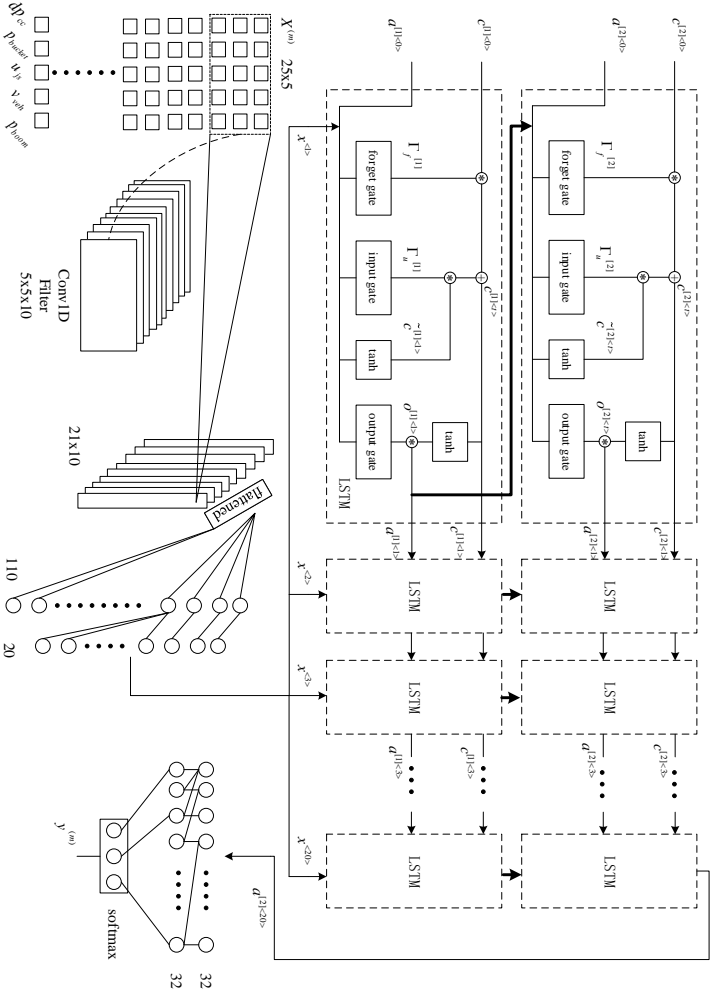
$$Y^{(1)} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \quad (5.1)$$

which demonstrates that the 1st sample is labelled as loading. In my dataset, 11.62% of all working time is in the loading process, and 7.86% is in the unloading process. Obviously, my dataset for truck loading process is skewed. That means, even if the system always predicts that it is neither in loading nor unloading process, it has a test accuracy at about 80%. To avoid it, Confusion Matrices (CM)s and micro average F1 scores should be used to evaluate the performance of my algorithm. Based on an exploratory training, the training cost without anti overfitting goes down to an extremely low level while the test cost goes firstly down and then explodes up. This indicated that my dataset has well considered the variance of Y cycle in different cases.

## 5.6 Combined Neural Networks

To have a better detection performance, I take the advantages of combined neural networks, and thus selected CRDNNs as my tool. In this section, I am going to present the combination of CNNs, RNNs, and DNNs. They all have limitations and advantages so that I believe that combined neural networks can be complementary for the disadvantages of each other. For example, LSTMs are good at temporal modelling while they cannot have a more significant number of hidden layers.

As shown in Fig. 5.4, I use one one-dimensional convolutional neural network (conv1D) at the beginning to provide better features for LSTMs. It is followed by two DNNs to reduce the dimension of the output of CNN. Further, I add two LSTMs since it is considered as an excellent tool for many time series applications. At the end, two DNNs are used to increase nonlinear hidden layers and thus increase the prediction performance by making a deeper mapping. The



**Figure 5.4:** Detailed description of CRDNN with two LSTMs. The notation I use is based on the Stanford University deep learning Lecture Notes, where  $c^{(j|<j>)}$  denotes the cell in  $i^{th}$  layer and for  $j^{th}$  time series.

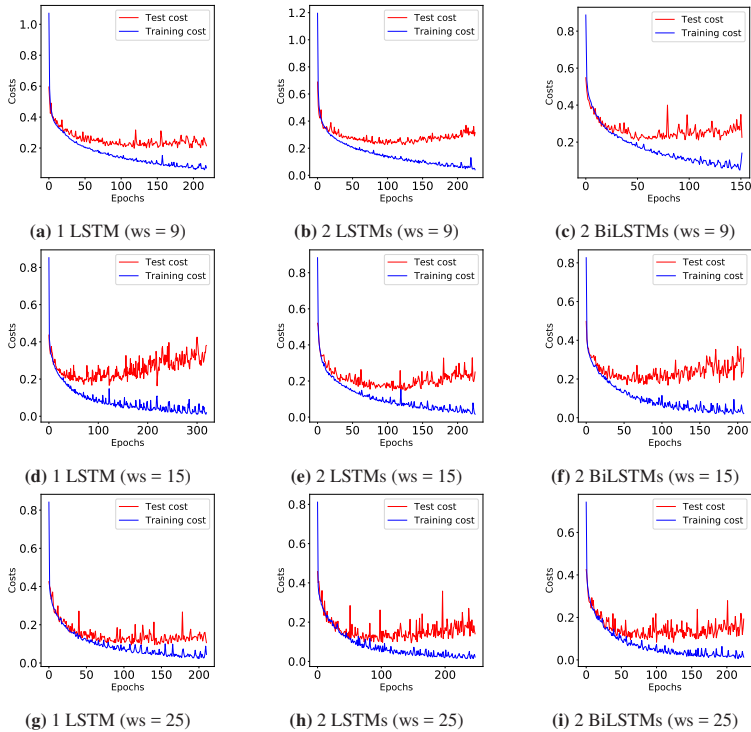
core of LSTMs is the update and forget gate to handle the long and short term data. Eq. 5.2 demonstrates the idea.

$$\left\{ \begin{array}{l} \tilde{c}^{(t)} = \tanh(W_c[a^{(t-1)}, x^{(t)}] + b_c) \\ \Gamma_u = \sigma(W_u[a^{(t-1)}, x^{(t)}] + b_u) \\ \Gamma_f = \sigma(W_f[a^{(t-1)}, x^{(t)}] + b_f) \\ \Gamma_o = \sigma(W_o[a^{(t-1)}, x^{(t)}] + b_o) \\ c^{(t)} = \Gamma_u * \tilde{c}^{(t)} + \Gamma_f * c^{(t-1)} \\ a^{(t)} = \Gamma_o * \tanh c^{(t)} \end{array} \right. \quad (5.2)$$

Generally, the learning ability is increasing as the number of hidden layers increase. However, more hidden layers result in much more training parameters that may be a heavy load for vehicle ECU. In this section, I evaluate the CRDNNs' test accuracy regarding the hidden layers, the units in a hidden layer, and time windows.

Theoretically, LSTMs can work without a slip window. However, I need to avoid the effect of the data before a disruptive event, such that the driver stops the vehicle to relax for a while, which affects the prediction performance. Therefore, I also use the slip windows for CRDNNs. The window size can affect the performance of neural networks since a larger window size allows the neural networks to consider a more extended period to make the decision.

Since I want to know which model has the best test accuracy and which one has a good test accuracy but with fewer training parameters, I show the different performances of different architecture with different window sizes. I supervise the training- and test costs over epochs and stop the optimization process when there is a noticeable tendency that test cost increases. The training cost versus iteration of different neural networks is shown in Fig. 5.5. For example, in the case of CRDNN with 2 LSTMs that is fed the data with a window size of 9, I stop the iteration at epoch 60.



**Figure 5.5:** Training- and test costs versus epochs.

To find out the suitable hyper-parameters of neural networks, I analyze the weights of each layers of neural networks. However, while people recognize the working process mainly by watching the pressure inside of bucket, CRDNNs do not pay too much attention to this variable since the absolute value of weight for it is no considerably larger than the others.

As shown in Fig. 5.5, the cost goes down to a certain level and then fluctuates if the regularization and drop-out method are used. Notice that the cost with anti-overfitting methods is higher since I use the regulation method: it does not mean the accuracy is worse than the one without anti-overfitting methods. Also, I add weights to the cost function. The weights can avoid a certain kind of error

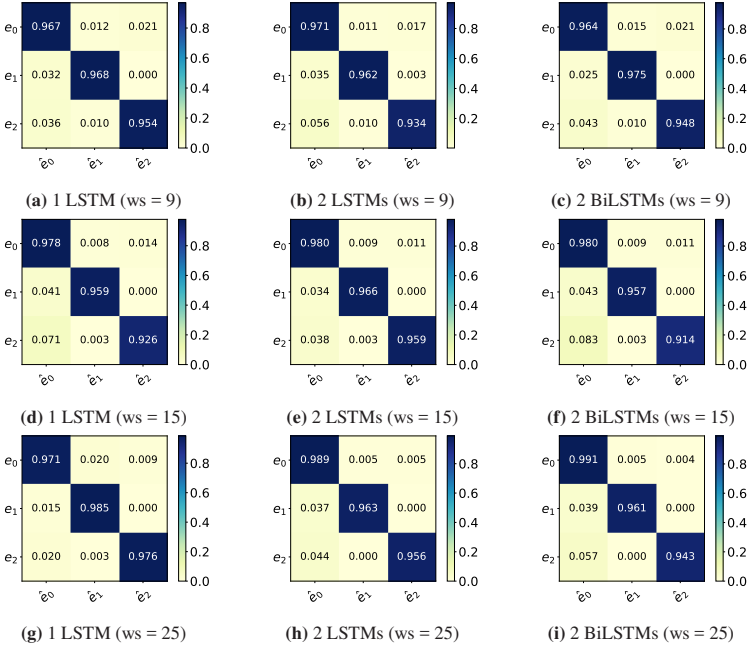


Figure 5.6: Confusion matrices of CRDNNs.

by recognizing. For instance, if the weight on loading is higher than the weight on the traveling process, the optimization process will take more attention to avoid the errors on loading rather than on the traveling process. Formally, see Eq. 5.3.

$$\begin{aligned}
 J(\theta) = & \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K [-y_k^{(i)} \log((h_{\theta} x^{(i)})_k) \\
 & - (1 - y_k^{(i)}) \log(1 - (h_{\theta} x^{(i)})_k)] W_k \\
 & + \frac{\lambda}{2m} \left[ \sum_{j=1}^{32} \sum_{k=1}^{32} (\theta_{j,k}^{(1)})^2 + \sum_{j=1}^3 \sum_{k=1}^{32} (\theta_{j,k}^{(2)})^2 \right]
 \end{aligned} \tag{5.3}$$

Obviously, compared to the cost function without regularization, the regularization might increase the total value of cost function. The  $W_k$  denotes the weight of  $k$  state. In my case, I recommend setting the weights as

$$\tilde{W} = \begin{bmatrix} 1 & 4 & 7 \end{bmatrix}^T \quad (5.4)$$

Since rectified linear unit (ReLU) has a constant gradient if the  $X > 0$ , I use ReLU as activation function so that the calculation effort can be reduced and thereby converging or learning much faster. The Hyper-parameters I used are shown in Tab. 5.1.

**Table 5.1:** Parameters of CRDNN

<b>Hyper-parameters</b>	<b>Value</b>
Window size	[9, 15, 25]
Batch size	128
Initial learning rate (decay during learning)	$1 \times 10^{-4}$
Num filter conv1D	10
Kernel size	5
Num units 1 <sup>st</sup> layer (RNN)	32
Num units 2 <sup>nd</sup> layer (RNN)	32
Num units 1 <sup>st</sup> layer (DNN)	32
Num units 2 <sup>nd</sup> layer (DNN)	32

Generally, I shall use the F1 micro average to evaluate and select the best suitable architecture. Nonetheless, since I am going to implement an operation strategy based on the learning algorithm later, the F1 score alone does not indicate whether a result is easy to correct or not, so I also use CMs to evaluate the results, see Fig. 5.6, where the abscissa indicates the predicted value and the ordinate indicates the ground truth label.  $e_0$ ,  $e_1$ ,  $e_2$  denote the travelling process, the loading process, and the unloading process appropriately. The F1 score is used as a subordinate criterion to select the better overall performance solution. Obviously, the CRDNN



with two bidirectional LSTMs has the best performance, which is similar to my assumption, with an overall accuracy at 98.5%<sup>2</sup>. Compared to simple DNNs, CRDNN has an improvement of about 2.4%. Bidirectional LSTMs make the decision using a relatively more prolonged-time period and can consider the data after the event so with no doubt it has better accuracy. The improvement compared to DNN is because LSTMs are good at dealing with long term problems so that I can use a larger window size to feed into CRDNNs. Another potential architecture is CRDNN with two LSTMs, which is only slightly worse than the one with bidirectional LSTMs but the training parameters are much fewer. An additional advantage of CRDNNs is that most of them have even fewer training parameters though they have a complicated architecture. Compared to the simple neural networks with two hidden layers and 128 units per layer, CRDNN with two layers of LSTMs has only 16,295 training parameters while the former has more than 30,000 training parameters, resulting in a much faster on-board calculation.

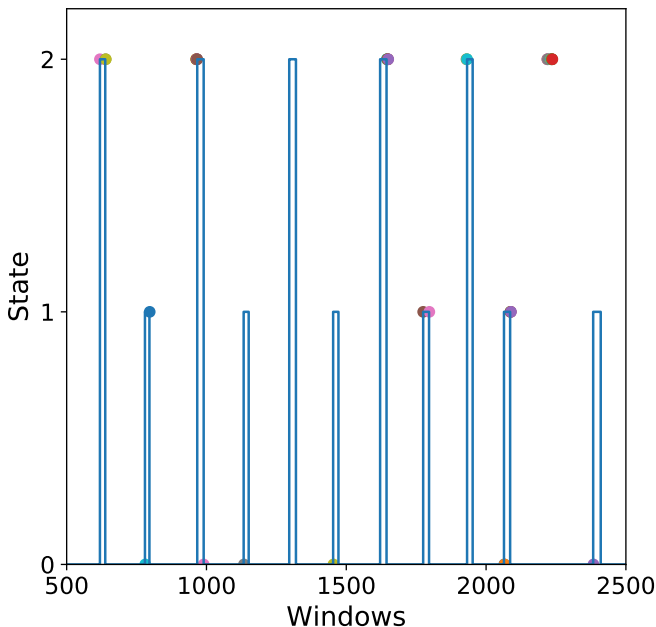
## 5.7 Evaluation of the Methods

As the results are shown in the last section, the improvement of test accuracy is almost stopped at 98%. To further improve the prediction accuracy, I draw out the place where my algorithm has made a mistake.

As shown in Fig. 5.7, I illustrate the ground truth and the mistakes made by CRDNN with two bidirectional LSTMs. The blue line denotes the ground truth label and the color points represent the place where CRDNN recognizes a different result as ground truth label and thus I say it makes a mistake. Obviously, the mistakes mainly occur at the time when the machine changes its state from one to another. However, I must point out that they are the states which are also controversial for humans to say whether the state should be loading, unloading,

---

<sup>2</sup> Note that the dataset consists 11.62% time proportion of the load process, 7.86% time proportion of the unload process, and the rest is the travelling process. Since the travelling process is the vast majority, the overall accuracy is close to the accuracy of the travelling process prediction.



**Figure 5.7:** Ground truth and prediction mistakes. As reminder, the state 0,1,2 represent the travelling, loading, and unloading process, respectively. Blue line denotes the ground truth. Color points show the predict results which are different as the ground truth. As can be seen, most of them are at the place where state changes. This indicates the further accuracy improvement may not make sense. Since the sample frequency is 5 Hz, the windows around 2,250th reveal the information about 450 s in the original data shown in Fig.5.3. The deep learning model is robust though there are some mistakes in ground truth dataset.

or travel since the features are vague in this region. When I further draw all the falsely recognized time windows, I find that almost all of the mistakes occur when the state is really fuzzy.

One exception is the windows around the 2,250th window in Fig. 5.7, corresponding to the measurement data at 450 seconds, see Fig. 5.3. As mentioned before, the time windows at 450 seconds are the windows that I consciously mislabeled the data to the travel process state. This proves that the CRDNN has a robust performance even if a few data is mislabeled. That is, the CRDNN with two

bidirectional LSTMs accurately identified that the process is actually an unloading process rather than a travel process. Therefore, I believe that about 98% is the best number since different engineers define the ground truth differently with their plausible reasons. Moreover, although a tiny number of mislabeling might harm the test accuracy. However, it cannot affect the prediction performance of the CRDNN.

## 5.8 Fast CRDNN

The CRDNN is a combined neural network that can accurately detect the truck loading cycles of torque based mobile working machines. On the one hand, it is a robust but offline learning algorithm so that it is more accurate and much quicker than the previous methods. However, on the other hand, its accuracy cannot always be guaranteed because of the diversity of the mobile machines industry and the nature of the offline method. To address the problem, I utilize the transfer learning algorithm and the IoT technology. Concretely, the CRDNN is first trained by computer and then saved in the on-board ECU. In case that the pre-trained CRDNN is not suitable for the new machine, the operator can label some new data by my App connected to the on-board ECU of that machine through Bluetooth. With the newly labeled data, I can directly further train the pre-trained CRDNN on the ECU without overloading since transfer learning requires less computation effort than training the networks from scratch. In this chapter, I prove this idea and show that CRDNN is always competent, with the help of transfer learning and IoT technology, by field experiment, even the new machine may have a different distribution. Also, I compared the performance of other SOTA multivariate time series algorithms on predicting the working state of the mobile machines, which denotes that the CRDNNs are still the most suitable solution. As a by-product, I build up a human-machine communication system to label the dataset, which can be operated by engineers without knowledge about AI. This paragraph is an abstract to help the reader to understand the context, the details will be described in the following sections.

### 5.8.1 What is CRDNN?

As aforementioned, CRDNN is a neural network that combines the CNN, RNN, and DNN. The combination brings the advantages of different kinds of neural networks together [139]. Pressure inside the bucket ( $p_{bu}$ ), vehicle velocity ( $v_{veh}$ ), vehicle direction signal on the joystick ( $u_{js}$ ), pressure inside of closed-circuit drivetrain ( $p_{cc}$ ), and pressure inside of bucket ( $p_{bo}$ ) are collected during the wheel loaders are working in Y cycles. I labeled the data with corresponding working state, traveling ( $e_0$ ), loading ( $e_1$ ), and unloading ( $e_2$ ). I then trained my neural network on the computer with these ground truth data. In order to find out the best model for the task, I have explored many different kinds of networks, such as CNNs, RNNs, DNNs, and their combinations. Among these neural networks, the combined neural networks CRDNN with two LSTM layers performs excellent test accuracy with relatively low training parameters<sup>3</sup>. Moreover, the robustness of this model to the small amount of mislabelled data is also the reason for the final selection. I saved the trained CRDNN in the on-board ECU, and CRDNN can rapidly identify the working state with high precision and recall. The model is built with Kereas API in Tensorflow [142]. A more detailed description of the CRDNN and how the dataset [143] was built up can be found in my previous study [63].

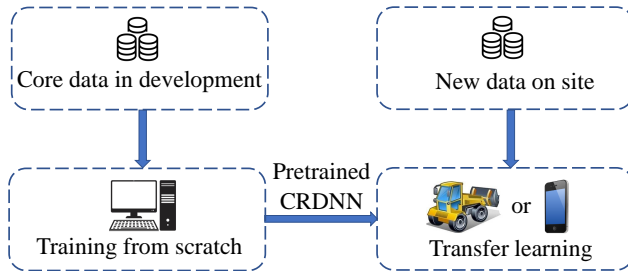
### 5.8.2 Motivation of Fast CRDNN

In the previous study [63], the CRDNN shows excellent performance in detecting the Y cycles of primary-torque-based mobile machines. To date, I believe that CRDNN is a promising method to solve the problem. Firstly, it is an offline approach that can be an order of magnitude faster than the other online learning methods. Also, it achieves a better performance on the challenging dataset by taking the time-series signal sequence into account. However, due to the diversity of mobile machines and driver behaviors, the accuracy of prediction is not always

---

<sup>3</sup> The graphical description is Fig. 5.4

so satisfying even the CRDNN is used. The performance of CRDNN decreases when it faces measured data from a driver with totally unseen behaviors, which means the distribution of data gathered from the new machines and drivers are different from the previous dataset used to train the CRDNN. The reasons are apparent. First and foremost, the CRDNN is an offline learning method that can not automatically adapt to the new tasks after it has been trained. Also, the gather of all the data in every scenario for the initial training is still challenging and, of course, economically impossible. Therefore, in this paper, I utilize the transfer learning and IoT technology to solve the problem. The pre-trained CRDNN will be further trained in case that the machines or drivers have totally different features, and the recognition system can then reach the expected performance. Apparently, establishing the communication interface between humans and machines plays a vital role in this approach. Therefore, this communication interface is also introduced in Section 5.13.



**Figure 5.8:** Graphical abstract. Here the core data is a large dataset that contains 119 Y cycles data from many wheel loaders. This core dataset is used to train the base network. Thanks to this base network, I can then use transfer learning to adapt the weights in this base network with the new data to improve the generalization ability, easy and quick. The method is proposed to solve the problem pointed out by many machine learning researchers, the distribution of the source data may differ from the target data since the collection of a comprehensive dataset is, in many cases, impossible.

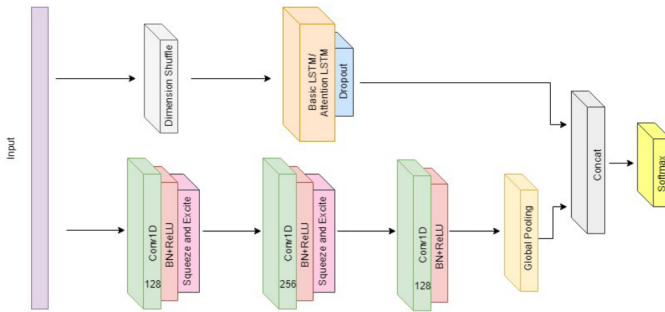
The main contributions of the following paragraphs in this chapter can be sum up as the following points:

- I compared the performance of the selected CRDNNs to another commonly used SOTA solution in the field of Time Series Classification (TSC), and show that CRDNNs is more suitable for the Y cycles detecting task, details see Tab. 5.6.
- I proposed that transfer learning should be used to enhance the generalization capability of CRDNNs.
- I recommend CRDNN with 2 LSTMs as the base network based on its micro F1, back- and forward propagation duration so that the networks can be further trained directly on the working site using transfer learning.
- I design an easy human-machine communication system for the data exchange between human and mobile machines.
- I proposed an approach to label the slip windows which can reduce the delay between the state occur and the state can be correctly predicted.

The rest of this chapter is organized as follows. Section 5.9 and Section 5.10 briefly introduce the prerequisite and background knowledge in fields of time series classification and IoT to understand this chapter since my readers might come from these fields. Next, the existing problems and proposed solutions are illustrated in Section 5.11. Then, the reasons why I adopt these solutions are provided in Section 5.12. After that, in Section 5.13, I describe the connection system between the human and the mobile machines. In Section 5.14, I show how the measurement setup. Followed by Section 5.16 and Section 5.17, I compare the variations of CRDNNs with the SOTA TSC solution, and the performance of different transfer learning methods. Finally, Section 5.18 gives conclusions of this chapter.

## 5.9 Long Short Term Memory Fully Convolutional Network: a SOTA Solution for TSC Tasks

Long Short Term Memory Fully Convolutional Network (LSTM-FCN) is designed for classifying univariate time series [144]. In order to apply this network to the multivariate time series classification problem, Karim extended the Squeeze-And-Excite (SAE) block to the case of 1D sequence models and augmented the fully convolutional blocks of the LSTM-FCN model to improve classification accuracy [5]. The network architecture is shown in Fig. 5.9.



**Figure 5.9:** LSTM-FCN with squeeze-and-excite block [5].

Fully Convolutional Networks (FCN) have proven to be an effective learning model for TSC problems [145], which comprised of three temporal convolutions, are typically used as feature extractors. Global average pooling [146] is used to reduce the number of parameters in the model before classification. The SAE is added after the FCN block which adaptively recalibrates the input feature maps [5].

This architecture has been tested on 35 benchmark datasets for TSC, and it outperforms the other SOTA models on at least 28 datasets [5]. Thus, I would

like to compare the CRDNN with this algorithm for the task of detecting mobile machines' Y cycles.

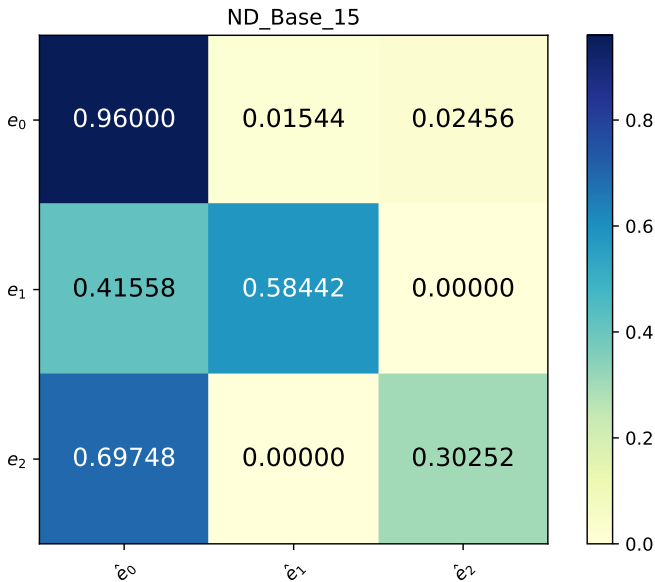
## 5.10 Wireless Human Machine Communication

To achieve the smart working site, effective communication among mobile machines is an inevitable vital step. Since the mobile machines are very likely to work at a place where there is outside of the coverage of the base station, I utilized the ad-hoc network as the first version for the fleet management of the mobile machines [40]. In that chapter, although the realtime communication system is proposed, the bidirectional communication between human and mobile construction machines is still a gap. Recently, many other scientists also emphasize the value of setting up the management system between operators and machines [147]. However, they did not consider the rapid development of the new technology on mobile smartphones and consequently did not develop core functions on the smartphone. Based on the research from Ignatov, the capability of the system on a chip (SoC) on cell phone grows extremely fast and research almost 40% velocity of Geforce GTX 1060 in terms of processing images [148]. Hence, I would like to build up a connection between cell phones and the mobile construction machines to take advantage of the cell phone SoC industry's development. The top SoCs until April 2020, A13 from Apple Inc, Snapdragon 865 5G from Qualcomm, Kirin 990 5G from Huawei, Exynos 990 5G from Samsung, claim that their SoCs can be about 20% faster compared to their last generation published in the last year. Also, the newest version SoCs equip with GPU to enhance the capability to deal with artificial intelligence tasks. All of them have Bluetooth 5.0 modules that can easily connect to the mobile construction machines onboard ECU. Apparently, the development of the computational performance of SoCs is much faster than the onboard ECU.



## 5.11 Problem Statement and Brief Description of the Solution

As my first version of CRDNNs, the CRDNNs can easily achieve predictive accuracy of about 98% based on the dataset of 119 Y cycles, which reaches human-level performance. However, when I consciously change the equipment, especially the shovel, of those mobile machines and then test the CRDNNs, the performance is degraded to an unacceptable level. Fig. 5.10 illustrates the performance of CRDNN when it faces measurement data from a driver with totally unseen behaviors, and the implements has been changed. The reason for that is the training data and the test data have a different distribution in both marginal distribution and conditional distribution.



**Figure 5.10:** The confusion matrix of CRDNN on new data. The  $e_i$  are the ground truth and the  $\hat{e}_i$  are the predictive state. As defined in my previous work, the  $e_{1,2,3}$  denote the state travelling, loading, and unloading, separately.

In addition, since the mobile machines are rent for construction tasks and count money by time, the robustness of the machines and the algorithms on the machines is a matter that cannot be negotiated for the contractors. Thus, either an approach that can always guarantee the performance of the algorithm without adjustment or an approach that only requires rapid and easy calibration is needed as a complementary solution. Even worse, OEMs are reluctant to share their data with each other resulting in a lack of training data for all of them. Based on the facts and challenges I analyzed, I select the approach of offline learning with online adaption. Concretely, instead of sharing the real measured data, transfer learning allows them to further train the pretrained base neural networks with a small new dataset and thus have a similar effect as they gain a series of data and train the neural networks.

As we know, data plays a critical role in deep learning. A large and highly diverse dataset improves the capability of machine learning methods. Also, the same distribution and feature between the training data and test data are a guarantee for the excellent performance when the neural networks are applied in practice. However, in the real world, there are many different kinds of construction machines and workplaces, which may lead to the change of the data distribution. Since the collection of the dataset from all kinds of construction machines is almost impossible, I adopt the transfer learning method to guarantee the same data distribution of the training and test data. Since there must be some similarities between the data collected from the previous wheel loaders and the new machines, it is likely to do fine-tuning with labeling a few datasets on the working site, and it will only take a few training steps to achieve the satisfying prediction results<sup>4</sup>. Thus, it is not computationally expensive and can be trained directly by the onboard ECU or smartphone. Notice that, whether the new data should be trained on the onboard ECU or the SoC in the cell phone is depending on the capability of them and the bandwidth of the connection. At present, I recommend further train the CRDNN on the onboard ECU since the transmit of the data from mobile

---

<sup>4</sup> Strictly speaking, this is a hypothesis until now. However, this will be proved in the following context.

construction machines to cell phones has a more massive amount of data as in reverse. However, the approach introduced in this chapter can be easily adapted to the version that trains the CRDNN on the cell phone at the time when the data transmission is proved as no more a problem.

## 5.12 Why Transfer-Learning Based Supervised Learning?

Traditional machine learning performs well by using training data and test data with the same input feature space and the same data distribution. When there is a difference in data distribution between the training data and test data, the results of a predictive learner is likely to be degraded [149, 150, 151]. In certain scenarios, obtaining training data that match the feature space and predicted data distribution characteristics of the test data could be difficult and expensive. Therefore, there is a need to create a high-performance learner for a target domain trained from a related source domain. This is the motivation for transfer learning [152]. Transfer learning is used to improve a learner from one domain by transferring information from a related domain [153, 154].

Since the transfer learning is a rapid developing subject, the terminology and definition have currently no consistency. In this chapter, I use the mathematical definition from Pan for further discuss, who defined that  $\mathcal{D}_s = (X_s, P(X_s))$  as source domain,  $\mathcal{D}_t = (X_t, P(X_t))$  as target domain,  $\mathcal{T}_s = (X_s, f_s(\cdot))$  as source task, and  $\mathcal{T}_t = (X_t, f_t(\cdot))$  as target task. Transfer learning aims to enhance the learning of the target predictive function  $f_t(\cdot)$  in  $\mathcal{D}_T$  using the knowledge in  $\mathcal{D}_s$  and  $\mathcal{T}_s$ , where  $\mathcal{D}_s \neq \mathcal{D}_t$ , or  $\mathcal{T}_s \neq \mathcal{T}_t$  [155].

In the past decade, transfer learning has been successfully implemented in the fields of image recognition [156, 157] and Natural Language Process [158]. In contract, scientists in the field of TSC believe that there has a lot of things should be proven or improved [159]. It is only recently that deep learning was proven to work well for some TSCs [160]. However, unlike image recognition [161],

the lack of a sizeable general-purpose dataset in TSC limits the development of transfer learning in TSC. Another well-known problem by implementing the transfer learning on the TSC task is the negative transfer. As we know, if one is good at handball, she or he can learn how to play basketball faster than the others who never played handball before. The reason is apparent: the knowledge about how to play handball and basketball well are similar. However, people usually have a negative evaluation of the people who give them a bad first impression ( $\mathcal{D}_s$ ), no matter how other people change ( $\mathcal{D}_t$ ). For the latter example, the first knowledge ( $\mathcal{D}_s$ ) does not contribute to the correct prediction ( $f_t(\cdot)$ ) and indeed has an adverse effect. This is a negative transfer. The negative transfer and how transferable are features are still very active research domain [162]. Fawaz has revealed that transfer learning can both improve or degrade the model prediction depending on the source dataset ( $\mathcal{D}_s$ ) [163], by testing the performance of FCN algorithm [145] with transfer learning and from scratch on a series of dataset. To the best of author's knowledge, the consensus is that transferring models between similar datasets improves the  $f_t(\cdot)$  performance. In contrast, Rosenstein empirically shows that if two tasks are too dissimilar, then brute-force transfer may hurt the performance of the target task [164]. Thus, Mahmud proved some theoretical bounds by analyzing the case of transfer learning using Kolmogorov complexity [165]. Furthermore, some previous works have been exploited to analyze relatedness among tasks by using clustering techniques, which provide the guideline about how to automatically avoid negative transfer [166, 167]. Keogh shows that dynamic time warping is a robust distance measure for time series, which can thus evaluate the similarity of the dataset [168]. Based on the literature recherche in the field of transfer learning, I can conclude that the more similarities between the ( $\mathcal{D}_s$ ) and ( $\mathcal{D}_t$ ), the better transfer learning can perform.

There are different strategies and implementations for solving a transfer learning problem. The majority of the homogeneous transfer learning solutions employ one of three general strategies which include trying to correct for the marginal distribution difference in the source  $P(X_s) \neq P(X_t)$ , trying to correct for the conditional distribution difference in the source  $P(Y_s|X_s) \neq P(Y_t|X_t)$ , or trying to correct both of them [169].

Some similar use cases for TSC with transfer learning can be found in many previous studies. For example, Hu explored first to train a model on the historical wind-speed data of an old farm and fine-tune it using the data of a new farm [170]. In addition, Peng propose a transfer-learning based approach to establish an anomaly detection model for dangerous actions of aircraft testing flights [171]. A transfer learning-based bi-directional long short-term memory model is proposed to predict the air quality by Ma [172]. The success of the implementation of transfer learning on TSC tasks encourages us to follow this concept.

In my transfer learning task, the data I used to pre-train the base network from scratch is the source domain ( $\mathcal{D}_s$ ), while the data I collect from the new machines are the target domain ( $\mathcal{D}_t$ ). Apparently, the solution to this problem is to correct both the marginal distribution and the conditional distribution difference in the source. It can be referred to as a parameter-transfer approach, which assumes that the source tasks and the target tasks share some parameters or prior distributions of the hyper-parameters of the models. My transfer learning approach is to recompute the trainable parameters in the neural network. The architecture of the base network will be kept the same.

Another potential approach is also mentioned, which could be used to detect Y cycles: semi-supervised sequence learning, which leverages the unlabeled data to further improve the predictive accuracy [173]. However, the performance of semi-supervised learning is quite difficult to outperform supervised learning [174]. This method is usually adopted for the private data task, where label the data is prohibited [175]. In the case of detecting Y cycles, obtain the new data is actually only a technical problem, and the data must be much easier to get in the era of IoT; thus, I would use supervised learning instead. To achieve the transfer-learning based supervised learning, I have designed a connection system between the mobile machines and human using smartphone.

## 5.13 Connection System Design

### 5.13.1 Choice of Wireless Communication Technology

There are mainly four common short-range wireless communication technologies in the field of IoT, namely Near-Field Communication (NFC), Radio-Frequency Identification (RFID), Bluetooth, WiFi. The comparison of their main specifications are shown in Tab. 5.2.

In order to enhance the generalization capability of CRDNN, I need to get the newly labeled data to train the pre-trained base network further. The new data is labeled through the mobile app, which connects ECU through the Bluetooth. With the new labeled data, the network is retrained on the ECU, and the accuracy of the retrained network can be shown in the app. When the test accuracy reached the expectation, the machine can be put into use.

Considering that most machine operators are not specialists in deep learning, I design the interface as naturally as possible. I find that only two tasks must be done manually: labeling the data and check the confusion matrix. The other steps will be done automatically either by the APP or the ECU.

Each of those technologies has its pros and cons, and can be implemented into different scenarios. NFC can be easily used for transactions, but not for on-site training due to the limited range, which is approximately 10 cm. RFID technology provides a reliable, efficient way to transmits the identity of an object [176], so that it is widely used in the area of the E-ZPass system [177]. However, RFID only supports the one-way transmission, and therefore it is not a solution for my use case. Compared to WiFi, Bluetooth has a lower energy consumption and more straightforward hardware implementation [178]. Therefore, I select Bluetooth for the on-site training. To date, the latest version in Bluetooth is Bluetooth 5.0, which is introduced by the Bluetooth Special Interest Group (SIG). This version offers significant enhancements compared to the previous specifications, regarding a broader range up to 200 m, a faster speed up to 2 Mbps, and more robust to interference [179].

Table 5.2: The main specifications of the selected wireless communication technology

Specifications	NFC	RFID	Bluetooth5.0	WiFi
Maximum Coverage Range	10 cm	3 m	100 m	100 m
Radio frequency	13.56 MHz	varies	2.4 GHz	2.45 GHz
Communication mode	2-way	1-way	2-way	2-way
Data Rate	106, 212, 424 Kbps	varies	2 Mbps	144 Mbps
Applications	Credit card related payments, E-ticket booking	Tracking items	Communication between phone and peripherals	Wireless internet

### 5.13.2 User Interface of the System

The on-site training system is presented in Fig. 5.11. Following, I describe the process that fine-tuning the model on the onboard ECU. The system consists of a mobile smartphone for labeling data manually and the mobile construction machine, which is equipped with Bluetooth Low Energy (BLE) transceiver chip for communicating with the mobile device. The construction machine operator installs my “Smart Working Site” app, which is demonstrated in Fig. 5.11. The app provides four perspectives, namely “Connect machines”, “Label the Data”, “Advanced Settings”, and “Test Accuracy”. At the beginning of the on-site training, the machine operator shall activate the Bluetooth of the smartphone and pair the construction machine, as long as the construction machine is situated within the Bluetooth coverage of the smartphone. Then, the operator observes and records the construction machine’s actions, as the driver starts the construction work. The machines’ working states are transmitted to the on-board ECU intermediately, once one action is labeled. This time series of labels indicates the current action of the machine and is served as ground truth for transfer training of the network. For those who are familiar with neural networks, they can tune the hyperparameters as well as different learning algorithms to retrain the network in the tab of “Advanced Setting”. However, use the model I recommend in this chapter can fix most of the problems; thus, I do not suggest to use the advanced function on the smartphone unless the operators are extremely confident. The hyperparameter “epochs” indicates the number of loops, in which all the training data are fed to the network. The other indicator “weights” means the priority of each working state to be correctly predicted. As the last step, Onboard ECU retrains the network work and transmits the accuracy back to the app, which is visible in “test accuracy”. Once the performance is satisfying, the retrained neural network is applied to the machine.



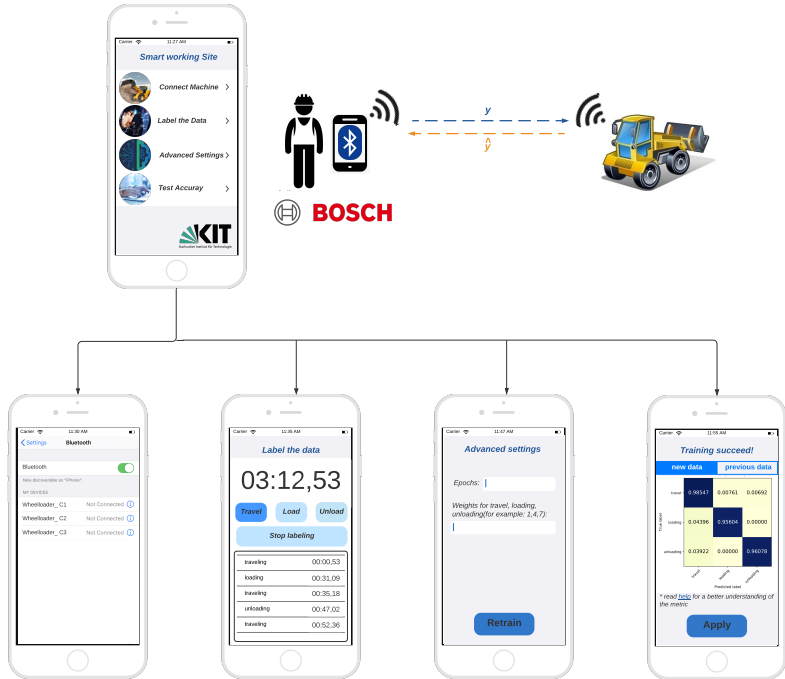


Figure 5.11: The sketch of the human-machine communication App system.

## 5.14 Measurement Setup

To simulate the situations which the OEMs are likely to meet, I consciously change the control algorithm of the implement, and also the size of the shovel. In fact, in order to adapt to different tasks, OEMs will modify a different control program to facilitate the driver's operation. Also, the machines have different sizes for the different working sites; among these differences, the most considerable distinction is the shovel sizes. Therefore, my measurement is set up based on these facts.

Fig. 5.12 shows the mobile machine which I used to gather the new data. Thanks to the dSpace, the control algorithm can be changed on this prototype mobile machine with ease.



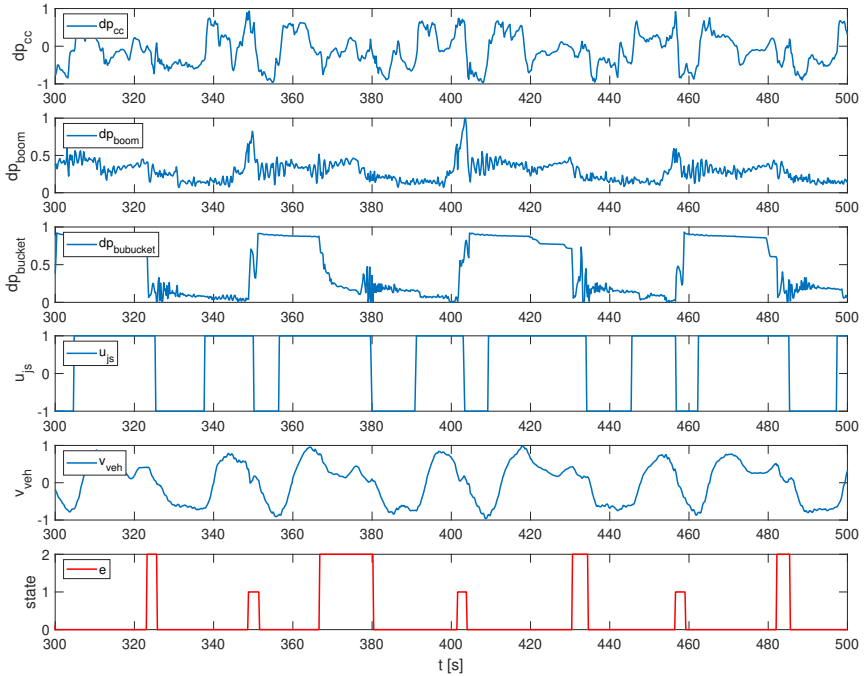
**Figure 5.12:** The mobile machine used for the measurement data.

The newly gathered measurement data, including 24 Y cycles, are partly shown in Fig. 5.13. By observing the newly gathered data  $\mathcal{D}_t$ , I find that the driver operated joystick differently<sup>5</sup>, compared to the driver who created the original dataset  $\mathcal{D}_s$ . The dataset is normalized to accelerate the training process so that the influence of the varying of the shovel dimension might not be shown clearly.

In order to simulate the fact that different engineers may have divergence on how to label the data since they have different standards or rules, I consciously label the newly gathered data in another way as the previous study [63]. For the new dataset, I label the sample into the state traveling whenever the  $dp_{bu}$  is still fluctuating, which is different from the previous approach. Consequently, the distribution of the new dataset has also changed, so the marginal distribution of the source data and target data is much different. To sum up, for the new

---

<sup>5</sup> The control strategy for the shovel is modified for other projects with other purposes. The system is more sensitive.



**Figure 5.13:** New measurement data with a different implement control algorithm and dimension of the shovel. The last subfigure shows the ground truth state of the measured data.

measurement, I purposefully chose a different driver, a different control algorithm for the implement, a shovel in a different size, and a different engineer to label the dataset. Although this makes the task more challenging, I believe it is closer to the reality and should be taken into consideration.

### 5.14.1 The Sliding Windows Labeling Method

After the raw data are gathered and labeled, I need to split the time series data into some small sliding windows to train the neural networks. I sample the data in 5 Hz to avoid overloading the ECU. Obviously, the window sizes affect the system performance; the larger the window sizes are, the more information will

be taken into consideration, and thus more accurate can be expected. However, a larger window size may result in a delay between the state occurs and the machine detects the state. Following, I illustrate the mechanism of this delay.

### **Labeling the Slide Windows Based on the Whole Data**

I do not use the state of the last sample data in the sliding window as the state of the slide windows, because the time point where the state changes are vague. Thus, I believe that I should not label the slide windows only based on one sample data in it. Another drawback of only using one sample data is, the consequently labeled sliding windows can make neural network confusion since most of the sample data in this sliding window might indicate another state.

Here I set the slide windows length as 15, which means a sliding window contains 15 sample data with the label. In order to label these sliding windows, I calculate the distribution of the samples. In this fashion, the slide windows must have an odd length. In case that one state has the majority, I can then set these windows as this state. For example, if 7th sample data is labeled as loading and 8th sample data is labeled as traveling, the sliding window will be labeled as traveling since traveling is the majority. However, in this case, the state traveling occurs at the 8th sample data, and the machine detects the sliding window as traveling when the 15th sample data is measured. Therefore, a delay exists principally by this method. The method can be explained by Fig. 5.14 concretely.

### **Labeling the Slide Windows Based on the Partial Data**

The previous labeling method supplies a reasonable method to label the slide windows. However, the larger the window sizes are, the longer the delay will be. In contrast, if I label the slide windows based on the partial sample data in the windows, the problem can partly be solved. Concretely, I use the last three or five sample data in the sliding window to label this sliding window, as shown in Fig. 5.15. In this vein, the delay can be reduced.

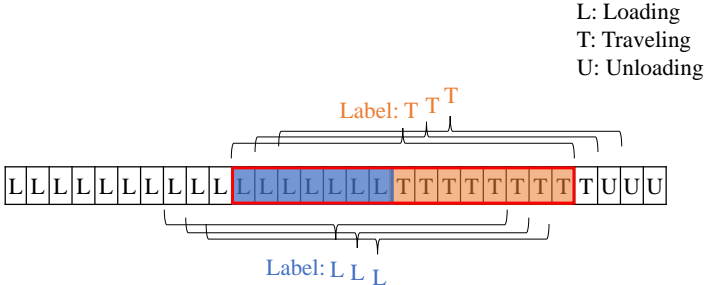


Figure 5.14: The diagram of the relabeling method.

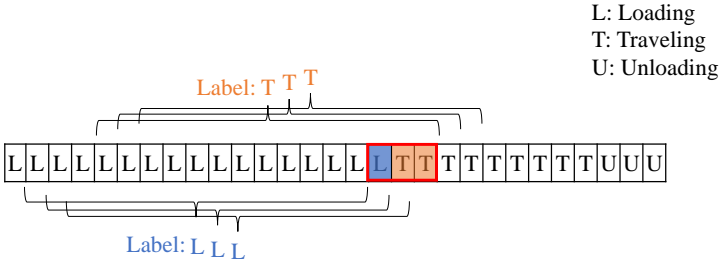


Figure 5.15: The diagram of the relabeling method.

## 5.15 Comparison Between CRDNN and Other SOTA Time Series Classification Neural Networks

Before I show the benefits of transfer learning, I first determine which neural networks should be used as the base network. As mentioned in Section 5.9, LSTM-FCN is considered as a SOTA solution for TSC tasks. In this section, I would like to compare the CRDNN with LSTM-FCN with respect to micro F1, training time, and test time. Here the training time indicates whether the algorithm is suitable for immediately fine-tuning on the working site. The test time shows if

the algorithm is appropriate for realtime detection. My base networks were trained on Nvidia GEFORCE GTX 1050 GPU. In order to find the global minimum rather than the local minimum, I use early stop and set the patient to 100, which means the training process will be stopped 100 epochs after finding the best predictor. To further avoid overfitting, I adopt the L2 regularization method the same as my previous study. The optimizer I use is ADAM [180]. Also, I use ReLU as my activation function since it can be trained faster as Sigmoid. In Tab. 5.3, I demonstrate the performance of different neural networks with different window sizes. Here I use the previous dataset to perform the process of selection of the base networks so that the selected base network can be directly used in the next section where the performance of transfer learning will be discussed. If the model mispredicts the unloading process into the loading process or in reverse, a complicated operation strategy must be designed. Therefore, I only select the models which do not make mistakes in classifying the loading state into the unloading process or in reverse. Among them, CRDNN with 2 LSTMs with WS 15 has the shortest training time and test time. The training time is 310.19 seconds. Compared to LSTM-FCN with WS 15, it needs only one third training time. Although the micro F1 is slightly worse than LSTM-FCN with WS 15, less than 1%, I believe that a much shorter training time conducive to a better performance in transfer learning with respect to efficiency. Moreover, in case that I want to increase the micro F1, I can either increase the WS, or use the other variances of CRDNNs, the one with bidirectional LSTM, to achieve almost the same micro F1, whose difference is less than 0.1%. Notice that I do not further pursue to increase the micro F1 since 98% is already the best performance, and thus a further increment might not make sense, i.e., achieving this value, the deep learning model only makes mistakes at the place where different engineers have different ideas to define the current ground truth state. Interestingly, although the micro F1 increases as the WS increases, the training time does not always increase as the WS increases. In short, based on the training results, the LSTM-FCN has a slightly better performance than the CLDNN with 2 LSTM layers and CLDNN with both one bidirectional LSTM layer and one LSTM layer; however, the training time of the LSTM-FCN is enormous pressure for the ECU when I make a transfer learning on the ECU. Thus, I select CRDNN with 2 LSTM layers

as base networks for transfer learning. Also, I select the WS as 15 according to the training results.

	5	9
<b>CRDNN (1 LSTM)</b>	419.69/ 93.92/ 0.0606/ No	214.99/ 96.65/ 0.0703/ No
<b>CRDNN (2 LSTMs)</b>	500.00/ 93.65/ 0.0835/ No	250.38/ 96.36/ 0.1112/ No
<b>CRDNN (1 LSTM, 1 BiLSTM)</b>	370.28/ 92.41/ 0.1111/ No	302.89/ 96.13/ 0.1645/ No
<b>CRDNN (2 LSTMs, SAE)</b>	768.97/ 95.21/ 0.1350/ No	442.94/ 96.88/ 0.1679/ No
<b>LSTM-FCN</b>	1095.00/ 96.17/ 0.1579/ No	1270.25/ 98.16/ 0.1698/ No

(a) First part of the table

	15	25
<b>CRDNN (1 LSTM)</b>	271.25/ 97.42/ 0.1008/ No	248.07/ 97.48/ 0.1338/ No
<b>CRDNN (2 LSTMs)</b>	310.19/ 97.21/ 0.1457/ Yes	410.01/ 98.25/ 0.2014/ Yes
<b>CRDNN (1 LSTM, 1 BiLSTM)</b>	385.26/ 97.31/ 0.2261/ Yes	479.69/ 98.34/ 0.3690/ Yes
<b>CRDNN (2 LSTMs, SAE)</b>	444.75/ 97.01/ 0.2041/ Yes	518.00/ 98.17/ 0.2671/ Yes
<b>LSTM-FCN</b>	882.13/ 98.14/ 0.1663/ Yes	747.39/ 98.32/ 0.2029/ Yes

(b) Second part of the table

**Table 5.3:** Performance analysis. The performance of the five network structures in respect of total training time (s), micro F1 (%), average test duration (ms), and whether it can never mistake an unloading into loading or in reverse

## 5.16 Transfer Learning Based CRDNNs

Since I do not change the model architecture, there are two potential transfer learning methods: either I can freeze the former parts of CRDNN and only further train the fully connected layers to save the training time, or I can use the pre-trained model's weights as the initial parameters for the further training of the total model. Obviously, the first vein is faster and can mitigate the ECU computational effort. Yet the second way may achieve a better recognition performance. Generally speaking, I can only use the newly gathered data as the validation set, just like other transfer learning tasks did. However, from the users' view, I evaluate the

performance both on previous data ( $\mathcal{D}_s$ ) and new data ( $\mathcal{D}_t$ ). To evaluate the accuracy of each approach, I first show the micro F1 value and then illustrate the CMs. Furthermore, to indicate whether a method is suitable for on-site transfer learning, I judge the approaches based on their training time (back-propagation) and test time (forward-propagation). Here I show the training time and test time on a CPU core i7 4720HQ@ 2.6 GHz since the results are more appropriate to be used as the benchmark for the onboard ECU. The hyper-parameters and the architecture are shown in Tab. 5.4, and the results are shown in Tab. 5.5, where the ND, PD, FS, FTF, OTF denotes newly gathered dataset, previous dataset, training from scratch, fully connected layers transfer learning, and overall transfer learning. For transfer learning, I reduce the patient to 50 for the purpose of achieving a relatively faster training process.

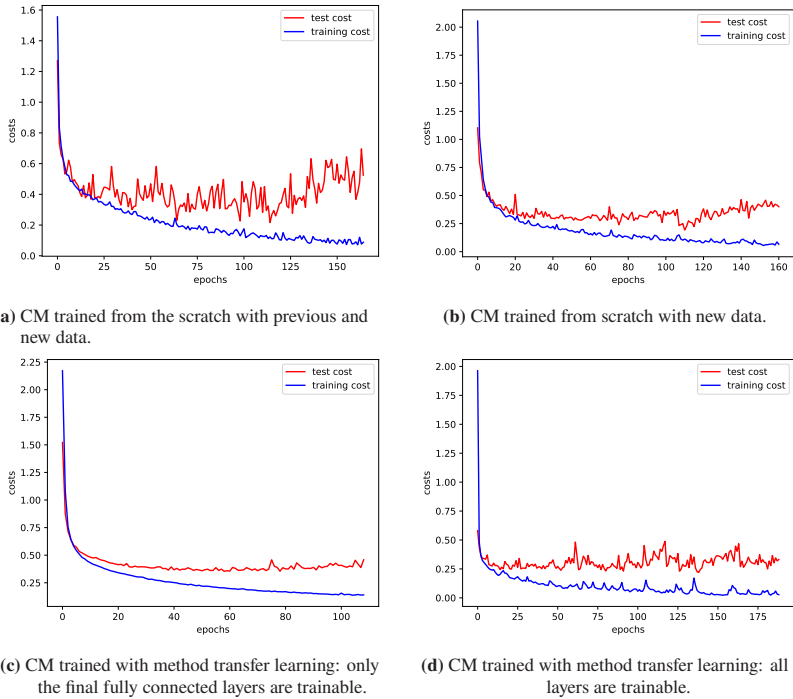
**Table 5.4:** Parameters of CRDNN with 2 LSTMs

<b>Hyper-parameters</b>	<b>Value</b>
<b>Window length</b>	15
<b>Batch size</b>	128
<b>Initial learning rate (decay during learning)</b>	$1 \times 10^{-4}$
<b>Num filter conv1D</b>	10
<b>Kernel size</b>	5
<b>Num units 1<sup>st</sup> layer (RNN)</b>	32
<b>Num units 2<sup>nd</sup> layer (RNN)</b>	32
<b>Num units 1<sup>st</sup> layer (DNN)</b>	32
<b>Num units 2<sup>nd</sup> layer (DNN)</b>	32

### 5.16.1 Training from the Scratch as Benchmark (ND+PD+FS)

In order to have a basic overview between the CRDNN trained from the scratch and the CRDNN trained by means of transfer learning, I demonstrate the training process of the CRDNN trained from the scratch and the CRDNN with the method





**Figure 5.16:** Cost versus epochs of CRDNNs from scratch and with transfer learning.

transfer learning, separately. The CRDNN from scratch is used as the benchmark to illustrate the benefits of transfer learning. By means of training from scratch, each epoch contains 143 Y cycles data since I mixed the newly gathered and the previous dataset together, and the training process can stop at about 75 epochs, as shown in Fig. 5.16(a).

### 5.16.2 Only Further Train the FCN (ND+FTF)

Because the DNNs can be trained faster than CNNs, I firstly only train the final fully connected layers in the CRDNN, and then analyze the performance. As can be seen, the model was further trained with the new dataset. Here each epoch

**Table 5.5:** The performance comparison between different training methods

	<b>F1 on ND</b>	<b>F1 on PD</b>	<b>Sample per epoch</b>	<b>Trainable parameters</b>	<b>Training time (s)</b>
<b>ND + FS</b>	0.9730	0.8432	5695	16295	174.70
<b>ND + FTF</b>	0.9540	0.8305	5695	2211	62.90
<b>ND + OTF</b>	0.9798	0.9038	5695	16295	224.04
<b>ND + PD + FS</b>	0.9655	0.9664	32473	16295	668.30

has only 24 Y cycles, and the training process stops at about 60 steps. After the transfer learning process, it can be seen that the prediction accuracy is much better than the results shown in Fig. 5.10. Concretely, the micro F1 is increasing to about 95%. However, I can utter that the results are satisfying but not perfect as the totally retrained CRDNN. As shown in Fig. 5.16(c), the current neural network is lack of learning ability for further improving the performance of the CRDNN since the validation cost does not change during the training cost goes down.

### 5.16.3 Train the Total Part of CRDNN (ND + OTF)

Fig. 5.16(d) is the result when I further train all the parts of CRDNN with the newly gathered data. Obviously, the CRDNN has a stronger learning capability compared to CRDNN with FTF since the test cost goes down deeper as the epoch increases. The micro F1 of the CRDNN with OTF is higher since the state traveling occupies a majority of my dataset. In order to let the newly trained model can also have a good performance on the previous data, I introduce the soft weight sharing method that uses different learning rates for different layers of neural networks. Concretely, I let the learning rate for the CNNs and RNNs smaller than the DNNs.

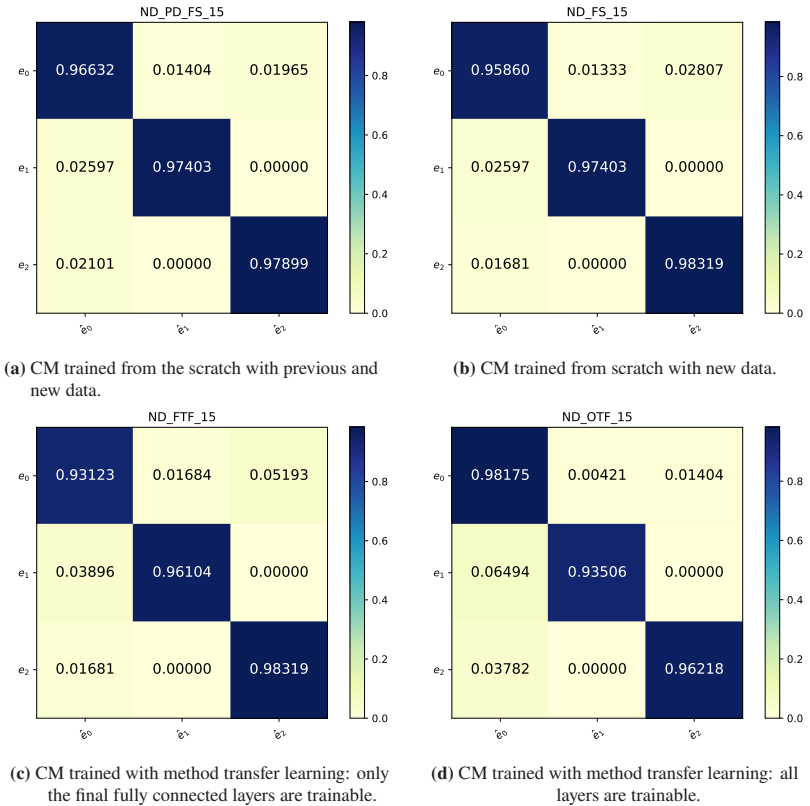


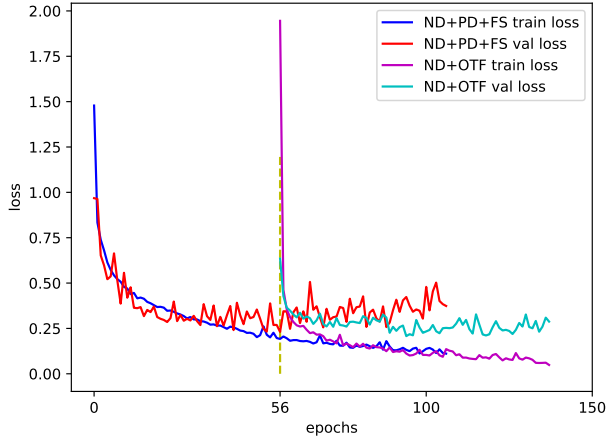
Figure 5.17: Confusion matrices of CRDNNs from scratch and with transfer learning.

### 5.16.4 Evaluation the Benefits of Transfer Learning

The performance of these four methods is shown in Tab. 5.5. Since the first three methods are trained on the newly gathered data, the samples per epoch are much fewer than the fourth methods. Also, in case that I only train the fully connected layers, the trainable parameters are the fewest. Both of them are good for reducing training time. Thus, the training time for ND+ FTF can reduce to one-tenth (10%) compared to ND+PD+FS, and one third (35%) compared to

ND+FS. Here the training time is 62.90 s; however, since the model can be trained on different onboard ECU or smartphone, the concrete numbers shown in the table are only made sense to be used as a benchmark to compare the performance of one approach to the other approaches. For instance, some ECUs on the mobile machines may have a relatively lower computational capability resulting in 10 times longer training time than the value here shown. Also, it is possible that the onboard ECU is even faster than this training time because mobile machines usually have a powerful energy source. Based on the comparison, I use the method ND+FTF as an emergency method to let the model can work with high accuracy immediately on the new task after new labeled data are fed into the model. ND+FS shows a great accuracy on the new data; however, since the newly gathered dataset is relatively small, the generalization capability of this approach is suspicious. The other transfer learning method, overall transfer learning, has the best performance on new data. It is also good at detecting the previous data, which indicates that it has a good generalization capability. Moreover, the training time is only one third (33%) compared to the ND+PD+FS. Therefore, I recommend using ND+FTF to train the network in the case that it is not so hurried or the mobile machine has a relatively powerful ECU. Note that the micro F1 of ND+PD+FS is slightly worse than the results in Tab. 5.3 because the patient is fewer. As shown in the CMs in Fig. 5.17, the models do not mistake the loading process with the unloading process, which denotes that all the models can mitigate the design of operation strategy; thus, all of them have the potential to be used, in case that OEMs have their special wish.

To illustrate the mechanism of time saved due to offline learning with online adaption compared to pure online learning, I show the training process of transfer learning. As shown in Fig. 5.18, the training and validation cost on precious explode at the epoch 63. The base network on the computer is finished at step 63 since the validation cost begins to grow. Hence, at this time point, I added the new labeled dataset to simulate the real scenario for transfer learning. Right after the new data are considered, both training and validation cost goes to an extremely high level since the dataset has an enormous variance. Consequently, the prediction results must be unsatisfying. Interestingly, only after a few steps of



**Figure 5.18:** The mechanism of transfer learning. The blue line is the training cost on ( $\mathcal{D}_s$ ), the red line is the validation cost ( $\mathcal{T}_s$ ), the purple line is the training cost on ( $\mathcal{D}_t$ ), and the cyan line is the validation cost ( $\mathcal{T}_t$ ).

further training on the on-board ECU, the cost goes dramatically down to the low level. As a result of that, the CRDNN is again suitable to predict the truck loading process even when the scenario is quite different from the original dataset.

Based on the results of this section, I find that transfer learning is a powerful tool to let the CRDNN be robust to the challenging Y cycles detection tasks. The transfer-learning based CRDNN with 2 LSTMs is the most appropriate model since it can be retrained much faster than LSTM-FCN with only 1% accuracy lost. Without transfer learning, the model can not guarantee excellent performance for the new target task ( $\mathcal{T}_t$ ); thus, I recommend using transfer-learning-based CRDNN for the detection of Y cycles.

## 5.17 The Advantages of This System from Engineers' View

Here I would like to sum up the main advantages of the transfer-learning based CRDNN and the corresponding IoT system as strong, fast, and easy.

### 5.17.1 Strong

This system is aimed to improve the efficiency of the novel torque-controlled hydrostatic mobile machines by correctly detecting the working process. This system can automatically recognize the working state without an additional button or human action, which offers essential information for the energy regeneration process. Thanks to the transfer learning, the system can be adapted to a new machine, even where there has a different distribution as the source dataset, without a complicated calibration process. The test accuracy of this working state recognition system can reach 98% on the challenging dataset [63], which achieves the human-level performance and guarantees accurate recognition. The strong ability of generalization of transfer-learning-based CRDNN is proven.

### 5.17.2 Fast

Usually, an excellent ability of generalization is based on the sacrifice of speed. However, the transfer-learning-based CRDNN is fast. It is an offline method with online adaption; thus, it is a realtime algorithm. Also, transfer learning needs much less computation effort resulting in the on-site training capability of CRDNN.

### 5.17.3 Easy

Generally speaking, an interface that controls an extensive system is complicated. However, the UI of the IoT system designed in this chapter is easy to use. The operators only need to give the data the appropriate label and check the model accuracy. The system automatically does most of the training steps.

At the end of this section, I demonstrate the performance of different approaches in Tab. 5.6, where the online learning approach was evaluated with the batch size is equal to 1.

**Table 5.6:** Performance comparison of different learning approach

	Offline	Online with adaption	Online
<b>Real time</b>	+	+	-
<b>Ability of generalization</b>	-	+	+
<b>Learning ability</b>	++	++	+

## 5.18 Conclusion

In this chapter, I have shown that CRDNN with bidirectional LSTMs has the best performance to detect the truck loading cycles, and the CRDNN with 2 LSTMs has the best performance-cost ratio if primary torque control concept is used. Because I use an offline learning strategy and the forward propagation is much faster than backward propagation, this method will not take up too much computational effort. By considering a period of 5 seconds, the test accuracy reaches 98.2%, and it never mistakes the loading process with the unloading process or vice versa, which makes the operation strategies easily to be implemented. Also, since I have a large dataset, a tiny mislabel could not harm the real performance of the CRDNNs. It is also worthy to point out, although CRDNN has only increased the test accuracy by 3%, it increases the most challenging 3% by successfully detecting the state of the data gathered when the drivers did not operate well.

Afterward, I update the naive CRDNN to the transfer-learning-based CRDNN. Thanks to the transfer learning, the generalization ability of CRDNN has been much enhanced so that it becomes a powerful solution for solving the high variance problem in detecting the truck loading process. Since transfer learning needs new data, I completed the IoT system of mobile machines by building a human-machine communication system on the smartphone for the purpose of gaining the data quickly. The model I recommend can be trained very fast so that the workers can adapt the model directly on the working site after gaining the new data rather than sending the data to the deep learning specialist. As the results shown, the proposed methods can always help the pre-trained CRDNN to achieve satisfactory performance with respect to precision and recall. Besides, the training time on the onboard ECU can reduce at least about 70% to 90% compared to if I retrain the neural network from scratch on the onboard ECU. Also, I use the new method to label the sliding window, so that I can partly solve the delay of the prediction results in the previous version of CRDNN.

As a result of successful detection of truck loading process, I envision the working process based motion prediction methods.



## 6 Visual Monitoring of Working Site<sup>1</sup>

Although the AI MAPF algorithm guides the machines to move efficiently, there is a potential risk. Since the machines' location are gathered with GPS/IMU system and then send to AI system, participants without localization equipment are ignored by the system. Consequently, tragedies might happen. Hence, a visual monitoring system is introduced in this chapter to monitor working sites. Current computer vision algorithms have shown excellent performance in detecting many common objects. By testing on well-known datasets, the best algorithm until 2020 was proved to achieve about 0.6 mean Average Precision (mAP) with the incredible innovation and effort of scientists. However, for commercial use systems regarding personal safety, this value is not satisfying. Considering that all machines and workers are working on a closed site, I attempt to increase the detection performance through reasonable overfitting. In light of that, I created the MOMA dataset, including eight classes of commonly used mobile machines, which can be used as a base dataset to be extended with the onsite collected data and then to train the SOTA algorithms to detect mobile construction machines. The view of the gathered images is outside of the mobile machines since I believe fixed cameras on the ground are more suitable if all the interesting machines are working on a closed site. Most of the images in the MOMA dataset are in a real scene, whereas some of the images are from the official website of top construction

---

<sup>1</sup> Except some tiny modifications, all the figures, text, and results of the presented work in this chapter have been published in my preprint publications [62]. My contribution to the papers is summarized as 100% in terms of conception and methodology, 90% of literature review, 40% of realization, 30% of data collection and labelling, 50% of results visualization, and 95% of formulation.

machine companies. Also, I have evaluated the performance of YOLOv3 [181] on the selected scenario, indicating that the SOTA computer vision algorithms already show an excellent performance for detecting the mobile machines on a specific working site. The visual monitoring system compensates for the system deficiency in recognizing the participants without a location system, and works as a safety system.

## 6.1 Introduction

The research on the fully and semi-automated driving mobile machines are prosperous in the past decades. Mostly, the introduction of novel technologies aims to increase productivity, enhance the safety of the workers, and reduce the cost of operation. Among these new contributions, computer vision has attracted the most significant attention. Thanks to the boom of the deep learning, recognition capability of artificial intelligence outperform human-level recognition for many tasks.

In the case of mobile machines, which usually work in a closed campus, making the autonomous driving of the mobile machines a level four task according to the standard from SAE [182]. Currently, there are a lot of significant deep learning methods to visually detect the objects of interest, such as YOLOv3 [181], Faster-RCNN [183], which achieve an appealing trade-off between speed and accuracy.

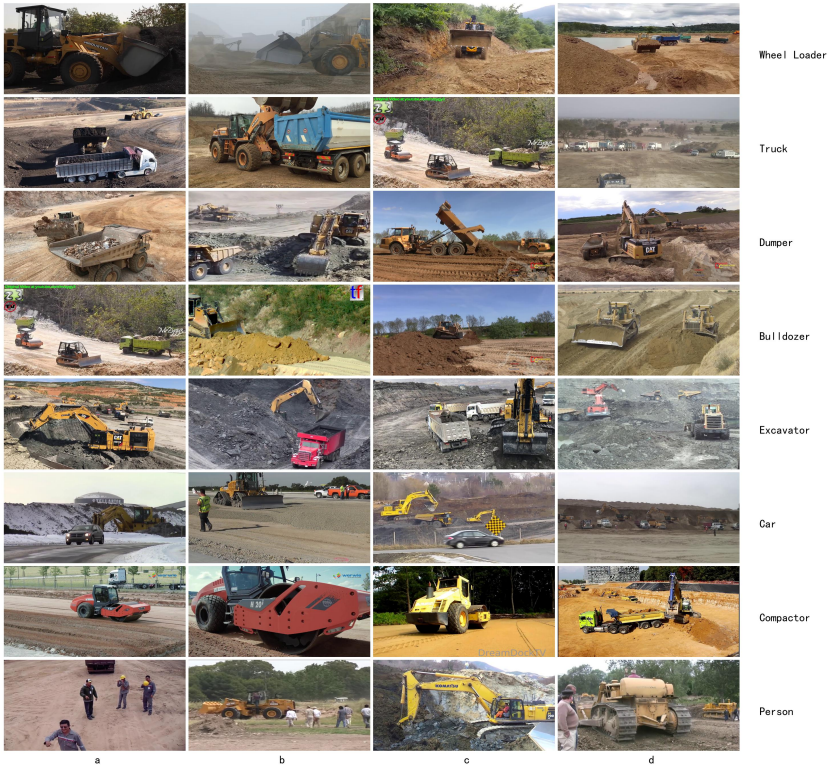
Without a doubt, a series of researchers in the field of construction machines have been explored the possibility of using computer vision technologies and deep learning to recognize mobile machines. Unfortunately, until today (July 2020), no well-known database containing common devices for mobile machines, such as excavators, wheel loaders, bulldozers, and dumpers, is published with easy access and can be downloaded directly. As we know, the success of deep learning mainly benefits from three aspects: the generation of large-scale datasets, the development of robust models, and a large number of computing resources. The

absence of the dataset limits the development of autonomous driving or working of the mobile machines.

To avoid the paucity of well-annotated images about mobile machines in current public datasets, a specific dataset for mobile construction machines is created: the MOMA dataset. Here images from varying viewpoints, poses, partial occlusions, and changing the depth of field were collected. A diversity of eight common categories across 5,663 images was organized in the standard PASCAL VOC dataset. 19,977 object instances were labeled for the research in the dataset. Based on my challenging dataset, by adding only a few newly gathered data onsite to achieve good detection within the closed campus becomes possible. I anticipate spurring the mobile machine detection to a higher level with a well-prepared based dataset. Fig. 6.1 illustrates the samples inside of my dataset.

The highlights of this chapter can be concluded as follow:

- A base dataset contributing to detecting the commonly used mobile machines is created. The structure of my dataset is similar to PASCAL VOC for the convenience of the researchers in the fields of computer vision and deep learning.
- The MOMA dataset is challenging: some instances are quite difficult to recognize and can only be detected with context information. There are many instances in figures since the working sites can be dense. I consciously selected these challenging figures to make the results tested on MOMA dataset plausible. The background of most of the figures in my dataset is from real construction sites. In this fashion, I guarantee the same distribution between the training data and the test data to the greatest extent; thus, guarantee the performance of the winning model in the real-world application.
- I show that the current SOTA solution for common objects can also have a high performance to detect mobile machines in restricted area with the help of my dataset, i.e., rather than considering the mobile machines as a specific



**Figure 6.1:** Sample images in dataset MOMA, including 7 classes of construction machines as well as person with varying poses on the working scenarios. Images in column (a): objects in iconic view; column (b) objects under partial occlusion; column (c) objects in varying poses; column (d) objects in non-iconic perspective.

task and try to develop more suitable algorithms, making an appropriate dataset can be an alternative to achieve the high-level detection task.

- As the tasks in the field of construction machines are level four, adding some custom figures of machines that need to be tested into my dataset can surely increase the predictor's performance. Thus, I also developed the program to analyze the modified dataset.

The rest of the chapter is organized as follows: I first briefly introduce the previous studies of computer vision based algorithms and datasets for common objects and construction machines. I then present the MOMA dataset in the following section with detail. Next, I analyze the performance of YOLOv3 on the MOMA dataset and give the current feasible solution for the construction industry, i.e., show how to leverage the dataset to detect mobile machines in practical. Finally, Section 6.6 gives conclusions of this chapter.

## **6.2 Related Works**

### **6.2.1 The Well-Known Datasets**

Data are the prerequisite cornerstone of deep learning because deep learning models directly get knowledge from the data. Although the importance of the dataset is not so significant before 2012 [184], the deep-learning community has the consensus that the data have been the vital driving force behind computer vision technology [185, 186]. To circumvent the bottleneck of limited data, both Acuna and Yu proposed a method to accelerate the human labeling process by their software [187] or a partially automated labeling scheme [188]. Besides these, Northcutt proposed an approach named Confident Learning (CL) to evaluate the quality of the data [189]. With the rapid development of computer vision, the dataset of image recognition is also enlarging at a rapid pace. For the classification, Caltech 256 is famous with more than 100 categories [190]. Also, many scientists published a classification dataset based on videos, such as [191]. Besides

that, more commonly applied datasets for general purpose are created, including PASCAL VOC dataset [192], Microsoft COCO [193], and ImageNet [194]. Also, there are a lot of specific datasets for specific tasks, including pedestrian [195], scene parsing [196, 197], human activity [198, 199], and face recognition [200]. In autonomous driving, KITTI is considered the pioneer, which contains objects of interest in the realistic scenarios of city Karlsruhe [201]. Followed by Cityscape [202], RobotCar [203] have also contributed to the autonomous driving community with their diverse dataset. Since the aforementioned automated driving datasets are collected in European countries, the scientists in the other parts of the world also published their dataset with much larger sizes, concretely they are BDD 100K [185] in the USA, ApolloScape [204] in China, and nuScenes [205] in both the USA and Singapore.

By the comprehensive literature review, it can be concluded that a benchmark dataset should be diverse, abundant, consistent with the actual scene, and online release.

## 6.2.2 Recent Object Detection Algorithms

Computer vision is a grand and long-standing subject. Before 2012, the most outstanding algorithms are based on hand-crafted features, such as Histogram of Oriented Gradient (HOG) [206] and Scale-Invariant Feature Transform (SIFT) [207]. In this period, a famous algorithm based on convolutional neural networks is LeNet [208]. However, due to the limitation of the level of computer computing technology at the time, it is quite shallow and with too few training parameters. Therefore, the advantages of deep learning at that time is not significant. Nevertheless, after the AlexNet [209] won the ImageNet challenge [194], so-called large scale recognition challenge, in 2012, the deep convolutional neural networks have attracted much research attention in recent years. In 2014, the VGG-16 [210] was proposed, and it is used as the base network for many applications. After that, the inception network [211], which combines the most of deep learning ideas, is designed. Among them, a particular form is called GoogleNet. Moreover,

He shows that the neural networks can even surplus the human-level recognition [212], and he invented the ResNet [213], including the concept skip connection, making the training of much deeper neural networks possible, because the identity function is easy for the residual block to learn, in the same year. Usually, deep neural networks have a large number of training parameters and thus need plenty of time to be trained. To address this problem, transfer learning has got attention. The training time on the specific tasks can be dramatically reduced through transfer learning compared to if the whole model is trained from scratch. Therefore, instead of directly training the total model, most of the researchers download the pre-trained ImageNet models. Until the time I wrote the content of this chapter, the most well known and successful computer vision algorithms to detect objects are YOLO, RCNN, SSD, and their variations. Redmon developed from YOLO [214] to YOLOv2 [215] and then to the YOLOv3 [181], whereas RCNN [216] was enhanced to fast R-CNN [217] and faster R-CNN [183]. The comparison among these algorithms was made and can be found in many scientific papers, such as [218]: thus, here I only make a brief summary. Since YOLOv3 is a one-stage method and solves the task as a regression problem, it is quicker and famous for realtime capability. In contrast, faster RCNN adopts the region proposal network and achieve slightly higher accuracy in most competitions and tasks. In this chapter, I adopt YOLOv3 for my system since YOLOv3 reduces the burden of hardware.

### **6.2.3 The Previous Contributions on Detecting Mobile Machines**

To date, besides some common purpose, computer vision is used in many specific applications, such as airplane detection [219], ship detection [220], and of course, mobile construction machines.

The idea of using a camera to recognize mobile machines visually is not novel. To the best of the authors' knowledge, the first research can be traced back to 1990 when Eldin want to use a camera to increase the productivity of construction

of a state prison in the USA. Before the rise of very deep neural networks, a series of researchers have already reached some achievements in these fields. Azar has developed a model for non-rigid equipment of excavators detection and pose estimation in construction images and videos [221]. In 2011, Chi used a background subtraction algorithm to extract motion pixels, which are then grouped into regions. After that, the group will be identified using classifiers [222]. The dataset, comprising of 750 images, is equally divided into three classes: skid steer loader, backhoe, and worker. It achieved overall classification errors of 3.9% with neural networks. The research also pointed out the similarities between loader and backhoe may cause worse performance. Both Park and Memarzadeh presented a method that can be concluded as a combination of HOG and the HSV color histogram, to localize construction workers or equipment in video frames [223, 224]. In 2014, Tajeen mentioned in their paper that they built an image dataset for construction equipment recognition, including 300 images [225]. After the Convolutional Neural Network (CNN) success, the application of the CNN-based object detection in detecting mobile machines and construction sites has been undertaking over the past decade. A consensus in the mobile construction machines industry has been built: for a variety of image recognition tasks, well-designed deep neural networks have far surpassed previous methods based on artificially designed image features. Fang uses Improved Faster Regions with Convolutional Neural Network Features (IFaster R-CNN) approach to detect the excavators and workers in realtime on their own dataset [226]. Kim did both the research about scene parsing [227] and objects detection [228] of construction machines. In their following researches, the estimated context information was used to reduce the cost of the earthmoving process [229]. In 2019, Son used a very deep neural network to detect the workers in the working site, which was claimed to have yielded an accuracy of 91% and 95%, exceeding the SOTA descriptor in image target detection methods at that time. In his paper, he emphasizes the importance of varying poses and changing background [230]. Also, Son points out that the visibility of the equipment operator is inherently poor [231], which is consistent with my point of view. Recently, Bang proposed an image augmentation method to enhance the performance of objects detector on construction sites,



achieving a recall of 66.76% and precision of 53.08% experimentally on the UAV-based resources [114].

Based on the literature review, I find that the research from Kim [228] who also aims to detect mobile machines is mostly similar to my research. Besides mobile machines detection, he claims that they have build up a dataset based on the images in the ImageNet. Since the R-FCN is powerful and the dataset is relatively large, i.e., 2,920 samples, no wonder they can achieve excellent performance. However, the dataset cannot be used as a based dataset for a reason. The background of the samples gathered from ImageNet is mostly not a real working site. This makes the winning algorithms in this dataset may not have an excellent performance in practice due to the dramatic domain shift.

## 6.3 Why I Created the MOMA Dataset?

As I mentioned, for a level 4 task, the generalization capability of the detection model for the objects outside of the specific area can be ignored. Obviously, reduce the difference between the training data and test data can definitely improve the performance of AI model. Also, the size of training dataset should not be too small to lose the necessary information. Hence, I created the MOMA as a base dataset which shall be used to be mixed with the newly gained data directly from the working site that will be monitored. In this fashion, the data from the MOMA give more general information while the newly gathered data provide more test-data-related information.

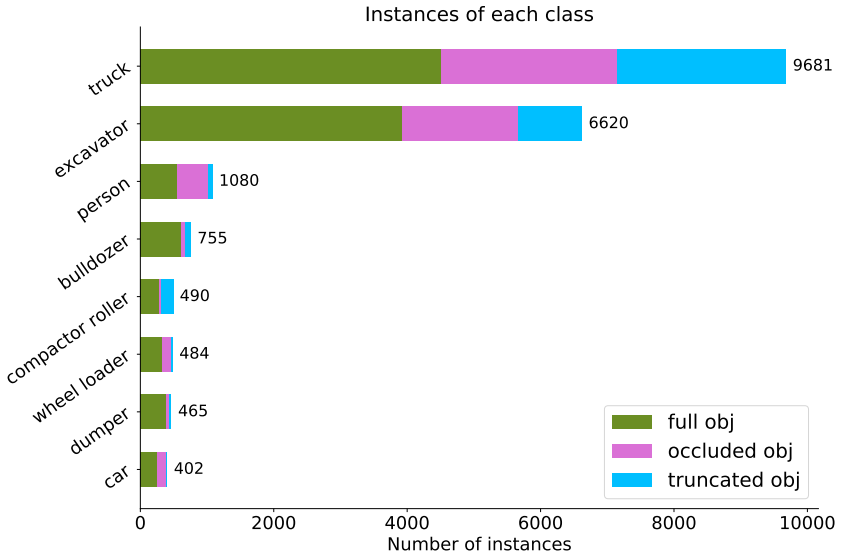
## 6.4 The MOMA Dataset

In this section, I first summarize my steps to build the dataset and then describe the details in subsections. The dataset MOMA is created as a specific dataset for commonly used mobile machines, which is challenging and diverse. There is one thing worth mentioning; I believe that the cameras on the construction site are

more likely to be fixedly installed on the ground than on the driving construction machines. Because in most cases, construction machinery works within a limited range, making the configuration that install the cameras on the vehicle no more an inevitable method. In addition, the advantages of fixing the cameras on the ground are obvious. First and foremost, the cameras installed on the ground can provide the depth information from the figures with the appropriate calibration of the cameras. Here is the calibration process relatively easy since the coordinate among cameras is constant without vibration. Also, a wider angle of view and a cleaner lens can be achieved. The machines are usually surrounded by the dust during working resulting in the limitation of the vision. Thus, I prefer to select the images gathered from a perspective outside of the mobile machines, which is quite different from the self-driving cars' training images. In this fashion, wireless communication should be developed for information sharing between machines and cameras. These researches can be found in [64, 40]. Consequently, a diversity of eight common categories across 5,663 images was organized in the form of the PASCAL VOC dataset. 19,977 object instances were labelled for the research in the chapter, see Fig. 6.2.

Based on the survey about the most vital participants in the working site, I clearly defined the categories that I should focus on as the first step before collecting data. Unlike other categories, mobile machines vary to a certain extent depending on the components and working conditions. On top of that, human beings must be included in the dataset since most of the researchers believe the accurate detection of humans in the working site can improve security. Therefore I limited the species of detection tasks to common representative groups: excavator, truck, dumper, bulldozer, wheel loader, car, compactor roller, and person.

In order to guarantee the algorithms trained on my dataset can really have the best performance in practice, I collected the candidate images both from video frames and the official website of construction machines. The streaming video files were collected under the different real scenarios, which makes my dataset closer to the actual situation on the working site, and I then cut them into images. Besides the images from the videos, I also gathered some figures directly from the website of famous construction machines companies, such as Caterpillar, Komatsu, with the



**Figure 6.2:** The statistics of the MOMA dataset.

help of chromedriver and web crawler, since I believe that introducing these figures can enhance the performance of the predictors. Apparently, the figures from the videos are in a non-iconic view like the figures in MS COCO. In contrast, the figures from the official website of construction machines companies are canonical perspective as the samples in Caltech. Both of them make significant contributions to ensure a relatively high recall. Finally, for the visual perception task, more than 25,000 images were gathered, from which 5,663 representatives were selected.

Following, I annotated the ground truths of determined classes in the selected images. I use the annotation tools “labellmg”, which is mainly for object detection labeling work from Lin [232]. The software can generate both XML files for Faster-RCNN and text files for YOLOv3. Since the XML file contains more information than txt, I save the dataset in XML and then transfer the XML into txt. As we know, labeling effort helps a dataset stand out in the training evaluation and detecting performance as well, whereas missing labels, false annotations,

even widely unbalanced instance distribution, and too many clutters impair the effectiveness and robustness of a dataset. Therefore, before the dataset is fed into training models, it is worth analyzing the dataset by means of statistics and subsequently split it into subsets aiming to train the predictor and cross-validation. Whenever the dataset shows a significant imbalance among the interested categories, it would probably weaken the performance as a result. In this case, countermeasures such as label deficiency examination and then moderate supplement must be taken to keep the predictor robust against all classes. After careful preparation, the MOMA dataset basically does not have such a problem; however, considering that I need add onsite gathered data into my dataset, I have created the tool to evaluate the balance of classes in the dataset.

Besides the balance among different classes, it is quite necessary to have the right balance between training and test set to gain a stable estimation of predictor performance. With less training data, the trained model tends to have a bias problem. In contrast, less testing data will lead to higher variance concerning the performance statistics. I randomly split the dataset into trainval, i.e., training and validation, and testing by a ration of 4:1.

Finally, the richly-annotated dataset is tested by the SOTA object detecting algorithm, concretely, YOLOv3. In the meantime, whenever I find that the detectors do not work well for a specific situation, I increase the number of labeled images in that case into my dataset. In this fashion, I increase the diversity and scene variation of the dataset. In addition, the metric mAP is used to evaluate the detection performance. Here I use the recommended parameters and thus set the threshold of Intersection over Union (IoU) as 0.5. An Average Precision (AP) comparison with the best parameter settings is conducted across all selected categories.

### 6.4.1 Data Acquisition

Thousands of images can be easily acquired as I have open access to a search engine and social media, e.g., Google and Flickr. Web images can be found and downloaded by crawling through websites. Hence, a scrapy crawler framework

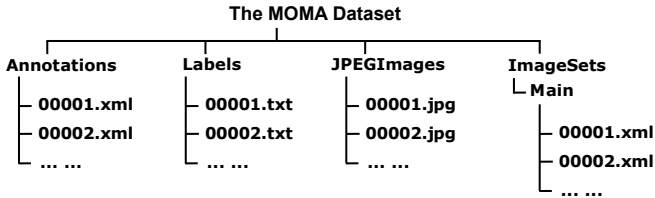
was built to grab pictures from Google search engine and mechanical engineering machinery websites. Special python scripts for each provider were created based on their site's HTML structure. By executing the python file created by Wang [233], images of interests from most pages on the website can be collected.

Nevertheless, most search engine based images present a canonical view of objects, which could bias the algorithm to assume mobile machines are always located at the center view. This may lead to a deviation from the predictors' optimal performance if they are trained only with these images. Despite their weakness in the real inference, the web-based images from various providers show diversity in size, luminance, resolution, color, background, as well as ambiguity and thus help models gain an understanding of essential object features. Moreover, in fact, most construction machines providers publish their new models timely on their website; thus, adding these figures can enhance the predictors' recognition capability. Since these figures are quite easy to be detected and thus may exaggerate the performance of detectors, they do not include in the main dataset of the MOMA. However, they are well prepared and saved in the additional file in my dataset for training.

By demonstrating the multi-angle and realtime working status of mobile machines, videos strengthen the generalization of predictors in realistic working surroundings. By appropriately extracting the images from videos every 50 fps records, I build up the non-iconic part of the dataset. In this part, I consciously select the videos varying the machinery working poses as well as the machine size due to the depth of the perspectives. Since the images are collected from realistic scenarios, occlusion and truncation are inevitable. In this fashion, thousands of images can be produced, making the detector feasible in various practical scenarios and, of course, realtime detection. A volume of 20,895 images was captured from 125 videos, and 5,663 from them were picked out for training the models and their validation.

## 6.4.2 Dataset Format

I use PASCAL VOC format as the exemplar dataset format for my task. Fig. 6.3 illustrates the structure of the MOMA dataset.



**Figure 6.3:** Hierarchical structure of the MOMA dataset, based on PASCAL VOC.

Similar to PASCAL VOC, directories “Annotations”, “labels”, “JPEGImages”, and subdirectory “Main” under “ImageSets” are the essential components, with relevant files in them. During the implementation of Faster-RCNN on the MOMA, file types such as XML, jpg, and files train.txt and test.txt in “Main” are in the necessity, while YOLOv3 detector will be trained with label text files, jpg files, and train.txt and test.txt in folder “Main”.

Labeling is non-trivial work, to avoid duplicating the creation of rectangular boxes and annotating them, the dataset was initially built only in the format of XML. SOTA object detection algorithms such as Faster-RCNN, SSD, YOLOv3, etc. require basically the same essential annotation information of targets of interests in two-dimensional images, including their coordinates and categories, which are generally expressed in the form of (left, top, width, height, and class). Although ground-truth targets were merely labeled in the format of XML to save labor work, text annotation can be transformed by program correspondingly. During the transformation, the location information for every objects is translated from  $(x_{min}, y_{min}, x_{max}, y_{max})$  to  $(x_{center}, y_{center}, w, h)$  to fit the two algorithms respectively. Besides, in the annotation, all the coordinates, width, and height are normalized, range from 0 to 1. Therefore attention should be paid whenever parameters for  $x, y, w, h$  are calculated. For instance, a constant image size must



**Figure 6.4:** Labelling tool. Label tool “labelImg” can load multiple images under a directory by clicking “Open Dir” on the menu, and save under pre-defined path. The shown bounding boxes that closely surround object ground truths, which the target instance is excavator in this figure, were made bold for the salience. Multi-class and multi-label for one single instance are possible; all marked labels are at the top right corners. As I consider the PASCAL VOC format as the standard format for the the MOMA dataset, all annotation files are saved in XML format.

be multiplied in the optimization process by k-means clustering of the annotated anchors, because the anchor centroids are measured in pixels.

### 6.4.3 Manual Annotation

Labeling is exhausting and costly to perform but is the prerequisite in the task of object detection; all the aforementioned annotation files such as XML files in the directory “Annotation” have been labeled manually. I used the label tool “labelImg”, which is a famous graphical image annotation tool available in GitHub repository from Lin [232], to accomplish the labeling job. I annotated every single object in an image with a bounding box, enclosing the ground truth of objects and marking the class each object belongs to. Fig. 6.4 illustrates the graphic interface of the label tool “labelImg” and an annotation sample.

The saved XML file for the image annotated as in Fig. 6.4 is represented in following code. It comprises all the ground truth information that I need to train the neural network with the samples.

```
<annotation>
  <folder>images</folder>
  <filename>sample.jpg</filename>
  <path>\path\to\the\sample.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1280</width>
    <height>720</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>excavator</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>561</xmin>
      <ymin>52</ymin>
      <xmax>1001</xmax>
      <ymax>382</ymax>
    </bndbox>
  </object>
  <object>
    <name>truck</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>394</xmin>
      <ymin>344</ymin>
      <xmax>704</xmax>
      <ymax>537</ymax>
    </bndbox>
  </object>
```



```
<object>
  <name>bulldozer</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  <bndbox>
    <xmin>694</xmin>
    <ymin>395</ymin>
    <xmax>950</xmax>
    <ymax>612</ymax>
  </bndbox>
</object>
</annotation>
```

Here I summarize the most decisive info tags in XML that should be kept when transforming the format into txt files.

- **Filename:** Name of the image file, in accordance with the text file under the path “MOMA/Annotations/”.
- **Size:** Width, height, and depth of the image. Depth refers to the three image color channels: red, green, and blue. Images of this size will be fed into the convolutional neural network models for training or detection.
- **Object:** Including the class of the object and location. The “bndbox” stands for the bounding box, which is expressed in four coordinates (xmin, ymin, xmax, ymax). If multiple objects fall into an image, then all their specific names and corresponding positions will be recorded in the XML file. For instance, in the XML I showed, there are three objects: excavator, truck, and bulldozer.
- **Difficult:** The objects in an image may sometimes be quite challenging to be detected only with the current image even for humans. For instance, if a truck is away from the camera, it will gradually become smaller and smaller so that it will become unclear in the end. To recognize the unclear spot, the previous frame must be used, i.e., the contextual information. In this case, I label such a sample as difficult.

To decrease possible interference by potential ubiquitous noise, every single object of interest in an image, including occlusions and truncated instances, was labeled with care whenever human eyes can spot them. Cases of occluded and truncated objects are also counted as ground truths. Here I follow the idea from Yu [185] that the images should be specially pointed out if the cases are occluded and truncated objects. Concretely, I annotated a truncated excavator as “excavator, t”, and an occluded excavator as “excavator, o”, since LabelImg does not have the function to give this selection. The purpose of this method is to propel more robust algorithms. For the implementation of YOLO or Faster RCNN, I create a program to cancel these suffixes.

To ensure the quality of my dataset, consistent rules were made for the labeling process as in the following items:

1. The label box size must be appropriate, i.e., the rectangular box should wrap the target closely. The rectangular box needs to contain information that distinguishes between different types of targets.
2. Although a ground truth target may be blocked, it still needs to be marked as long as the human eye can identify the target. This improves the generality of the model. In the actual application scenario, there will be many obscured targets that the model, even so, should detect.
3. Small size targets cannot be missed if they are identifiable. SOTA detection algorithms are capable of multi-scale object recognition; thus, annotated tiny size ground truths boost the detection performance consequently. Fig. 6.6 depicts a labeling specimen of a far-off excavator behind two persons.
4. Targets that human observers cannot recognize should be ignored. Otherwise, they will mislead the neural network. Only when humans can recognize them with the help of context information, I will label them and mark them as difficult.



**Figure 6.5:** Label example. Two excavators and two trucks should be labeled in this image. They can all be distinguished by eyesight even though they are partially blocked.



**Figure 6.6:** Label example. Four objects can be clearly seen in the image; even the excavator in the distance is much smaller than the one nearby. Moreover, the two standing workers can be recognized as well.

## 6.4.4 Dataset Splits

As mentioned above, the dataset is meant for training as well as testing. Therefore I grouped it into four subsets randomly to ensure the training set and test set coincide in the data distribution. In order to achieve a stable estimation of model performance, a reasonable balance between training and test set is required. Depending on the volume of the database, it is quite flexible in determining the partitioning scale. Practically it gains better performance with a smaller proportion of testing set when the size of the dataset is larger. Based on the data amount, I split the dataset into approximately 80% for training and validation, and 20% for testing.

The arrangement of the data division is illustrated in Fig. 6.3. In txt file like trainval.txt image file names with a suffix are stacked. Literately, all images in accordance with names in trainval.txt are intended for training and validation. Likewise, I test predictors using the images regarding the names in test.txt. Data inside trainval can be further split into training and validation subsets. Above all, these four groups work together to make full use of the complete database in order to gain satisfying predicting performance.

## 6.4.5 Data Preprocess

As a consensus, clean data helps improve detection performance. Prior to the implementation of CNNs, the dataset is analyzed statistically to strike out ineffective labels and ensure its conformity with the working scenarios. This is an essential step since the dataset will be modified to better suit other tasks.

To list annotated labels by the annotation tool “labelImg”, a specific Python script was written. By executing the script, a list of class/count pairs would be printed, e.g. (excavator 536). In the case of typo error labels such as “excavater”, further steps must be taken to rewrite the revised class into the XML files. The program that can automatically find the error was built [233].



**Figure 6.7:** Label example. Rectangular correctly encompass ground truths: a dumper, an excavator as well as two persons. However, other trucks can be inferred from context stream frames but are not identifiable in the single image. The label should be ignored if YOLOv3 is used to detect mobile machines.

Moreover, difficult spotted instances should be averted for YOLO and faster RCNN. Fig. 6.7 depicts annotations of unrecognizable trucks and person, which are marked as difficult and need to be removed for YOLO and faster RCNN [233]. In contrast, for other algorithms, these marks may be useful.

In this section, a special dataset MOMA for the CNN-based visual perception of mobile machines was created, and preprocessing in necessity is also introduced. Instead of using my dataset to train the predictors from scratch, I use Darknet-53 trained on ImageNet as a base network.

## 6.5 Evaluation of the Recent Computer Vision Algorithm Performance on the MOMA Dataset

I would like to encourage the engineers from construction machines to use the similar idea and take advantage of the computer vision technologies for their application. Here, I evaluate the effectiveness of the visual-based safety system and show the model setup. Since many mobile machines predictors have been built with Faster RCNN, I do not show the setup of Faster RCNN again to avoid redundancy. Here I only demonstrate the implementation of YOLOv3.

Object detection tasks demand high computational power, and for some practical cases such as video stream recognition, powerful computing devices are needed. Since Google offers a graphic computing platform on which both models can be trained much faster than on a commonly used local laptop, I tested my monitoring system trained on Google colab where the GPU is free to use. This is to prove that the high performance computer is not necessary. Nvidia GPUs boost the calculation by taking advantage of CUDA. Since mAP performance does not differ much, up to 1%, between different GPU series even with different image scales or non-identical mini-batch sizes, the trained weights can be used on other platforms. All the environments, including GPU, are set up in the configuration files. The framework Faster-RCNN can reach a Frame Per Second (FPS) of 5, while YOLOv3 at about 45 on Tesla k80, which is offered for free by Google. As a comparison, a Nvidia GTX 1050 used in a mediocre laptop can achieve an FPS of 10 with YOLOv3.

The original YOLO algorithm was uploaded by Joseph Redmon on his website. Afterward, several revised versions came out in different programming languages and updated in quite different aspects. In my work, mostly the original version is applied. However, to extend some features, another prevailing repository is



**Figure 6.8:** Prediction samples with optimized dataset and algorithm YOLOv3.

preferred as well, concretely, I use the version from Bochkovskiy, whose code can be found on his Github<sup>2</sup>.

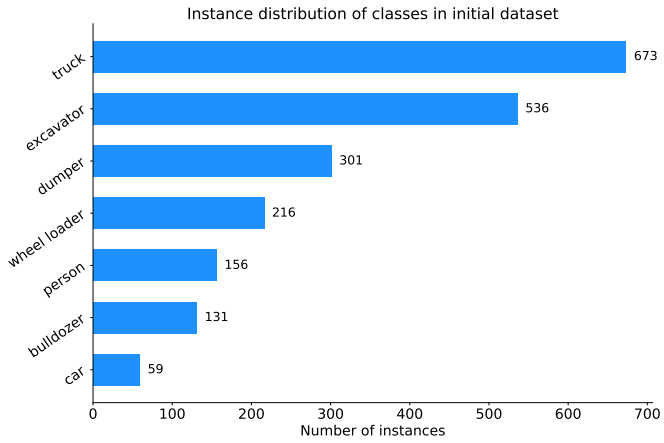
Ideally, for each category to detect, there should be at least one similar object in the training set, which should comprise likeness of shape, relative size, point of view, tilt, illumination, etc. of the targets. From that perspective, the larger the dataset, the better the detectors will be. However, it may take a long time to train the large dataset with the default settings in the configuration file. On this point, it might make the construction machine engineers flinch from the chance to use computer vision algorithms. Also, even with SOTA solutions, based on the test results on

<sup>2</sup> [https://github.com/AlexeyAB/darknet/tree/darknet\\_yolo\\_v3\\_optimal](https://github.com/AlexeyAB/darknet/tree/darknet_yolo_v3_optimal)

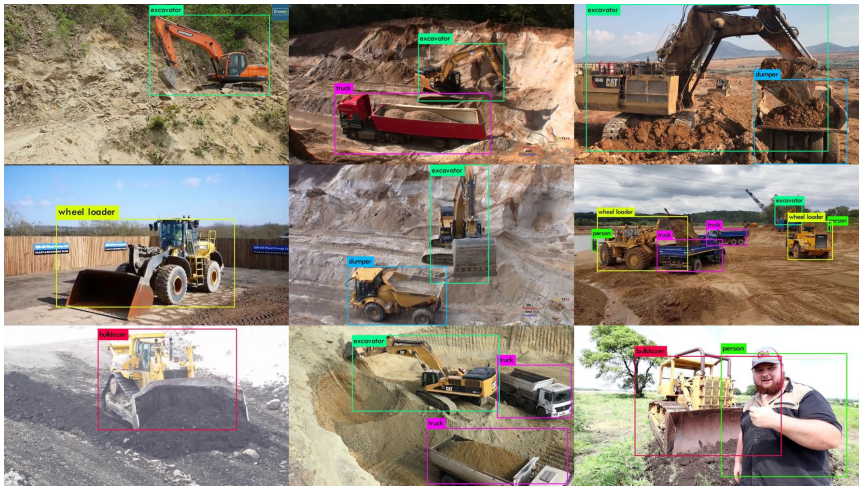
MS COCO with IoU of 0.5, the best mAP is about 50%. Since my dataset is easier than MS COCO, the test results go to 85%. However, it is surely unacceptable for the construction machines industry due to the safety reasons; it seems like those SOTA solutions should be improved for the detection of construction machines the same as the detection of cars. Alternatively, since the autonomous driving of construction machines is a level four task, which provides the possibility to increase the prediction performance by means of scarfifying the generalization capability, i.e., the performance of the predictor for the specific working site with only limited kinds of mobile machines inside is more important than its performance to detect all the mobile machines in the world; thus, I focus on finding a current feasible solution for the construction machines industry in the following context. Generally speaking, if the distribution among the training, validation, and test dataset are the same, the predictor will perform its best performance. Besides, the mAP of the predictor can increase when I add some similar objects from different scenarios in the training data. Therefore, I recommend adding some additional annotated images of target mobile machines into the base dataset and further train the model to get the optimal predictor for the level four task detection. To validate the idea, concretely, I take 666 well-annotated images from the MOMA dataset into the network for training as well as validation. The basic idea of this approach is to increase the recognition rate of the target mobile machines by adding some samples of the target machines to be detected in a relatively small dataset to reduce the difficulty of detection. This approach is based on the assumption that no unexpected mobile machines will come to the working site. Obviously, for a closed working site, this assumption is reasonable. The training time is dramatically reduced, and the prediction results are illustrated in Fig. 6.8. The selected ground truth instances are plotted in the histogram in Fig. 6.9.

On the images in Fig. 6.10, every single inference is marked with a bounding box in a different color to specify its category. Categories are labeled in the bounding box over the top left corner. The model appears to have satisfying performance on those images since they are in a canonical view and thus not so challenging.





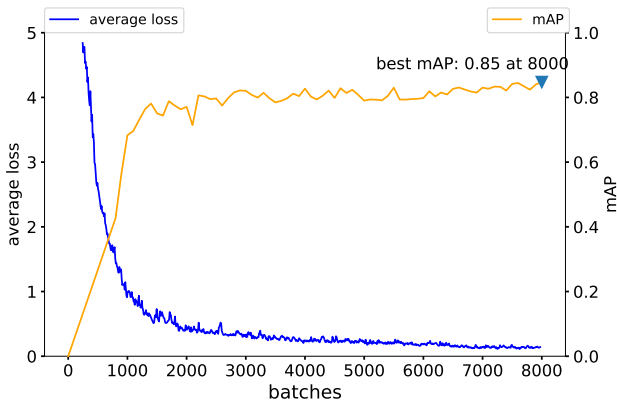
**Figure 6.9:** Class distribution on 666 images: the instances number of truck and excavator outnumber bulldozer and car since they attract more interest.



**Figure 6.10:** Sample of inference results by the 8,000th predictor on images in iconic view.

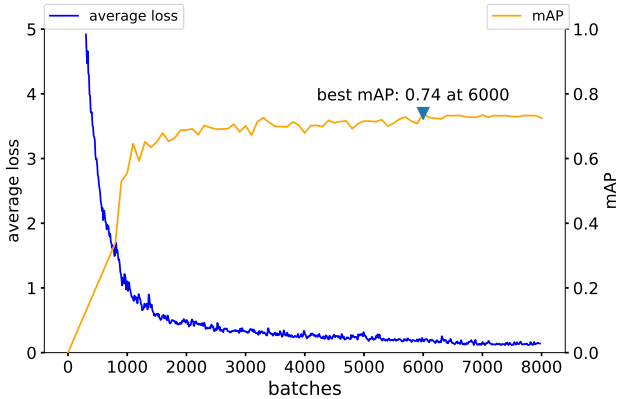
Although the predictor with default configuration can easily achieve excellent accuracy on iconic images, it cannot have a satisfying performance on images in non-iconic view, which are not taken from a normal perspective, i.e., with truncated, or blocked by other objects. An image can also become non-canonical when the whole image is obscured or ambiguous, or targets such as excavators are working surprisingly, e.g., sitting in the water. With the default setting of YOLOv3, the optimal performance may not be achieved. To address this problem, here I would like to share some useful tricks to improve the training process and the mAP of the YOLOv3 algorithm.

First of all, by comparing the results from Fig. 6.11 and Fig. 6.12, higher mAP performance can be achieved with a relatively balanced training dataset, i.e., the quantity of each class should not differ too much.



**Figure 6.11:** mAP over batches, trained with a balanced dataset.

Second, according to the setting of YOLO, the multi-scale prediction is applied in feature maps. To narrow down the computing without hurting the prediction performance, k-means was implemented [233] to cluster the centroids of the positions of all the labeled objects. Instead of using the default anchor for the dataset COCO, I generate nine anchors as (16.0,26.0), (40.0,40.2), (30.8,84.4), (71.8,84.2), (119.6,124.2), (105.0,219.0), (191.6,175.2), (200.0,290.6), (322.6,346.6).



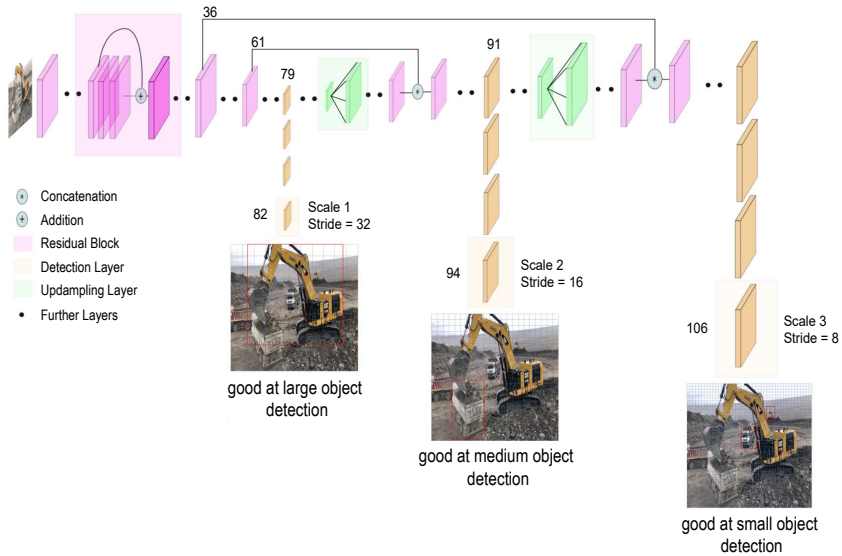
**Figure 6.12:** mAP over batches, trained with an unbalanced dataset.

Third, following the expectation that more training batches but with smaller learning rates could improve detection performance, I decay the training steps after the average loss begins to fluctuate. Concretely, I set the learning rate as follows,

```
steps=8000,10500,12000
scales=.5,.1,.1
```

Here step learning rate decay of 0.5, 0.1, and 0.1 are applied at the 8,000th, 10,500th and 12,000th step, respectively. Usually, it would be sufficient with 2,000 batches for each class, and no less than 4,000 iterations in total, training work can be then stopped. Also, the learning process can also be stopped when the average loss no longer decreases. After 12,000 training steps, the average loss function converges to no more than 0.1, a quite adequate condition to stop.

With the new predictor, I run inferences on the no-iconic images, which are shown in Fig. 6.14. Mobile Machines like excavators are usually in large size, and predictors may quickly get used to that dimension; however, if target excavators are zoomed out or seen from an irregular perspective, they can be small objects as well. Based on my experiments, the inference ability concerning classification and localization of the predictor on the first three images has been greatly enhanced



**Figure 6.13:** Hierarchy of predictor with skip connections, e.g. 94th layer, responsible for detecting medium-size targets, relates to the 61st layer before downsampling. Likewise, 36th layer is directly connected to 91st layer by a short cut.

compared to the predictor with the default setting that only found out the large objects in the middle of the figures. It remains blind to the excavator in the last image of Fig. 6.14. Regrettably, nothing is found even though human observers can easily discern the mobile machine (an excavator) on the left. Possible reasons for that are the lacking of remarkable characteristics and its tiny size. More images of this size and pose should be added to improve the identification capability. Although some instances are still not detected, the holistic performance of the predictor is satisfying since a shorter range deserves more attention.

Further differences among the three predictors at the 1,900th, 8,000th, and 12,000th are shown in the  $2 \times 4$  image grid. In Fig. 6.15, a and b are raw images, a1 and b1 are predicted by the first predictor at the 1,900th batch. Similarly, a2 and b2 are from the second predictor at 8,000th. At bottom a3 and b3 are from the third predictor at 12,000th batch. Apparently, the bounding box



**Figure 6.14:** Inference by predictor at the 12,000th batch made on images with a non-iconic view.

surrounds “dumper” closer as the training steps increase in Fig. 6.15, indicating that the predictor has 12,000 batches acquired a more powerful ability to localize the targets. Besides that, the dumper, which is in the blue bounding box, is recognized, and the fictitious noise of the truck is eliminated, which implies that the class confidence increases with the more trained predictor.

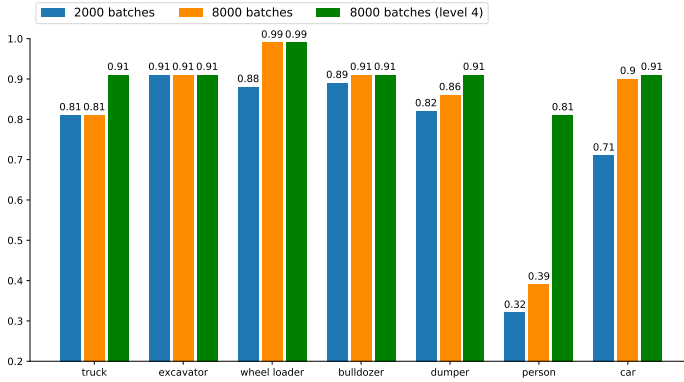
Fig. 6.16 shows the specific AP values on each class predicted by predictors under the different situations. Their trend illustrates AP increases with more batches for most of classes. Here is the test data quite similar to the validation data; hence, the predictors may overfit to the mobile machines that exist in the training and validation data. Although these results exaggerate the algorithm’s real ability, it can accurately reflect its performance on the fourth level of autonomous driving.

Although it might make no sense to show the generalization capability of the predictors since the assumption that no unexpected mobile machines are in the working site is reasonable, I further tested the predictor with 8,000 batches on other 5,663 images in the MOMA because I would like to show the performance of my method if the level 4 condition do not hold true. From Fig. 6.17, it can

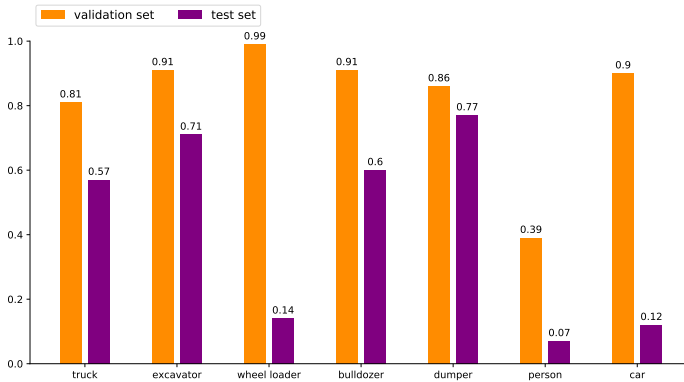


**Figure 6.15:** Prediction performance contrast by three predictors, which were made on images with both iconic view and non-iconic view.

be seen that the AP for each class goes much lower. Although the classes person and car are the lowest, it is predictable since I have fewer samples in these two classes. As a counter measurement, samples from other datasets can be added in the base dataset, and thus it cannot lead to a problem. The other colossal gap is the wheel loader. By analyzing the precision-recall curve, I found that the false-positive dramatically increases as the confidence threshold decreases, resulting in an extremely low AP. Moreover, I further analyzed the false detected samples. I found that most mistakes are the excavators with a shovel facing forwards since they are reconfigured for mines, or some trucks are very close to the cameras so that the wheels are extremely large. These features are not including in the small



**Figure 6.16:** Individual prediction AP for each class made by predictors at batch 1,900 in blue, 8,000 in orange. The green column demonstrates the performance if I take the assumption that only known mobile machines are working in the working site. i.e. level 4 autonomous driving case.



**Figure 6.17:** Individual prediction AP for each class on the other 5,663 images.

subset dataset of the MOMA; thus, this typical wheel loader's features let the model believe it encounters a wheel loader. Based on the analysis, I add some mispredicted samples into the training and validation set, and the AP of wheel loader increases then to 0.6. In this way, I rely on a minimal data set to achieve good results on a specific site, though overfitting occurs.

## 6.6 Conclusion

In this work, I validated the feasibility of creating a visual monitoring system increasing the safety of participants in the closed working site. To create the monitoring system, I build the MOMA dataset, a large-scale and diverse construction machines detection dataset with ground truth label. Most of the images are gained in real scenarios on the working site, while some other images are downloaded directly from the official website of construction machine companies. Instead of gathering the images in the drivers' view, I collect the samples from the outside view of the mobile machines since I believe it is more in line with the actual situation of autonomous driving of construction machines. With my dataset, YOLOv3 is possible to detect mobile machines with mAP of 85% (Fig. 6.17) in general, which is much better than the previous works without using the deep learning algorithms. Notice that I only compared the researchers who have confidently published their code. Also, without considering the instances outside of the specific working site, the mAP goes to almost 90.7% (Fig. 6.16), which indicates that the predictor is ready for a level four autonomous driving task. Since YOLO is more friendly to real time applications, I recommend adopting this algorithm for the recognition task of construction machines. Finally, recognition performance depends on the dataset quality and how people train the algorithms. By further expanding the data collection and annotation, more satisfying results can be expected. Hence, I also recommend adding the images of interest, such as the excavators or dumpers that are going to be detected, into my MOMA dataset and further train the pretrained model to get a predictor, which is the best suit for the specific level four task.



## 7 Wireless Communication System<sup>1</sup>

The fleet management of mobile working machines with the help of connectivity can increase not only safety but also productivity. However, rare commercial mobile working machines have taken advantage of V2X communication. Current mature wireless communication technology can be roughly divided into ad-hoc network and cellular network. In this chapter, I suggest that both IEEE 802.11p and 5G should be implemented for fleet management. In the first part, I proposed an analytical model for machines to estimate the ad-hoc network performance, i.e., the delay and the packet loss probability in realtime based on the simulation results I made in  $ns - 3$ . The model of this part can be further used for determining when shall ad-hoc or cellular network be used in the corresponding scenarios. Afterward, I demonstrated the scenarios where 5G can have a significant effect on the construction machines industry. Also, based on the simulation I made in  $ns - 3$ , I compared the performance of 4G and 5G for the most relevant construction machines scenarios. Last but not least, I showed the feasibility of remote-control and self-working construction machines with the help of 5G.

---

<sup>1</sup> Except some tiny modifications, e.g., sequence of the text, all the figures, text, and results of the presented work in this chapter have been published in my publication [40, 41]. My contributions to these paper are summarized as 90% and 80% in terms of conception and methodology, 90% and 90% of literature review, 80% and 40% of coding, 80% and 40% of results visualization, and 95% and 95% of formulation, respectively.

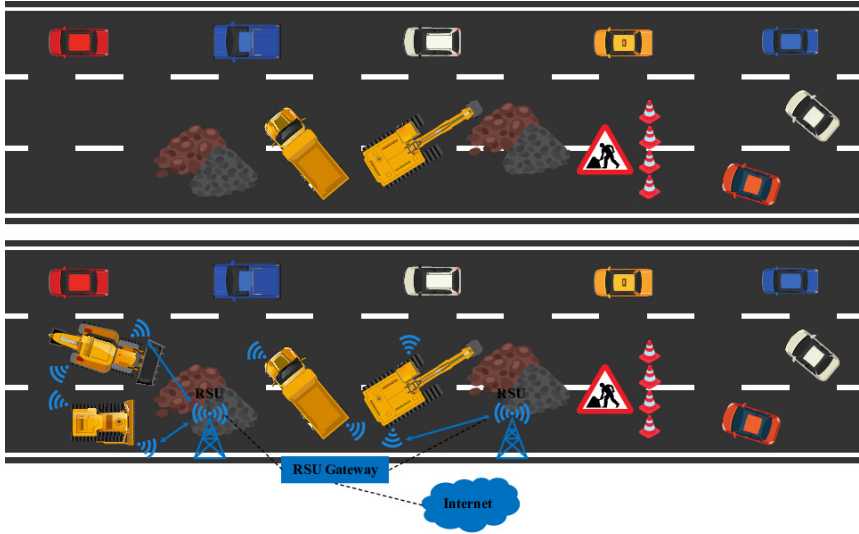
## 7.1 Introduction

Besides artificial intelligence [63], the fleet management of mobile machines is the principal research direction of the IoT in the fields of mobile working machinery. Currently, the mobile machines are distributed sparsely in the working site and working at low transport speed to avoid a collision. With the vehicle-to-everything (V2X), the information about current position, speed or even destination and task are exchanged periodically between individual mobile machines. Since the intentions of neighbor mobile machines within sensing range are known, the working machine can work more densely and transport the material more efficiently. The most challenging and research-worthy use case can be described as the task of repairing the highway. During repairing the highway, a traffic congestion is usually expected. According to the study from Triantis, traffic congestion causes significant economic losses [234]. Apparently, by investing more machines with the help of V2X technology in a particular site can surely improve the working productivity, so that the economy lost due to the congestion can be diminished. Assuming that, all or part of the vehicles are equipped with V2X, a high channel load of V2X network occurs in the traffic congestion. Thus, the V2X performance decreases, manifesting in larger delay and packet loss probability. In this chapter, I first evaluate the performance of the IEEE 802.11p standard for varying node density rates by means of simulations using  $ns - 3^2$  [235]. Since the simulation model is computationally expensive, I then propose an fast estimation model for mobile machines to predict the mean delay and package loss probability of the IEEE 802.11p-based V2X network.

Fig. 7.1 illustrates the benefits of the implementation of V2X technology on mobile machines.

---

<sup>2</sup> Ns-3 is one of the most widely used software for network simulation. It is open-source, scalable, and actively developed by the scientific community. Moreover, its documentation is excellent. Its popularity and flexibility make me selected it as the tool for my research. Other network simulators usually mentioned are OMNET++, SWANS, NetSim, QualNet.



**Figure 7.1:** Comparison the working site with/without V2X: More mobile working machines on the site, much higher productivity.

## 7.2 Current Wireless Communication for V2X

### 7.2.1 Ad-hoc Networks

The time-efficient and reliable message exchange among vehicles have been a longstanding issue for Intelligent Transportation System (ITS), which aims at enhancing the driving safety management as well as fulfilling requirement for infotainment service. Currently, there are two common used technologies for V2X, IEEE 802.11p and 3GPP Cellular-V2X [236]. IEEE 802.11p is the first standard for vehicular communication [237]. Both ITS-G5 and the Wireless Access in Vehicular Environments (WAVE), which is proposed by the EU and the US separately, amend the IEEE 802.11 standard for vehicular use [238].

In the last two decades, the tremendous evolution of wireless communication technique has paved the way for the materialization of ITS. In 1999, 75 MHz of free

but licensed spectrum at 5.850-5.925 GHz was allocated by US Federal Communications Commission (FCC) for implementation of the Dedicated Short Range Communications (DSRC) exclusively for the vehicle to vehicle/infrastructure communications. In the US, the spectrum is divided into seven 10 MHz channels with 6 Service Channels (SCHs) and a Control Channel (CCH). Compared with the US, the European Union (EU) introduced five channels (5.875-5.925 GHz), where CCH is restricted to safety usage only [239], i.e., Cooperative Awareness Message (CAM). CAM is a periodic broadcast message which contains safety-relevant information, such as position, speed, acceleration. Until the time when the author writes this thesis, the final version of the IEEE 802.11p is the version published in 2010 [238]. IEEE 802.11p is an ad-hoc network that has a mesh topology and thus has shortages such as a limitation to the short communication range, the medium mobility, as well as the contention. The coverage of IEEE 802.11p mainly depends on the transmit power [240], path loss, signal fading, delay spread, Doppler spread, and angular spread [241]. The delay is unbounded, caused by Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [242].

## 7.2.2 Cellular Networks

In comparison with the WLAN-based IEEE 802.11p, C-V2X uses the cellular networks and thus the communication relies on base stations. C-V2X uses 3GPP standardized 4G Long Term Evolution (LTE) or 5G mobile cellular connectivity [243]. As Vukadinovic pointed out, the C-V2X is a developing technology, from 3G to 5G [244]. With a supervised star topology, the collision of information is avoided. However, an obvious shortage of cellular network is the relative high delay even under a low channel load due to the round-trip between transceiver nodes and the base station. In release 14, 3GPP introduced direct Vehicle-to-Vehicle (V2V) communication outside of coverage under LTE-V mode 4 [245]. However, the distributed scheduling for LTE-V mode 4 is principally cannot totally avoid collisions. As the best of author's known, a congestion avoidance mechanism from 3GPP doesn't outperform IEEE 802.11p.

2020 is considered the first year of the 5G era in the wireless community since 5G is commercially employed in this year. To date, 5G is still a fast-developing research subject; thus, opposite views exist simultaneously. To avoid exaggerate the 5G technology, I only take the parameters and data that more than at least half of the community agree with into account. Although some controversies, I do not distinguish between 4G and LTE according to Dahlman [246].

To overcome the shortcoming of 4G [247], the basic requirements for the 5G are drawn by [248, 249, 250, 251, 252]: higher transmission rate, shorter latency, higher reliability, and more User Equipment (UE) connection. Correspondingly, the big 3 concepts: enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), massive Machine Type Communications (mMTC) [253], were proposed. According to the 3rd Generation Partnership Project (3GPP) 38.101 agreement [254], 5G New Radio (NR) mainly uses two frequency bands: FR1 frequency band and FR2 frequency band. The frequency range of the FR1 band is 450 MHz-6 GHz, which is also called the sub 6 GHz frequency band; the frequency range of the FR2 band is 24.25 GHz-52.6 GHz, usually called millimeter wave (mmWave) band. Currently, the most influential providers in the field of 5G are Huawei for sub 6 GHz band and Qualcomm for the mmWave band, separately. Other competitors mentioned quite often are Samsung, Ericsson, Datang, Nokia, Telecom, Intel, and ZTE. As we know, the higher the frequency, the closer the characteristic is to the light. That is, the propagation of the signal will be more similar to the light, which only goes straightforward so that the obstacles can easily block it. Also, the energy loss increases dramatically as the propagation distance increases, and proportionally to the square of the frequency. Consequently, the coverage problem, which restricts the promotion of the high-frequency spectrum 5G, occurs due to the nature of the mmWave. For this reason, most countries, such as China, Japan, and Korea, give priority to the sub 6 GHz band since the coverage is much larger, and thus more people can benefit from 5G technology. Compared to 4G, which only has 20 MHz channel bandwidth, 5G is allocated about 100 MHz in the sub 6 GHz area. Moreover, thanks to the novel Multiple-Input Multiple-Output (MIMO) technology, more antennas are used simultaneously to achieve a much higher transmission rate than

the previous 4G technology. Compared to the 4G handsets, which only have 2\*2 or 4\*4 antennas, 5G base stations and UEs have antenna array to increase the spectrum utilization [255, 256]. However, since such 5G UEs also use the sub 6 GHz band, there is principally not greatly different than 4G, and thus some serious problems are inevitable. First of all, because the sub 6 GHz area is also used by 2G, 3G, 4G, and thus already very crowded, a further increase of the bandwidth is almost impossible. Although some communication operators give 5G more channel bandwidth, which was belongs to 2G and 3G to increase the bandwidth of 5G further, the bandwidth is surely not enough for the future potential requirements. In addition, the configuration of the antenna depends on the signal frequency. At sub 6 GHz, the wavelength is more than 1 cm, so that the number of the antenna in the UE, in this case, is also limited. Therefore, soon after sub 6 GHz was promoted, how to use the higher FR2 frequency regions, i.e., higher than 28 GHz, has become a hot topic. Compared to the sub 6 GHz region, it is quite easy to have 1 GHz channel bandwidth in the FR2 region so that the transmission ratio is expected to be much higher. In the mmWave frequency band, taking the 28 GHz frequency band as an example, the available spectrum bandwidth has reached 1 GHz, while the available signal bandwidth of each channel in the 60 GHz frequency band is 2 GHz [254]. In the case of constant spectrum utilization, if the mmWave frequency band is selected, the data transmission rate can be doubled by directly doubling the bandwidth. Since 3GPP has decided to continue to use Orthogonal Frequency Division Multiplexing (OFDM) technology for 5G NR [254], mmWave technology has become the biggest novel idea of 5G. Although mmWave is already used by satellite, they were considered as infeasible for the daily life scenarios. Until recently, the novel technology unlocks the high-frequency spectrum. Concretely, thanks to antennas array, which constitutes a large number of antennas and the beamforming technology [257], the energy can be concentrated in small regions. Moreover, because the antennas for mmWave can be designed much smaller than the microwave antennas, the antennas in the mmWave antenna array are much denser and achieve a larger number for the same geometrical apparatus. Along with a certain number of small cell base stations, mmWave comes to the forefront of commercial applications. The introduction of other important 5G technologies, such as new numerology, LDPC/Polar codes,

etc., can let OFDM technology better extend to the mmWave band. To adapt to the large bandwidth characteristics of mmWave, 5G defines multiple sub-carrier intervals, of which the larger sub-carrier intervals are specifically designed for mmWave, whereas the lower is for the compatibility of previous system deriving from the 4G era. One of the main goals of 5G is to support URLLC services with stringent requirements for reliability and delay. LTE achieves a user plane two-way wireless delay of less than 10 ms, and the design goal of 5G is to reduce this delay by at least 5 times, that is, less than 2 ms. According to the 3GPP TS 38.211 protocol [254], the 5G NR physical layer provides multiple sub-carrier spacing configurations [258]. By increasing the sub-carrier spacing, the duration of OFDM symbols is reduced, thereby reducing the duration of a single time slot and reducing delay. The 3GPP protocol claims that the sub-carrier spacing is inversely proportional to the OFDM symbol duration, which is an inherent attribute of OFDM. For the current network communication technology, the key capability indicators of the 5G system have been greatly improved. The information transmission delay of the 5G network can reach milliseconds, which meets the stringent requirements of the network and guarantees the safety of controlled UE. The peak rate of 5G can reach 10-20 Gbit/s, and the number of connections can reach 1 million/ $km^2$  [259]. Apparently, although the technology can overcome the difficulties of implementing the mmWave, the base stations for mmWave are energy-consuming equipment. Thus, Heterogeneous Network (HetNet) is also essential in the 5G era, i.e., most scientists in the wireless community believe that both sub- and above 6 GHz networks will coexist in a long time. The same as LTE, 5G also has device to device network to solve the problem when UEs are outside of the coverage of base stations [260].

## 7.3 IEEE 802.11p

### 7.3.1 Why I Use the IEEE 802.11p?

Despite the fact that LTE has a series of advantages, I would like to adopt the IEEE 802.11p as the first version for connected mobile machines due to the following reasons. First of all, to fully make the advantages of C-V2X, mobile machines need a base station nearby, which varies from 10 m until 10 km [261]. However, for the fleet of mobile machines that are working far away from urban, they might fail to find a base station nearby. Moreover, the usage of 802.11p is free of charge. Different from the cellular network which the users must pay for the service from the network operators, the 5.9 GHz band is a free but licensed spectrum [237]. In addition, IEEE 802.11p is well designed for the vehicle industry so that no additional modification is needed for vehicle onboard ECU [262]. Thus, the compatibility of IEEE 802.11p is better than cellular networks for the mobile machine which is designed without the consideration of V2X. Usually, mobile machines drive at a relatively low speed. Furthermore, the communication between other onboard units, for instance, driving cars and mobile machines is not essential; thus, the under-performed ability to deal with vehicle mobility by IEEE 802.11p, based on the analysis of Alasmary's study [263], can be ignored. Although there have no consensus about which wireless technology is the more promising technology, scientists from both sides agree that the combination of LTE and 802.11p have a certain improvement in performance compared to if only one technology is used [236, 240, 262, 264]. Thus, I would like to use IEEE 802.11p as the communication technology for the initial version fleet management. Even though the passenger car industry adopts cellular technology in the future, the idea of using IEEE 802.11p for mobile machines is still sensible, because the congestion of the channel is consequently alleviated.



## 7.3.2 Modelling

Mecklenbräuker has shown the common scenarios in their paper [241]. Unfortunately, for mobile machines that have the task to repair the highway, the scenario does not belong to these common ones. Firstly, there has usually no buildings around the working site, but the traffic is congested. Secondly, instead evaluate the communication among all the participants in the ad-hoc network, only communication among mobile machines is essential.

## 7.3.3 Propagation Model

In [265], a comparative analysis between different propagation models is performed. Based on Stoffer's study, there is no best model for all cases, and the users should select the model depending on the concrete use case. Because engineers are mainly interested in delay and packet loss resulting from congestion control algorithms at MAC layer and the highway is more similar to an urban scenario, I used a log-distance path loss model proposed by [266]. It is denoted as

$$PL(dB) = PL(d_0) + 10 \cdot n \cdot \log\left(\frac{d}{d_0}\right) \quad (7.1)$$

where  $PL(d_0)$  is defined as the path loss at the reference distance ( $d_0$ ), and  $PL(d_0) = 46.6777$ dB.  $n$  refers to the path loss distance exponent varying from the propagation environment, and  $n = 3$ .

Since the single factor that influences receive power is the distance from the transmitter, in the following simulations, the dynamic mobility model is not applied to vehicles. Still, the relative positions of the vehicles are randomly initialized.

### 7.3.4 CAM' Generation Model

Venel presented that CAMs are generated at a rate in a range of 2 to 20 packets/second corresponding to multiple factors such as driver's reaction time and vehicle speed [267]. Thereby, I apply a mean value from them, namely 10 packets/second (10 Hz). In addition, the length of a packet varies from different applications in real-world vehicular communications. In the following simulations, packet length is set to be 450 bytes, which ensures the necessary information for the safety-related application<sup>3</sup>. Since the generation rate and CAM length are constant throughout the simulation, the channel load is only depended on the number of nodes in the scenario.

### 7.3.5 CSMA/CA and Enhanced DCF Channel Access (EDCA)

CSMA/CA algorithm is specified in IEEE 802.11 to schedule transmissions over a single channel by differing the access attempt with a random back-off time. In the meantime, EDCA introduces Interframe Spaces (IFS) and different contention window size to prioritize access categories and to improve quality-of-service (QoS) [269].

Since the primary emphasis of this chapter is on the congestion control algorithms at MAC layer and CAM length is constant, the term delay in the following part will always refer to the back-off time between the time point that a node requests for channel access and the packet is forwarded from the MAC layer to the PHY layer, neglecting the transmission time depending on packet length and propagation time depending on distance. Tab. 7.1 contains the vital parameter settings that I use.

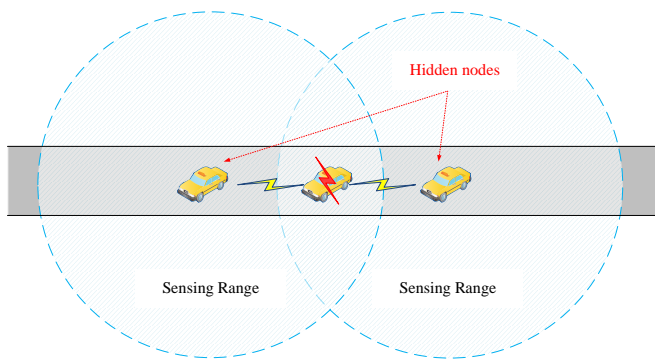
---

<sup>3</sup> Based on the *Survey on ITS-G5 CAM statistics CAR 2 CAR Communication Consortium*, CAM size is a design parameter and 30% of the messages are above 450 bytes. The typical V2X messages' size falls within the range of 60-800 bytes [268].

**Table 7.1:** Simulation parameters

<b>Parameters</b>	<b>Value</b>	<b>Unit</b>
<b>TxPower</b>	17	dBm
<b>Packet length</b>	450	Bytes
<b>Packet generation rate</b>	10	Hz
<b>Channel width</b>	10	MHz
<b>Data rate (BPSK)</b>	3	Mbps
<b>Data rate (QPSK)</b>	6	Mbps
<b>CWmin</b>	15	-
<b>AIFSN</b>	7	-
<b>Time Slot</b>	13	$\mu$ s
<b>SIFS</b>	32	$\mu$ s
<b>EIFS</b>	120	$\mu$ s

There are two ranges, i.e. transmission range and sensing range for each transmitter, since the CAM header and payload are modulated with different schemes and have different immunities against noise and channel fading. The Physical Layer Convergence Protocol (PLCP) header is modulated with Binary Phase Shift Keying (BPSK) [270] and the payload is transmitted in the form of Quadrature Phase Shift Keying (QPSK) modulation. Here I did an experiment 200 times and each time I let the distance between the transmitter and the receiver gradually increase. Simulation results show that, the transmission range is equal to 115 m corresponding to a SINR level at 6.49825 dB and the sensing range is equal to 175 m. That is, in case more than 115 m, the receiver cannot decode the content of the messages, and in case the distance is more than 175 m, the receiver cannot even get the headers. Once two transmitters are distanced more than 175 m, they can send packets simultaneously, being unconscious of the busy channel status. In this case, as shown in Fig. 7.2, they are called “Hidden Node”. Multiple arbitrary packets may collide at the receivers who are visible and connectable to both hidden nodes. The interference between each other results in transmission failures.



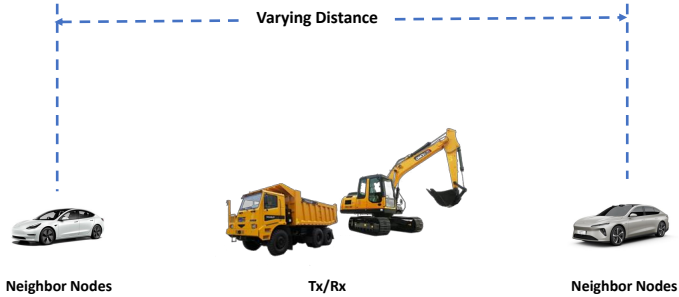
**Figure 7.2:** The hidden node problem.

In short, the scenario I analyzed is a working site on the highway where the communication performance among mobile machines under the interference from cars nearby.

### 7.3.6 Evaluation of Hidden Node Problem

To evaluate the impact of hidden node problem on vehicle network, a set of simulations is considered as follows: I set the transmitter and receiver, i.e., the dumper and the excavator in the figure, very closer to each other. A total of 80 neighbor nodes is equally divided into two groups, which are symmetrically distributed on both sides of Transmitter/Receiver pair (Tx/Rx). Concretely, twelve simulations are executed, with the distance between two groups of neighbors increases by 20 m from 0 to 220 m and 300 CAMs are sent per each node. Here the most critical performance is whether the receiver can get the information sent by the transmitter under the distribution of the neighbor nodes. The simulation setup for testing the hidden node problem is shown in Fig. 7.3.

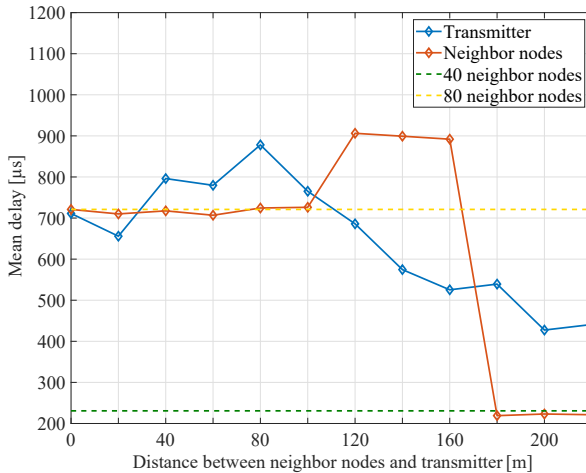
How the different distances of two neighbor groups impact the mean delay, packet collision probability, and packet loss probability of transmitter and neighbor are



**Figure 7.3:** Schematic view of simulation scenarios.

demonstrated in Fig. 7.4, Fig. 7.5, and Fig. 7.6, individually. Performance observed at the transmitter and neighbors are illustrated with blue and red curve, respectively. As reference, the yellow and green dotted lines indicate the simulation results in which 40 and 80 neighbors are located at the same position as the transmitter.

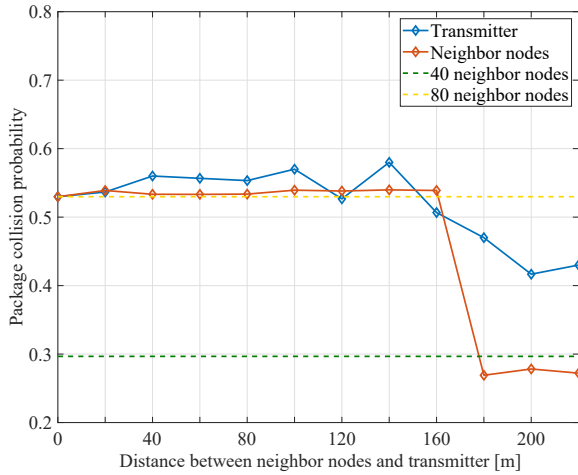
With respect to mean delay in Fig. 7.4, the curve for neighbors remains stable within 115 m and then rises in the sensing range owing to the additional Extended Inter-Frame Spacing (EIFS) appended to Arbitrary Inter-frame Space (AIFS). Finally, it sinks significantly when the two groups are more than 175 m apart from each other. In this case, they are hidden to each other. Therefore, the delay in each group is approximate to the scenario with just 40 neighbor nodes in the transmission range. In the meanwhile, the curve for transmitter fluctuates slightly. The reason is that the mean delay of transmitter is averaged by 300 packet in contrast with  $80 \times 300$  packets of neighbors. The mean delay of the transmitter decreases when two neighbor groups are in each others' sensing range because the higher delay of the neighbors provides the transmitter a higher probability to access the channel. When the neighbors are hidden to each other, transmissions from hidden nodes overlap with each other, the whole channel busy time decreases. As a result, mean delay of transmitter declines.



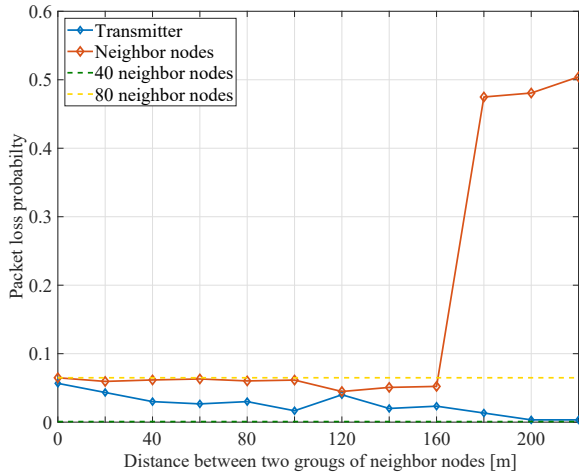
**Figure 7.4:** Mean delay [ $\mu\text{s}$ ] versus distance between two groups of neighbor nodes [m].

Similarly, the packet collision probability, which solely depends on the number of sensible nodes, are shown in Fig. 7.5. The red curve for neighbors remains coincident with 80 neighbors' scenario and grows down rapidly to the 40 neighbors' level as the two groups become hidden nodes to each other. In the meanwhile, the collision probability of the transmitter keep steady until the neighbours become hidden nodes. Since more idle channel is released due to overlapped transmissions, as mentioned in the previous paragraph, the packet collision probability of the transmitter declines, as wells as its packet loss probability, which is shown in Fig. 7.6.

The overlapped transmissions from hidden nodes packets are collided and corrupt at the receiver, resulting in a dramatic growth on packet loss probability of the neighbor nodes, which can be clearly seen from the red curve in Fig. 7.6. In the meanwhile, the transmitter has less collided transmissions. In brief, the transmitters benefits from the appearance of neighbor nodes in form of hidden nodes in pairs, in terms of less mean delay, packet collision probability, and packet loss probability.



**Figure 7.5:** Packet collision probability versus distance between two groups of neighbor nodes [m].



**Figure 7.6:** Packet loss probability versus distance between two groups of neighbor nodes [m].

The factor number of neighbors has a significant impact on the network performance, particularly in the case that packet length and generating rate is fixed.

### 7.3.7 Empirical Model for Fast Estimation of Ad-hoc Network Performance

Although  $n_s - 3$  can simulate the V2X performance regarding the delay and the probability of lost packet, I still need a quick estimation method, so that onboard ECU can obtain V2X performance in realtime and evaluate the plausibility of V2X data. Therefore, I build an empirical model to fast estimate the network performance based on the results from  $n_s - 3$ . Since the contention behavior due to CSMA/CA in corresponding ranges should follow the same roles, which highly depends on the number of neighbors, I introduce the analytical model as follows.

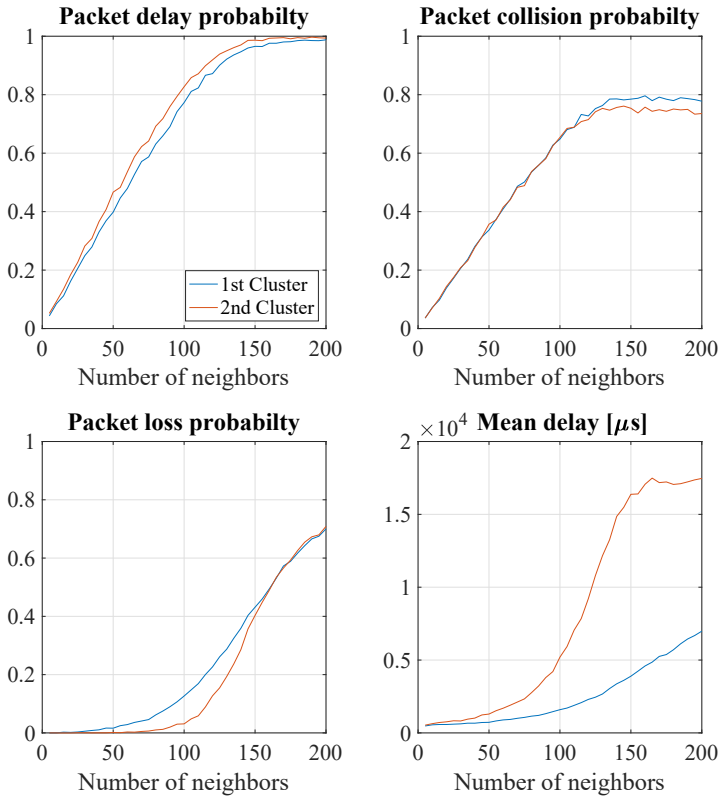
#### 7.3.7.1 LUT Generation

For each cluster, e.g., the area within the transmission range and the area between the transmission and sensing range, I generate a Lookup-Table (LuT) in advance, which contains a set of crucial performance indicators in relationship with varying number of neighbors. To reduce the effect of randomness, I average the results from a large number of CAM transmissions.

To generate LuT for 1. cluster, I execute the following simulations. The neighbors are located at the same position with 60 m away from the transmitter. The number of neighbors varies from 5 to 200, with a step of 5 in each scenario. Furthermore, for each of the 40 scenarios, 5 simulations are conducted, in which every single node schedules 1,000 transmissions. The same simulations are executed for the 2. LuT, only the neighbors are 140 m away from the transmitter.

Four metrics of the transmitter are measured, as shown in Fig. 7.7, e.g., collisions probability ( $P_c$ ), packet delay probability ( $P_d$ ), packet loss probability ( $P_l$ ), and mean delay ( $t_{md}$ ). The term collision indicates the access attempt occurs during





**Figure 7.7:** Packet delay probability, packet collision probability, packet loss probability and mean delay measured with varying number of neighbors in 2 clusters are included in the LuT.

the duration, in which another node is transmitting. Moreover, the access attempt can also be differed due to the on-going AIFS, which follows the previous transmission, even though the channel is idle. Therefore, the percentage of delayed packets is slightly higher than the percentage of collisions. The metrics packet delay probability and mean delay indicate how probable the packet would be delayed due to an access contention, and once the delay occurs, what would be the average duration.

### 7.3.7.2 Performance Estimation

For each on board unit in the scenario, the number of neighbors located in each of the two Clusters are measured. The analytical result is derived from the sum of two values that are interpolated and extracted from LuTs. Furthermore, the upper limit for an analytical percentage is equal to 1. Eq. 7.2 and Eq. 7.3 demonstrates this idea,

$$\hat{\Phi}_{A,t,n} = LuT_{t,1}(n_T) + LuT_{t,2}(n_S) \quad (7.2)$$

$$\hat{\Phi}_{A,p,n} = \min(1, LuT_{p,1}(n_T) + LuT_{p,2}(n_S)) \quad (7.3)$$

where  $\hat{\Phi}_{A,n}$  is the naive estimation of the performance of the ad-hoc using the analytical model, the footnote  $t$  and  $p$  denote the estimation in terms of time and probability, respectively.  $n_T$  is the number of nodes inside of the transmission range,  $n_S$  is the number of nodes inside of the sensing range.

### 7.3.8 Validation and Calibration

In this section, I first validate the viability of the analytical model and then introduce the correction factor to eliminate the error between the naive LuT and the realistic simulation results.

In the validation simulation, the traffic scenario is set to be a 1,500 m long highway with 3 lanes in each direction. 500 onboard units equipped with 802.11p devices are located statically. A congested traffic due to a highway worksite is assumed. The simulation is set up with a total simulation time of 100 s, in which the vehicles are randomly distributed on the road.

The delay relevant metrics are simulated and estimated among all onboard units. This is because each transmission has a unique channel access time, which is independent of reception. In the meanwhile, for each onboard unit, the packet loss probability is measured on a single receiver, which is located randomly within its 15 m range, corresponding to two cooperating mobile machines.

Fig. 7.8 represents the correlation coefficients for each performance metric, which evaluate the strength of the association between simulated and analytical results. For an optimum fitting, the blue dots are supposed to be correctly distributed along the diagonal line, which denotes a correlation coefficient of 1. The correlation coefficients for the mean delay, packet delay probability, and packet loss probability are 0.9417, 0.9277 and 0.9167, which manifest a strong correlation and satisfying estimation ability of the analytical model.

To optimize the estimation performance of the proposed analytical model, the term correction factor ( $f_c$ ) is introduced,

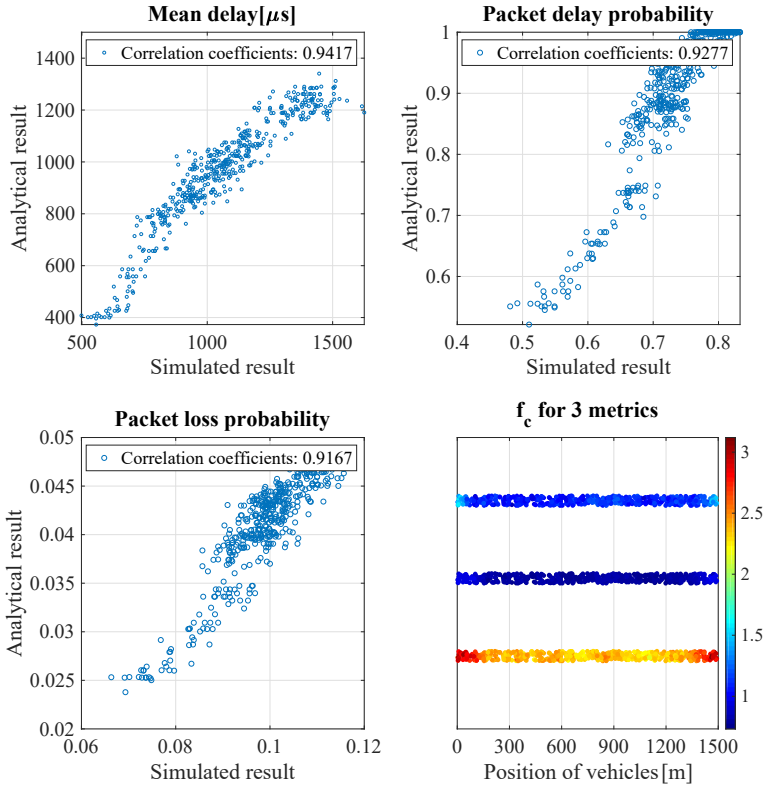
$$\tilde{f}_c = \frac{\tilde{\Phi}_S}{\hat{\Phi}_A} \quad (7.4)$$

where  $\tilde{\Phi}_S$ ,  $\hat{\Phi}_A$  are the performance matrix from the simulation and the analytical model regarding the  $t_{md}$ ,  $P_d$ ,  $P_l$ , separately.

Obviously, my goal can be demonstrated as Eq. 7.5:

$$\min(J) = \sum_i^{n=N} (\tilde{f}_c \cdot \hat{\Phi}_A - \tilde{\Phi}_S)^2 \quad (7.5)$$

where N denotes the total number of vehicles.



**Figure 7.8:** Correlation coefficients of 3 metrics are close to 1, which indicate a good feasibility of analytical estimation. To increase estimation accuracy, I introduce  $f_c$ .

The  $\tilde{\Phi}_S/\hat{\Phi}_A$  is shown in the bottom right sub-figure in Fig. 7.8. The three curves from top to bottom indicate the  $f_c$  for mean delay, packet delay probability and packet loss probability. The uniform color in the center area indicates that the naive analytical estimation method has stable performance and thus can be adjusted by multiplying appropriate correction factor  $f_c$ . Among 3 metrics, packet loss probability is dramatically underestimated and needs a larger  $f_c$ . This is because, in the LuT generation scenario, a reception is failed only due to multiple transmitter attempts to access the channel simultaneously, without consideration of hidden node. However, in the realtime simulation, the transmissions from the hidden nodes cause interference at the receiver. Consequently, the reception is more like to corrupt due to lower SINR.

The correction factor differs in the discontinuous edge of the scenario, where hidden node problem is not obvious. In this case, I introduce another correction factor. Tab. 7.2 records the correction factor in the center ( $f_{c,c}$ ) and the correction factor at the edge ( $f_{c,e}$ ), where the results are calculated based on Eq. 7.5.

**Table 7.2:** Correction factors

$f_c$	<b>center</b>	<b>edge</b>
<b>Mean delay</b>	1.0857	1.3048
<b>Packet delay probability</b>	0.7516	0.9671
<b>Packet loss probability</b>	2.2617	2.9121

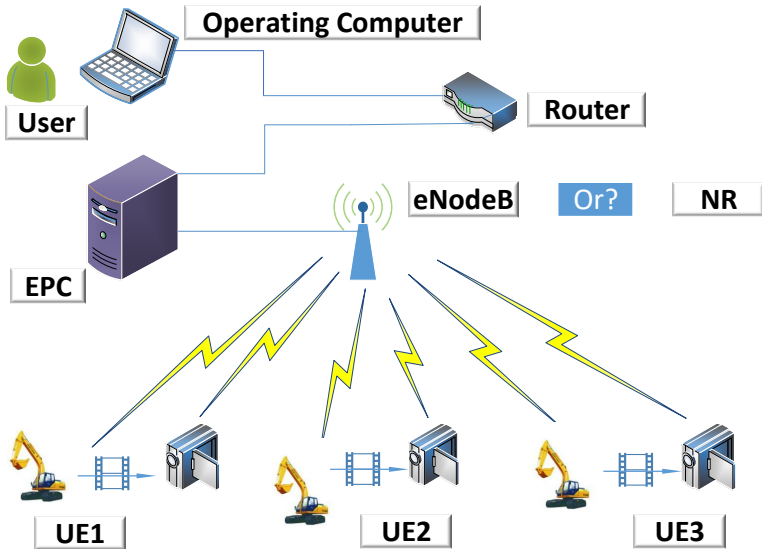
After using the correction factors, the analytical model outputs a very similar result to the simulation model. Furthermore, the LuT is portable to scenarios with different PHY parameters and path loss models, by re-calculating the transmission and sensing range size, since the contention mechanism due to CSMA/CA stays the same.

## 7.4 The Fifth-Generation Mobile Networks

The fleet management of mobile machines is an interesting research direction of the Internet of Things (IoT) in the construction machines industry. Besides using the ad-hoc network as the first version for mobile machines [40], 5G attracts huge attention to be expected to achieve even higher-quality communication. As mentioned in the earlier part of this chapter, WiFi technology can accomplish realtime communication among mobile machines so that they will work denser and faster. As a consequence, engineers can increase productivity and therefore reduce the duration of the construction projects. This is meaningful for the cases of repairing projects on the highway, mining projects, and transportation in harbors. Since mobile machines are usually working surrounded by dust and Lidars are quite sensitive to this case, cameras are a more robust and promising approach towards self-working machines or remote control of mobile construction machines. As we know, as the videos' resolution increases, both image recognition algorithms and humans can acquire information easier and more accurate. However, the capacity, especially the uplink capacity of WiFi technology, limits the introduction of wireless HD video transmission for construction machines. As I did not find comprehensive research indicating how can 5G change the mobile construction machines industry, I first analyze the potential use cases for the implementation of 5G for the construction machines industry in this chapter. Followed by illustrating the benefits by utilizing 5G with my simulation results by means of  $n_s - 3$  [235]. Last but not least, I show the blueprint of future smart working sites based on the simulation results. Fig. 7.9 and Fig. 7.10 demonstrate the potential use cases of 5G in the field of mobile construction machines.

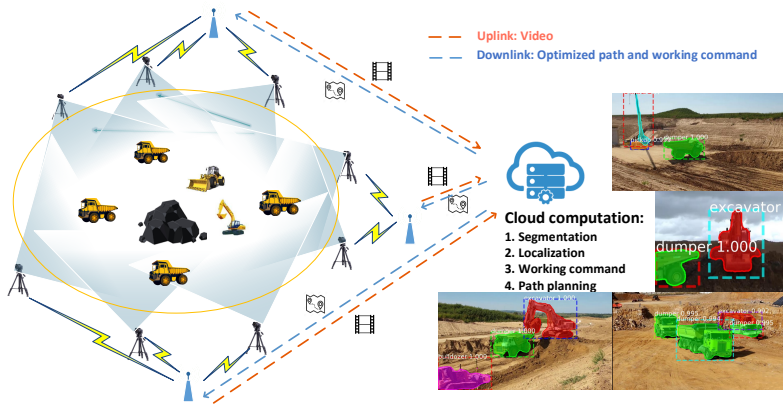
### 7.4.1 Where Can Working Sites be Benefited from 5G?

According to GSMA's outlook in 2020, mmWave can roughly make economic benefits 212 billion dollars only in the Asia Pacific region in 2034. Among them, 3% to 9% of the amount will come from the agriculture and mining industry.



**Figure 7.9:** Remote control with live streaming: here cameras will be installed on the mobile machines while the driver sits in a comfortable room to operate the machines remotely. Thanks to 5G, HD video streaming can be sent with low delay and high reliability.

To date, 5G mmWave need a lot of micro base stations, and they are also energy-consuming [272] and cost-consuming. Moreover, the shortcoming of mmWave will be amplified by the harsh environment on the working site, such as the blockage of dust and giant machines. However, it did not stop the engineers to adopt this new technology in the construction field. Currently, most people believe that IoT technologies will endow the mobile construction machines industry with the ability, such as predictive maintenance, data analytics, and visualization and notification. Besides these wisdom, other scenarios are remote control and self-working mobile machines with which previous communication technology cannot do well. In some dangerous traditional industries, such as remote maintenance of underground pipelines, remote rescue of landslides, underground mine excavation, etc., these industries' operating environment is hazardous and harmful to the



**Figure 7.10:** Self-working mobile machines: here, cameras will be fixed on the ground instead of being installed on the machines to avoid the obstruction of vision. The stream will be uploaded to the center commander and be processed on the cloud. Based on the stream from more than two cameras, the depth information and motion of machines can be acquired. Afterward, the command signal will be sent directly to the machines. The research about instance segmentation of construction machines can be found in [271].

human body. Although remote control is achieved with a wired network for nowadays projects, the flexibility is limited by the cable connected to the vehicle so that remote control is only used in some particular cases. Thanks to 5G, the remote control can be performed without the limitation of cables so that 5G accelerates the usage of remote control. In this case, the cameras are usually installed on the machines to collect the surrounding environment information [273, 274, 275]. Since they typically need more than three cameras to get the information, and the transmission rate of WiFi is limited, they cannot install more cameras to create the depth information resulting in lower productivity even with the very best operators [276]. Considering virtual reality technology will be adopted with 5G, the difficulty of the remote control will be dramatically reduced. Better than the earlier network technologies, 5G guarantees the efficiency and accuracy of the remote control. Another major expected application is self-working machines. Cooperating with deep learning-based image processing models [62], the image can be further processed on the local cloud. The command can then be directly



sent to the machines. To avoid the additional cost, many scientists point out a smartphone can be used as an intermediary to transmit information instead of installing additional equipment [64].

Although 5G shows excellent progress compared to 4G and WiFi, for end customers to accept a new technology, a sudden colossal improvement is always necessary. Currently, most people believe that IoT technologies will endow the mobile construction machines industry with the ability, such as predictive maintenance, data analytics, and visualization and notification. However, I find that they are actually nice-to-have technologies. Since 5G may need a lot of micro base stations, and they are also energy-consuming [272], the value created by predictive maintenance is quite difficult to compensate for the additional cost of 5G. In many cases, preparing some backup vehicles can be a more effective and money-saving solution. Moreover, the shortcoming of mmWave will be amplified by the harsh environment on the working site, such as the blockage of dust and giant machines. Thus, I believe more realistic scenarios are remote control and self-working mobile machines since 5G achieves something engineers cannot do well before. In some dangerous traditional industries, such as remote maintenance of underground pipelines, remote rescue of landslides, underground mine excavation, etc., these industries' operating environment is hazardous and harmful to the human body. Although remote control is achieved with a wired network for nowadays projects, the flexibility is limited by the cable connected to the vehicle so that remote control is only used in some particular cases. Thanks to 5G, the remote control can be performed without the limitation of cables so that 5G accelerates the usage of remote control. In this case, the cameras are usually installed on the machines to collect the surrounding environment information [273, 274, 275]. Since they typically need more than three cameras to get the information, and the transmission rate of WiFi is limited, they cannot install more cameras to create the depth information resulting in lower productivity even with the very best operators [276]. Considering virtual reality technology will be adopted with 5G, the difficulty of the remote control will be dramatically reduced. Better than the earlier network technologies, 5G guarantees the efficiency and accuracy of the remote control. Another major expected application is self-working machines.

Cooperating with deep learning-based image processing models [62], the image can be further processed on the local cloud. The command can then be directly sent to the machines. To avoid the additional cost, many scientists point out a smartphone can be used as an intermediary to transmit information instead of installing additional equipment [64].

In the above scenario, there are three key technologies for remotely controlling or self-working construction machinery. The first one is the high-speed data transmission rate. In order to enable the AI or human to fully understand the situation in realtime, construction machinery will be under the sight of HD cameras or wear the cameras for an operator to get the video streaming data collection. The transmission of HD video requires a large bandwidth to ensure the fluency and realtime transmission of video content. The second is the low delay in receiving information. The realtime issuance of interactive behavior between operators and controlled construction machinery requires the network to have low latency to ensure that the controller's command can be executed in realtime through actuators. The third is the rapid and convenient communication network deployment between the construction machinery and the operators. If a wired network is used between the construction machinery and the controller, although the network delay and bandwidth can be guaranteed to a certain extent, the cable makes the activity of the construction machinery limited. Moreover, the rapid deployment of networks between construction machinery and controllers cannot be easily achieved. If a 4G-LTE wireless cellular network is used, due to the limitation of the transmission rate and delay of the 4G-LTE network, the bandwidth and delay of the existing wireless network may not stably meet some high-rate and low-delay scenarios. These technical bottlenecks make the remote control collaboration project encounter many difficulties in the industry's practical application. No wonder so far, it has not been able to achieve widespread development and deployment. The large bandwidth and low delay technologies of the 5G network can solve these technical bottlenecks. 5G is bringing new opportunities for the industrial development of remote-controlled construction machinery.

## 7.4.2 Problem Statement and Goal

In the previous study from Bermudez [277], they tested the performance of the LTE network by the transmission of video data. Their article evaluated two protocols' behaviors, Realtime Messaging Protocol (RTMP) and Realtime Streaming Protocol (RTSP), in a 4G environment. Based on their results, I find that the performance of LTE to transfer the HD video from the working side to the operator side in realtime is good but not fully satisfying.

Also, the throughput of LTE is in a steady-state growth situation. That means, the simulation parameters of Bermudez [277] missed the extreme working critical condition. Whether the LTE network can always have an excellent performance in a more stringent remote-control situation was not shown. Therefore, the comparison between LTE and 5G for video transmission in construction scenarios is necessary. For remote control, the delay is always a significant indicator because it equals to the accuracy and reliability of the job and the safety of the controlled machine [273]. Inspired from this and to fill this research gap, I compare the performance from one of the new 5G technologies, mmWave, with the LTE network's performance for construction machinery in remote-control and self-working scenarios. Meanwhile, I give the simulation a more stringent critical environment. Under the goal of finding out whether 5G network is more suitable for remote control or self-working construction machinery than LTE or not and if so, how good it is, a similar research is not in existence.

## 7.4.3 Modelling

In the scenarios shown in Fig. 7.9 or Fig. 7.10, that my UEs, i.e., construction machines, are under the sight of HD cameras or with the HD cameras. Here I assume the construction machines and cameras are both connected with the base station and the operator. The operator will give the construction machines commands. Meanwhile, they will collect the video streaming data from cameras. In case that the cameras are stick to the machine, the operator will give the order

and receive the video data simultaneously. Compared to the instruction from the operator, video streaming data will occupy a much larger bandwidth. Therefore, in the research, I use video streaming as the media, which can verify the performance of both networks. Obviously, video streaming with different resolution occupies different network bandwidth. Depending on the different resolution requirements of video streaming, different pressure will be applied to the network.

For the research, I use  $ns - 3$  [235, 278] as the simulation tool. To perform LTE simulation, I directly call the LTE module inside  $ns - 3$  because there is already a complete set of simulation modules and processes in  $ns - 3$  for 4G [279]. On the other hand, for the 5G network, since it is still quite novel,  $ns - 3$  has not yet developed an official simulation platform with all 5G modules. Fortunately, because  $ns - 3$  is an open-source platform, many professional network simulation users can contribute to this platform based on their requirements, such as rewriting the algorithm, adding patch packages, or doing other upgrades. Among them, I selected the model from Mezzavilla [280] to simulate the 5G mmWave performance. The following paragraphs will present some basic architecture details and model settings for both network models. Basic parameters are shown in Tab. 7.3 and Tab. 7.4.

**Table 7.3:** LTE Network Parameters, from 3GPP TS-36101 [281]

<b>Parameters</b>	<b>Value</b>
<b>Bandwidth</b>	25 MHz
<b>Downlink Earfcn</b>	100
<b>Uplink Earfcn</b>	18100
<b>Scheduler</b>	<i>PfFfMacScheduler</i>

**Table 7.4:** 5G Network Parameters, From 3GPP TS-38101 [254]

Parameters	Value
<b>Band</b>	n257
<b>Downlink NR-ARFCN</b>	2054167 - 2104165
<b>Uplink NR-ARFCN</b>	2054167 - 2104165
<b>Scheduler</b>	<i>MmWaveMacScheduler</i>

## 7.4.4 Model Parameters

### 7.4.4.1 Propagation Model

For LTE, I use *FriisPropagationLossModel* [282]. Given an unobstructed visual path between the transmitter and receiver, the free-space propagation model can predict the strength of the received signal. According to Friis [283], the received signal strength can be described as,

$$P_r(d) = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d^2 \cdot L} \quad (7.6)$$

where  $P_r(d)$  is defined as received signal power,  $P_t$  is transmit power,  $G_t$  is transmit antenna gain,  $G_r$  is receive antenna gain,  $\lambda$  is wavelength(m),  $d$  is the distance, and  $L$  is the system loss.

As for 5G, I use *MmWavePropagationLossModel* [284]. This mmWave model presents two kinds of path loss models. The first one is the one that I used, which is in a statistical characteristic of the Line of Sight (LOS) state. The other one is *BuildingsObstaclePropagationLossModel* [285], adding the obstacle between the gNB and the UE. Further path-loss models of mmWave can be found in [286].

#### **7.4.4.2 Transmission Control Protocol/Internet Protocol (TCP/IP)**

The network transmission adopts the TCP/IP protocol. The core protocols of the TCP/IP protocol are the transport layer protocol for TCP and User Datagram Protocol (UDP) and the network layer protocol for IP, which are usually implemented in the kernel of the operating system. Because the purpose of TCP is to achieve reliable data transmission, it has a set of handshake mechanism, send - confirmation, timeout - resend [287]. In the case of video streaming, the network spending of TCP transmission is too large, thus impairing image quality and latency. Therefore, the UDP transmission method is preferred for realtime live streaming [288, 289].

#### **7.4.4.3 Hybrid Automatic Repeat Request (HARQ)**

For 4G and 5G, they both have two levels of retransmission mechanisms: HARQ at the MAC layer and ARQ at the Radio Link Control (RLC) layer [290, 291]. For 4G, the retransmission of lost or erroneous data is mainly handled by the HARQ mechanism of the MAC layer and supplemented by the ARQ of the RLC. The HARQ mechanism of the MAC layer can provide fast retransmission, and the ARQ mechanism of the RLC layer can provide reliable data transmission. In contrast, for 5G, the uplink HARQ mechanism is the same as the downlink, and both are asynchronous HARQ. There will be two kinds of changes [292]. First, the scheduling timing is more flexible, especially in TDD mode, resulting in more resource allocation flexibility. Second, the pressure of data buffering will increase. Unlike LTE's uplink synchronous HARQ, asynchronous HARQ may have a longer retransmission interval. During this time, the UE must buffer the unACKed data, which will increase the buffering pressure.

#### **7.4.4.4 Scenarios**

Three scenarios were setup for both network environments. In the first scenario, I choose 2 Mbps as the video streaming volume. 2 Mbps is nearly the level of 720P

video streaming bandwidth requirement [293]. Then I change the UE number from 2 to 20. In the second scenario, I set the UE number as a constant condition. By changing the data volume to realize new scenario, from 1 Mbps to 8 Mbps, which includes the bandwidth requirement of 720P (3 Mbps), 1080P (5 Mbps), and 3D 1080P (6 Mbps) videos [294], I tested the network performance with a varying resolution of the video. At last, I let UEs move to acquire the knowledge of how mobility condition affects the networks. For scenarios 1 and 2, the mobile machines will be under the sight of those HD cameras. Those cameras will collect the working video data and transfer it to the operator. The UEs in scenario 3 will be cameras installed on mobile machines. Here they will change their position together with the construction machinery as collecting the video streaming.

**Table 7.5:** Network Scenario 1

<b>Data Volume (Mbps)</b>	2
<b>UE Number</b>	2, 4, 6, 8, 10, 12, 14, 16, 18, 20

**Table 7.6:** Network Scenario 2

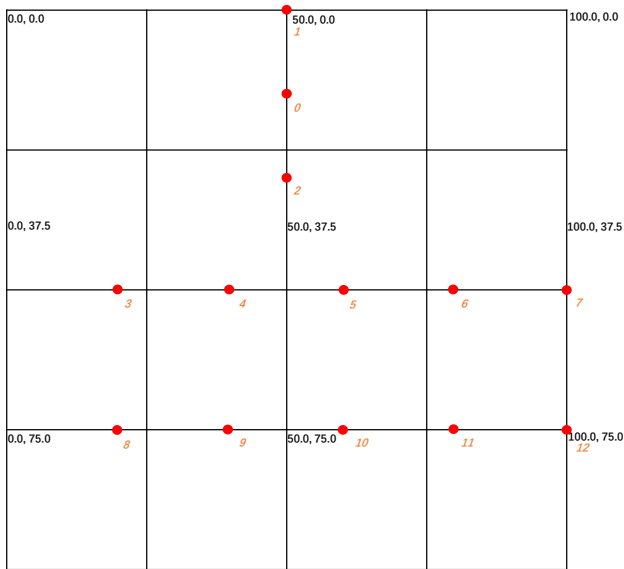
<b>UE Number</b>	8
<b>Data Volume (Mbps)</b>	1, 2, 3, 4, 5, 6, 7, 8

**Table 7.7:** Network Scenario 3

<b>Data Volume (Mbps)</b>	2
<b>UE Number</b>	8
<b>UE velocity(km/h)</b>	from 10 to 60

### 7.4.5 Simulation Results

This section presents the results of the simulated network scenarios in terms of throughput, packet loss rate, and delay. As for both network environments, I performed the simulation repeatedly and got the average value to improve accuracy.

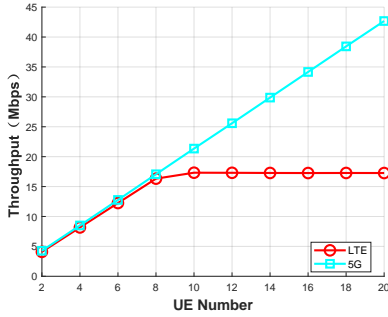


**Figure 7.11:** Network topology. In this figure, the upper left corner is the origin of the coordinates. The side length of each square grid is set as 25 m.

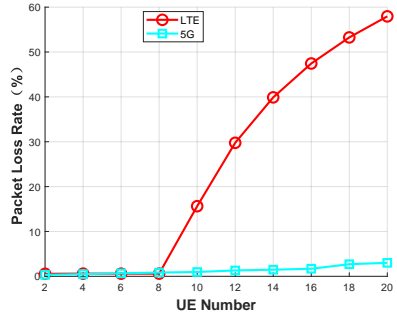
The network topology is shown in Fig. 7.11. From node 3 to node 12 represent a set of remote devices, i.e., cameras, and the transmission data represents the video data sent by the camera avatar, which is finally sent to the user terminal (node 1) through eNodeB (node 0) or gNB (node 2), with Evolved Packet Core (EPC) or NR.

With the increase of the number of UE, the throughput simulation results are shown in Fig. 7.12(a). In the beginning, the throughput of LTE and 5G networks has increased rapidly, and the throughput matches the total data volume, which

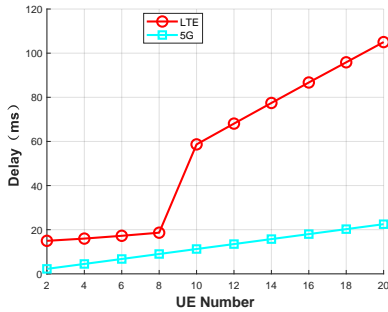




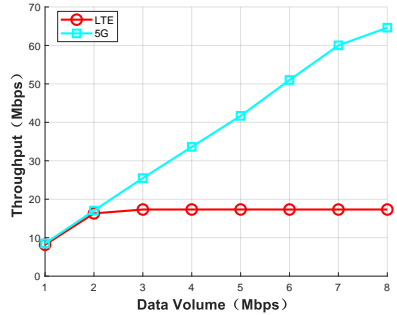
(a) Average throughput with increasing UE.



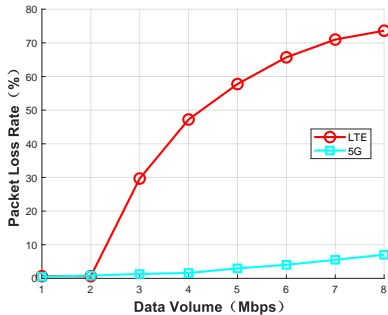
(b) Packet loss rate with increasing UE.



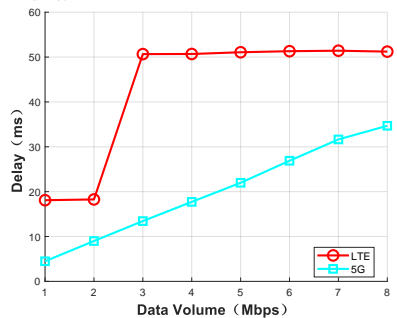
(c) Average delay with increasing UE.



(d) Average throughput with increasing Data Volume.



(e) Packet loss rate with increasing Data Volume.



(f) Average delay with increasing Data Volume.

**Figure 7.12:** Simulation results of scenario 1 and scenario 2.

means both of them can complete the transmission of the video streaming task.

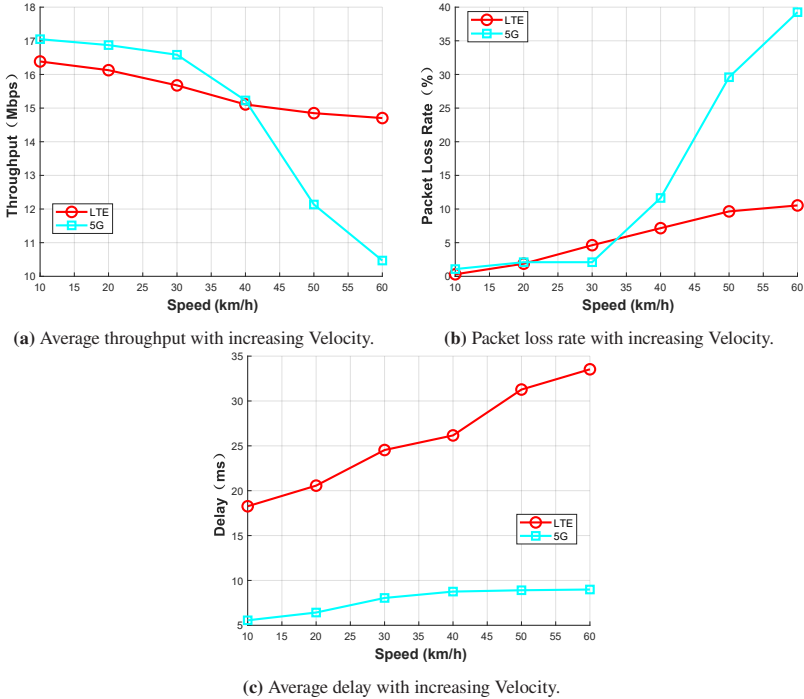


Figure 7.13: Simulation results of scenario 3.

As the number of UEs further increases, the 5G network can still transmit video service data better; however, the LTE network cannot provide enough transmission capacity for video service data, reaching a state of business saturation. It can be observed that the throughput remains basically unchanged as the UE number grows, about 17 Mbps. In Fig. 7.12(b) simulation results of the packet loss rate as the UE number increase are shown. When the UE number is small, both LTE and 5G networks can keep the packet loss rate at a low level, i.e., almost no packet loss occurs. If the UE number increases, the 5G network can still maintain the network with an almost low packet loss rate. Still, the LTE network will have more packet loss due to its network resource constraints. It must discard the video service's data packets, causing the transmitted video to lose frames, freeze

or completely lose the result of the video image, which will seriously affect the operator's performance of the construction machinery. Besides that, high latency will make a video to be out of sync. In these cases, the operator cannot grasp the on-site working environment in realtime, resulting in the operator to make wrong judgments about the working environment, which is very dangerous for the work task and the construction machinery. The average delay of the 5G network is lower than that of the LTE network, as shown in Fig. 7.12(c). This is because the 5G network can provide larger network bandwidth, increase network transmission speed, and reduce data packet delay. If the UE number is small, the average delay of the LTE network is about twice that of the 5G network; however, when the number of users is large, the average delay of the LTE network is much higher than that of the 5G network. At this time, the LTE network cannot guarantee the video streaming service.

In the second simulation scenario, the number of UE number is fixed to 8, and the video service data is increased from 1 Mbps to 8 Mbps. The simulation result of throughput with increasing video service rate is shown in Fig. 7.12(d). When the video service rates are 1 Mbps and 2 Mbps, the throughput of the LTE network and the 5G network can meet video streaming services' requirements. However, when the video service rate exceeds 3 Mbps, the throughput of the LTE network does not continue to increase, and the throughput of the 5G network still increases with the video service rate, which can guarantee the transmission of the video service. The simulation results of the packet loss rate are demonstrated in Fig. 7.12(e), where can be seen that the 5G network has been able to maintain the packet loss rate at a low level. However, severe packet loss will occur for LTE networks when a higher video service rate is required. In case that the video service rate is 5 Mbps, the packet loss rate of LTE exceeds 50%. The average delay of video services is presented in Fig. 7.12(f). 5G network continues to increase with the increase of data volume, and they are all maintained at a low level, even when the video service rate is 5 Mbps, the average delay still does not exceed 25 ms. The video service average delay of the LTE network is significantly higher than that of the 5G network. In short, as the video service rate goes higher, the improvement with 5G will be more significant.

In the third scenario, I want to simulate the case that construction machines carry the cameras with them when they change their positions. Here the video service data rate is 2 Mbps, and the number of remote devices is still 8. I simulate the longest distance up to 200 m since the longest propagation distance of mmWave is considered as 200 m [295]. The simulation results of throughput with increasing speed are shown in Fig. 7.13(a). Due to lower frequency bands, LTE network performances are affected only slightly with mobility. Also, when the UE velocity is lower than 40 km/h, the throughput of the 5G network is still in a relatively stable decline stage. In contrast, when the UE velocity exceeds 40 km/h, the throughput of the 5G network drops dramatically, and thus the transmission of video services cannot be guaranteed at this time. Fig. 7.13(b) presents, as the velocity increases, the packet loss rate is rising slowly for LTE networks. However, the 5G network will suffer a fast increasing packet loss rate when the UE moves faster than 30 km/h. In Fig. 7.13(c), both the delay of the LTE network and 5G network increase steadily with the growth of velocity. Noteworthy, the delay of the 5G network still much advantageous compared with LTE.

To sum up, 5G mmWave has significant advantages in terms of throughput, packet loss, and latency if the UEs are fixed. Although one of the requirements for 5G is the capacity to deal with high mobility, the mmWave 5G may still have a problem if the beamforming technology, concretely, tracking algorithm, is not perfect. In contrast, since 4G uses a lower frequency band, this problem is not so apparent for 4G, which hints the suitability of using sub 6 GHz band 5G.

## 7.5 Conclusion

In this chapter, I suggest that the IEEE 802.11p is a preferable solution for the first version of the fleet management of mobile working machines based on the analysis of the ad-hoc network and the cellular network. Moreover, I propose an analytical model to let mobile working machines have a realtime sense of the packet delay probability, mean delay, and the probability of packet loss in the ad-hoc network. That is, the machine can estimate how probable its transmission

can be delayed, how long its transmission can be delayed and how many packets can be lost in realtime. Thanks to V2X technology, mobile machines can work closer and be driven faster so that the productivity of the working site can be increased dramatically.

Afterward, I indicate that 5G can be employed in the construction machines industry to improve the remote control operation and work as an essential component to achieve self-working construction machines. By taking the remote-control and self-working of construction machinery as the scenes and using video streaming transmission as the medium, I compared the LTE network's performance and the 5G mmWave network. Based on my research, I found that 5G has the capability to accomplish a better quality of live streaming so that both scenes can be significantly improved. Especially, 5G can let more cameras in the same network, indicating the possibility to acquire depth information from the video. Besides, since it is not difficult to let the machines always under the cameras' vision, I suggest letting the cameras unmoved avoid the shortcoming of mmWave. Otherwise, more robust beamforming, i.e., dynamic beamforming, algorithm is needed.



## 8 Conclusions and Future Directions

This chapter gives a summary of conclusions that were made throughout this dissertation. Also, I specify the blank and blind spots of the conducted research, and delineate perspective on future directions. To avoid repetition, the conclusions shown in the previous individual chapters will not be shown again.

This thesis has proposed a novel concept of the smart working site initially focusing on increasing the productivity of working sites. Also, by integrating the AI and IoT technologies developed in this thesis, the safety performance and cost of the working site are expected to be ameliorated in the meantime. The expected applications are construction sites and mining sites, where currently tortured by low productivity caused by waiting, high-risk potential, and lack of laborers. An individual technology cannot achieve the goal since working sites are complicated and should be optimized as holistic systems.

To provide an alternative, I have presented the fleet management solution. Considering a group of mobile machines as a whole, I showed the blueprint of a future working site using five complementary technologies to make this concept closer to reality: multi-working machines pathfinding algorithm, multi GPS/IMU SLAM system to offer terrain information, working process detection algorithm, visual monitoring system, and wireless communication system.

The validity of the proposed model has been substantiated by comprehensive experiments. Since I expect the smart working site concept to make a sudden change in the industry, the experiments' results were gained with commodity

hardware, or the parameters setting of the validation simulations were done based on affordable sensor's datasheet.

Besides the contributions are shown in the Chapter 1 that I have pushed forward the SOTA solutions for the individual presented task, I believe another main contribution of this thesis is that I quantitatively evaluated and proved the feasibility of future smart working considering the cutting edge AIoT technologies.

Regarding future extensions, although I did some contributions, the concept of smart working site is a comprehensive topic and cannot be completed with only one dissertation. In this section, I point out the shortcomings of our research and give directions for improvements.

**Path Planning:** As I mentioned in the literature review, path planning is a fast developing and prosperous research field; thus, I did not fully consider all the improved methods for our method's initial version. For instance, I did not add in the mega-agents concept, which merges the agents together based on some specific rules to reduce the conflicts and thus speed up the searching process. Since I confirm that our method can be combined with these methods, I expect the searching process to be accelerated further.

In addition, although Huoshenshan's working site proved the concept, the more machines invested, the faster is the project, which is also consistent with our subjective imagination, a comprehensive study on the quantitative relationship between the number of machines invested and the productivity of the working site has not been done. I encourage the experts in civil engineering to propose some challenging scenarios and test our algorithm on them.

**SLAM:** In our research, I show the method to create a map with only one mobile machine. However, in a real working site, many mobile machines work simultaneously on the construction site, indicating the possibility of creating a map even faster if the machines can share the information. Thus, I encourage the researchers to enable the cooperative map drawing approach by means of WiFi or 5G.



**Motion Prediction:** Although the proposed deep learning algorithm can successfully handle the time series problem to know the machine's working process, a combination with video technology can surely improve the motion prediction accuracy. This combination has not been done in this thesis. Another regret is that I did not spend much time optimizing the CRDNNs to detect truck loading processes due to the limited time. With further optimization, at least the training parameters can be reduced so that an even faster CRDNN can be expected.

**Human Machine Communication:** The IoT system designed for human-machine communication shall be further developed due to its potential. The connected mobile machine is undoubtedly a research focus shortly. While the Bluetooth technology is considered as a cheap and reliable communication solution for human and machines interaction, I believe the next generation communication tools should have access to cellular networks (4G or 5G) since the other components, such as hydraulic pump and hydraulic motor, of the mobile machines also have the requisite to connect to the communication networks for components monitoring, which might overload the Bluetooth. Moreover, I believe fleet management can facilitate the industry of mobile machines. Therefore, in the next generation of the connection system, I will take advantage of 5G to achieve a fully connected working site. Thanks to the cloud, CRDNN can be further trained with newly gathered data whenever a customer label the new dataset for their newly developed mobile construction machines and thus become even more reliable.

**The MOMA Dataset:** The task of object detection relates to a wide range of knowledge, experience, and hardware allocations. A further in-depth study of mobile machine detection algorithms to promote their performance in precision and fps is highly recommended. Current MOMA dataset is relatively small and only suitable for level-four tasks. To achieve better performance, the size of the dataset should be increased. Besides algorithmic improvement, some improvements in the dataset can be concluded as follows. In this dataset, the mobile machines are treated as a whole, whereas perceiving component or sub-assembly of mobile machines makes sense somehow, for instance, bucket or backhoes of an excavator. In addition, collect extra data of mobile machines in

extreme poses if needed. The majority of mobile machines work in typical poses, for instance, an excavator sits on the ground or even in the water, with its bucket moving around; a wheel loader loads coal and unloads it. However, machines must work in extreme poses in some situations, e.g., a dumper deposits earth or a wheel loader buried in the earth but still feebly recognizable by human. By collecting more images like this may expand the scope of model application. Finally, besides object detection, computer vision is also trending to image segmentation. Pixel-level semantic segmentation can also improve the detection performance of predictors.

**V2X Communication – 5G:** Since I use video as the medium to test the performance of the two networks, future work shall refine video factors and explore how the structure of the different encoding video styles will affect the networks. Besides, starting from the video phase, through the networks, and finally to the control operator, a simulation analysis of the entire link can be carried out to improve this article's content. Moreover, as the 6G technology is on the way [296, 297, 298], the possibility of benefiting the construction machine industry from 6G technology shall be explored.

Obviously, the research about smart working site is just at the beginning.

# List of Figures

2.1	An example of a smart construction site illustrated by Komatsu. . . . .	8
2.2	The critical topics for smart working site summarized by Štefanič [17] and us. . . . .	10
2.3	Selected works about smart working site [14, 25, 20, 23, 22, 16, 24, 26, 27]. . . . .	12
3.1	Overview of the smart construction site concept. I introduce to use artificial intelligence to unify the scheduling of construction machinery on construction sites. Maximize the use of construction machinery on specific construction sites by avoiding conflicts among construction machinery and thus ultimately improve the productivity of the construction sites. . . . .	17
3.2	A grid map with terrain weight based on a real construction site, drawn by Liu [81]. In this map, the green, orange, and red grids demonstrate the easy, normal, and difficult to pass terrain, separately. The blue cells denote the place where it is considered impossible to pass, such as occupied by the obstacles. On a real construction site, the obstacles can be the place to store construction materials temporarily. . . . .	26

3.3 An example of multilayered grid maps. My approach depends on multilayered grid maps to offer data of different types of information to make the best path. Concretely, every grid saves a  $1*3$  matrix, including location information and the corresponding terrain information. The map, which can be visualized in the BIM system, is saved as a  $2*m*n*3$  tensor, where  $m$  is the max displacement in the  $x$ -direction, while  $n$  is the max displacement in the  $y$ -direction. In case a place is unknown, it will be marked as NaN to denote the uncertainty of the regions and be treated as obstacles. . . . . 28

3.4 Detail description of a layer in the grid-based map. . . . . 28

3.5 The basic requirements of the path planning algorithm. As shown in the left subfigure, the vehicle should take the lowest-cost path to reach its goal. The middle subfigure shows that the vehicle with lower priority should wait until the vehicle with higher priority pass through if there is no other bypass possibility. Last but not least, the right subfigure indicates that a good path should not be too close to dangerous objects. . . . . 30

3.6 Illustration of the benefit of bidirectional search with heuristics. I use squares and triangles to represent the start point and the goal point separately. Here the grey region denotes the whole region of the map, and the green region shows the region that the algorithm must search before finding the shortest path for agents. Apparently, bidirectional search with heuristics cannot be slower than its antecessors. . . . . 33

3.7 Map example drawn by Liu [81]. Another map based on a real construction site which has more narrow corridor. . . . . 38

3.8 The planned path for each map. In the clockwise direction, the subfigures demonstrate the final solutions, including the best path for each machine for maps 1, 3, 4, 5. The left bottom point is defined as the original point (0,0), and the horizontal axis is the first axis. The layout of map 2 is the same as map 1; however, the difference is that there are more agents on map 2. Due to its huge amount of information, I give the planned schedule in Tab. 3.2 instead of using figures. . . . . 40

- 
- 3.9 The place where the agents intend to pass through and the resulting conflicts on the second map. As it shown from the points allocation, initial conflicts have a great relevance to the conflicts occurring during the conflicts avoidance process. Notice that I did the experiment 50 times and the conflicts shown in these figures are the average number of these 50 experiments. . . . 42
- 3.10 The place where the agents intend to pass through and the resulting conflicts on the fourth map. As we can see from the points allocation, initial conflicts still have a great relevance to the conflicts occurs during the conflicts update process. . . . . 43
- 3.11 Statistics of the conflicts made by corresponding agents on map 2. Here blue histogram denotes the conflicts found by initial bidirectional search, and the orange histogram shows the conflicts solved while updating the solution with unidirectional search. 44
- 3.12 Statistics of the conflicts made by corresponding agents on map 4. . . 44
- 4.1 Mobile machines perform tasks more efficiently or safer according to their location and surroundings information. The short-term goal of SLAM is to prevent construction machinery from always working in low-efficiency areas for safety reasons, whereas the long-term goal is to increase the productivity of the working site with the help of path planning. The chapter focuses on affordable SLAM technology for construction machines. . . . . 50
- 4.2 Wheel loader model in Gazebo: once the models had been developed in Solidwork, they were converted to Unified Robotic Description Format (URDF), using a 3rd party URDF conversion tool called “*sw\_urdf\_exporter*”, which allows for conveniently export SW Parts and Assemblies into a URDF file. Gazebo enables us to obtain sensors’ simulation such as IMUs, GPSs, encoders, cameras, and stereo cameras through *gazebo\_plugins*, which can be used to attach into ROS messages and service calling the sensor outputs, i.e., the *gazebo\_plugins* create a complete interface (Topic) between ROS and Gazebo. . . . . 56
- 4.3 The dynamic system simulated by URDF file on ROS. . . . . 58

4.4 A grid map created by SLAM. My approach uses multilayered grid maps to store data for different types of information. Concretely, every grid saves a  $1 \times 3$  matrix including location information and resistance or grade, depending on which layer it is. A grid with site information will be created after the vehicle passes by. The map is saved as a  $2 \times m \times n \times 3$  tensor, where  $m$  is the max displacement in the  $x$ -direction, while  $n$  is the max displacement in the  $y$ -direction. In case a grid does not be occupied once by the vehicle, it will be marked as NaN to denote the unknown regions. . . . . 66

4.5 The ground truth map with dimensions. The simulation environment I used in Gazebo was modeled based on a real construction site, and the parameters are selected according to material characteristics. Since simulating a small construction site may cause system error and thus lack plausibility, I augmented this real construction site's dimensions in Gazebo. . . . . 69

4.6 The IMU + Odometry estimation method yields for both EKF and UKF the RMSE over 70 m. Except for the IMU + Odometry methods, the other results can be divided into four performance levels according to the error scale. The worst level is EKF with 1 GPS, where the RMSE is about  $3 \sim 4$  m. The second level is EKF with 2 GPSes and EKF with 3 IMUs and 3 GPSes. In these cases, the RMSEs are about  $2 \sim 3$  m. A better level is UKF with 1 GPS and 1 IMU, where the RMSE of the third level can be achieved about  $2 \sim 2.5$  m. Finally, my experiment's best class is UKF with 2 GPSes and 1 IMU, and UKF with 3 IMUs and 3 GPSes, which reduce the RMSE to about 1.2 m. . . . . 72

4.7 Quantitative evaluation of different methods. Here (a) and (b) show the accumulative error, i.e., RMSE, while (c) and (d) demonstrate the current Euclidean distance error of each sensor arrangement. There are some noticeable instantaneous position changes every ten seconds due to infrequent GPS signal loss. Generally, UKF shows better performance than EKF for both RMSE and euclidean distance error. . . . . 73

4.8	The ground truth and estimated maps. In the ground resistance map and road grade map plotted by EKF with 1 IMU and 1 GPS, the spikes caused by infrequent GPS are quite obvious. With additional GPS sensor fused in Kalman filter, the spikes improve a lot.	75
4.9	Difference between ground truth and the estimated map. Here I compare the predefined areas and the plotted path, where the white pixels are the wrong plotted grids. (a),(b),(c) are the Friction map results, and (d),(e),(f) are the road grade map results. UKF shows more accurate positioning capabilities than EKF, and with two GPSes fused in the Kalman filter, the wrong located grid is less than just with one GPS signal. . . . .	77
5.1	A typical truck loading process (Y cycle). . . . .	81
5.2	Segment of control concepts [121]. . . . .	81
5.3	Normalized measurement data with label. It is important to highlight that I consciously mislabel the data around 450 s to test the model robustness. . . . .	87
5.4	Detailed description of CRDNN with two LSTMs. The notation I use is based on the Stanford University deep learning Lecture Notes, where $c^{[i]<j>}$ denotes the cell in $i_{th}$ layer and for $j_{th}$ time series.	90
5.5	Training- and test costs versus epochs. . . . .	92
5.6	Confusion matrices of CRDNNs. . . . .	93
5.7	Ground truth and prediction mistakes. As reminder, the state 0,1,2 represent the travelling, loading, and unloading process, respectively. Blue line denotes the ground truth. Color points show the predict results which are different as the ground truth. As can be seen, most of them are at the place where state changes. This indicates the further accuracy improvement may not make sense. Since the sample frequency is 5 Hz, the windows around 2,250th reveal the information about 450 s in the original data shown in Fig.5.3. The deep learning model is robust though there are some mistakes in ground truth dataset. . . .	96

5.8 Graphical abstract. Here the core data is a large dataset that contains 119 Y cycles data from many wheel loaders. This core dataset is used to train the base network. Thanks to this base network, I can then use transfer learning to adapt the weights in this base network with the new data to improve the generalization ability, easy and quick. The method is proposed to solve the problem pointed out by many machine learning researchers, the distribution of the source data may differ from the target data since the collection of a comprehensive dataset is, in many cases, impossible. . . . . 99

5.9 LSTM-FCN with squeeze-and-excite block [5]. . . . . 101

5.10 The confusion matrix of CRDNN on new data. The  $e_i$  are the ground truth and the  $\hat{e}_i$  are the predictive state. As defined in my previous work, the  $e_{1,2,3}$  denote the state travelling, loading, and unloading, separately. . . . . 103

5.11 The sketch of the human-machine communication App system. . . . 111

5.12 The mobile machine used for the measurement data. . . . . 112

5.13 New measurement data with a different implement control algorithm and dimension of the shovel. The last subfigure shows the ground truth state of the measured data. . . . . 113

5.14 The diagram of the relabeling method. . . . . 115

5.15 The diagram of the relabeling method. . . . . 115

5.16 Cost versus epoches of CRDNNs from scratch and with transfer learning. . . . . 119

5.17 Confusion matrices of CRDNNs from scratch and with transfer learning. . . . . 121

5.18 The mechanism of transfer learning. The blue line is the training cost on ( $\mathcal{D}_s$ ), the red line is the validation cost ( $\mathcal{T}_s$ ), the purple line is the training cost on ( $\mathcal{D}_t$ ), and the cyan line is the validation cost ( $\mathcal{T}_t$ ). . . . . 123



---

6.1	Sample images in dataset MOMA, including 7 classes of construction machines as well as person with varying poses on the working scenarios. Images in column (a): objects in iconic view; column (b) objects under partial occlusion; column (c) objects in varying poses; column (d) objects in non-iconic perspective.	130
6.2	The statistics of the MOMA dataset.	137
6.3	Hierarchical structure of the MOMA dataset, based on PASCAL VOC.	140
6.4	Labelling tool. Label tool “labelImg” can load multiple images under a directory by clicking “Open Dir” on the menu, and save under pre-defined path. The shown bounding boxes that closely surround object ground truths, which the target instance is excavator in this figure, were made bold for the salience. Multi-class and multi-label for one single instance are possible; all marked labels are at the top right corners. As I consider the PASCAL VOC format as the standard format for the the MOMA dataset, all annotation files are saved in XML format.	141
6.5	Label example. Two excavators and two trucks should be labeled in this image. They can all be distinguished by eyesight even though they are partially blocked.	145
6.6	Label example. Four objects can be clearly seen in the image; even the excavator in the distance is much smaller than the one nearby. Moreover, the two standing workers can be recognized as well.	145
6.7	Label example. Rectangular correctly encompass ground truths: a dumper, an excavator as well as two persons. However, other trucks can be inferred from context stream frames but are not identifiable in the single image. The label should be ignored if YOLOv3 is used to detect mobile machines.	147
6.8	Prediction samples with optimized dataset and algorithm YOLOv3.	149
6.9	Class distribution on 666 images: the instances number of truck and excavator outnumber bulldozer and car since they attract more interest.	151
6.10	Sample of inference results by the 8,000th predictor on images in iconic view.	151
6.11	mAP over batches, trained with a balanced dataset.	152
6.12	mAP over batches, trained with an unbalanced dataset.	153

6.13 Hierarchy of predictor with skip connections, e.g. 94th layer, responsible for detecting medium-size targets, relates to the 61st layer before downsampling. Likewise, 36th layer is directly connected to 91st layer by a short cut. . . . . 154

6.14 Inference by predictor at the 12,000th batch made on images with a non-iconic view. . . . . 155

6.15 Prediction performance contrast by three predictors, which were made on images with both iconic view and non-iconic view. . . . . 156

6.16 Individual prediction AP for each class made by predictors at batch 1,900 in blue, 8,000 in orange. The green column demonstrates the performance if I take the assumption that only known mobile machines are working in the working site. i.e. level 4 autonomous driving case. . . . . 157

6.17 Individual prediction AP for each class on the other 5,663 images. . . 157

7.1 Comparison the working site with/without V2X: More mobile working machines on the site, much higher productivity. . . . . 161

7.2 The hidden node problem. . . . . 170

7.3 Schematic view of simulation scenarios. . . . . 171

7.4 Mean delay [ $\mu s$ ] versus distance between two groups of neighbor nodes [m]. . . . . 172

7.5 Packet collision probability versus distance between two groups of neighbor nodes [m]. . . . . 173

7.6 Packet loss probability versus distance between two groups of neighbor nodes [m]. . . . . 173

7.7 Packet delay probability, packet collision probability, packet loss probability and mean delay measured with varying number of neighbors in 2 clusters are included in the LuT. . . . . 175

7.8 Correlation coefficients of 3 metrics are close to 1, which indicate a good feasibility of analytical estimation. To increase estimation accuracy, I introduce  $\hat{f}_c$ . . . . . 178

7.9 Remote control with live streaming: here cameras will be installed on the mobile machines while the driver sits in a comfortable room to operate the machines remotely. Thanks to 5G, HD video streaming can be sent with low delay and high reliability. 181

---

7.10	Self-working mobile machines: here, cameras will be fixed on the ground instead of being installed on the machines to avoid the obstruction of vision. The stream will be uploaded to the center commander and be processed on the cloud. Based on the stream from more than two cameras, the depth information and motion of machines can be acquired. Afterward, the command signal will be sent directly to the machines. The research about instance segmentation of construction machines can be found in [271].	182
7.11	Network topology. In this figure, the upper left corner is the origin of the coordinates. The side length of each square grid is set as 25 m. . . . .	190
7.12	Simulation results of scenario 1 and scenario 2. . . . .	191
7.13	Simulation results of scenario 3. . . . .	192



# List of Tables

3.1	Weight table. The weights I use to describe the complicated terrain of construction sites . . . . .	37
3.2	The schedule of MAPF task on map 2. In this task, the algorithm should plan the path for 50 agents . . . . .	41
3.3	The average algorithm searching duration (s). “None” denote that there is no need to optimize since the response is already quick enough . . . . .	46
4.1	Sensor configuration. In my simulation, I ignore the mirror difference caused by the slightly different installation position of sensors. Therefore, the various configurations are reduced from 64 to 16. Since odometry is robust and necessary for many applications, I do not consider the case without an encoder . . . . .	70
4.2	Localization errors of each approach . . . . .	74
4.3	Localization errors of each approach during the plotting . . . . .	76
5.1	Parameters of CRDNN . . . . .	94
5.2	The main specifications of the selected wireless communication technology . . . . .	109
5.3	Performance analysis. The performance of the five network structures in respect of total training time (s), micro F1 (%), average test duration (ms), and whether it can never mistake an unloading into loading or in reverse . . . . .	117
5.4	Parameters of CRDNN with 2 LSTMs . . . . .	118
5.5	The performance comparison between different training methods . . . . .	120
5.6	Performance comparison of different learning approach . . . . .	125
7.1	Simulation parameters . . . . .	169

7.2	Correction factors . . . . .	179
7.3	LTE Network Parameters, from 3GPP TS-36101 [281] . . . . .	186
7.4	5G Network Parameters, From 3GPP TS-38101 [254] . . . . .	187
7.5	Network Scenario 1 . . . . .	189
7.6	Network Scenario 2 . . . . .	189
7.7	Network Scenario 3 . . . . .	189

# Bibliography

- [1] A. Burke and S. Fishel, “A coal elimination treaty 2030: Fast tracking climate change mitigation, global health and security,” *Earth System Governance*, vol. 3, no. 1, p. 100046, 2020.
- [2] F. Geng and J. H. Saleh, “Challenging the emerging narrative: Critical examination of coalmining safety in china, and recommendations for tackling mining hazards,” *Safety Science*, vol. 75, no. 2003, pp. 36–48, 2015.
- [3] D. D. Landen, J. T. Wassell, L. McWilliams, and A. Patel, “Coal dust exposure and mortality from ischemic heart disease among a cohort of U.S. coal miners,” *American journal of industrial medicine*, vol. 54, no. 10, pp. 727–733, 2011.
- [4] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, “Conflict-based search for optimal multi-agent path finding,” *Artificial Intelligence*, vol. 219, pp. 40–66., 2015.
- [5] F. Karim, S. Majumdar, H. Darabi, and S. Harford, “Multivariate LSTM-FCNs for time series classification,” *Neural Networks*, vol. 116, pp. 237–245, 2019.
- [6] R. Flanagan and W. Lu, “Making informed decisions in product-service systems,” in *IMEchE Conference, Knowledge and Information Management Through-Life*, London, UK, 2008, pp. 1–8.
- [7] K. Chen, W. Lu, Y. Peng, S. Rowlinson, and G. Q. Huang, “Bridging BIM and building: From a literature review to an integrated conceptual

- framework,” *International Journal of Project Management*, vol. 33, no. 6, pp. 1405–1416, 2015.
- [8] J. D. Goedert and P. Meadati, “Integrating construction process documentation into building information modeling,” *Journal of Construction Engineering and Management*, vol. 134, no. 7, pp. 509–516, 2008.
- [9] C. M. Eastman, Y. S. Jeong, R. Sacks, and I. Kaner, “Exchange model and exchange object concepts for implementation of national BIM standards,” *Journal of Computing in Civil Engineering*, vol. 24, pp. 25–35, 2010.
- [10] E. J. Jaselskis and T. El-Misalami, “Implementing radio frequency identification in the construction process,” *Journal of Construction Engineering and Management*, vol. 129, no. 6, pp. 680–688, 2003.
- [11] K. Domdouzis, B. Kumar, and C. Anumba, “Radio-frequency identification (RFID) applications: A brief introduction,” *Advanced Engineering Informatics*, vol. 21, no. 4, pp. 350–355, 2007.
- [12] W. Lu, G. Q. Huang, and H. Li, “Scenarios for applying RFID technology in construction project management,” *Automation in Construction*, vol. 20, no. 2, pp. 101–106, 2011.
- [13] R. Flanagan, C. Jewell, W. Lu, and M. K. Pekerikli, *Auto ID-Bridging the physical and the digital on construction*. Bracknell, Berkshire, UK: Chartered Institute of Building, 2014.
- [14] T. Kawakami, B. L. N. Ly, S. Takeuchi, Y. Teranishi, K. Harumoto, and S. Nishio, “Distributed sensor information management architecture based on semantic analysis of sensing data,” in *International Symposium on Applications and the Internet*. Turku, Finland: IEEE, 2008, pp. 353–356.
- [15] M. P. Kolba and L. M. Collins, “Sensor management with uncertain performance characteristics,” in *Proc. Fourth IEEE Workshop on Sensor Array and Multichannel Processing, 2006*. Waltham, MA, USA: IEEE, 2006, pp. 456–460.



- [16] E. R. Azar and V. R. Kamat, "Earthmoving equipment automation: a review of technical advances and future outlook." *Journal of Information Technology in Construction (ITcon)*, vol. 22, no. 13, pp. 247–265, 2017.
- [17] M. Štefanič and V. Stankovski, "A review of technologies and applications for smart construction," *Proc. of the Institution of Civil Engineers - Civil Engineering*, vol. 172, no. 2, pp. 83–87, 2019.
- [18] Y. Pan and L. Zhang, "Roles of artificial intelligence in construction engineering and management: A critical review and future trends," *Automation in Construction*, vol. 122, no. 9, p. 103517, 2021.
- [19] K. M. Lundeen, S. Dong, N. Fredricks, M. Akula, J. Seo, and V. R. Kamat, "Optical marker-based end effector pose estimation for articulated excavators," *Automation in Construction*, vol. 65, pp. 51–64, 2016.
- [20] Y. Turkan, F. Bosche, C. T. Haas, and R. Haas, "Automated progress tracking using 4D schedule and 3D sensing technologies," *Automation in Construction*, vol. 22, pp. 414–421, 2012.
- [21] W. Ren and Z. Wu, "Real-time anticollision system for mobile cranes during lift operations," *Journal of Computing in Civil Engineering*, vol. 29, no. 6, p. 04014100, 2015.
- [22] D. Liu, Y. Wu, S. Li, and Y. Sun, "A real-time monitoring system for lift-thickness control in highway construction," *Automation in Construction*, vol. 63, pp. 27–36, 2016.
- [23] A. Braun, S. Tuttas, A. Borrmann, and U. Stilla, "A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds," *Journal of Information Technology in Construction*, vol. 20, no. 8, pp. 68–79, 2015.
- [24] H. Kim, H. S. Lee, M. Park, B. Chung, and S. Hwang, "Automated hazardous area identification using laborers' actual and optimal routes," *Automation in Construction*, vol. 65, pp. 21–32, 2016.

- [25] A. Borrmann and E. Rank, “Topological analysis of 3D building models using a spatialquery language,” *Advanced Engineering Informatics*, vol. 23, no. 4, pp. 370–385, 2009.
- [26] Y. Niu, W. Lu, K. Chen, G. Q. Huang, and C. Anumba, “Smart construction objects,” *Journal of Computing in Civil Engineering*, vol. 30, no. 4, p. 04015070, 2016.
- [27] H. Zhou, H. Wang, and W. Zeng, “Smart construction site in mega construction projects: A case study on island tunneling project of Hong Kong-Zhuhai-Macao bridge,” *Frontiers of Engineering Management*, vol. 5, no. 1, pp. 78–87, 2018.
- [28] H. Luo, J. Liu, C. Li, K. Chen, and M. Zhang, “Ultra-rapid delivery of specialty field hospitals to combat COVID-19: Lessons learned from the Leishenshan hospital project in wuhan,” *Automation in Construction*, vol. 119, p. 103345, 2020.
- [29] L.-K. Chen, R.-P. Yuan, X.-J. Ji, X.-Y. Lu, J. Xiao, J.-B. Tao, X. Kang, X. Li, Z.-H. He, S. Quan *et al.*, “Modular composite building in urgent emergency engineering projects: A case study of accelerated design and construction of Wuhan Thunder God Mountain/Leishenshan hospital to COVID-19 pandemic,” *Automation in Construction*, vol. 124, p. 103555, 2021.
- [30] Y. Xiang, K. Liu, T. Su, J. Li, S. Ouyang, S. Mao, and M. Geimer, “An extension of BIM using AI: A multi working-machines pathfinding solution,” *IEEE Access*, vol. 9, pp. 124 583–124 599, 2021.
- [31] A. Borrmann, M. König, C. Koch, and J. Beetz, in *Building Information Modeling*. Cham: Springer International Publishing, 2018, pp. 1–24.
- [32] K. Barlish and K. Sullivan, “How to measure the benefits of BIM — a case study approach,” *Automation in Construction*, vol. 24, pp. 149–159, 2012.
- [33] Y. Liu, S. van Nederveen, and M. Hertogh, “Understanding effects of BIM on collaborative design and construction: An empirical study in china,”

- 
- International Journal of Project Management*, vol. 35, no. 4, pp. 686–698, 2017.
- [34] R. Volk, J. Stengel, and F. Schultmann, “Building Information Modeling (BIM) for existing buildings — literature review and future needs,” *Automation in Construction*, vol. 38, pp. 109–127, 2014.
- [35] S. Song and E. Marks, “Construction site path planning optimization through BIM,” in *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*. American Society of Civil Engineers Reston, VA, 2019, pp. 369–276.
- [36] A. R. Soltani, H. Tawfik, J. Y. Goulermas, and T. Fernando, “Path planning in construction sites: performance evaluation of the Dijkstra, A\*, and GA search algorithms,” *Advanced Engineering Informatics*, vol. 16, no. 4, pp. 291–303, 2002.
- [37] A. Khalafallah and K. El-Rayes, “Automated multi-objective optimization system for airport site layouts,” *Automation in Construction*, vol. 20, no. 4, pp. 313–320, 2011.
- [38] M. Yahya and M. P. Saka, “Construction site layout planning using multi-objective artificial bee colony algorithm with levy flights,” *Automation in Construction*, vol. 38, pp. 14–29, 2014.
- [39] H. Ma, W. Hönig, T. S. Kumar, N. Ayanian, and S. Koenig, “Lifelong path planning with kinematic constraints for multi-agent pickup and delivery,” in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 33, Hilton Hawaiian Village, Honolulu, Hawaii, USA, 2019, pp. 7651–7658.
- [40] Y. Xiang, T. Su, C. Brach, X. Liu, and M. Geimer, “Realtime estimation of IEEE 802.11p for mobile working machines communication respecting delay and packet loss,” in *Proc. IEEE Intelligent Vehicle Symposium*, Las Vegas, USA, 2020, pp. 1516–1521.
- [41] Y. Xiang, B. Xu, T. Su, C. Brach, S. S. Mao, and M. Geimer, “5G meets construction machines: Towards a smart working site,” in *Proc. IEEE*

- International Conference on Computing and Communications Applications and Technologies (I3CAT)*, Virtual, 2021, pp. 1–8.
- [42] K. Whitlock, F. H. Abanda, M. B. Manjia, C. Pettang, and G. E. Nkeng, “BIM for construction site logistics management,” *Journal of Engineering, Project, and Production Management*, vol. 8, no. 1, p. 47, 2018.
- [43] J. H. Woo, “BIM (building information modeling) and pedagogical challenges,” in *Proc. of the 43rd ASC national annual conference*, Flagstaff, AZ, USA, 2006, pp. 12–14.
- [44] S. Azhar, “Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry,” *Leadership and Management in Engineering*, vol. 11, no. 3, pp. 241–252, 2011.
- [45] F. P. Rahimian, S. Seyedzadeh, S. Oliver, S. Rodriguez, and N. Dawood, “On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning,” *Automation in Construction*, vol. 110, p. 103012, 2020.
- [46] R. Sacks, M. Girolami, and I. Brilakis, “Building information modelling, artificial intelligence and construction tech,” *Developments in the Built Environment*, vol. 4, p. 100011, 2020.
- [47] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, “A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends,” *Automation in Construction*, vol. 101, pp. 127–139, 2019.
- [48] G. Sullivan, S. Barthorpe, and S. Robbins, *Managing construction logistics*. Oxford, U.K: John Wiley & Sons, 2011.
- [49] S. Hedborg Bengtsson, “Coordinated construction logistics: an innovation perspective,” *Construction Management and Economics*, vol. 37, no. 5, pp. 294–307, 2019.

- 
- [50] M. Hirayama, J. Guivant, J. Katupitiya, and M. Whitty, “Path planning for autonomous bulldozers,” *Mechatronics*, vol. 58, pp. 20–38, 2019.
- [51] —, “Artificial intelligence in path planning for autonomous bulldozers: Comparison with manual operation,” *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol. 15, pp. 825–844, 2019.
- [52] Z. Sun, M. Nakatani, and Y. Uchimura, “Optimal routing control of a construction machine by deep reinforcement learning,” in *Proc. IEEE 15th International Workshop on Advanced Motion Control (AMC)*, Tokyo, Japan, 2018, pp. 187–192.
- [53] S. K. Kim, J. S. Russell, and K. J. Koo, “Construction robot path-planning for earthwork operations,” *Journal of Computing in Civil Engineering*, vol. 17, no. 2, pp. 97–104, 2003.
- [54] S. H. Lee and T. M. Adams, “Spatial model for path planning of multiple mobile construction robots,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 19, no. 4, pp. 231–245, 2004.
- [55] A. R. Soltani and T. Fernando, “A fuzzy based multi-objective path planning of construction sites,” *Automation in Construction*, vol. 13, no. 6, pp. 717–734, 2004.
- [56] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [57] P. Hart, N. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [58] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, MA, USA: MIT press, 1992.
- [59] T. Cheng, U. Mantripragada, J. Teizer, and P. A. Vela, “Automated trajectory and path planning analysis based on ultra wideband data,” *Journal of Computing in Civil Engineering*, vol. 26, no. 2, pp. 151–160, 2012.

- [60] G. Bohács, A. Gyimesi, and Z. Rózsa, “Development of an intelligent path planning method for materials handling machinery at construction sites,” *Periodica Polytechnica Transportation Engineering*, vol. 44, no. 1, pp. 13–22, 2016.
- [61] C. J. Turner, J. Oyekan, L. Stergioulas, and D. Griffin, “Utilizing industry 4.0 on the construction site: Challenges and opportunities,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 746–756, 2021.
- [62] Y. Xiang, H. Wang, T. Su, R. Li, C. Brach, S. S. Mao, and M. Geimer, “KIT MOMA: A mobile machines dataset,” *arXiv*, 2020. [Online]. Available: arXiv:2007.04198
- [63] Y. Xiang and M. Geimer, “Optimization of operation strategy for primary torque based hydrostatic drivetrain using artificial intelligence,” in *Proc. 12th International Fluid Power Conference*, Dresden, Germany, 2020, pp. 55–65.
- [64] Y. Xiang, T. Tang, T. Su, C. Brach, L. Liu, S. S. Mao, and M. Geimer, “Fast CRDNN: Towards on site training of mobile construction machines,” *IEEE Access*, vol. 9, pp. 124 253–124 267, 2021.
- [65] I. Kurinov, G. Orzechowski, P. Hamalainen, and A. Mikkola, “Automated excavator based on reinforcement learning and multibody system dynamics,” *IEEE Access*, vol. 8, pp. 213 998–214 006, 2020.
- [66] Y. Xiang, D. Li, T. Su, Q. Zhou, C. Brach, S. S. Mao, and M. Geimer, “Where am I? SLAM for mobile machines on a smart working site,” *arXiv*, pp. 1–14, 2020. [Online]. Available: arXiv:2011.01830
- [67] Q. Zhou, J. Li, B. Shuai, H. Williams, Y. He, Z. Li, H. Xu, and F. Yan, “Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle,” *Applied Energy*, vol. 255, p. 113755, 2019.
- [68] Q. Zhou, Y. Zhang, Z. Li, J. Li, H. Xu, and O. Olatunbosun, “Cyber-physical energy-saving control for hybrid aircraft-towing tractor based on

- online swarm intelligent programming,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4149–4158, 2017.
- [69] R. Stern, “Multi-agent path finding – an overview,” in *Artificial Intelligence - 5th RAAI Summer School*. Dolgoprudny, Russia: Springer, 2019, pp. 96–115.
- [70] N. R. Sturtevant, “Benchmarks for grid-based pathfinding,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 2, pp. 144–148, 2012.
- [71] J. Švancara, M. Vlk, R. Stern, D. Atzmon, and R. Barták, “Online multi-agent pathfinding,” in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7732–7739.
- [72] A. Madkour, W. G. Aref, F. U. Rehman, M. A. Rahman, and S. Basalamah, “A survey of shortest-path algorithms,” *arXiv*, 2017. [Online]. Available: arXiv:1705.02044
- [73] A. Felner, R. Stern, S. E. Shimony, E. Boyarski, M. Goldenberg, G. Sharon, N. Sturtevant, G. Wagner, and P. Surynek, “Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges,” in *Tenth Annual Symposium on Combinatorial Search (SoCS)*, Stockholm, Sweden, 2017, pp. 29–37.
- [74] E. Boyarski, A. Felner, G. Sharon, and R. Stern, “Don’t split, try to work it out: Bypassing conflicts in multi-agent pathfinding,” in *Proc. of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS)*, Jerusalem, Israel, 2015, pp. 47–51.
- [75] A. Felner, J. Li, E. Boyarski, H. Ma, L. Cohen, T. K. S. Kumar, and S. Koenig, “Adding heuristics to conflict-based search for multi-agent path finding,” in *Proc. of the International Conference on Automated Planning and Scheduling (ICAPS)*, Delft, The Netherlands, 2018, pp. 83–87.
- [76] J. Li, A. Felner, E. Boyarski, H. Ma, and S. Koenig, “Improved heuristics for multi-agent path finding with conflict-based search,” in *28th International*

- Joint Conference on Artificial Intelligence (IJCAI)*, Macao, China, 2019, pp. 442–449.
- [77] J. Li, D. Harabor, P. J. Stuckey, A. Felner, H. Ma, and S. Koenig, “Disjoint splitting for multi-agent path finding with conflict-based search,” in *International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 29, Berkeley (CA), USA, 2019, pp. 279–283.
- [78] G. Gange, D. Harabor, and P. J. Stuckey, “Lazy CBS: Implicit conflict-based search using lazy clause generation,” in *International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 29, Berkeley (CA), USA, 2019, pp. 155–162.
- [79] W. Hönig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, “Conflict-based search with optimal task assignment,” in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Stockholm, Sweden, 2018, pp. 757–765.
- [80] M. Barer, G. Sharon, R. Stern, and A. Felner, “Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem,” in *International Symposium on Combinatorial Search (SoCS)*, Prague, Czech Republic, 2014, pp. 19–27.
- [81] K. Liu, “An Extension of BIM Using AI: A Multi Working-Machines Pathfinding Solution,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2021.
- [82] P. Fankhauser and M. Hutter, “A universal grid map library: Implementation and use case for rough terrain navigation,” in *Robot Operating System (ROS)*. Springer, 2016, vol. 625, pp. 99–120.
- [83] T. Ikeda, M.-Y. Hsu, H. Imai, S. Nishimura, H. Shimoura, T. Hashimoto, K. Tenmoku, and K. Mitoh, “A fast algorithm for finding better routes by AI search techniques,” in *Proc. of VNIS’94 - 1994 Vehicle Navigation and Information Systems Conference*. Yokohama, Japan: IEEE, 31 Aug.-2 Sept. 1994, pp. 291–296.



- 
- [84] A. Vahidi, A. Stefanopoulou, and H. Peng, "Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments," *Vehicle System Dynamics*, vol. 43, no. 1, pp. 31–55, 2005.
- [85] H. Tan and J. Huang, "DGPS-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 415–428, 2006.
- [86] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [87] S. Jung, J. Kim, and S. Kim, "Simultaneous localization and mapping of a wheel-based autonomous vehicle with ultrasonic sensors," *Artificial Life and Robotics*, vol. 14, no. 2, p. 186, 2009.
- [88] S. Wang, Z. Deng, and G. Yin, "An accurate GPS-IMU/DR data fusion method for driverless car based on a set of predictive models and grid constraints," *Sensors (Basel, Switzerland)*, vol. 16, no. 3, p. 280, 2016.
- [89] D. Perea, J. Hernandez-Aceituno, A. Morell, J. Toledo, A. Hamilton, and L. Acosta, "MCL with sensor fusion based on a weighting mechanism versus a particle generation approach," in *Proc. 16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, The Hague, Netherlands, 2013, pp. 166–171.
- [90] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 572–578, 1999.
- [91] S. Iwataki, H. Fujii, A. Moro, A. Yamashita, H. Asama, and H. Yoshinada, "Visualization of the surrounding environment and operational part in a 3DCG model for the teleoperation of construction machines," in *Proc. IEEE/SICE International Symposium on System Integration (SII)*. The Hague, Netherlands: IEEE, 2015, pp. 81–87.

- [92] I. Skog and P. Handel, “In-car positioning and navigation technologies—a survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 4–21, 2009.
- [93] H. L. Isa, S. A. Saad, A. Hisham, and M. Ishak, “Improvement of GPS accuracy in positioning by using DGPS technique,” in *Modeling, Design and Simulation of Systems*, M. S. Mohamed A., H. Wahid, N. A. Mohd Subha, S. Sahlan, M. A. Md. Yunus, and A. R. Wahap, Eds. Singapore: Springer Singapore, 2017, pp. 3–11.
- [94] J. Kim, J. Song, H. No, D. Han, D. Kim, B. Park, and C. Kee, “Accuracy improvement of DGPS for low-cost single-frequency receiver using modified flächen korrektur parameter correction,” *ISPRS International Journal of Geo-Information*, vol. 6, no. 7, p. 222, 2017.
- [95] E. Zhang and N. Masoud, “Increasing GPS localization accuracy with reinforcement learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2615–2626, 2021.
- [96] G. Soatti, M. Nicoli, N. Garcia, B. Denis, R. Raulefs, and H. Wymeersch, “Implicit cooperative positioning in vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3964–3980, 2018.
- [97] M. Skoglund, T. Petig, B. Vedder, H. Eriksson, and E. M. Schiller, “Static and dynamic performance evaluation of low-cost RTK GPS receivers,” in *Proc. IEEE Intelligent Vehicles Symposium (IV)*. Gotenburg, Sweden: IEEE, 2016, pp. 16–19.
- [98] J. Toledo, J. D. Piñeiro, R. Arnay, D. Acosta, and L. Acosta, “Improving odometric accuracy for an autonomous electric cart,” *Sensors (Basel, Switzerland)*, vol. 18, no. 1, pp. 1–15, 2018.
- [99] X. Qi, W. Wang, M. Yuan, Y. Wang, M. Li, L. Xue, and Y. Sun, “Building semantic grid maps for domestic robot navigation,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, p. 172988141990006, 2020.

- [100] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to non-linear systems,” in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., Orlando, FL, USA, 1997, pp. 182–193.
- [101] E. A. Wan and R. van der Merwe, “The unscented Kalman filter for non-linear estimation,” in *Proc. of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Lake Louise, AB, Canada: IEEE, 2000, pp. 153–158.
- [102] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA: University of North Carolina at Chapel Hill, 1995.
- [103] L. C. Bento, U. Nunes, F. Moita, and A. Surrecio, “Sensor fusion for precise autonomous vehicle navigation in outdoor semi-structured environments,” in *Proc. IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, 2005, pp. 245–250.
- [104] F. Zhang, H. Stahle, G. Chen, C. Chen, C. Simon, C. Buckl, and A. Knoll, “A sensor fusion approach for localization with cumulative error elimination,” in *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Hamburg, Germany, 2012, pp. 1–6.
- [105] C. Li, B. Dai, and T. Wu, “Vision-based precision vehicle localization in urban environments,” in *Chinese Automation Congress*, Changsha, Hunan, China, 2013, pp. 599–604.
- [106] R. W. Wolcott and R. M. Eustice, “Visual localization within LIDAR maps for automated urban driving,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, Illinois, USA, 2014, pp. 176–183.
- [107] E. Ward and J. Folkesson, “Vehicle localization with low cost radar sensors,” in *Proc. IEEE Intelligent Vehicles Symposium (IV)*. Gothenburg, Sweden: IEEE, 2016, pp. 864–870.

- [108] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer LIDAR," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 420–429, 2016.
- [109] J. A. del Peral-Rosado, J. A. Lopez-Salcedo, S. Kim, and G. Seco-Granados, "Feasibility study of 5G-based localization for assisted driving," in *International Conference on Localization and GNSS (ICL-GNSS)*. Barcelona, Spain: IEEE, 2016, pp. 1–6.
- [110] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, 2018.
- [111] J. Larsson, M. Broxvall, and A. Saffiotti, "An evaluation of local autonomy applied to teleoperated vehicles in underground mines," in *Proc. IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010, pp. 1745–1752.
- [112] J. M. Roberts, E. S. Duff, and P. I. Corke, "Reactive navigation and opportunistic localization for autonomous underground mining vehicles," *Information Sciences*, vol. 145, no. 1-2, pp. 127–146, 2002,.
- [113] R. Gu, R. Marinescu, C. Seceleanu, and K. Lundqvist, "Formal verification of an autonomous wheel loader by model checking," in *Proc. ACM/IEEE Conference on Formal Methods in Software Engineering*, Beijing, China, 2018, pp. 74–83.
- [114] S. Bang, F. Baek, S. Park, W. Kim, and H. Kim, "An image augmentation method for detecting construction resources using convolutional neural network and UAV images," in *36th International Symposium on Automation and Robotics in Construction*, M. Al-Hussein, Ed. Banff Alberta, Canada: International Association for Automation and Robotics in Construction (IAARC), 2019.

- 
- [115] Z. Shang and Z. Shen, “Real-time 3D reconstruction on construction site using visual SLAM and UAV,” *arXiv*, 2019. [Online]. Available: arXiv:1712.07122
- [116] Y. Xiang, S. Mutschler, N. Brix, C. Brach, and M. Geimer, “Optimization of hydrostatic-mechanical transmission control strategy by means of torque control,” in *Proc. 12th International Fluid Power Conference*, Dresden, Germany, 2020, pp. 421–431.
- [117] F. Samie, L. Bauer, and J. Henkel, “IoT technologies for embedded computing: A survey,” in *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*. Pittsburgh, USA: IEEE, 2016, pp. 1–10.
- [118] D. Li, “Where am I? SLAM for mobile machines on a smart working site,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.
- [119] T. Moore and D. Stouch, “A generalized extended Kalman filter implementation for the robot operating system,” in *Intelligent autonomous systems 13*. Cham: Springer, 2016, pp. 335–348.
- [120] S. Mutschler, N. Brix, and Y. Xiang, “Torque control for mobile machines,” in *Proc. 11th International Fluid Power Conference*, H. Murrenhoff, Ed. Aachen, Germany: RWTH Publications, 2018, pp. 186–195.
- [121] H. Murrenhoff, *Servohydraulik – Geregelte Hydraulische Antriebe*, 3rd ed. Aachen: Shaker Verlag, 2008.
- [122] G. Bauer and M. Niebergall, *Ölhydraulik: Grundlagen, Bauelemente, Anwendungen*. Wiesbaden, Germany: Springer-Verlag, 2020.
- [123] L. Ge, L. Quan, X. Zhang, J. Huang, and B. Zhao, “High energy efficiency driving of the hydraulic excavator boom with an asymmetric pump,” in *Proc. 11th International Fluid Power Conference*, Aachen, Germany, 2018, pp. 347–357.

- [124] L. Ge, L. Quan, X. Zhang, B. Zhao, and J. Yang, “Efficiency improvement and evaluation of electric hydraulic excavator with speed and displacement variable pump,” *Energy Conversion and Management*, vol. 150, pp. 62–71, 2017.
- [125] M. Vukovic, R. Leifeld, and H. Murrenhoff, “STEAM – a hydraulic hybrid architecture for excavators,” in *Proc. 10th International Fluid Power Conference*. Dresden: Technische Universität Dresden, 2016, pp. 151–162.
- [126] Y. Xiang, R. Li, C. Brach, X. Liu, and M. Geimer, “A novel algorithm for hydrostatic-mechanical mobile machines with a dual-clutch transmission,” in *Proc. 11th IEEE Global Fluid Power Society PhD Symposium*, Guilin, China, 2020, pp. 1–8.
- [127] M. W. Sprengel, “Influence of architecture design on the performance and fuel efficiency of hydraulic hybrid transmissions,” Dissertation, Purdue University, West Lafayette, USA, 2015.
- [128] C. Pohlandt, *Intelligentes Gesamtmaschinenmanagement für elektrische Antriebssysteme*, ser. Karlsruher Schriftenreihe Fahrzeugsystemtechnik. Karlsruhe, Germany: KIT Scientific Publishing, 2018, vol. 60.
- [129] L. Brinkschulte, K. Daiß, and M. Geimer, “Der nutzen einer fahr- und arbeitszustandserkennung am beispiel eines radladers,” *Land. Technik - Agricultural Engineering - Das Forum für agrartechnische Innovationen, Leinfelden-Echterdingen*, vol. VDI Bericht 2332, pp. 41–50, 2018.
- [130] T. Nilsson, P. Nyberg, C. Sundström, E. Frisk, and M. Krysander, “Robust driving pattern detection and identification with a wheel loader application,” *International Journal of Vehicle Systems Modelling and Testing*, vol. 9, no. 1, pp. 56–76, 2014.
- [131] N. J. Keller, M. Ivantysynova, A. Vacca, Y. X. Sun, Y. F. Zhou, and Lin. G., “Classification of machine functions: A hydraulic excavator case study,” in *16th Scandinavian International Conference on Fluid Power*, K. Huhtala, S. Tikkanen, and J. Uusi-Heikkilä, Eds. Tampere: Tampere University, 2019, pp. 410–422.

- [132] M. Starke and F. Will, “Automatic process pattern recognition for mobile machinery,” in *16th Scandinavian International Conference on Fluid Power*, K. Huhtala, S. Tikkanen, and J. Uusi-Heikkilä, Eds. Tampere: Tampere University, 2019, pp. 450–461.
- [133] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [134] S. Hagen, S. George, and B. Kingsbury, “The IBM attila speech recognition toolkit,” in *Proc. IEEE Spoken Language Technology Workshop*, D. Hakkani-Tür and M. Ostendorf, Eds. Berkeley, USA: IEEE, 2010, pp. 97–102.
- [135] T. N. Sainath, A. R. Mohamed, B. Kingsbury, and B. Ramabhadran, “Deep convolutional neural networks for LVCSR,” in *Proc. IEEE International Conference on Acoustics*, IEEE Signal Processing Society, Ed., Piscataway, NJ, USA, 2013, pp. 8614–8618.
- [136] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [137] S. Hochreiter, “Untersuchungen zu dynamischen neuronalen netzen,” Dissertation, Technische Universität München, München, 1991.
- [138] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [139] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia: IEEE, 2015, pp. 4580–4584.

- [140] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Fifteenth Annual Conference of the International Speech Communication Association*, Singapore, 2014, pp. 1915–1919.
- [141] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," 2014. [Online]. Available: arXiv:1312.6026
- [142] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and Isard, M. and Kudlur, M., "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, USA, 2016, pp. 265–283. [Online]. Available: <https://www.usenix.org/conference/osdi16/program>
- [143] T. Tang, "Working state identification of torque based mobile machines using combined neural networks," Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.
- [144] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.
- [145] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *International Joint Conference on Neural Networks (IJCNN)*. Anchorage, Alaska, USA: IEEE, 2017, pp. 1578–1585.
- [146] M. Lin, Q. Chen, and S. Yan, "Network in network," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014*, 2014. [Online]. Available: arXiv:1312.4400
- [147] A. Villa-Henriksen, G. T. Edwards, L. A. Pesonen, O. Green, and C. A. G. Sørensen, "Internet of things in arable farming: Implementation, applications, challenges and potential," *Biosystems Engineering*, vol. 191, pp. 60–84, 2020.



- [148] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. van Gool, “AI benchmark: Running deep neural networks on android smart-phones,” in *Proc. of the European Conference on Computer Vision (ECCV)*, Glasgow, UK, 2018, pp. 1–27.
- [149] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of Statistical Planning and Inference*, vol. 90(2), pp. 227–244, 2000.
- [150] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. Cambridge, MA, USA: The MIT press, 2009.
- [151] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, “Hybrid heterogeneous transfer learning through deep learning,” in *Twenty-eighth AAAI conference on artificial intelligence*, Québec City, Québec, Canada, 2014, pp. 2213–2220.
- [152] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *arXiv*, 2019. [Online]. Available: arXiv:1911.02685
- [153] C. M. Bishop, *Pattern recognition and machine learning*. New York, USA: Springer, 2006.
- [154] W. Dai, Q. Yang, G. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proc. of the 24th international conference on Machine learning*, Corvalis Oregon, USA, 2007, pp. 193–200.
- [155] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [156] G. Csuska, *Domain Adaptation in Computer Vision Applications*. Cham: Springer International Publishing, 2017.

- [157] A. Rozantsev, M. Salzmann, and P. Fua, “Beyond sharing weights for deep domain adaptation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 801–814, 2019.
- [158] J. Jiang and C. Zhai, “Instance weighting for domain adaptation in NLP,” in *Proc. of the 45th annual meeting of the association of computational linguistics*, Prague, Czech Republic, 2007, pp. 264–271.
- [159] J. Gamboa, “Deep learning for time-series analysis,” *arXiv*, 2017. [Online]. Available: arXiv:1701.01887
- [160] A. Le Guennec, S. Malinowski, and R. Tavenard, “Data augmentation for time series classification using convolutional neural networks,” in *ECM-L/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, Riva Del Garda, Italy, 2016, pp. 1–9.
- [161] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [162] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in neural information processing systems*. Montreal, Quebec, Canada: MIT press, 2014, pp. pp. 3320–3328.
- [163] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Transfer learning for time series classification,” in *Proc. IEEE International Conference on Big Data (Big Data)*. Seattle, WA, USA: IEEE, 2018, pp. 1367–1376.
- [164] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, “To transfer or not to transfer,” in *NIPS 2005 workshop on transfer learning*, vol. 898, Vancouver, British Columbia, Canada, 2005, pp. 1–4.
- [165] M. M. Mahmud and S. Ray, “Transfer learning using Kolmogorov complexity: Basic theory and empirical evaluations,” in *Advances in neural*

- 
- information processing systems*, Vancouver, British Columbia, Canada, 2008, pp. 985–992.
- [166] S. Ben-David and R. Schuller, “Exploiting task relatedness for multiple task learning,” in *Learning Theory and Kernel Machines*. Berlin, Heidelberg: Springer, 2003, pp. 567–580.
- [167] B. Bakker and T. Heskes, “Task clustering and gating for bayesian multitask learning,” *Journal of Machine Learning Research*, vol. 4, pp. 83–89, 2003.
- [168] E. Keogh and C. A. Ratanamahatana, “Exact indexing of dynamic time warping,” *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.
- [169] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.
- [170] Q. Hu, R. Zhang, and Y. Zhou, “Transfer learning for short-term wind speed prediction with deep neural networks,” *Renewable Energy*, vol. 85, pp. 83–95, 2016.
- [171] P. Xiong, Y. Zhu, Z. Sun, Z. Cao, M. Wang, Y. Zheng, J. Hou, T. Huang, and Z. Que, “Application of transfer learning in continuous time series for anomaly detection in commercial aircraft flight data,” in *Proc. IEEE International Conference on Smart Cloud*. York City, NY, USA: IEEE, 2018, pp. 13–18.
- [172] J. Ma, J. C. Cheng, C. Lin, Y. Tan, and J. Zhang, “Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques,” *Atmospheric Environment*, vol. 214, no. 116885, 2019.
- [173] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” in *Advances in Neural Information Processing Systems 2019*, 2019, pp. 5050–5060.

- [174] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.
- [175] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in neural information processing systems*, Montreal, Quebec, Canada, 2015, pp. 3079–3087.
- [176] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of things (IoT): A literature review," *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164–173, 2015.
- [177] B. Nath, F. Reynolds, and R. Want, "RFID technology and applications," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 22–24, 2006.
- [178] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does BTLE measure up against WiFi? a comparison of indoor location performance," in *European Wireless 2014*, Barcelona, Spain, 2014, pp. 1–6.
- [179] C. Mario, P. Giovanni, T. Timothy, and O. K. Tonguz, "Bluetooth 5: A concrete step forward toward the IoT," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 125–131, 2018.
- [180] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, San Diego, CA, USA, 2015, pp. 1–15.
- [181] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, 2018. [Online]. Available: [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
- [182] SAE international, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *SAE International (J3016)*, 2016. [Online]. Available: [https://saemobilus.sae.org/content/j3016\\_201609](https://saemobilus.sae.org/content/j3016_201609)
- [183] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, Montreal, Quebec, Canada, 2015, pp. 91–99.

- 
- [184] X. Zhu, C. Vondrick, D. Ramanan, and C. C. Fowlkes, “Do we need more training data or better models for object detection?” in *British Machine Vision Conference (BMVC)*, vol. 3, no. 5, Surrey, UK, 2012, pp. 1–11. [Online]. Available: <https://www.cs.cmu.edu/~deva/papers/moredata.pdf>
- [185] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “BDD100K: A diverse driving dataset for heterogeneous multitask learning,” *arXiv*, 2018. [Online]. Available: arXiv:1805.04687v2
- [186] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “MOT16: A benchmark for multi-object tracking,” *arXiv*, 2016. [Online]. Available: arXiv:1603.00831v2
- [187] D. Acuna, H. Ling, A. Kar, and S. Fidler, “Efficient interactive annotation of segmentation datasets with polygon-RNN++,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Salt Lake City, Utah, USA, 2018, pp. 859–868.
- [188] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop,” *arXiv*, 2015. [Online]. Available: arXiv:1506.03365v3
- [189] C. G. Northcutt, L. Jiang, and I. L. Chuang, “Confident learning: Estimating uncertainty in dataset labels,” *arXiv*, 2019. [Online]. Available: arXiv:1911.00068v2
- [190] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” *Gray Literature*, 2007. [Online]. Available: <https://authors.library.caltech.edu/7694/1/CNS-TR-2007-001.pdf>
- [191] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, “YouTube-8M: A large-scale video classification benchmark,” *arXiv*, 2016. [Online]. Available: arXiv:1609.08675

- [192] M. Everingham, L. van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [193] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European conference on computer vision*. Zurich, Switzerland: Springer, Cham., 2014, pp. 740–755.
- [194] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, USA: IEEE, 2009, pp. 248–255.
- [195] S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA, 2017, pp. 3213–3221.
- [196] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, 2014, pp. 487–495.
- [197] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 633–641.
- [198] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. of the IEEE conference on computer vision and pattern recognition*, Boston, MA, USA, 2015, pp. 961–970.
- [199] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv*, 2012. [Online]. Available: arXiv:1212.0402

- [200] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," *Workshop on Faces in "Real-Life" Images: Detection, Alignment, and Recognition*, pp. 1–11, 2008.
- [201] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [202] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016, pp. 3213–3223.
- [203] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
- [204] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The ApolloScape dataset for autonomous driving," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Salt Lake City, Utah, USA, 2018, pp. 954–960.
- [205] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nusscenes: A multimodal dataset for autonomous driving," *arXiv*, 2019. [Online]. Available: arXiv:1903.11027
- [206] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. San Diego, California, USA: IEEE, 2005, pp. 886–893.
- [207] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the Seventh IEEE International Conference on Computer Vision*. Corfu, Greece: IEEE, 1999, pp. 1150–1157.

- [208] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [209] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, vol. 25, Lake Tahoe, Nevada, USA, 2012, pp. 1097–1105.
- [210] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv*, 2014. [Online]. Available: arXiv:1409.1556v6
- [211] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016, pp. 1–10.
- [212] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. of the IEEE international conference on computer vision*, Santiago, Chile, 2015, pp. 1026–1034.
- [213] —, “Deep residual learning for image recognition,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, Las Vegas, Nevada, USA, 2016, pp. 770–778.
- [214] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, Las Vegas, Nevada, USA, 2016, pp. 779–788. [Online]. Available: arXiv:1609.08675
- [215] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, USA, 2017, pp. 7263–7271. [Online]. Available: arXiv:1612.08242v1



- 
- [216] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Portland, Oregon, USA, 2013, pp. 580–587.
- [217] R. Girshick, “Fast R-CNN,” in *Proc. of the IEEE international conference on computer vision*, Santiago, Chile, 2015, pp. 1440–1448. [Online]. Available: <https://arxiv.org/abs/1504.08083>
- [218] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [219] Z. Chen, T. Zhang, and C. Ouyang, “End-to-end airplane detection using transfer learning in remote sensing images,” *Remote Sensing*, vol. 10, no. 1, p. 139, 2018.
- [220] Y. Wang, C. Wang, and H. Zhang, “Combining a single shot multibox detector with transfer learning for ship detection using sentinel-1 SAR images,” *Remote Sensing Letters*, vol. 9, no. 8, pp. 780–788, 2018.
- [221] E. R. Azar and B. McCabe, “Part based model and spatial–temporal reasoning to recognize hydraulic excavators in construction images and videos,” *Automation in Construction*, vol. 24, pp. 194–202, 2012.
- [222] S. Chi and C. H. Caldas, “Automated object identification using optical video cameras on construction sites,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 5, pp. 368–380, 2011.
- [223] M. W. Park and I. Brilakis, “Construction worker detection in video frames for initializing vision trackers,” *Automation in Construction*, vol. 28, pp. 15–25, 2012.
- [224] M. Memarzadeh, M. Golparvar-Fard, and J. C. Niebles, “Automated 2D detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors,” *Automation in Construction*, vol. 32, pp. 24–37, 2013.

- [225] H. Tajeen and Z. Zhu, “Image dataset development for measuring construction equipment recognition performance,” *Automation in Construction*, vol. 48, pp. 1–10, 2014.
- [226] W. Fang, L. Ding, B. Zhong, P. E. Love, and H. Luo, “Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach,” *Advanced Engineering Informatics*, vol. 37, pp. 139–149, 2018.
- [227] H. Kim and K. Kim, “Data-driven scene parsing method for recognizing construction site objects in the whole image,” *Automation in Construction*, vol. 71, pp. 271–282, 2016.
- [228] H. Kim, Y. W. Hong, and H. Byun, “Detecting construction equipment using a region-based fully convolutional network and transfer learning,” *Journal of Computing in Civil Engineering*, vol. 32, no. 2, p. 04017082, 2018.
- [229] H. Kim, S. Bang, H. Jeong, and Y. Ham, “Analyzing context and productivity of tunnel earthmoving processes using imaging and simulation,” *Automation in Construction*, vol. 92, pp. 188–198, 2018.
- [230] H. Son, H. Choi, H. Seong, and C. Kim, “Detection of construction workers under varying poses and changing background in image sequences via very deep residual networks,” *Automation in Construction*, vol. 99, pp. 27–38, 2019.
- [231] H. Son, H. Seong, H. Choi, and C. Kim, “Real-time vision-based warning system for prevention of collisions between workers and heavy equipment,” *Journal of Computing in Civil Engineering*, vol. 33, no. 5, p. 04019029, 2019.
- [232] T. Lin, “Labelimg,” Github, 2015. [Online]. Available: <https://github.com/tzutalin/labelImg>

- [233] H. Wang, “Evaluating state-of-the-art object detector on challenging mobile machine dataset,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.
- [234] K. Triantis, S. Sarangi, D. Teodorović, and L. Razzolini, “Traffic congestion mitigation: Combining engineering and economic perspectives,” *Transportation Planning and Technology*, vol. 34, no. 7, pp. 637–645, 2011.
- [235] T. R. Henderson, M. Lacage, and G. F. Riley, “Network simulations with the ns-3 simulator,” *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.
- [236] Z. H. Mir and F. Filali, “LTE and IEEE 802.11p for vehicular networking: a performance evaluation,” *EURASIP Journal on Wireless Communications and Networking*, vol. 89, no. 1, pp. 1–15, 2014.
- [237] D. Jiang and L. Delgrossi, “IEEE 802.11p: Towards an international standard for wireless access in vehicular environments,” in *Proc. VTC Spring 2008-IEEE Vehicular Technology Conference*. Singapore: IEEE, 2008, pp. 2036–2040.
- [238] A. Festag, “Standards for vehicular communication—from IEEE 802.11p to 5G,” *e & i Elektrotechnik und Informationstechnik*, vol. 132, no. 7, pp. 409–416, 2015.
- [239] Z. Wang and H. Mahbub, “How much of DSRC is available for non-safety use?” in *Proc. of the fifth ACM international workshop on Vehicular Inter-NETworking*, V. Sadekar, Ed. New York, NY: ACM, 2008, pp. 23–29.
- [240] S. Chen, J. Hu, Y. Shi, and L. Zhao, “LTE-V: A TD-LTE-Based V2X solution for future vehicular network,” *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 997–1005, 2016.

- [241] C. F. Mecklenbräuker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, “Vehicular channel characterization and its implications for wireless system design and performance,” *Proc. of the IEEE*, vol. 99, no. 7, pp. 1189–1212, 2011.
- [242] K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, “On the ability of the 802.11p MAC method and STDMA to support real-time vehicle-to-vehicle communication,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 347, 2009.
- [243] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, “Vehicle-to-everything (v2x) services supported by LTE-based systems and 5G,” *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.
- [244] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister, and I. Thibault, “3GPP C-V2X and IEEE 802.11p for vehicle-to-vehicle communications in highway platooning scenarios,” *Ad Hoc Networks*, vol. 74, pp. 17–29, 2018.
- [245] R. Molina-Masegosa and J. Gozalvez, “LTE-V for Sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.
- [246] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Waltham, MA, USA: Academic press, 2013.
- [247] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, “LTE-advanced: next-generation wireless broadband technology,” *IEEE Wireless Communications*, vol. 17, no. 3, pp. 10–22, 2010.
- [248] S. Chen and J. Zhao, “The requirements, challenges, and technologies for 5G of terrestrial mobile telecommunication,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 36–43, 2014.
- [249] A. Gupta and R. K. Jha, “A survey of 5G network: Architecture and emerging technologies,” *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

- [250] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogenssen, "From LTE to 5G for connected mobility," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 156–162, 2017.
- [251] M. N. Tahir, K. Maenpaa, and T. Sukuvaara, "Evolving wireless vehicular communication system level comparison and analysis of 802.11p, 4G 5G," in *2nd International Conference on Communication, Computing and Digital systems (C-CODE)*. Islamabad, Pakistan: IEEE, 2019, pp. 48–52.
- [252] A. Vinel, J. Breu, T. H. Luan, and H. Hu, "Emerging technology for 5G-enabled vehicular networks," *IEEE Wireless Communications*, vol. 24, no. 6, p. 12, 2017.
- [253] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. de Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017.
- [254] 3GPP, "3gpp specification series: 38series," 08/29/2020 22:00:45. [Online]. Available: <https://www.3gpp.org/DynaReport/38-series.htm>
- [255] J. Deng, O. Tirkkonen, R. Freij-Hollanti, T. Chen, and N. Nikaein, "Resource allocation and interference management for opportunistic relaying in integrated mmWave/sub-6 GHz 5G networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 94–101, 2017.
- [256] Y. Li, C. Sim, Y. Luo, and G. Yang, "12-port 5G massive MIMO antenna array in Sub-6GHz mobile handset for LTE bands 42/43/46 applications," *IEEE Access*, vol. 6, pp. 344–354, 2018.
- [257] S. Han, C. I. Z. Xu, and C. Rowell, "Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 186–194, 2015.
- [258] J. Campos, "Understanding the 5G NR physical layer," 2017. [Online]. Available: [https://www.3g4g.co.uk/5G/5Gtech\\_video0022\\_01.pdf](https://www.3g4g.co.uk/5G/5Gtech_video0022_01.pdf)

- [259] E. Mohyeldin, "Minimum technical performance requirements for IMT-2020 radio interface(s)," in *ITU-R Workshop on IMT-2020 Terrestrial Radio Interfaces*, Munich, Germany, 2016, pp. 1–12.
- [260] R. I. Ansari, C. Chrysostomou, S. A. Hassan, M. Guizani, S. Mumtaz, J. Rodriguez, and Rodrigues, J. J. P. C., "5G D2D networks: Techniques, challenges, and future prospects," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3970–3984, 2018.
- [261] I. Siomina, P. Varbrand, and D. Yuan, "Automated optimization of service coverage and base station antenna configuration in UMTS networks," *IEEE Wireless Communications*, vol. 13, no. 6, pp. 16–25, 2006.
- [262] A. Bazzi, B. M. Masini, A. Zanella, and I. Thibault, "On the performance of IEEE 802.11p and LTE-V2V for the cooperative awareness of connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 419–10 432, 2017.
- [263] W. Alasmay and W. Zhuang, "Mobility impact in IEEE 802.11p infrastructureless vehicular networks," *Ad Hoc Networks*, vol. 10, no. 2, pp. 222–230, 2012.
- [264] W. Viriyasitavat, M. Boban, H. Tsai, and A. Vasilakos, "Vehicular communications: Survey and challenges of channel and propagation models," *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 55–66, 2015.
- [265] M. Stoffers and G. Riley, "Comparing the ns-3 propagation models," in *Proc. IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Washington, DC, USA: IEEE, 2012, pp. 61–67.
- [266] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE Journal on selected areas in communications*, vol. 17, no. 7, pp. 1205–1211, 1999.

- [267] A. Vinel, Y. Koucheryavy, S. Andreev, and D. Staehle, “Estimation of a successful beacon reception probability in vehicular ad-hoc networks,” in *Proc. of the 2009 International Conference on Wireless Communications and Mobile Computing Connecting the World Wirelessly - IWCMC 09*, M. Guizani, P. Mueller, K. P. Fährnich, A. Vasilakos, Y. Zhang, and J. Zhang, Eds. New York, NY, USA: ACM Press, 2009, p. 416.
- [268] L. M. Lopez, C. F. Mendoza, J. Casademont, and D. Camps-Mur, “Understanding the impact of the pc5 resource grid design on the capacity and efficiency of LTE-V2X in vehicular networks,” *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–13, 2020.
- [269] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, “IEEE 802.11e wireless LAN for quality of service,” in *Proc. European Wireless*, vol. 2, 2002, pp. 32–39.
- [270] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, England: Cambridge University Press, 2004.
- [271] Y. Xiang, Y. Huang, Z. Zhang, T. Su, C. Brach, S. Mao, and M. Geimer, “KIT MOMA v2: towards instance segmentation of construction machines,” 2020, unpublished.
- [272] W. Roh, J. Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, “Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 106–113, 2014.
- [273] H. Yoshida, T. Yoshimoto, T. Innami, K. Ohashi, H. Furuya, and N. Mori, “Improving efficiency of remote construction by using adaptive video streaming,” in *Proc. the 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, USA, 2019, pp. 1–2.
- [274] S. Dadhich, U. Bodin, and U. Andersson, “Key challenges in automation of earth-moving machines,” *Automation in Construction*, vol. 68, pp. 212–222, 2016.

- [275] I. Johansson, S. Dadhich, U. Bodin, and T. Jönsson, “Adaptive video with SCReAM over LTE for remote-operated working machines,” *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–10, 2018.
- [276] S. Dadhich, U. Bodin, F. Sandin, and U. Andersson, “From tele-remote operation to semi-automated wheel-loader,” *International Journal of Electrical and Electronic Engineering and Telecommunications*, vol. 7, no. 4, pp. 178–182, 2018.
- [277] H. F. Bermudez, R. Sanchez-Iborra, and J. L. Arciniegas, “Performance validation of ns3-LTE emulation for live video streaming under QoS parameters,” in *Proc. IEEE 13th International Conference on Wireless and Mobile Computing Networking and Communications (WiMob)*, Rome, Italy, 2017, pp. 300–307. [Online]. Available: <http://ieeexplore.ieee.org/servlet/opac?punumber=8106925>
- [278] L. F. Perrone, C. Cicconetti, G. Stea, and B. C. Ward, “On the automation of computer network simulators,” in *Proc. of the 2nd International Conference on Simulation Tools and Techniques*, Rome, Italy, 2009, pp. 1–10.
- [279] ns3, “LTE module — model library,” 08/24/2020 14:43:54. [Online]. Available: <https://www.nsnam.org/docs/models/html/lte.html>
- [280] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, “End-to-end simulation of 5G mmwave networks,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2237–2263, 2018.
- [281] 3GPP, “3gpp specification series: 36series,” 08/29/2020 22:01:22. [Online]. Available: <https://www.3gpp.org/DynaReport/36-series.htm>
- [282] M. Assyadzily, A. Suhartomo, and A. Silitonga, “Evaluation of X2-handover performance based on RSRP measurement with Friis path loss using network simulator version 3 (ns-3),” in *2nd International Conference on Information and Communication Technology (ICoICT)*. Bandung, Indonesia: IEEE, 2014, pp. 436–441.



- 
- [283] H. T. Friis, "A note on a simple transmission formula," in *Proc. of the IRE*, vol. 34(5), 1946, pp. 254–256.
- [284] G. R. MacCartney, J. Zhang, S. Nie, and T. S. Rappaport, "Path loss models for 5G millimeter wave propagation channels in urban microcells," in *Proc. IEEE Global Communications Conference (GLOBECOM)*. Atlanta, GA, USA: IEEE, 2013, pp. 3948–3953.
- [285] ns3, "ns-3: ns-3 documentation," 08/29/2020 22:25:39. [Online]. Available: <https://www.nsnam.org/doxygen/>
- [286] 3GPP, "Study on channel model for frequency spectrum above 6 GHz." [Online]. Available: [https://www.etsi.org/deliver/etsi\\_tr/138900\\_138999/138900/14.02.00\\_60/tr\\_138900v140200p.pdf](https://www.etsi.org/deliver/etsi_tr/138900_138999/138900/14.02.00_60/tr_138900v140200p.pdf)
- [287] J. Wiebelitz, C. Kunz, S. Piger, and C. Grimm, "TCP-AuthN: TCP inline authentication to enhance network security in grid environments," in *Proc. IEEE the Eighth International Symposium on Parallel and Distributed Computing*, Lisbon, Portugal, 2009, pp. 237–240.
- [288] D. Madhuri and P. C. Reddy, "Performance comparison of TCP, UDP and SCTP in a wired network," in *International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2016, pp. 1–6.
- [289] M. Sinky, A. Dhamodaran, B. Lee, and J. Zhao, "Analysis of H.264 bitstream prioritization for dual TCP/UDP streaming of HD video over WLANs," in *Proc. 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, USA, 2015, pp. 576–581.
- [290] M. Kim, S. Kim, and Y. Lim, "An implementation of downlink asynchronous HARQ for LTE TDD system," in *Proc. IEEE Radio and Wireless Symposium*. Carolina, USA: IEEE, 2012, pp. 271–274.
- [291] A. Anand and G. de Veciana, "Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2411–2421, 2018.

- [292] J. Yeo, S. Park, J. Oh, Y. Kim, and J. Lee, "Partial retransmission scheme for HARQ enhancement in 5G wireless communications," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Singapore, 2017, pp. 1–5.
- [293] C. Summerson, "How much data does Netflix use?" *How-To Geek*, 2018/1/15. [Online]. Available: <https://www.howtogeek.com/338983/how-much-data-does-netflix-use/>
- [294] K. Bode, "Netflix starts offering 'super HD' and 3D streams - but only through ISPs that use their CDN," 2020. [Online]. Available: <http://www.dslreports.com/shownews/Netflix-Starts-Offering-Super-HD-and-3D-Streams-122670>
- [295] C. Seker, M. T. Güneser, and T. Ozturk, "A review of millimeter wave communication for 5G," in *2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Kızılcahamam, Ankara, Turkey, 2018, pp. 1–5.
- [296] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.
- [297] B. Peng and T. Kuerner, "Three dimensional angle of arrival estimation in dynamic indoor terahertz channels using forward-backward algorithm," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3791–3811, 2016.
- [298] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

# List of Publications

## Journal Articles

**Xiang, Y.**, K. Liu, T. Su, J. Li, S. Ouyang, S. Mao, and M. Geimer, “An extension of BIM using AI: A multi working-machines pathfinding solution,” *IEEE Access*, vol. 9, pp. 124 583–124 599, 2021.

**Xiang, Y.**, T. Tang, T. Su, C. Brach, L. Liu, S. S. Mao, and M. Geimer, “Fast CRDNN: Towards on site training of mobile construction machines,” *IEEE Access*, vol. 9, pp. 124 253–124 267, 2021.

**Xiang, Y.**, D. Li, T. Su, Q. Zhou, C. Brach, S. S. Mao, and M. Geimer, “Where am I? SLAM for mobile machines on a smart working site,” *arXiv*, pp. 1–14, 2020. [Online]. Available: [arXiv:2011.01830](https://arxiv.org/abs/2011.01830)

**Xiang, Y.**, H. Wang, T. Su, R. Li, C. Brach, S. S. Mao, and M. Geimer, “KIT MOMA: A mobile machines dataset,” *arXiv*, 2020. [Online]. Available: [arXiv:2007.04198](https://arxiv.org/abs/2007.04198)

J. Cai, J. Zhao, **Xiang, Y.**, J. Liu, G. Chen, Y. Hu, and J. Chen, “Can I trust you? Estimation models for e-bikers stop-go decision before amber light at urban intersection,” *Journal of advanced transportation*, vol. 2020, no. 6678996, pp. 1–17, 2020.

Y. Liu, J. Qiao, Y. Hu, T. Fang, T. Xu, **Xiang, Y.**, and Y. Han, “Determination of curve speed zones for mountainous freeways,” *Mathematical Problems in Engineering*, vol. 2020, no. 8844004, pp. 1–11, 2020.

J. Li, W. Hong, and **Xiang, Y.**, “A short review on data modelling for vector fields,” *arXiv*, 2020. [Online]. Available: arXiv:2009.00577

## Conference Contributions

**Xiang, Y.**, T. Su, C. Brach, X. Liu, and M. Geimer, “Realtime estimation of ieeec802.11p for mobile working machines communication respecting delay and packet loss,” in *Proc. IEEE Intelligent Vehicle Symposium*, Las Vegas, USA, 2020, pp. 1516–1521.

**Xiang, Y.**, R. Li, C. Brach, X. Liu, and M. Geimer, “A novel algorithm for hydrostatic-mechanical mobile machines with a dual-clutch transmission,” in *Proc. 11th IEEE Global Fluid Power Society PhD Symposium*, Guilin, China, 2020, pp. 1–8.

**Xiang, Y.**, S. Wang, T. Su, J. Li, S. S. Mao, and M. Geimer, “KIT bus: A shuttle model for CARLA simulator,” in *Proc. IEEE Industrial Electronics and Applications Conference (IEACon)*, Virtual, 2021, pp. 1–6.

**Xiang, Y.**, B. Xu, T. Su, C. Brach, S. S. Mao, and M. Geimer, “5G meets construction machines: Towards a smart working site,” in *Proc. IEEE International Conference on Computing and Communications Applications and Technologies (I3CAT)*, Virtual, 2021, pp. 1–8.

**Xiang, Y.** and M. Geimer, “Optimization of operation strategy for primary torque based hydrostatic drivetrain using artificial intelligence,” in *Proc. 12th International Fluid Power Conference*, Dresden, Germany, 2020, pp. 55–65.

**Xiang, Y.**, S. Mutschler, N. Brix, C. Brach, and M. Geimer, “Optimization of hydrostatic-mechanical transmission control strategy by means of torque control,” in *Proc. 12th International Fluid Power Conference*, Dresden, Germany, 2020, pp. 421–431.

L. Liu, B. Li, G. Götting, **Xiang, Y.**, Q. Salem, M. Hamid, and J. Xie, “Loss minimization of traction systems in battery electric vehicles using variable dc-link

voltage technique—experimental study,” in *Proc. 22nd European Conference on Power Electronics and Applications (EPE'20 ECCE Europe)*, Lyon, France, 2020, pp. P1–P8.

S. Mutschler, N. Brix, and **Xiang, Y.**, “Torque control for mobile machines,” in *Proc. 11th International Fluid Power Conference*, H. Murrenhoff, Ed. Aachen, Germany: RWTH Publications, 2018, pp. 186–195.

J. Yang, F. Lin, **Xiang, Y.**, P. Katranuschkov, and R. Scherer, “Fast crack detection using convolutional neural network,” in *Proc. 28th International Workshop on Intelligent Computing in Engineering*, Berlin, Germany, 2021, pp. 540–549.

## Supervised Theses

I did accept **all** the applications if the students were at their final phase of the study. The selections **were not** based on their previous grades, gender, and nationality. All the theses which were done at KIT were under the supervision of **Yusheng Xiang** and **Marcus Geimer**. In addition, the thesis from Mr. Julian Kreis were supervised by **Yusheng Xiang** and **Klaus Allmendinger**. The order is based on the date of our on-site interview. I have only listed the theses for which I am responsible for the quality of the theses.

**Li, Ruoyu**, “Optimization of control strategy for dual clutch transmission on torque based mobile machines,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2019.

**Julian, Kreis**, “Optimierung eines schaltalgorithmus für ein doppelkupplungsgetriebe in einer mobilhydraulikapplikation, (english title: Optimization of a shift algorithm for a dual clutch transmission in a mobile hydraulic application),” Bachelor thesis, Department of Mechanical Engineering, Technische Hochschule Ulm, Germany, 2020.

**Wang, Hongzhe**, “Evaluating state-of-the-art object detector on challenging mobile machine dataset,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Tang, Tian**, “Working state identification of torque based mobile machines using combined neural networks,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Wang, Long**, “Use-case-basierte Lastkollektive Entwicklung für Mildhybridanwendung auf 48V Basis (English title: Use-case-based load collective development for a mild hybrid 48V application),” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Su, Jiamei**, “Evaluation of SSD object detector on challenging mobile machines dataset,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Li, Dianzhao**, “Where am I? SLAM for mobile machines on a smart working site,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Sun, Bosheng**, “An analysis of the sensors arrangement for driverless city buses,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Mei, Tao**, “Arrangement of the sensors for driverless city buses in challenging road scenarios,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Cui, Qiangguo**, “Estimation of working machines mass and road grade with recursive least square method,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Xu, Bing**, “5G meets construction machinery: Towards a smart working site,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Xiong, Xinyi**, “A mobile machine with legged system,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Fu, Meiqi**, “Driving speed of torque based mobile machines by means of MPC controller,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2020.

**Zhao, Hengping**, “An approach for multi-construction machinery pathfinding in construction site,” Bachelor thesis, Karlsruhe Institute of Technology and Beijing Institute of Technology, Germany and China, 2020.

**Zhao, Yang**, “Performance evaluation of IEEE 802.11p for mobile working machines communication respecting delay and packet loss,” Bachelor thesis, Karlsruhe Institute of Technology and Beijing Institute of Technology, Germany and China, 2020.

**Chen, Yucheng**, “The advantages of application of CBS in construction sites,” Bachelor thesis, Karlsruhe Institute of Technology and Beijing Institute of Technology, Germany and China, 2020.

**Gao, Xiaochen**, “Design and control of a walking mechanism of an excavator in complex scenarios,” Bachelor thesis, Karlsruhe Institute of Technology and Beijing Institute of Technology, Germany and China, 2020.

**Zhang, Zhenduo**, “Instance segmentation of engineering plants in real construction environment based on Mask-RCNN,” Bachelor thesis, Karlsruhe Institute of Technology and Beijing Institute of Technology, Germany and China, 2020.

**Huang, Yanliang**, “Object recognition on the construction site based on Mask R-CNN,” Bachelor thesis, Karlsruhe Institute of Technology and Beijing Institute of Technology, Germany and China, 2020.

**Hu, Qiankun**, “KIT MOMA V2: Towards instance segmentation of construction machines,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2021.

**Wang, Shuo**, “A passenger bus model for CARLA simulator and implement simulation,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2021.

**Niu, Zhuo**, “Research on a rapid recognition method of construction machinery based on the improved YOLO-V4 algorithm,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2021.

**Liu, Kailun**, “An Extension of BIM Using AI: A Multi Working-Machines Pathfinding Solution,” Master thesis, Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Germany, 2021.