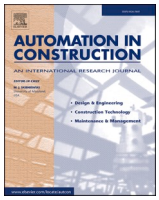
Automation in Construction 166 (2024) 105592

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Computer vision-based excavator bucket fill estimation using depth map and faster R-CNN

Bobo Helian [a,b,*], Xiaoqian Huang [a], Meng Yang [c,d,**], Yongming Bian [c,d], Marcus Geimer [a]

[a] Institute of Mobile Machines, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
[b] State Key Laboratory of Fluid Power and Mechatronic System, Zhejiang University, 310027 Hangzhou, China
[c] School of Mechanical Engineering, Tongji University, 201804 Shanghai, China
[d] Shanghai Engineering Research Center for Safety Intelligent Control of Building Machinery, 200032 Shanghai, China

ARTICLE INFO

Keywords:
Excavator automation
Bucket volume estimation
Computer vision
Depth map
Faster R-CNN
Custom loss function

ABSTRACT

Excavators are crucial in the construction industry, and developing autonomous excavator systems is vital for enhancing productivity and reducing the reliance on manual labor. Accurate estimation of the volume of the excavator bucket fill is key for monitoring and evaluating system automation performance. This paper presents the use of 2D depth maps as input to a Faster Region Convolutional Neural Network (Faster R-CNN) deep learning model for bucket volume estimation. This structure enables high estimation accuracy while maintaining fast processing speed. An excavator operation monitoring test bench was established, and the datasets used in the study were self-generated for training. A loss function is proposed, combining Cross Entropy with Root Mean Squared Error to improve generalization and precision. Comparative results indicate that the proposed approach achieves 96.91% accuracy in fill factor estimation and predicts in real-time at about 10 fps, highlighting its potential for practical use in automated excavator operations.

1. Introduction

Excavators are widely applied in construction and mining industries [1], and they are commonly operated in harsh and challenging environments that require experienced manual labor. Thereby, the automation of excavators has practical significance in enhancing operation safety and increasing productivity while reducing the need for highly skilled operators [2].

The payload volume filled in the bucket is a significant factor indicating the productivity of earth-moving machines [3]. Real-time estimation of bucket filling volume is one of the key tasks of automatic construction machines [4], which can contribute to monitoring and improving productivity by observing the effective volume of material dug and moved by buckets. Meanwhile, it can also prevent the construction machines (e.g. excavators) from potential hazards due to contact with uncertain working conditions. To achieve a reliable bucket fill estimation system, high estimation accuracy and efficient high real-time estimation performance are mandatory.

1.1. Related works

Numerous studies have been conducted on the estimation of material fill in earth-moving industries [5]. In the tasks of material fill estimation, weight-based estimation is often utilized in industrial settings, this method estimates fill by weighing containers. Dadhich and Bodin proposed a method in which the fill factor can be quantified using a weighing scale system integrated within the machine during the lifting of the bucket. A weighing scale system uses the pressure in the cylinders to calculate the loaded weight [6]. This approach allows the direct measurement of the load, however, in practical scenarios, the fill level of the excavator bucket varies with multiple factors such as particle density, and moisture content [7]. In contrast, measuring the volume of the excavator bucket directly reflects its fill status without the need to consider the density of the material particles. Considering that in dynamic systems, volume is difficult to obtain directly through measurements, computer vision-based volume measurement is an effective means of obtaining space information. In the task of volume estimation, computer vision-based measurement of volumes would commonly be deployed in scenarios in which 3D information is required [8]. Usually, a

* Correspondence to: B. Helian, Institute of Mobile Machines, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany.
** Correspondence to: M. Yang, School of Mechanical Engineering, Tongji University, 201804 Shanghai, China.
E-mail addresses: bobo.helian@kit.edu (B. Helian), yangmeng@tongji.edu.cn (M. Yang).

https://doi.org/10.1016/j.autcon.2024.105592
Received 6 December 2023; Received in revised form 25 June 2024; Accepted 26 June 2024
Available online 4 July 2024

depth sensor can be utilized to obtain 3D spatial information [9]. Commonly used are stereo cameras, LiDAR, or sonar. These sensors provide spatial information, typically in the form of 3D point clouds or 2D depth maps.

1.1.1. Non-learning-based 3D object volume estimation

Non-learning methods can achieve high accuracy in volume estimation tasks when reliable three-dimensional spatial information is available as input. Volume estimation tasks benefit from accurate measurements of objects' shape and size. Non-learning methods do not require the construction of datasets and training networks, but obtain the volume of 3D objects by processing 3D information such as point clouds [3]. For example, in scenarios where high-quality point cloud data is available, projection-based techniques, as detailed in [10], stand out as the preferred choice for 3D object detection tasks. Chang and Wu presented a method for object volume estimation using a 3D point cloud. This method incorporates slicing coupled with the least squares approach. Specifically, it applies least squares curve fitting to determine the contour of each slice [11]. Guevara presented a method using point clouds to represent the volume of the skid steer. The volume is estimated with point cloud data by capturing the bucket state and then matching it with a pre-built model of the empty bucket [12]. J. Lu introduced a 3D point cloud-based method coupled with a position sensor to estimate the fill volume of wheel loaders [13]. Although achieving high precision, it had low processing speeds of up to 2 s, which limited the real-time monitoring performance. In addition, the processing of 3D point clouds requires significant computational effort [14], which could limit its application in industries.

1.1.2. Learning-based volume estimation

A load-weight-based bucket fill study by S. Dadhich implemented a time-delayed neural network (TDNN) for estimating the fill factor in a loader's bucket filled with medium coarse gravel [5], utilizing pressures in the lift and tilt hydraulic cylinders as input features. While this approach achieves efficient fill estimation for loaders, its generalization ability may be limited when applied to excavators, which typically handle more diverse excavation materials.

Computer vision-based methods can be effective solutions for bucket fill estimation. Estimating the volume of objects from 2D images is a challenging task due to the loss of depth information. However, with the advancement of computer vision and deep learning techniques, it has become feasible to approximate the volume of objects using single or multiple 2D images [15]. Monocular vision-based 2D object identification (produced by RGB cameras) has been employed to enhance numerous automated monitoring systems, benefiting both safety and productivity [16]. Since there is no depth information available, traditional computer vision methods can only be applied with certain restrictions. Prasad introduced a method for single-view reconstruction to model smooth shapes from their apparent contours [17]. Choy and colleagues introduced a comprehensive method for 3D reconstruction using 3D convolutional networks and LSTM, which produces a 3D voxel-based representation of the corresponding objects [18].

Furthermore, binocular images generated from stereo vision that contain depth information are simply called depth maps. Representing 3D objects using depth maps involves the process of capturing the three-dimensional structure of objects or environments based on depth information [19,20]. In contrast to traditional RGB images, depth maps contain information about the distance of objects from the camera, allowing a more direct estimation of 3D structures. A typical explored area is human pose estimation using neural networks [21], where 2D images serve as input and the output consists of the three-dimensional orientations of human limbs [22].

In the field of deep-learning applications for estimating bucket fill of construction equipment, Alam [23] presented a deep learning-based approach for estimating earth volume in large-scale engineering projects. The models were trained and evaluated using RGB images of trucks loaded with varying amounts of fill. However, the evaluation did not include tests with data outside the training set, and the training data was collected exclusively from a toy truck, leaving the generation ability of the model questionable. Lu introduced a learning-based method to estimate the fill factor of the machine [24], in which a faster region convolutional neural network (Faster R-CNN) [25] is used for feature extraction and classification. The final value for the bucket fill factor is obtained through classification and probabilistic-based methods in the post-processing stage, in which large data sets are required during training for higher accuracy. In the study by W. Guan [4], Mask R-CNN was employed for estimating the fill factor of a wheel loader, achieving high accuracy. Despite constructing a large dataset of over 75,000 images, the test set required more volume division categories. This issue typically arises when fill factor estimation is treated as a classification task.

In summary, volume prediction based on 2D images can significantly reduce the computation time and achieve fast processing speed, which is crucial for real-time dynamic estimation in industrial applications. As 2D images, the depth map contains 3D information and is a suitable input for the neural network. However, accurate volume estimation requires a large amount of training data to ensure the estimation performance, which is a typical challenge for learning-based computer vision techniques, especially in the presence of data outside the training set (i.e., out-of-distribution data). Therefore, it is significant to improve the generation capability of deep learning-based methods.

1.1.3. Deep learning methods in autonomous detection

In the field of using deep learning-based approaches in autonomous target detection, recent publications [26] emphasize the significance of achieving high Mean Average Precision (mAP) values and real-time processing performance (i.e., frames per second, FPS). Among the most popular target detection models, YOLO (You Only Look Once) is widely used in autonomous driving and engineering machinery due to its efficient real-time processing, although it may sacrifice accuracy, especially for small objects [27,28].

In contrast, models with a two-stage architecture, such as the above-introduced Faster R-CNN, often achieve higher accuracy and better localization, making them effective in scenarios with challenging small objects or instances that require precise localization [29]. In addition, in previous research [4], ResNet is often chosen as the backbone of the learning framework, facilitating efficient feature extraction from input images which are subsequently used by the Faster R-CNN framework. Furthermore, a study by T. Mahendrakar compared Faster R-CNN and YOLOv5 and showed that Faster R-CNN performed better in terms of accuracy and YOLOv5 had faster inference rates [30].

In the task of this study, the requirement for high real-time processing speed for volume estimation is fulfilled by using depth map, so accuracy is a higher priority when selecting the deep learning network. Therefore, Faster R-CNN proves to be a suitable choice as it provides acceptable real-time performance while ensuring high accuracy.

1.2. Contributions

This study proposes a solution for estimating the volume of excavator bucket fill. It combines the Faster R-CNN deep learning architecture with depth maps as input. This framework is specifically designed for classification-regression tasks, enabling the neural network model to achieve accurate prediction for data not included in the training set. Meanwhile, it reduces computational complexity for higher real-time system performance.

1. To improve the processing speed of fill level estimation, 2D depth maps are used as the input for the neural network. A Faster-RCNN network with ResNet50 as its backbone is implemented, facilitating the detection of the excavator bucket and estimation of its fill factor. Compared to traditional methods of processing 3D point clouds, this

approach significantly enhances efficiency while maintaining a high level of estimation accuracy.

2. A computer vision recording platform integrated with an excavator test bench has been developed, enabling the collection of 1562 images using a stereo camera. This dataset captures a wide range of fill degrees and various excavator poses, closely replicating real-world operating conditions. Furthermore, a comprehensive dataset annotation and distribution method has been established to facilitate effective training and testing of the model.
3. Typically, deep learning-based volume fill factor estimation approaches depend heavily on large training datasets to achieve high accuracy and generalization, often resulting in prolonged convergence times and weakness with data outside the training dataset.

   To address this issue, this study proposes a loss function that integrates cross-entropy with Mean Squared Error, treating the bucket fill estimation as a combined classification-regression task. This approach prioritizes classification to categorize fill levels, and then it refines the estimation through regression within the identified classes, resulting in high accuracy and efficiency, particularly for out-of-distribution data.

Below is a detailed breakdown of the methodology into subsections, accompanied by a workflow diagram as presented by Fig. 1. (1) Depth Map Acquisition and Dataset: Experiments were conducted using an excavator test bench setup, and depth maps were acquired using stereo cameras. The dataset was organized into training and testing sets for model development and evaluation. (2) Methodology: A Faster R-CNN architecture was employed as the backbone of the neural network model for excavator bucket fill estimation, with a custom loss function integrating Cross Entropy and Mean Squared Error to guide the training process effectively. (3) Comparative Experiments and Analysis: The performance of the model was evaluated during the training phase, assessing metrics such as loss function convergence and accuracy. The trained model was then tested on unseen data to assess its generalization capability and accuracy in estimating excavator bucket fill levels.

2. Depth map acquisition and dataset

2.1. Depth map acquisition

By employing 2D depth maps as input to the neural network, the processing time of the estimation can be shortened, thereby ensuring real-time estimating performance. To obtain a depth map, an excavation scene, including the fill level of the bucket, is continuously captured by two laser sensors, as shown in Fig. 2. Depth maps are generated using

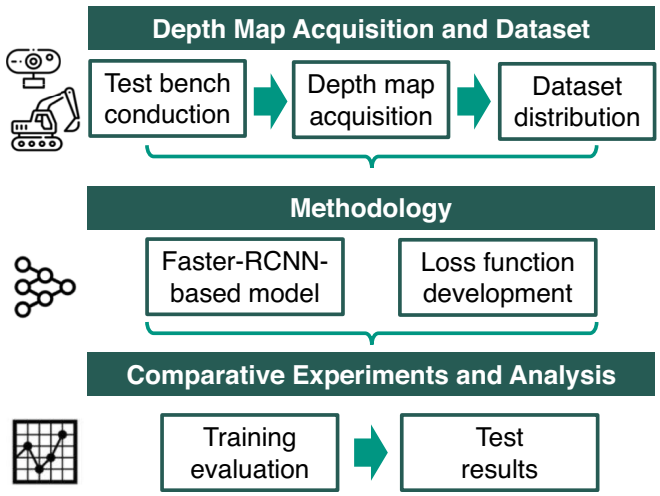


Fig. 1. Depth image with various filling factors.

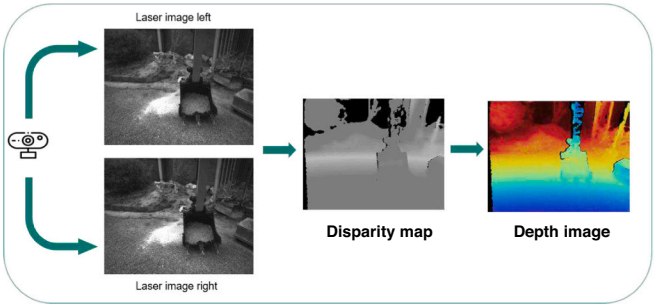


Fig. 2. Depth map acquisition with a stereo camera.

stereo vision techniques. By analyzing the disparity of the objects' projections onto each sensor, the distance to these objects can be calculated, as shown in [31]. Finally, the bucket-fill factor can be estimated by post-processing the neural network output, which will be introduced in the next section.

To prepare the dataset for training the volume estimation model, it's imperative to capture a variety of fill levels. To facilitate data collection, a custom data collection platform is implemented as shown in Fig. 3. The excavator used for this task is a Sany SY16C with a standard bucket capacity of 0.04 m^3. Sand was chosen as the fill material because it tends to minimize volume discrepancies due to compression during transportation. To ensure standardized ground truth for each loading operation, a fixed-volume container is used, as shown in the figure, with dimensions of 34.5 cm × 24.5 cm × 17.5 cm, allowing for accurate measurement of the volume of the fill material. In addition, Fig. 4 shows a brief overview of the dataset annotation, including the depth map images aligning with fill factors. A Realsense d435i stereo camera is used to capture depth maps alongside RGB data. This camera is connected to a PC positioned in front of the excavator cabin and securely mounted on a tripod. This setup provides an advantageous vantage point for monitoring bucket orientation and fill factor, closely matching the operator's perspective.

The data recording process includes the following steps: 1) Fill factor selection: Select the desired fill factor and load the corresponding material into the bucket after confirming it with the ground truth; 2) Complete operation cycle: Perform a complete operation cycle for each fill factor, including rotation, sweeping, and dumping to simulate real excavator operations; 3) Operation recording: Record the excavation process with the depth camera, and store the data on the connected PC for future analysis and model training; 4) Data augmentation: For the training dataset, the following augmentation was applied to create 3 versions of each source image:

- Random brightness adjustment of between −10% to +10%
- Random exposure adjustment of between −25% to +25%
- Salt and pepper noise was applied to 2% of the pixels

Fig. 3. Data acquisition platform.

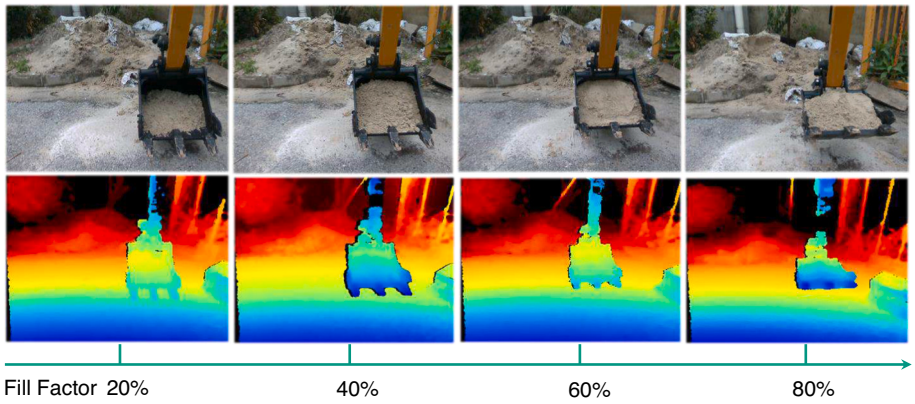


Fig. 4. Depth map annotation with bucket fill factor.

2.2. Dataset annotation and distribution

Following the data acquisition phase, the next steps involve annotating and distributing the datasets.

To begin with, it is important to determine the specific labels or annotations required for the dataset. Images are categorized based on the bucket fill factor, with each fill level also including different bucket positions under different working poses.

To improve the generalization ability of the network, the fill factors are distributed over a specific interval of 5% between each of the two closed classes. As shown in Fig. 5, the annotation for the training data within the 5% interval starts from 15% to 80%, such as 15% and 25%, and is used exclusively for training (and validation during training). Meanwhile, the data with the annotation for testing starts from 17.5% to 77.5% also with 5% interval, such as 22.5% and 27.5%. This distribution strategy ensures that the network can perform accurate predictions for fill factors that are not explicitly covered during the training phase. Each class in the test set, which includes 13 classes in total, consists of 20 images.

In total, through the above steps, datasets of 1562 images are built and distributed as follows: a training dataset of 1251 depth images, a validation dataset of 51 depth images (used in the training phase), and an out-of-distribution (OOD) test dataset of 260 depth images. The images were processed at a resolution of 640 × 480 pixels, which is a standard compressed size that reduces the computational complexity in real-time processing without compromising significant accuracy.

Deep learning models are susceptible to overfitting, which is a common challenge due to dataset limitations. To address this issue, as explained above, the dataset was carefully designed to avoid excessive homogeneity. Specifically, the dataset is small but evenly distributed across different bucket fill levels, which helps prevent the model from becoming overly specialized in any one class. Furthermore, an effective way to verify whether overfitting occurs is to test the model on comprehensive OOD data. This step is critical because it demonstrates the model's ability to generalize well to new, unseen data, confirming that it is not simply memorizing the training set, but rather learning the underlying patterns necessary to make accurate predictions in diverse scenarios. These are the concerns behind the dataset annotation and distribution detailed above.

3. Method

3.1. Faster-RCNN-based neural network

3.1.1. Neural network framework

This study proposes a customized Faster-RCNN network architecture for the computer vision-based bucket-fill estimation. Furthermore, a loss function is designed that integrates Cross Entropy with Mean Squared Error to improve the accuracy and versatility of the neural network in handling classification-regression tasks.

The Faster R-CNN is an advanced object detection algorithm that builds on the foundations of previous architectures such as R-CNN and Fast R-CNN [32]. Its primary innovation is the integration of the Region Proposal Network (RPN), a component that shares full-image convolutional features with the detection network, effectively eliminating the computational overhead associated with region proposal generation. The RPN is trained end-to-end to produce high-quality region proposals, which are then used by the Fast R-CNN for the final detection step. As a result of this design, the Faster R-CNN excels at object detection, achieving a remarkable combination of accuracy and speed.

In this section, a customized network architecture is proposed to meet the specific requirements of the real-time estimation task, as shown

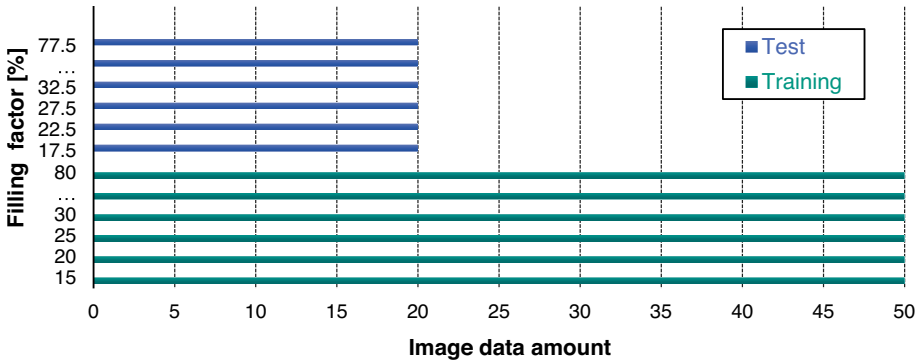


Fig. 5. Dataset distribution.

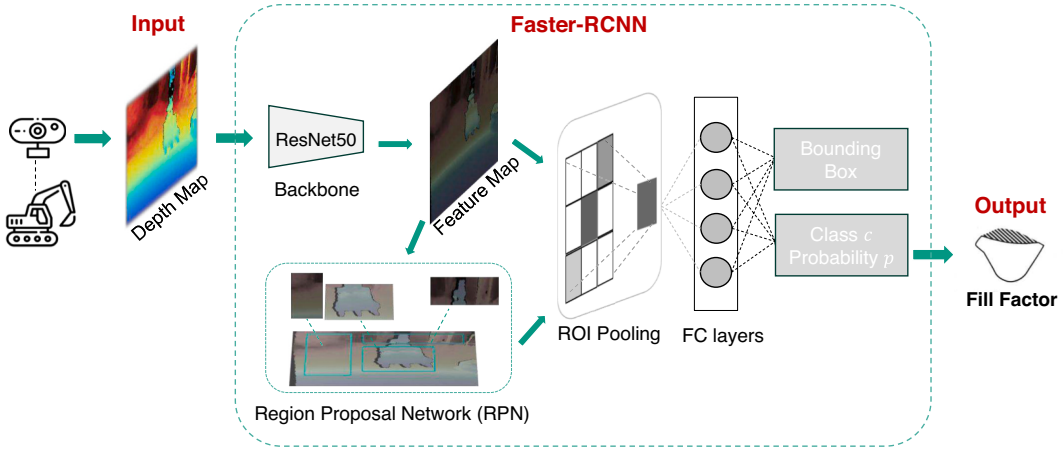


Fig. 6. Faster-RCNN-based fill volume estimation architecture.

in Fig. 6. The modified network architecture is primarily based on the Faster R-CNN framework. Depth maps are used as the primary input data. These images undergo feature extraction using the ResNet-50 backbone [33], a deep convolutional neural network known for its ability to capture intricate patterns and fine details. The ResNet50 backbone is responsible for extracting high-level features from the input images, and these features are then used by the Faster R-CNN framework for further fill volume estimation. There are various options for backbone selection, such as VGG16 and ResNet101. The reason why ResNet50 is selected is that it strikes a balance between model depth and computational efficiency.

In addition, the extracted features are used by the Region Proposal Network (RPN), which collaborates with the Region of Interest (RoI) pooling layer, ensuring that both components leverage shared features to improve estimation performance. Following RoI pooling, the network uses a fully connected (FC) layer to derive the classification results. The outputs include bounding box coordinates and predicted object classes (i.e., real-time fill factor), and their associated probability scores, providing a comprehensive solution to the bucket-fill estimation task.

3.1.2. *Loss function design*

As mentioned above, the primary outputs include the bounding box coordinates and the predicted object classes, especially regarding the real-time fill factor.

In the context of classical classification tasks, the typical output, as shown in Table 1, typically consists of one or more classes, each accompanied by an associated probability score. This is facilitated by the use of specialized activation functions tailored to the needs of classification, such as sigmoid activation and softmax, which ensure that the resulting probabilities sum to one and remain in the [0,1] range. However, a significant challenge arises when confronted with the fill factor, a continuous numerical value that inherently transforms the classification task into a regression. While a straightforward approach might involve linear regression with a linear activation function, the unique characteristics of our depth features, which lack linearity, could potentially lead to convergence problems during the training process. To address this challenge, an innovative approach by introducing a customized loss function is proposed for the estimation. This unique loss function enables the network to predict continuous values, such as the fill factor, while maintaining the robustness and structure of the classification model.

Table 1
Classification and regression in machine learning.

Task	Activation function	Loss function	Output type
Regression	Linear	Mean Square Error	Continuous Value
Classification	Sigmoid/ Softmax	Cross-Entropy	Discrete Classes & Probabilities

In Fig. 6, the Faster R-CNN classifier processes the region proposals generated by the RPN. Each proposal undergoes pooling to achieve a fixed size through RoI pooling, resulting in two significant output components derived from the RoI:

- **Class scores**: This output determines the probability of the proposed region belonging to each of the possible classes.
- **Bounding box**: For object (i.e., bucket) detection, the network adjusts the coordinates of the proposal's box to fit the actual object.

Based on the output of Faster R-CNN, the loss function integrates two individual losses, detailed as follows:

1) **Bounding box localization**: In bounding box localization, the regression loss measures the accuracy of the model in predicting the bounding box coordinates for an object. The regression loss is typically calculated using the Mean Squared Error (MSE), a simple and widely used loss function that measures the deviation between the predicted bounding box adjustments and the ground-truth adjustments:

$$L_{\text{bbox}}(t,v) = \frac{1}{N}\sum_{i=1}^{N}(t_i - v_i)^2 \tag{1}$$

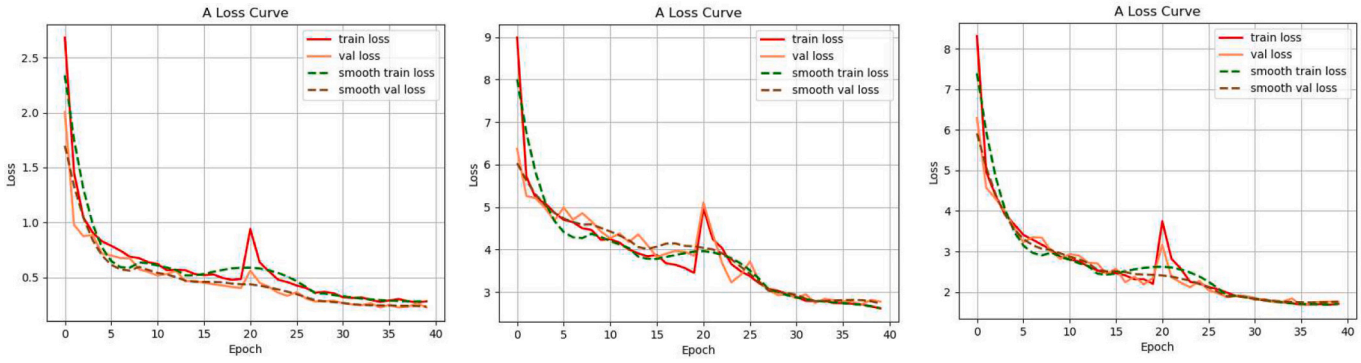
where t represents the predicted bounding box coordinates, v represents the ground-truth bounding box coordinates, and N is the number of coordinates.

2) **Multi-Class Classification**: The bucket fill estimation can be considered as a multi-class classification task, in which Categorical Cross-Entropy (CME) loss is used:

$$L_{\text{CME}}(y_c,p) = \sum_{c=1}^{M} y_c log(p) \tag{2}$$

where M is the number of classes, y_c is a binary indicator (0 or 1) denoting whether the class label is the correct classification, and p is the predicted probability that the observation belongs to the class.

In simple terms, for each observation, the loss is the negative logarithm of the predicted probability for the true class. A high predicted probability for the true class (i.e., close to 1) results in a low loss, and vice versa.



(a) CME loss (b) RMSE loss (c) CUM loss

Fig. 7. Loss comparison.

The design concept of the Classification-Regression Framework arises from the inherent nature of the bucket-fill estimation task, which involves both classification and regression components. Classifying the volume fill degree to a specific level based on the limited training data available is a typical multi-class classification task while determining the specific volume value within each class constitutes a regression task. Therefore, this study proposes a methodology that combines the classification and regression components to compensate for the limitations of a single type of loss.

Specifically, in this study, the predicted score (i.e., the fill factor) is a continuous value in real-time, which is typical for regression tasks, although the training set is a distinct category. To ensure accurate predictions on the training set while achieving good generalization to other fill factors within the range, a custom loss function based on the Root Mean Squared Error (RMSE) for the classification task is used by

$$L_{\mathrm{RMSE}}(y_i, \widehat{y}_i) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2} \tag{3}$$

where y_i is the ground truth value for the ith sample, indicating the ground truth fill factor, $\widehat{y}_i$ is the predicted value for the corresponding sample, and N is the number of fill factor classes. In the context of a regression problem, RMSE provides a measure of how well the predictions align with the true values.

Ultimately, the customized loss function of the classification task for estimating fill factor is a linear combination of two parts given by

$$L_{\mathrm{CUM}} = \alpha \cdot L_{\mathrm{RMSE}} + \beta \cdot L_{\mathrm{CME}} \tag{4}$$

where α and β are weighting factors appropriately balancing the classification and regression objectives.

The methodology behind the selection of weighting factors defined in Eq. (4) is based on the principle that classification loss (CME) should have a higher weight than regression loss (RSME). This strategy ensures that the classification component takes precedence and guides the model to categorize the fill level into discrete classes, as introduced in Fig. 5, thereby narrowing the search space for the regression component. The regression component then fine-tunes the estimation within the identified classes. This combined approach allows the model to achieve high accuracy and efficiency, especially for data not included in the training set. In addition, giving the weighting factor for CME to be higher is beneficial in balancing initial loss values, as RMSE tends to have a larger initial loss, as shown in Fig. 7.

3.2. Evaluation metrics

To assess the effectiveness of the trained neural network, two metrics are introduced to evaluate the estimation precision.

The first metric is the Mean Average Precision (mAP). This metric is critical in evaluating the accuracy of the classification component of the model. It provides a comprehensive measure of how accurately the model predicts the correct class at various thresholds. In essence, mAP quantifies the model's ability to correctly classify the objects, taking into account both false positives and false negatives. The mAP is calculated as follows:

$$\mathrm{mAP} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{R_q}\sum_{k=1}^{R_q}P(k) \times \mathrm{rel}(k) \tag{5}$$

where Q is the total number of classes, R_q is the number of retrieved items for class q, $P(k)$ is the precision at cut-off k in the list of retrieved items, $\mathrm{rel}(k)$ is an indicator function that equals 1 if the item at rank k is a relevant item (correctly classified), and 0 otherwise.

The second metric is the Mean Absolute Error (mAE). This metric is particularly relevant to the regression component of the model, where the goal is to predict the fill factor. The mAE calculates the absolute difference between the predicted fill factor values and the true values. It provides a clear indication of the model's accuracy in estimating the fill factor, with lower mAE values indicating better performance, which is given by

$$\mathrm{mAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y}_i| \tag{6}$$

Together, these metrics provide a comprehensive evaluation of the model's performance, encompassing both its classification and regression capabilities.

Table 2
Networks with comparative loss.

ID	Loss function
CME	Cross-entropy: *loss* = CME (i.e., Eq. (2))
RMSE	RMSE: *loss* = RMSE (i.e., Eq. (3))
CUM	Custom: *loss* = 5·CME + 0.5·RMSE (i.e., Eq. (4))

4. Comparative experiments and analysis

In this section, the comparative experimental results are presented and analyzed to evaluate the performance of the models trained with different loss functions. A comparative set of experiments is outlined in Table 2, where the three loss functions (i.e., CME, RMSE, and CUM introduced in the above section) for each model are specified. Each model is identified by the name of its corresponding loss function.

4.1. Comparative training evaluation

Throughout the training phase, transfer learning techniques are used to expedite the process. Initially, specific layers of the model are frozen for the first 20 epochs, after which these layers are unfrozen to allow for weight updates.

Aiming at validating the advantage of the selected Faster R-CNN training structure in the task of this study, a comparison between Faster R-CNN and YOLOv5 using the same training dataset is performed. To ensure a fair comparison between these two deep learning network structures, Faster R-CNN utilized the default cross-entropy loss, while YOLOv5 utilized the default typical binary classification loss. As shown in Fig. 8, after 40 epochs, Faster R-CNN achieved a mAP of around 0.99, which was higher than YOLOv5's mAP of 0.77.

Fig. 7 shows a comparison of the loss results for the three network models, including the training and validation sections. While the integration of a new loss component leads to an initial rise in the loss value in the early stages of training, the loss function generally converges around the 40th epoch.

In terms of precision during training, there is a noticeable distinction in mean average precision between the customized loss CUM (defined in Eq. (4)), the original cross-entropy loss CME (defined in Eq. (2)), and the original root mean square loss RMSE (defined in Eq. (3)). As shown in Fig. 9, the mAP of the CUM model outperforms the RMSE model, indicating the effective incorporation of the cross-entropy component in improving accuracy.

In addition, regarding the evaluation of the volume fill factor estimation accuracy, Fig. 10 shows that all three models achieve a satisfactory mean absolute error (mAE) converging to a low value of 0.25% after 35 epochs. It is clearly shown that the CUM achieves the fastest mAE convergence. Overall, CUM shows the most exceptional performance among the three models during training.

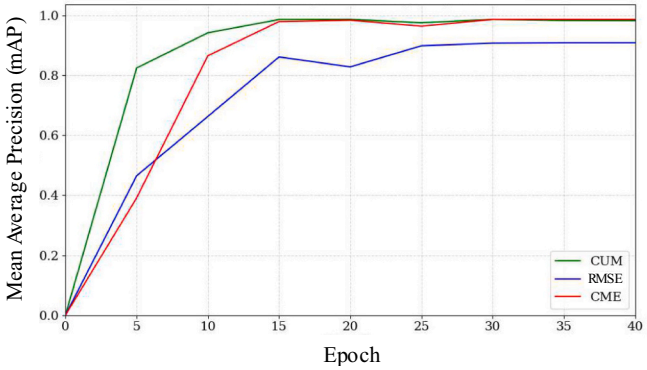


Fig. 9. Mean average precision of training.

4.2. Test results

Comparative tests are performed in this section with the three selected loss functions. It is worth noting that the test data is independent of the training dataset. This distributional approach, as shown in Fig. 5, ensures a rigorous evaluation of the model's generalization capabilities when exposed to out-of-distribution data. It provides significance for its real-world applicability and robustness. In addition, a computer equipped with an NVIDIA RTX 3060 GPU is used for real-time processing. This setup allows for concurrent processing of RGB and depth maps, ensuring efficient real-time performance of industrial mobile machines.

The comparison of the estimation results is presented in Fig. 11. It shows that the model using the proposed loss function CUM achieves significantly lower error rates on the test data compared to the cross-entropy loss model CME. This result indicates that the custom loss function improves the generalizability of the model. The combination of RMSE and cross-entropy further improves the model's adaptability to data beyond the training set.

The average errors (eAE) of the comparative models are shown in Fig. 12. A satisfactory average error of 3.09% is achieved by the proposed CUM, indicating an accuracy of 96.91% in its estimates. In contrast, the CME has a significantly higher error rate of 22.64%. This significant difference underscores the effectiveness of the custom loss function in improving the accuracy of the test, which demonstrates the strength of the model with the proposed custom loss function in learning

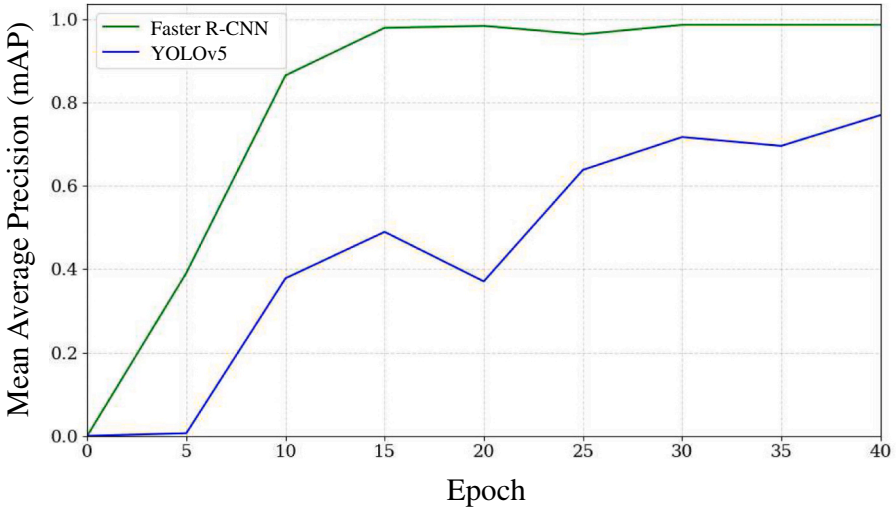


Fig. 8. Results of mAP with comparative learning.

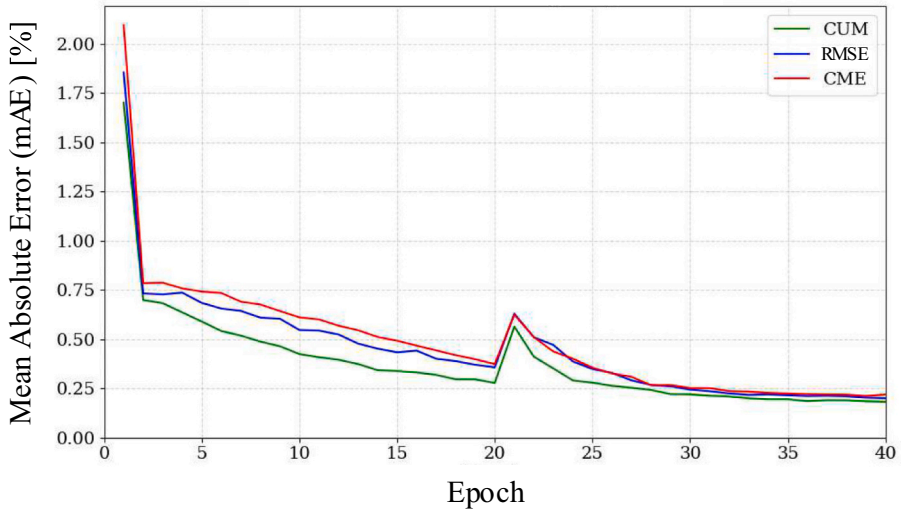


Fig. 10. Mean absolute error of training.

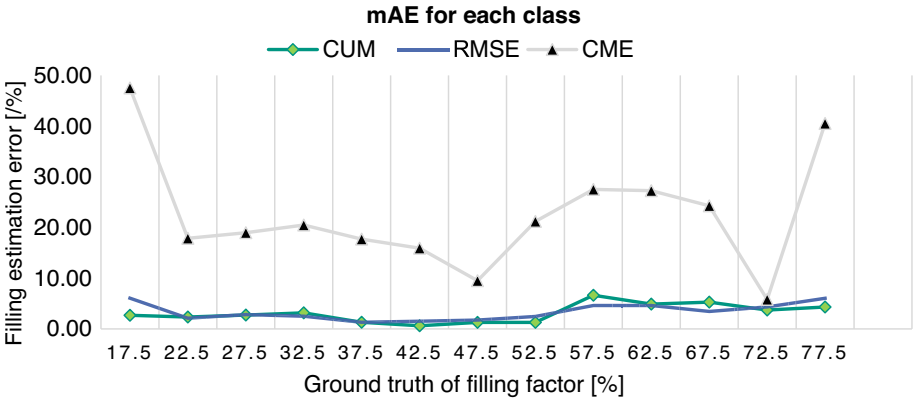


Fig. 11. Mean absolute error of the test.

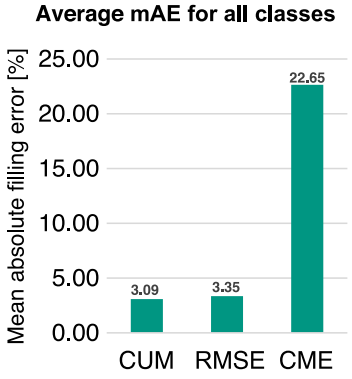


Fig. 12. Average mAE of the test.

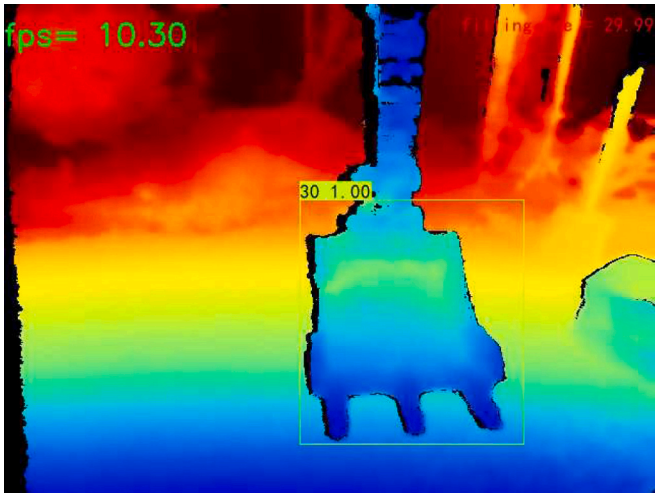
the underlying data patterns and achieving high estimation accuracy in practical applications.

It is worth noting that the high accuracy of the proposed approach was achieved with a low requirement for the size of training data. As introduced in Section 1.1.2, the study by Lu [24] used a dataset of 41,610 samples with 20,882 samples allocated for training, achieving a volume estimation accuracy of 95.25%. In contrast, our method utilized a significantly smaller dataset of 1251 samples and achieved a testing accuracy of 96.91% even with OOD data. This comparison highlights our method's higher sample efficiency and better generalization capability with a substantially smaller dataset.

Furthermore, achieving high real-time processing speeds is crucial for the practical implementation of computer vision-based systems in real-world applications. Using the depth stream as input to the proposed network training model, the output results (including bounding boxes, predicted fill class, and estimated fill factor) are then overlaid on the corresponding RGB stream for visualization and in-depth analysis. In the real-time recording shown in Fig. 13, the model achieves a fast real-time prediction speed of approximately 10 frames per second (fps) (with each frame processed within 100 ms). This processing speed is significantly faster compared to 3D point cloud-based approaches (e.g., [13] as discussed in the Introduction). This performance demonstrates the proposed model's exceptional real-time capabilities, ensuring reliable fill factor estimation in excavator operations.

4.3. Discussion

This study, as fundamental research, aims to provide a solution for real-time volume estimation during earth-moving operations. The method uses 2D depth maps as input to a Faster R-CNN deep learning model, which achieves high fill factor estimation accuracy while maintaining real-time processing speed. In addition, a custom loss function tailored to the classification-regression task is introduced,



(a) Depth stream

(b) RGB stream

Fig. 13. Real-time estimation result.

which further enhances the generalization capabilities of the network. The results demonstrate the feasibility and effectiveness of the proposed approach in addressing the challenges associated with automated excavator operations. By accurately estimating the volume of material in the excavator bucket, the method contributes to improving productivity and worker safety in the construction industry.

5. Conclusion and limitations

In this paper, depth maps are used as input in combination with Faster-RCNN to estimate the fill factor (volume) of an excavator bucket, achieving high estimation accuracy in real-time. A customized dataset containing 1562 depth maps with the different poses of the excavator during the working cycle was created for network training. In particular, a customized loss foundation was designed for the deep-learning neural network to specifically solve the classification-regression task of this study, improving the versatility of the neural network. A remarkable average mean absolute accuracy of 96.91% is achieved by the proposed approach in comparative experiments without training with a large quantity of data. Furthermore, the proposed approach has superior computational efficiency, achieving high processing speed in real-time and maintaining a frame rate over 10 FPS due to the application of depth maps. This combination of high accuracy and speed underscores the potential of depth images and deep learning to revolutionize volume estimation tasks in construction and excavation scenarios, providing a reliable operation evaluation approach for automated excavator systems.

The designed framework and the loss function possess universality and repeatability. However, despite the promising results, this study is constrained by the availability of only one excavator testbench (i.e., the Sany SY16C), which limits the demonstration of the generalizability of the proposed approach. The dataset used in the study is tailored to a specific excavation setup, indicating the need for more diverse datasets to strengthen the robustness of the approach in different construction environments. In future work, a broader range of excavator models should be used to further improve and validate the applicability of the proposed method to different construction scenarios. In addition, due to the characteristics of depth cameras, depth map generation is compromised at low lux levels. Considering that operations in low-light conditions during rainy or snowy days are typically avoided, as excavators are generally not used in such scenarios. For night-time operations, additional lighting would be required to be configured.

Research elements

The dataset customized in this study and the codes can be accessed at: https://gitlab.kit.edu/bobo.helian/bucket-fill-estimation.git.

CRediT authorship contribution statement

Bobo Helian: Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Conceptualization. **Xiaoqian Huang:** Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation. **Meng Yang:** Writing – review & editing, Resources, Project administration. **Yongming Bian:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Marcus Geimer:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This study is funded by the Open Foundation of the State Key Laboratory of Fluid Power and Mechatronic Systems under Grant GZKF-202202 and GZKF-202317. This study is also supported by the National Natural Science Foundation of China (52205279) and the National Key Research and Development Program of China (2023YFF0613200).

References

[1] M. Geimer, Mobile Working Machines, SAE International, Warrendale, Pennsylvania (USA), 2020, https://doi.org/10.4271/9780768094329.
[2] S.K. Baduge, S. Thilakarathna, J.S. Perera, M. Arashpour, P. Sharafi, B. Teodosio, A. Shringi, P. Mendis, Artificial intelligence and smart vision for building and construction 4.0: machine and deep learning methods and applications, Autom. Constr. 141 (2022) 104440, https://doi.org/10.1016/j.autcon.2022.104440.
[3] J. Guevara, T. Arevalo-Ramirez, F. Yandun, M. Torres-Torriti, F.A. Cheein, Point cloud-based estimation of effective payload volume for earthmoving loaders, Autom. Constr. 117 (2020) 103207, https://doi.org/10.1016/j.autcon.2020.103207.
[4] W. Guan, Z. Chen, S. Wang, G. Wang, J. Guo, Z. Liu, A deep learning approach for construction vehicles fill factor estimation and bucket detection in extreme

environments, Comput. Aided Civ. Inf. Eng. 38 (13) (2023) 1857–1878, https://doi.org/10.1111/mice.12952.
[5] S. Dadhich, F. Sandin, U. Bodin, U. Andersson, T. Martinsson, Field test of neural-network based automatic bucket-filling algorithm for wheel-loaders, Autom. Constr. 97 (2019) 1–12, https://doi.org/10.1016/j.autcon.2018.10.013.
[6] S. Dadhich, U. Bodin, U. Andersson, Key challenges in automation of earth-moving machines, Autom. Constr. 68 (2016) 212–222, https://doi.org/10.1016/j.autcon.2016.05.009.
[7] R. Horn, M. Lebert, Soil compactability and compressibility, in: Developments in Agricultural Engineering vol. 11, Elsevier, 1994, pp. 45–69, https://doi.org/10.1016/B978-0-444-88286-8.50011-8.
[8] A. Assadzadeh, M. Arashpour, H. Li, R. Hosseini, F. Elghaish, S. Baduge, Excavator 3d pose estimation using deep learning and hybrid datasets, Adv. Eng. Inform. 55 (2023) 101875, https://doi.org/10.1016/j.aei.2023.101875.
[9] E. Yurtsever, J. Lambert, A. Carballo, K. Takeda, A survey of autonomous driving: Common practices and emerging technologies, in: IEEE, 2024, pp. 58443–58469, https://doi.org/10.1109/ACCESS.2020.2983149. Access 8.
[10] E. Arnold, O.Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, A. Mouzakitis, A survey on 3d object detection methods for autonomous driving applications, IEEE Trans. Intell. Transp. Syst. 20 (10) (2019) 3782–3795, https://doi.org/10.1109/TITS.2019.2892405.
[11] W.-C. Chang, C.-H. Wu, Y.-H. Tsai, W.-Y. Chiu, Object volume estimation based on 3D point cloud, in: 2017 International Automatic Control Conference (CACS), 2024, pp. 1–5, https://doi.org/10.1109/CACS.2017.8284244.
[12] J. Guevara, T. Arevalo-Ramirez, F. Yandun, M. Torres-Torriti, F.A. Cheein, Point cloud-based estimation of effective payload volume for earthmoving loaders, Autom. Constr. 117 (2020) 103207, https://doi.org/10.1016/j.autcon.2020.103207.
[13] J. Lu, Q. Bi, Y. Li, X. Li, Estimation of fill factor for earth-moving machines based on 3d point clouds, Measurement 165 (2020) 108114, https://doi.org/10.1016/j.measurement.2020.108114.
[14] K. Mirzaei, M. Arashpour, E. Asadi, H. Masoumi, Y. Bai, A. Behnood, 3d point cloud data processing with machine learning for construction and infrastructure applications: a comprehensive review, Adv. Eng. Inform. 51 (2022) 101501, https://doi.org/10.1016/j.aei.2021.101501.
[15] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. NieÃŸner, R. Klein, A. Kolb, State of the art on 3d reconstruction with rgb-d cameras, Comp. Graph. Forum 37 (2) (2018) 625–652, https://doi.org/10.1111/cgf.13386.
[16] J. Shen, W. Yan, P. Li, X. Xiong, Deep learning-based object identification with instance segmentation and pseudo-lidar point cloud for work zone safety, Comput. Aided Civ. Inf. Eng. 36 (12) (2021) 1549–1567, https://doi.org/10.1111/mice.12749.
[17] M. Prasad, A. Fitzgibbon, Single view reconstruction of curved surfaces, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) Vol. 2, 2024, pp. 1345–1354, https://doi.org/10.1109/CVPR.2006.281.
[18] C. Choy, J. Gwak, S. Savarese, 4D spatio-temporal convnets: Minkowski convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3075–3084, https://doi.org/10.1109/CVPR.2019.00319.
[19] P. Li, B. Zhou, C. Wang, G. Hu, Y. Yan, R. Guo, H. Xia, Cnn-based pavement defects detection using grey and depth images, Autom. Constr. 158 (2024) 105192, https://doi.org/10.1016/j.autcon.2023.105192.
[20] S. Foix, G. Alenya, C. Torras, Lock-in time-of-flight (tof) cameras: a survey, IEEE Sensors J. 11 (9) (2011) 1917–1926, https://doi.org/10.1109/JSEN.2010.2101060.
[21] S. Li, A.B. Chan, 3D human pose estimation from monocular images with deep convolutional neural network, in: Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1–5, 2014, Revised Selected Papers, Part II 12, Springer, 2015, pp. 332–347, https://doi.org/10.1007/978-3-319-16808-1_23.
[22] D. Maji, S. Nagori, M. Mathew, D. Poddar, Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2637–2646, https://doi.org/10.48550/arXiv.2204.06806.
[23] F. Alama, H. Koa, H. Leea, C. Yuanb, Deep learning approach for volume estimation in earthmoving operation, Int. J. Industr. Eng. Manag. (IJIEM) 14 (1) (2023), https://doi.org/10.24867/IJIEM-2023-1-323.
[24] J. Lu, Z. Yao, Q. Bi, X. Li, A neural network–based approach for fill factor estimation and bucket detection on construction vehicles, Comput. Aided Civ. Inf. Eng. 36 (12) (2021) 1600–1618, https://doi.org/10.1111/mice.12675.
[25] R. Girshick, Fast r-cnn, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, https://doi.org/10.1109/ICCV.2015.169.
[26] Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: a survey, Proc. IEEE 111 (3) (2023) 257–276, https://doi.org/10.1109/JPROC.2023.3238524.
[27] J. Yao, D. Cai, X. Fan, B. Li, Improving yolov4-tiny rsquo;s construction machinery and material identification method by incorporating attention mechanism, Mathematics 10 (9) (2022), https://doi.org/10.3390/math10091453. URL, https://www.mdpi.com/2227-7390/10/9/1453.
[28] Y. Xiang, J. Zhao, W. Wu, C. Wen, Y. Cao, Automatic object detection of construction workers and machinery based on improved yolov5, in: W. Guo, K. Qian (Eds.), Proceedings of the 2022 International Conference on Green Building, Civil Engineering and Smart City, Springer Nature Singapore, Singapore, 2023, pp. 741–749, https://doi.org/10.1007/978-981-19-5217-3_74.
[29] P. Soviany, R.T. Ionescu, Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction, in: 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2018, pp. 209–214, https://doi.org/10.1109/SYNASC.2018.00041.
[30] T. Mahendrakar, A. Ekblad, N. Fischer, R. White, M. Wilde, B. Kish, I. Silver, Performance study of yolov5 and faster r-cnn for autonomous navigation around non-cooperative targets, in: 2022 IEEE Aerospace Conference (AERO), 2022, pp. 1–12, https://doi.org/10.1109/AERO53065.2022.9843537.
[31] R.A. Hamzah, H. Ibrahim, et al., Literature survey on stereo vision disparity map algorithms, J. Sens. 2016 (2016), https://doi.org/10.1155/2016/8742920.
[32] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149, https://doi.org/10.1109/TPAMI.2016.2577031.
[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.