Deep Learning for Environmental Remote Sensing Image Understanding: Analyzing Dust and Anthropogenic Objects Across Varying Scales and Densities

---

Andreas Michel

Doctoral Thesis 2025

Deep Learning for Environmental Remote Sensing Image Understanding: Analyzing Dust and Anthropogenic Objects Across Varying Scales and Densities

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für
Bauingenieur-, Geo- und Umweltwissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Andreas Michel

Tag der mündlichen Prüfung: 26.02.25

Referent: Prof. Dr.-Ing. habil. Stefan Hinz
Institut für Photogrammetrie und Fernerkundung (IPF)
Karlsruher Institut für Technologie (KIT)

Korreferent: Priv.-Doz. Dr.-Ing. Martin Weinmann
Institut für Photogrammetrie und Fernerkundung (IPF)
Karlsruher Institut für Technologie (KIT)

Korreferent: apl. Prof. Dr. techn. Franz Rottensteiner
Institut für Photogrammetrie und GeoInformation (IPI)
Leibniz-Universität Hannover (LUH)

Karlsruhe (2025)

Abstract

Steady advancements in sensor technologies and platforms have significantly increased the collection of electro-optical data over large areas of the Earth's surface. To manage this vast amount of data, deep learning methods offer promising solutions for various tasks such as classification, object detection, semantic segmentation, and density estimation. Despite ongoing challenges, this work focuses on analyzing objects and particles at different scales and densities.

Analyzing airborne dust is particularly suitable for this purpose. However, developing an efficient and robust algorithm for airborne dust analyzing entails several hurdles. Airborne dust can be opaque or translucent, exhibit considerable variation in density, and possess indistinct boundaries. Moreover, distinguishing dust from other atmospheric phenomena, such as fog or clouds, is particularly challenging.

In this work, multiple neural networks are developed for the analyzing of dust. The initial focus is on density estimation, but to improve the approach, other methods and tasks are considered, particularly object detection. Not only are neural network architectures examined, but attention is also given to training and inference optimization. Instead of dust, anthropogenic objects such as ships or airplanes are primarily considered, which can vary greatly in their scale and object density.

One of the strengths of this study is the combination of methods from density estimation and object detection. It is demonstrated that the integration of both tasks can lead to improved object detection, better object counting, and ultimately enhance dust density estimation. The proposed methods are evaluated on multiple datasets and compared with state-of-the-art techniques.

This work aspires to advance the research of analyzing objects and particles in remote sensing across various object sizes and scales, especially for dust monitoring. Analyzing of airborne dust holds substantial potential benefits for climate protection, environmentally sustainable construction, scientific research, and various other fields.

Kurzfassung

Stetige Fortschritte in Sensortechnologien und Sensorplattformen haben die Erfassung elektro-optischer Daten über große Gebiete der Erdoberfläche erheblich gesteigert. Um diese enorme Datenmenge zu bewältigen, bieten Deep-Learning-Methoden vielversprechende Lösungen für verschiedene Aufgaben wie Klassifikation, Objekterkennung, semantische Segmentierung und Dichteschätzung. Trotz anhaltender Herausforderungen konzentriert sich diese Arbeit auf die Analyse von Objekten und Partikeln in unterschiedlichen Maßstäben und Dichten.

Die Entwicklung eines effizienten und robusten Algorithmus zur Analyse von luftgetragenem Staub ist von großer Bedeutung. Allerdings stellt die Detektion von Staubpartikeln eine besondere Herausforderung dar, da Staub opak oder transluzent sein kann, erhebliche Variationen in der Dichte aufweist und unscharfe Grenzen besitzt. Zudem ist die Unterscheidung von Staub gegenüber anderen atmosphärischen Phänomenen wie Nebel oder Wolken besonders schwierig. In dieser Arbeit werden mehrere neuronale Netzwerke für die Analyse von Staub entwickelt. Der anfängliche Fokus liegt auf der Dichteschätzung; um den Ansatz jedoch zu verbessern, werden weitere Methoden und Aufgaben berücksichtigt, insbesondere die Objekt- erkennung. Es werden nicht nur neuronale Netzwerkarchitekturen untersucht, sondern auch Optimierungen im Training und bei der Inferenz vorgenommen. Zusätz- lich werden, anstelle von Staub, hauptsächlich Objekte wie Schiffe oder Flugzeuge betrachtet, die in ihrem Maßstab und ihrer Objektdichte stark variieren können.

Eine der Stärken dieser Studie ist die Kombination von Methoden der Dichteschätzung und Objektdetektion. Es wird gezeigt, dass die Integration beider Aufgaben zu verbesserter Objekterkennung, präziserer Objektzählung und letztlich zu einer optimierten Staubdichteschätzung führen kann. Die vorgeschlagenen Methoden werden an mehreren Datensätzen evaluiert und mit aktuellen Referenzmethoden verglichen.

Diese Arbeit strebt an, die Forschung zur Analyse von Objekten und Partikeln in der Fernerkundung über verschiedene Objektgrößen und Maßstäbe hinweg voranzutreiben, insbesondere hinsichtlich der Staubüberwachung. Die Analyse von in der Luft befindlichem Staub bietet potenzielle Vorteile für den Klimaschutz, umweltverträgliches Bauen, wissenschaftliche Forschung und verschiedene andere Bereiche.

Contents

1 Introduction

> *Wer nichts verändern will, wird auch das verlieren, was er bewahren möchte.*
>
> — **Heinemann, Gustav**
> (Politician)

Continuous advancements in sensor technologies and platforms have significantly enhanced the collection of vast amounts of electro-optical data over extensive areas of the Earth's surface. However, the sheer volume of these data renders comprehensive manual analysis impractical. Consequently, Computer Vision (CV) methods offer effective automated solutions to this challenge without manual intervention. In recent years, deep learning techniques have emerged as particularly powerful tools, achieving remarkable success in tasks such as classification [69], object detection [120], semantic segmentation [122], density estimation [177], and natural language processing [152]. These methods have also been effectively applied to remote sensing scenarios [180].

Remote sensing presents unique challenges compared to traditional computer vision [94]. The area covered by the images is often significantly larger, covering extensive geographical areas, and the image resolution can vary widely. The data is often multimodal and requires specialized domain knowledge for accurate interpretation. Objects of interest in remote sensing data, e. g. buildings, vehicles, and weather phenomena, can appear at varying densities, scales, and orientations, which complicates monitoring efforts. Additionally, the complex backgrounds inherent in such data further hinder accurate detection and analysis. For example, ships may exhibit very high object density within a harbor, where numerous vessels are clustered together, making individual identification challenging. Conversely, in an open ocean scenario, ships may be sparsely distributed, posing different detection challenges due to their relatively small size against the vast expanse of the sea. This latter scenario is particularly crucial for monitoring dark fishing fleets, as highlighted by [110].

Similarly, monitoring smaller particles, such as those present in dust, poses its own set of challenges. Dust exposures are only detectable at high densities without

specialized optical equipment, monitoring airborne dust emissions remains essential due to its wide-ranging impacts, affecting everything from climate to human health [46, 146]. Comprehensive monitoring of dust can facilitate the identification of environmental hazards and the development of effective mitigation strategies [128]. However, conventional dust measurement such as by *in-situ* monitoring equipment or 3D LiDAR scanning devices is typically expensive and limited in their capacity to monitor the spatial characteristics of dust. Vision-based camera systems present a potential tool for measuring these spatial characteristics, yet the automatic detection of airborne dust within these images has not been extensively researched in comparison to other fields. In particular, regression approaches, such as dust density estimation, are underexplored.

Therefore, there is a need for methods that focus on a more holistic methodology to monitor objects and particles at different scales, which would be advantageous for applications in environmental monitoring, disaster management, and resource exploration.

In addition, recent advancements in deep learning show that successful methodologies in one domain can often be adapted effectively to others. Notable examples include the U-Net [122] architecture, initially designed for medical imaging, which has proven advantageous across various applications in computer vision. Similarly, the transformer architecture has garnered significant success in Natural Language Processing (NLP), and its adaptation for computer vision through the Vision Transformer (ViT) [32] exemplifies this trend. Consequently, it is prudent to explore diverse techniques to address the challenges at hand.

1.1 Main Objective and Research Goals

This work focuses on the analyzing of objects and particles across a wide range of sizes and scales. The aim of this work is to monitor small particles like dust as well as larger anthropogenic objects such as cars, planes, and ships. To achieve this, neural network architectures for two-dimensional object detection, instance segmentation, and density estimation are analyzed and improved within the context of remote sensing.

While multi-task training optimization methods exist [60], the scarcity of panoptic datasets [67] in remote sensing necessitates a more traditional training approach in this work.

Firstly, the primary focus is on analyzing airborne dust exposures. Most related research concentrates on satellite imagery [74] or addresses related problems such as binary segmentation of smoke [173]. Analyzing dust density is particularly challenging due to the variable nature of dust particles, which can range from opaque to translucent and lack consistent visual cues. The density and appearance of dust vary widely, often resulting in images with indistinct boundaries and low spatial contrast. Traditional algorithms struggle because dust can resemble other atmospheric phenomena, such as fog or clouds. The diversity of dust sources and the influence of environmental conditions further complicate detection. Moreover, the absence of consistent color schemes exacerbates the problem. Therefore, these reasons could explain why the analyzing of airborne dust emissions remains underexplored.

Some efforts have been made in binary segmentation, such as dust road emission analyzing [29] or object detection for dust analyzing at construction sites [156]. Due to the small size of dust particles, it is not feasible to detect each particle individually; instead, attention is directed toward dust plumes. While detecting dust plumes can provide localization information, it does not offer precise details about dust intensity. While a semantic segmentation approach could be used to analyze different dust levels, dust boundaries are often fuzzy and exhibit smooth transitions. Therefore, a regression approach like density estimation is more suitable for capturing the nuances of dust distribution. To the best of our knowledge, no prior work has addressed dust density estimation within deep learning. Consequently, we present the development of multiple neural network architectures specifically designed for dust density estimation.

Subsequently, attention is directed toward larger objects, applying training and inference optimization techniques. While much of the existing research focuses on neural network architectures, this study also addresses non-architectural optimizations. Post-processing techniques, such as box-based non-maximum suppression (NMS) [50], often suffer from issues like double detections, partial detections, clusters of objects, overlapping concave objects, and diagonally-aligned objects. The proposed method addresses these challenges by utilizing semantic masks and emphasizing object size, which is particularly important in scenarios with high object densities. Additionally, selecting the correct training and inference scheme can be a significant challenge, greatly impacting detection performance. This aspect is also analyzed extensively.

Finally, having focused on density estimation and object detection separately, the subsequent step involves integrating both approaches. In this endeavor, the previously described challenges are further addressed. Specifically, the tasks of object detec-

tion and counting, as well as dust density estimation, are undertaken, particularly considering varying object and particle densities and scales.

This work addresses three major research goals:

1. Developing and studying dust density estimation techniques.
2. Developing non-architectural optimizations for object detection techniques.
3. Fusing the results from the previous steps by combining object detection techniques with density estimation methods to improve object counting, object detection, and density estimation.

Fusion approaches require a fundamental understanding of the methods used, particularly their strengths and weaknesses. Therefore, it is necessary to focus on the methods individually before combining them. By applying the proposed methods to a wide range of tasks, their versatility is demonstrated while still using the analyzing of airborne dust emissions as a case study.

1.2 Outline and Thesis Contributions

This dissertation is structured into seven chapters, as illustrated in Figure 1.1, which provides an overview of the structure.

Chapter 2 introduces the fundamental concepts underlying this work. It begins with a basic overview of machine learning, then addresses feature extractors by examining Convolutional Neural Networks (CNNs) and, subsequently, the more recent Transformer architectures. The strengths and weaknesses of each method are highlighted.

Chapter 3 focuses on Dust Density Estimation and traces the evolution of proposed architectural frameworks from CrowdFPN to DustNet++, including intermediate developments like DeepDust and DustNet. This historical perspective highlights advancements in algorithmic design and emphasizes the iterative nature of research in this field. After presenting the datasets used for training and testing the models and their configurations, the chapter provides a comprehensive overview of the experimental results. The chapter concludes with a critical discussion of the findings, synthesizing insights from the experiments and placing them in the broader context of the field.

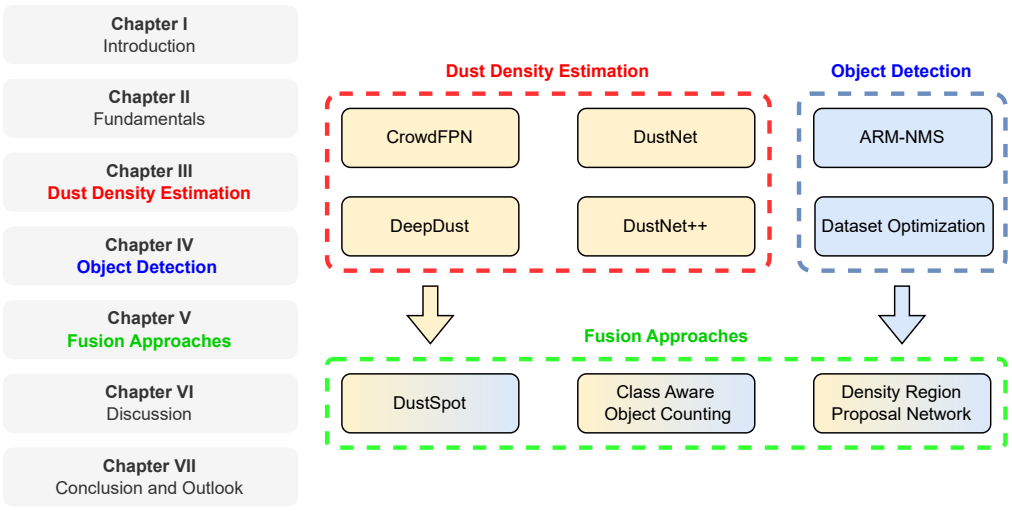


Figure 1.1: Structure of this Thesis. This figure shows the most important building blocks of this thesis: Dust Density Estimation is indicated in **red**, Object Detection in **blue** and Fusion Approaches in **green**.

Chapter 4 illustrates that non-architectural object detection techniques can significantly enhance object detection performance in complex scenes with considerable variability in scale and density. A novel filtering technique for instance segmentation results is presented and tested on a relevant dataset. Additionally, a three-stage training scheme for maritime object detection is introduced, leveraging multiple representations of input data during inference, outperforming existing methods without modifying the model's architecture. Overall, this chapter demonstrates that factors such as training schemes, data handling, and post-processing techniques are as crucial as architectural optimization in improving object detection performance.

Chapter 5 demonstrates how fusion approaches between density estimation and object detection can enhance detection and counting performance in scenes characterized by significant variability in scale and density. Three distinct fusion approaches are introduced, each designed to improve the effectiveness of object detection, object counting, and dust density estimation, respectively. By combining object detection with density estimation, the analyzing of remote sensing objects and particles at different scales is enhanced.

Chapter 6 discusses the primary research topics of this work. It begins with an analysis of dust density research, then addresses object detection and object counting in contexts characterized by significant variance in object appearances and densities. Finally, it explores the broader societal implications of dust research.

Chapter 7 concludes the work by summarizing the research conducted. The conclusion is followed by an outlook on future studies based on the presented findings.

2 Fundamentals

This chapter begins with a comprehensive summary of the recent history of Deep Learning in the realm of computer vision (CV), providing essential context and highlighting significant advancements within the field. The focus is on topics related to CV, as they are foundational to remote sensing methodologies. Following this introduction, the fundamental concepts of vision-based deep learning are discussed. Furthermore, the important topic of regularization is addressed, as it is crucial for understanding the nuances of this work. An in-depth examination of Convolutional Neural Networks (CNNs) is presented, exploring their foundational principles and their pivotal role in processing visual data. Subsequently, attention is directed toward the more recent Vision Transformer (ViT) architecture, which is analyzed in terms of its innovative approach to addressing CV tasks and its advantages over traditional methods. This overview does not aim to provide an exhaustive review. Rather, it seeks to enhance the comprehensibility of the design process within this work.

2.1 Recent History of Deep Learning for Image Processing

The origins of neural networks can be traced back to the early 1940s [95], when neurons in the brain were modeled using simple electrical circuits. Building upon this foundational idea, researchers developed the first perceptrons [124]. Subsequent advancements led to the development of the Multi-Layer Perceptrons (MLP) algorithm [123, 59, 4, 126]. The next significant step was backpropagation, popularized by Rumelhart et al. in 1986 [126], which serves as the default learning mechanism for nearly all modern deep learning algorithms. In conjunction with backpropagation, the first Convolutional Neural Network (CNN) layers were introduced in 1989 [72]. This pioneering work successfully applied CNNs to the task of reading postal service zip codes, demonstrating the practical potential of neural networks in image recognition tasks. Nearly a decade later, LeNet-5 was presented

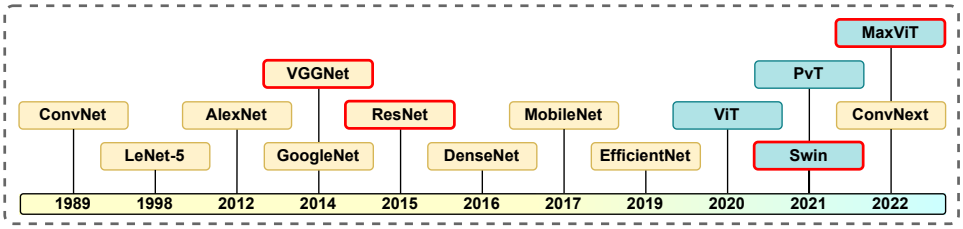


Figure 2.1: Recent History of Deep Learning for Visual Tasks. This figure depicts major milestones regarding the evolution of deep learning algorithms. Networks enclosed in blanched almond-colored boxes represent convolutional neural network-based architectures, while networks highlighted in light blue denote transformer-based architectures. Algorithms marked with a red border are those employed and modified in this thesis. Although algorithms developed after 2022 have also been examined for this work, it cannot be determined at the time of writing whether they will gain widespread acceptance.

by Yann LeCun and colleagues in 1998 [73], proposing a larger and more refined network architecture. This development process is illustrated in Figure 2.1.

Despite these earlier developments, significant progress remained limited until the introduction of AlexNet [69], which won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [127] and marked a pivotal milestone in CV. AlexNet revolutionized the field by being among the first to utilize the Rectified Linear Unit (ReLU) as an activation function, combined with normalization layers and extensive data augmentation. These innovations, along with leveraging graphics processing unit (GPU) acceleration for training, enabled the network to be significantly deeper and more effective, leading to unprecedented performance in image classification tasks.

Building on the success of earlier architectures, the Visual Geometry Group (VGG) network [137] demonstrated that stacking multiple small convolutional kernels, specifically 3×3 filters, could effectively achieve a large effective receptive field (ERF) while reducing the number of learnable parameters and increasing computational efficiency. This approach allowed for deeper networks without incurring prohibitive computational costs, resulting in improved performance on image recognition tasks. Another winner of the ILSVRC was GoogLeNet [143], which introduced parallel convolutional structures known as inception modules. These modules enabled simultaneous multi-scale processing by performing parallel convolutions with filters of different sizes within the same layer. This efficient design utilized computational resources effectively and reduced the number of parameters, allowing for a deeper and wider network without increasing computational expense.

A further milestone was the introduction of Residual Networks (ResNet) by He et al. [48], which introduced residual skip connections to improve the flow of information and gradients through network layers. These skip connections mitigated the vanishing gradient problem that affects very deep networks in a negative way by allowing inputs to bypass layers via identity mappings. This innovation facilitated the training of much deeper architectures that achieved excellent results across various CV challenges.

DenseNet [57] proposed further innovations by introducing a novel connectivity pattern that reduces the issue of vanishing gradients even more. By connecting each layer directly to every other layer in a feed-forward manner, DenseNet strengthened feature propagation and encouraged feature reuse throughout the network. MobileNet [55], through the introduction of depthwise separable convolutions, and EfficientNet [144], incorporating squeeze-and-excitation blocks along with a compound scaling method, aimed to enhance network efficiency. These latter advancements focus primarily on making neural networks more efficient and smaller, allowing edge devices to benefit from the technical advancements in CV.

Prior to 2017, the field of natural language processing (NLP) was primarily dominated by recurrent neural networks, notably Long Short-Term Memory (LSTM) networks [53] and Gated Recurrent Units (GRUs) [24]. The introduction of the Transformer architecture by Vaswani et al. [152] marked a groundbreaking development that profoundly reshaped NLP. By leveraging self-attention mechanisms to model long-range dependencies in sequential data efficiently, the Transformer established a new benchmark for state-of-the-art methodologies. Central to this architecture are the Multi-Headed Self-Attention (MSA) and Cross-Attention (MCA) mechanisms, which utilize multiple sequences of scaled dot-product attention to facilitate complex attention processes within the network.

Leveraging the significant success of Transformers in natural language processing (NLP), Dosovitskiy et al. introduced the ViT [32], adapting the Transformer architecture for CV applications. This model segments an image into a series of fixed-size patches, treating each patch as a token, and employs global self-attention to analyze these visual components. In contrast to CNNs, which possess strong inductive biases such as locality and translation invariance through convolutional filters and pooling operations, the ViT lacks these biases. This absence of inductive bias enables the ViT to scale more effectively [145], but it necessitates training on larger datasets to achieve competitive performance. Furthermore, when processing large images, the computational demands of global self-attention increase quadratically with the

number of patches, rendering the basic ViT model computationally intensive and less practical.

To address the scalability challenges inherent in processing large images, particularly the computational intensity of global self-attention in the ViT, Liu et al. proposed the Swin Transformer [90]. This architecture introduces a hierarchical structure with shifted windows, effectively reducing computational complexity from quadratic to linear with regard to image size. By computing self-attention within local windows that shift between layers, the Swin Transformer achieves a balance between computational efficiency and modeling capacity, enhancing the practicality of Transformers across a diverse range of CV applications.

An enhanced version, the Swin Transformer V2 [89], further refines the model by substituting the scaled dot-product attention with cosine attention, offering improved performance at larger image scales. Similarly, MaxViT [149] advances transformer-based approaches in CV by combining window-based and grid-based attention mechanisms. This combination maintains a sparsely connected yet globally ERF with linear complexity. Extending the application of the Swin Transformer to temporal data, the Video Swin Transformer [91] modifies the original architecture to link patches spatially and temporally. This adaptation facilitates efficient processing of video sequences, demonstrating the versatility of the Swin Transformer in various CV contexts.

Further noteworthy approaches include DeiT [147], which significantly improves the training of ViT through knowledge distillation. Moreover, numerous hybrid architectures combine ViT and CNN [157, 163, 27], aiming to leverage the strengths of both models.

2.2 Foundations of Image Understanding

This section focuses on the fundamental concepts of deep learning applied to image understanding. Figure 2.2 provides a comprehensive overview of the general architecture of contemporary deep learning algorithms for 2D image understanding. Neural networks typically consist of two types of parameters: non-trainable parameters, known as hyperparameters, and trainable parameters, which are adjusted during the training process in accordance with a loss function, referred to as weights.

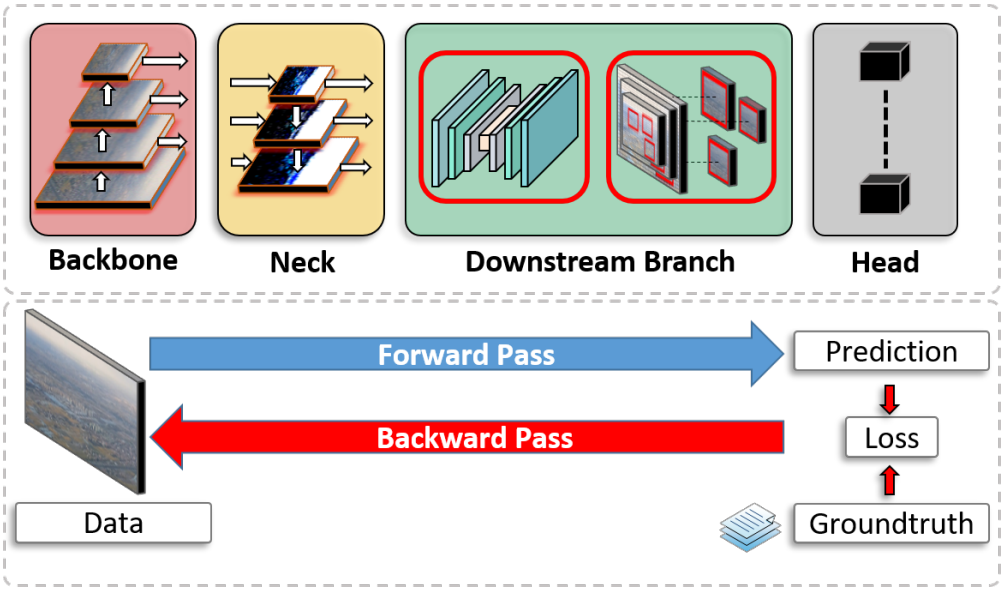


Figure 2.2: Basic Components of a Vision-based Neural Network for Image Understanding. This figure illustrates the fundamental structure of a vision-based deep learning network. The primary components include a backbone, which serves as the feature extractor, an optional supplementary neck to enhance feature flow, a downstream branch, and the corresponding head. The configuration of the downstream branch depends on the specific task, such as object detection, semantic segmentation, or density estimation. All components are usually trained end-to-end.

In the context of supervised training, images are processed, and the results are compared with a ground truth, which serves as the basis for calculating the loss. Through backpropagation [126], the loss is utilized in the backward pass to adjust the weights, aiming to minimize the loss. This is achieved by computing the gradients, which indicate the extent to which each weight should be adjusted to reduce errors effectively. This iterative optimization process is crucial for enhancing the performance of neural networks.

The main components of the architecture for 2D image understanding include the backbone, an optional neck, the downstream branch, and the head:

Backbone. This is the foundational network responsible for feature extraction. The backbone typically consists of a stem and multiple backbone blocks. The stem is generally not very deep, consisting of only two CNN layers with a stride of 2 to downsample the input image effectively. The subsequent blocks are composed of multiple layers, typically showing a decrease in spatial size as depth increases, while simultaneously conveying a higher level of semantics. This hierarchical structure enables the network to capture features at varying levels of abstraction, facilitating a comprehensive understanding of the input image. In the context of multimodal

data, the architecture may include multiple backbones to process different types of input data independently.

Neck. This component is optional and often incorporates a structure similar to a Feature Pyramid Network (FPN) [82]. The neck enhances the feature flow by facilitating downward propagation of features, thereby allowing for better integration of multi-scale information. More complex architectures may include multiple downsampling and upsampling layers, which further refine the feature representation. This design enables the network to leverage both high-resolution features from earlier layers and semantically rich features from deeper layers, ultimately improving the model's ability to perform downstream tasks.

Downstream Branch. The downstream branch can vary significantly and is heavily dependent on the specific task. For 2D image understanding tasks, common applications include object detection, panoptic segmentation, instance segmentation, semantic segmentation, and density estimation. In cases where the task is straightforward image classification, this component may not be required. Instead, the output can be directly derived from the features extracted by the backbone. However, for more complex tasks, the downstream branch is essential as it processes the extracted features to generate the necessary outputs, tailoring the model's predictions to the specific requirements of each application. This adaptability is a key strength of modern deep learning architectures, allowing them to address a wide range of visual perception tasks.

Head. The head is the final module that processes the output from the downstream branch to produce the final predictions. It can be distinguished between inference and training modes, as the latter incorporates loss functions for optimization. Additionally, it is possible to have multiple heads, particularly in scenarios involving multi-task learning or tasks such as instance segmentation. This allows the model to simultaneously perform different predictions, leveraging shared features for improved efficiency. Furthermore, auxiliary heads [143] can be utilized during training to enhance the learning process, although they are typically not employed during inference. These auxiliary heads can provide additional supervision, helping to refine the model's performance on primary tasks. Overall, the head plays a critical role in converting the learned representations into useful outputs, tailored to the specific objectives of the application.

2.3 Regularization

The primary goal of regularization in machine learning is to prevent overfitting, which occurs when a model excessively adapts to the training data, ultimately leading to a decline in performance during inference. Overfitting results from a model capturing noise and specific patterns in the training set rather than the underlying general trends that are present in the data.

Regularization can be achieved through a variety of techniques, each designed to impose constraints on the model's complexity or to enhance its ability to generalize to unseen data. Some notable regularization methods include:

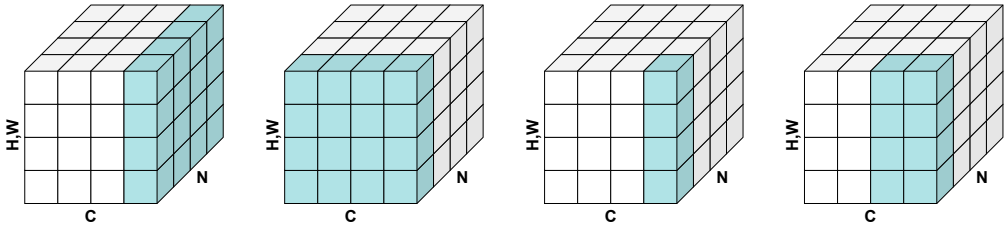
Early Stopping. This technique is relatively straightforward yet highly effective [170]. During the training process, if the validation performance metrics begin to degrade, the training is stopped. This is typically monitored by evaluating the model's performance on a separate validation dataset at regular intervals.

Dropout and Drop Path Regularization. Commonly used in neural networks, dropout [141] involves randomly deactivating a subset of neurons during training. This prevents the model from becoming overly reliant on specific neurons, promoting a more robust representation of the data. Drop Path [71] expands this idea by deactivating neurons and whole branches in a given neural network while training. This leads to more robust learned weights.

Data Augmentation. During training, the diversity of the dataset can be enhanced through various simple data transformations, such as rotation, scaling, and flipping, applied to the input data [135]. These augmentations effectively increase the variability within the training set, which in turn enhances the model's generalization capabilities. By exposing the model to a wider range of scenarios, data augmentation helps to learn more robust features that are less sensitive to specific instances in the training data.

However, it is crucial that the chosen data augmentation techniques are contextually appropriate for the specific application. For instance, in a maritime setting, applying a vertical flip transformation may not be sensible, as ships do not hover in the air, and their orientation is significant. Such an inappropriate transformation could lead to misleading representations and degrade the model's performance.

Normalization. Normalization techniques can improve regularization and reduce Internal Covariate Shift [58]. Internal Covariate Shift is a phenomenon observed in neural networks during training, characterized by the changing distribution of inputs



(a) Batch Norm. **(b)** Layer Norm. **(c)** Instance Norm. **(d)** Group Norm.

Figure 2.3: Normalization Layers. An overview of fundamental normalization layers employed in deep learning architectures. The variables C, N, H, and W represent the channel, batch, and spatial dimensions, respectively. The colored parts indicate the components included in a normalization group.

to a specific layer as the parameters of preceding layers are updated. As the model learns and modifies its weights, the outputs produced by earlier layers fluctuate, consequently impacting the input distribution for later layers. This continuous alteration can introduce instability in the training process, as each layer must consistently adjust to the evolving input distribution. Such instability can hinder the convergence of the model, thereby reducing performance. Figure 2.3 presents an overview of normalization techniques.

Batch Normalization (BN) [58] operates by gathering all data points within a batch and normalizing them to have a common mean and standard deviation. This technique enhances robustness during training to variations in the scale and shift of the input data, and it is also robust to the scale of the weight vectors. However, BN is only stable when the batch size is sufficiently large; smaller batch sizes can lead to inaccurate estimates of the mean and standard deviation, resulting in decreased performance. Additionally, the dependence on mini-batch statistics means that the behavior during training and inference can differ, potentially causing inconsistencies. Consequently, BN is not well-suited for recurrent networks or other sequential models where batch sizes are often small or variable, and where the sequential dependencies make batch-wise normalization challenging.

Layer Normalization (LN) [77] normalizes the inputs across the features instead of across the batch dimension. This feature-wise normalization makes LN particularly effective for small mini-batch sizes and recurrent neural networks (RNNs), especially when processing sequential data. In these contexts, the statistical properties of the batch are less informative due to the small size or sequential dependencies, and LN provides a more stable normalization. However, LN may not perform as well with CNNs, where BN often yields better results by leveraging batch-level statistics to improve learning and generalization.

Instance Normalization (IN) [151] focuses exclusively on individual feature maps, normalizing each pixel map independently. This method is particularly effective for style transfer tasks, as it allows each instance to retain unique style characteristics without being influenced by batch-level statistics.

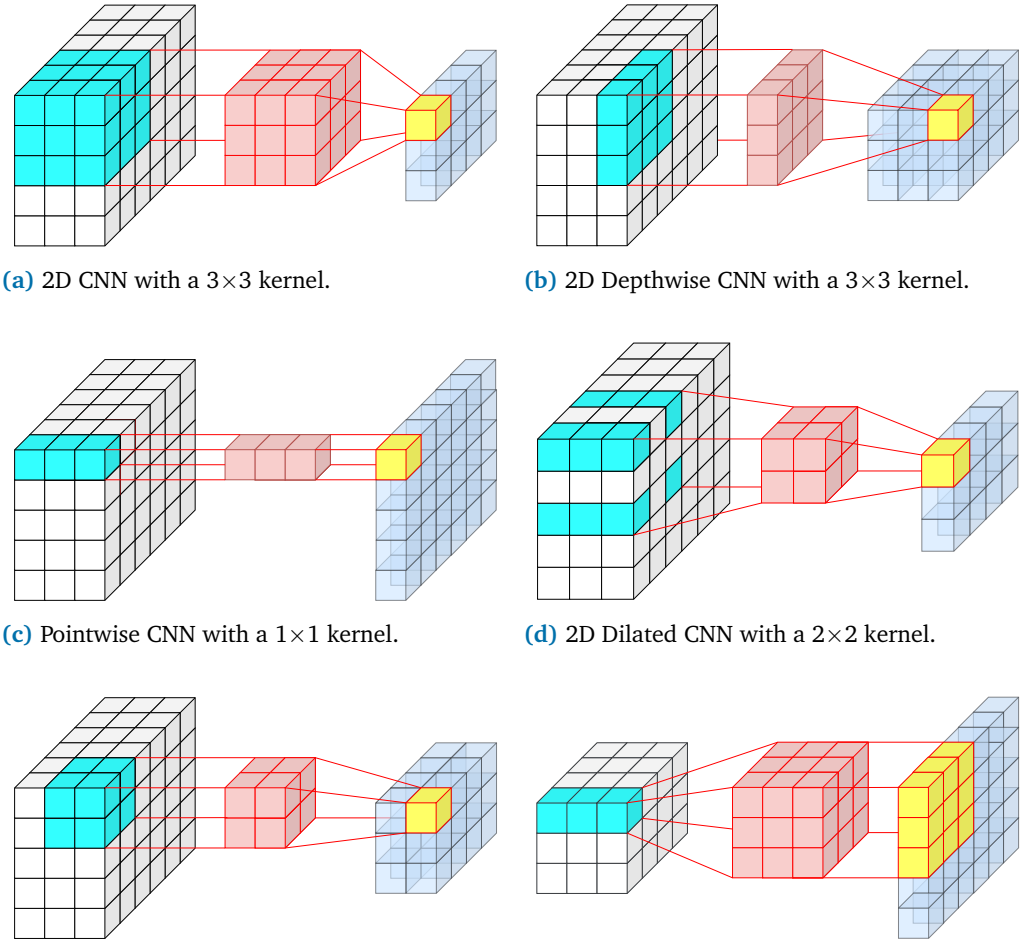
Group Normalization (GN) [164] serves as a trade-off between Instance Normalization and Layer Normalization. By dividing the channels of each layer into groups and computing the mean and variance within each group, GN preserves spatial dependencies more effectively than LN, making it better suited for CNNs. GN operates independently of the batch size by normalizing along the channel dimension, which makes it particularly effective when training with small or variable batch sizes. In many cases, GN performs comparably to BN for CNNs, providing a viable alternative, especially when batch sizes are small or variable.

Weight Normalization (WN) [130] operates differently from methods like Batch Normalization (BN). Instead of normalizing the activations, WN normalizes the weights of the neural network layers. By reparameterizing the weight vectors to have a fixed Euclidean norm, it decouples the length of the weight vectors from their direction. This approach makes WN computationally cheaper than BN because it avoids the need to compute statistics over mini-batches during training. In terms of training error, WN performs similarly to BN; however, it often yields worse testing error. This discrepancy suggests that while WN can facilitate optimization during training, it may not generalize as well to unseen data compared to BN.

Incorporating these normalization strategies can significantly enhance a model's ability to generalize to new, unseen data, ultimately leading to improved performance in practical applications. Normalization remains a critical aspect of model training in machine learning, as it aligns the model's complexity with the available data while mitigating the risks associated with overfitting.

2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) serve as fundamental building blocks in the creation of neural networks for image analysis tasks. The spatial bias inherent in CNNs is a crucial factor contributing to their effectiveness, as it enables the network to capture and exploit spatial hierarchies in visual data. CNN architectures are often combined with pooling layers, activation functions, and normalization layers.



(a) 2D CNN with a 3×3 kernel.

(b) 2D Depthwise CNN with a 3×3 kernel.

(c) Pointwise CNN with a 1×1 kernel.

(d) 2D Dilated CNN with a 2×2 kernel.

(e) 3D CNN with a 2×2 kernel.

(f) 2D Transposed CNN 3×3 kernel.

Figure 2.4: Convolutional Layers. This figure illustrates the fundamental architecture of convolutional layers within a CNN. All layers depicted are configured without padding and employ a stride of one. The white, red, and blue cubes symbolize the inputs, weight matrices, and outputs, respectively.

Figure 2.4 illustrates the basic structure of convolutional layers. The core concept of a convolutional layer is that a kernel slides over the image, performing element-wise multiplication with corresponding input values and aggregating the products to produce a single output value at each step. The kernel values, known as weights, are learnable parameters and are implemented in parallel structures referred to as filters. This operation, called the convolution operation, results in a feature map emphasizing specific patterns or features within the image, such as edges, textures, or particular shapes, depending on the used kernel. CNNs are usually configured by a stride, the step size of the sliding window, supplementary padding operations, kernel size, the size of the window, and further settings like dilation. Dilated convolutions have a larger ERF, but slightly worse performance in capturing narrow

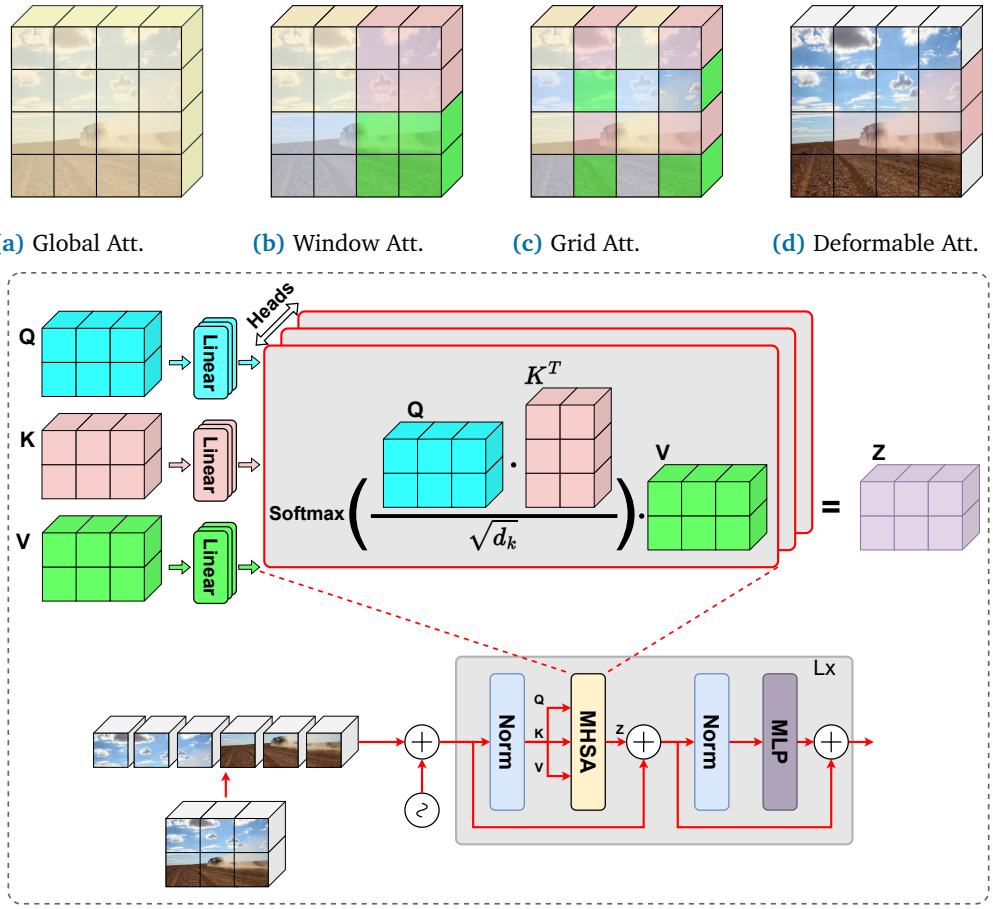
correlations between pixels. Furthermore, convolutional layers can be transposed and used to learn upsampling operations. This enables better upsampling at a higher computational cost than traditional methods like bilinear upsampling. Usually, all channels are processed through a convolutional kernel, but in the case of a 2D depthwise convolutional kernel, each input channel is processed separately by its own kernel. In combination with a pointwise CNN layer, which has a kernel size of 1×1, this results in a depthwise separable convolutional layer [55]. These require fewer weights and fewer computations, resulting in significantly enhanced efficiency. However, they may lose accuracy as a result. Other notable convolutional layers are dynamic kernels [175] and dynamic convolutions [19].

2.5 Vision Transformer

CNN architectures are characterized by a strong inductive spatial bias, while transformer architectures, in comparison, exhibit less. This difference typically makes transformers more difficult to train and necessitates larger datasets; however, it also allows for enhanced scalability. Furthermore, transformers exhibit a notably large Effective Receptive Field (ERF), particularly when employing global attention mechanisms. This characteristic allows them to capture long-range dependencies within the data more effectively. Hybrid models that integrate both methodologies present a trade-off. However, pure transformers can achieve performance comparable to hybrid architectures or higher when a substantial number of training samples are available.

Figure 2.5(e) presents the basic idea behind ViT. An input image is first tokenized, meaning that multiple pixels are converted into a token, which is then by default processed by a 2D CNN layer by default and subsequently flattened. These patch embeddings are further supplemented with positional embeddings. This addition is necessary because the resulting vectors are fed into the ViT as a sequence. Transformers are, by nature, invariant to the order of the input; thus, additional positional information must be modulated onto the patch embeddings.

In the original ViT [32], fixed sinusoidal positional embeddings are utilized. These embeddings do not require learned parameters and can handle sequences longer than those seen during training; however, they encounter challenges with more complex positional dependencies compared to learned positional embeddings. Learned embeddings can mitigate this issue but come with a fixed maximum length and are computationally expensive.



(a) Global Att. **(b)** Window Att. **(c)** Grid Att. **(d)** Deformable Att.

(e) Core principle of a Vision Transformer.

Figure 2.5: Vision Transformer. The Vision Transformer (ViT) [32] describes a methodology that involves partitioning an image into multiple patches, transforming these patches into tokens through linear embedding, and subsequently computing self-attention among them. Notably, ViT utilizes only the Transformer encoder. Initially, ViT employs global attention, which allows each token to attend to all other tokens across the entire image. Alternatives to global attention include window attention [90], grid attention [149], and deformable attention [181].

Relative positional embeddings [133] operate differently, as they aim to encode the relative distances between tokens rather than their absolute positions. These can be more effective for varying context lengths, but they introduce additional computational overhead. Newer methods, such as Rotary Positional Embeddings (RoPE) [142], enable longer contexts but are more challenging to implement and require fine-tuning.

ViT is based on the transformer encoder, which employs self-attention mechanisms. The architecture of ViT comprises multiple blocks, each consisting of several layers.

An overview of a layer is presented in Figure 2.5e. The patch embeddings are processed through a Layer Normalization (LN), after which the patches are split into three matrices: Query Q, Key K, and Value V. Each vector is passed through a linear layer and then fed into a Multi-Headed Self Attention MHSA module.

In this module, the vectors are distributed across multiple heads, analogous to the filters in CNNs. For each head, Q is first multiplied by K^T, yielding the attention scores A. The attention scores A are then normalized by dividing them by the square root of the dimensionality of K $\sqrt{d_k}$. Subsequently, a softmax operation is applied to the normalized A, which is then multiplied by the vector V. The results from all heads are concatenated and added to the identity of the layer input.

This vector is normalized by a Layer Normalization (LN) and passed through a MLP, after which it is summed with its identity. The output of this layer can then be utilized as the input for the subsequent layer.

When all tokens are utilized for self-attention, this is referred to as global attention. This approach offers the advantage of a vast effective receptive field (ERF). However, the computational requirements increase quadratically with image size, making it less feasible for large images. To address this issue, Liu et al. proposed shifted window attention [90]. This method calculates attention only over a portion of the image. In the case of shifted window attention, the attention windows are shifted to incorporate all neighboring relationships.

This concept is further enhanced by integrating grid attention [149] with window attention. Grid attention operates similarly to dilated CNN layers, providing a sparse attention mechanism across the entire image. This approach enables the capture of both local and global correlations. Additionally, deformable attention [181], introduced for DETR, significantly improves training efficiency. In this framework, the sparse attention mechanism can focus on relevant tokens by utilizing reference points, significantly enhancing the model's convergence time.

3 Dust Density Estimation

In the following, quotations to related publications are highlighted by colored bars placed on the outer border of the text, even though minor editorial changes have been made. The color encoding used is as follows:

- **Michel, A.**, Mispelhorn, J., Schenkel, F., Gross, W. & Middelmann, W. (2020, September). An approach to improve detection in scenes with varying object densities in remote sensing. In Image and Signal Processing for Remote Sensing XXVI (Vol. 11533, pp. 107-113). SPIE. Reprinted with permission. It is cited as [98]. ■
- **Michel, A.**, Weinmann, M., Schenkel, F., Gomez, T., Falvey, M., Schmitz, R., Middelmann, W. & Hinz, S. (2023, July). Terrestrial visual dust density estimation based on deep learning. In IGARSS 2023 – 2023 IEEE International Geoscience and Remote Sensing Symposium (pp. 4923-4926). IEEE. Reprinted with permission. It is cited as [104]. ■
- **Michel, A.**, Weinmann, M., Schenkel, F., Gomez, T., Falvey, M., Schmitz, R., Middelmann, W. & Hinz, S. (2023, September). DustNet: Attention to Dust. In DAGM German Conference on Pattern Recognition (pp. 211-226). Cham: Springer Nature Switzerland. Reprinted with permission. It is cited as [101]. ■
- **Michel, A.**, Weinmann, M., Kuester, J., AlNasser, F. , Gomez, T., Falvey, M., Schmitz, R., Middelmann, W. & Hinz, S. (2025, February). DustNet++: Deep Learning-Based Visual Regression for Dust Density Estimation. In International Journal of Computer Vision (pp. 1-25). Springer. Reprinted with permission. It is cited as [100]. ■

This chapter provides a comprehensive overview of the research efforts dedicated to the development of density estimation algorithms throughout the course of this thesis, utilizing the specific example of dust monitoring as a focal point. As illustrated in Figure 3.1, which visually represents the development process, the initial sections of the chapter will delve into the problem statement, elucidating

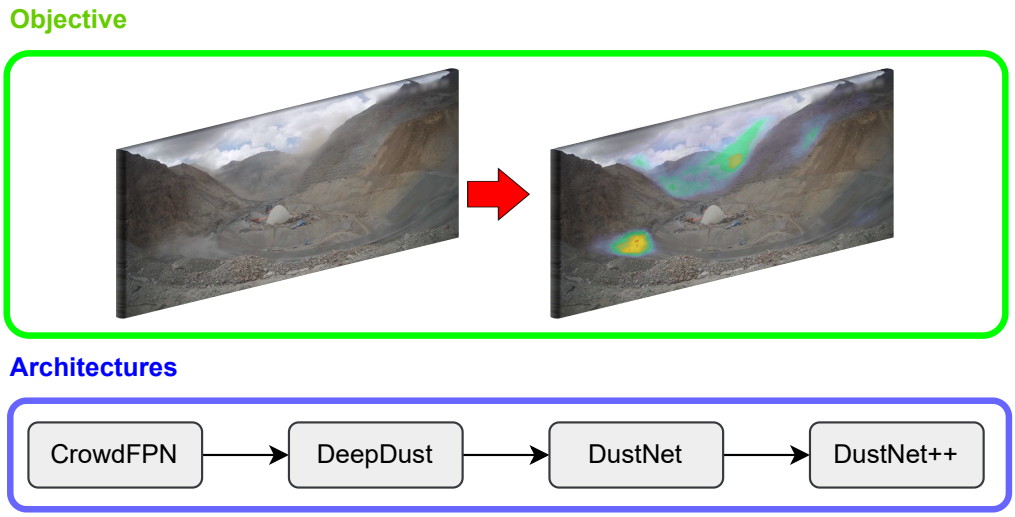


Figure 3.1: Topics of Chapter 3. This figure details the process of developing and enhancing neural network architectures specifically for dust density estimation. The primary focus includes articulating the motivation behind this research and presenting the objectives pursued. Furthermore, the chapter traces the evolutionary trajectory of the various approaches developed and employed throughout this study. An extensive evaluation is conducted using two distinct dust datasets, providing a comprehensive analysis of the models' performance and effectiveness in addressing the challenges associated with dust density estimation.

the complexities and challenges associated with monitoring dust concentrations in various environments. Moreover, it will articulate the motivation underlying this research endeavor, highlighting the significance of effective dust monitoring for public health, environmental protection, and regulatory compliance. Recognizing that the problem of dust monitoring has not been thoroughly explored in the existing literature, this discussion will extend to encompass related issues that employ pixel-wise regression techniques. Specifically, the chapter will address pertinent topics such as crowd counting and monocular depth estimation. These areas serve as illustrative examples of how density estimation can be applied in different contexts, thereby providing a broader understanding of the methodologies involved. Subsequently, the chapter will trace the evolution of various architectural frameworks, beginning with CrowdFPN and advancing to DustNet++, while also including intermediary developments such as DeepDust and DustNet. This historical perspective will not only delineate the advancements in algorithmic design but will also underscore the iterative nature of research and development in this field. Once the datasets utilized for training and testing the models, as well as the configurations of these models, have been thoroughly presented, the chapter will proceed to offer a comprehensive overview of the experimental results obtained from the research. This analysis

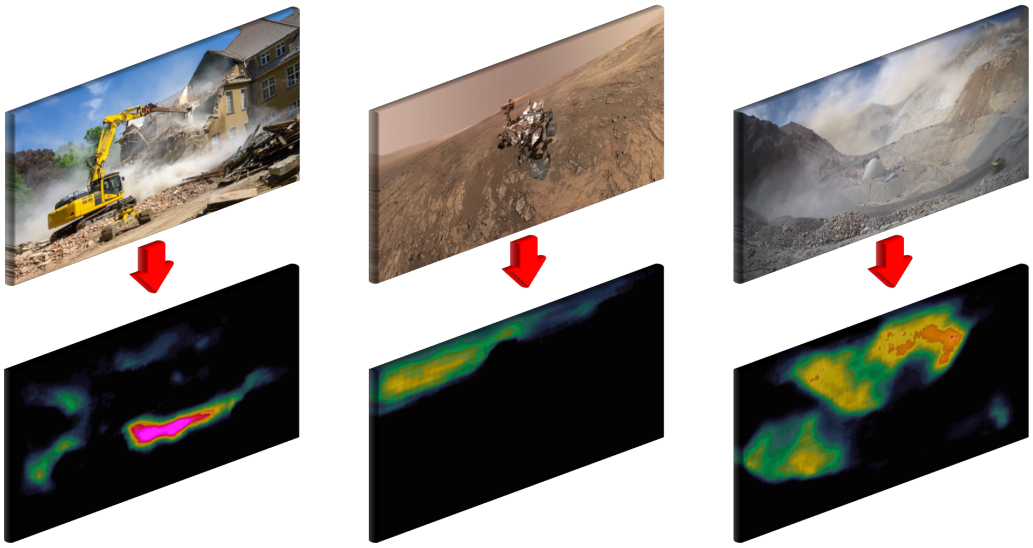


Figure 3.2: Objective. Our methods aim to estimate the level of dust in an RGB image or sequence. The images on the left display a construction site [42], the central image depicts a scene from the surface of Mars [108], and the image on the right illustrates an opencast mine. For each image, our DustNet++ model predicts a corresponding dust density map. It is important to note that the goal of our model is not to achieve physically accurate dust estimation, but rather to mimic a human observer who describes the dust levels in a given scene.

will include quantitative and qualitative assessments of the models' performance, providing insights into their efficacy in dust monitoring applications. Finally, this chapter will conclude with a critical discussion of the findings, synthesizing the insights gleaned from the experimental results and placing them within the broader context of the field. The concluding section will summarize the key conclusions drawn from the research, presenting their implications for future work and potential applications in environmental monitoring and beyond.

3.1 Motivation

Monitoring airborne dust emissions is an essential endeavor due to its profound impact on climate, human health, infrastructure, buildings, and various socio-economic sectors [146, 65, 51, 78]. The source of airborne dust particles can stem from natural phenomena such as strong winds, wildfires, and seismic activities, as well as from human activities [172, 136, 65]. Predominant anthropogenic sources include construction sites, vehicular traffic, and mining operations. While the complete eradication of dust emissions is unattainable, targeted suppression measures can be implemented. These measures might include but are not limited to watering

untreated roads, decelerating vehicles, or curtailing mining activities. However, effective and economical monitoring of dust emissions is imperative to enhance dust mitigation strategies. Conventional instrument-based *in-situ* monitoring equipment does not focus on identifying emissions but on concentrations of particles.

On the other hand, while insightful, advanced remote sensing technologies such as 3D LiDAR scanning are not economically viable on a large scale and often yield noisy, ambiguous data, particularly in complex terrains. Consequently, visual monitoring through camera-based systems stands out as a more feasible option for the detection of airborne dust emissions. Nonetheless, the field of visual dust density estimation remains significantly under-explored. The scarcity of research in this domain is largely attributable to the intricate challenge of detecting dust in images, a task that is highly ill-posed. Numerous factors contribute to the complexity of dust detection; dust can vary greatly in density and can appear both opaque and translucent. The variation in dust density can be imbalanced, with dense dust from dust storms occurring less frequently in arid regions than the more transparent dust brought about by mild winds, often manifesting sporadically during specific meteorological conditions. Additionally, dust can originate from a multitude of locations and due to various factors. The transparency of dust means that its visual appearance is heavily influenced by environmental conditions, leading to indistinct boundaries in images. Consequently, images depicting dust often appear partially blurred and typically exhibit low spatial contrast. Classical algorithms struggle to capitalize on these partial blur effects because similar effects are also produced by other atmospheric phenomena, such as fog or clouds. Furthermore, while human observers may find it easier to identify dust across a sequence of images, algorithmically harnessing temporal data to improve detection accuracy poses a significant challenge. Moreover, the absence of a consistent color scheme for dust complicates its detection based on visual cues alone. For instance, opaque dust may exhibit a brownish hue during a dust storm or appear black in the aftermath of a mining explosion. Collectively, these characteristics underscore the necessity for a more sophisticated approach to detect and monitor airborne dust emissions effectively.

In the last decade, deep learning has had huge success in various tasks like classification [69], object detection [120], neural linguistic processing [152], and remote sensing [180]. However, airborne dust monitoring, obstructed by the aforementioned challenges, is not well researched, and most scientific papers focus on satellite images [74], or related tasks like smoke binary segmentation [173].

Recently, De Silva et al. published the Unsealed Roads Dust Emissions (URDE) dataset [29] representing a binary dust segmentation dataset. While this can be

seen as a first important step towards dust monitoring, we believe that a regression approach could be more beneficial. In contrast to semantic segmentation, which aims to predict labels on a per-pixel basis, the continuous range of dust densities rather suits a regression strategy. Furthermore, the vague boundaries of dust make it challenging to create discrete hard labels.

Accordingly, this research centers on a novel task: dust density estimation, as illustrated in Figure 3.2. The primary objective of our proposed model is to estimate the level of dust present in an RGB image or a sequence of images on a per-pixel basis. It is crucial to emphasize that the aim of our model is not to achieve a physically precise quantification of dust levels; rather, it seeks to emulate a human observer's description of dust concentrations within a given scene. This approach allows for a more intuitive understanding of dust presence, aligning our model's output with human perceptual experiences.

To achieve this objective, we outline the evolution of various architectural frameworks, beginning with CrowdFPN and advancing to DustNet++, while also highlighting intermediary developments such as DeepDust and DustNet. This historical overview will not only clarify the advancements made in algorithmic design but also emphasize the iterative process inherent in research and development within this domain. In order to validate the effectiveness of our approach, we compare the results achieved to those of visual density estimation techniques originating from other domains, including monocular depth estimation (MDE) and crowd counting. MDE aims at estimating the scene depth on a per-pixel basis, whereas crowd counting is the task of approximating the number of people in a given image. Though both tasks differ strongly from dust density estimation, our method is heavily influenced by ideas of both domains.

In summary, the key contributions of this chapter are as follows:

1. We investigate the relatively unexplored area of airborne dust density estimation and propose several neural network architectures.

2. We introduce CrowdFPN [98], which leverages multiscale feature maps by employing FPN structures to estimate density maps.

3. We present DeepDust [103], which builds upon the principles of CrowdFPN while enhancing feature flow and incorporating new convolutional structures.

4. We introduce DustNet [101], which integrates attention-based and convolutional FPN structures to effectively merge local and global features. Addition-

ally, DustNet C addresses the fusion of temporal features within the context of dust density estimation.

5. We unveil a novel architectural model, DustNet++ [100], characterized by a more streamlined design that outperforms its predecessor presented in the initial conference version. DustNet++ utilizes an innovative cross-multi-axis feature pyramid network that facilitates interactions across different resolutions and semantic levels of feature maps, while maintaining both global and local interactions within a single feature map.

6. To demonstrate the efficacy of our proposed neural network architectures, we compare their performance with existing methods from the crowd counting and MDE domains using the Meteodata dust dataset.

7. To assess the generalization capability of our approach, we provide a quantitative analysis utilizing the URDE dataset.

3.2 Related Work

This section focuses on methodologies from other domains that have inspired the approach to dust density estimation. Given that dust density estimation is an area that has been under-explored, we have drawn inspiration from established techniques in adjacent fields, specifically crowd counting and monocular depth estimation. These domains offer valuable insights and methodologies that can be adapted to enhance our understanding and capabilities in estimating dust density. We will pay particular attention to the PixelFormer [2], which has played a crucial role as the foundational model for DustNet.

3.2.1 Crowd Counting

Density estimation methods have been used successfully in crowd counting [177, 131, 81, 88, 93]. The objective of crowd counting is to predict a coarse density map of the relevant target objects, e.g. people. The ground truth is generated by smoothing center points with a multi-dimensional Gaussian distribution. Recent approaches are focused on increasing the spatial invariance [93] or dealing with noise in the density maps [22]. Most works are designed for individual images, but Avvenuti et al. [5] take advantage of the temporal correlation between consecutive frames in order to reduce localization and count error. CrowdFPN [98] was significantly

inspired by CanNet [88]. CanNet introduces an end-to-end trainable architecture that effectively integrates multi-scale features without requiring explicitly defined patches. Instead of relying on predetermined structures, it learns to assign weights to these features on a per-pixel basis, facilitating a robust adaptation to rapid scale variations. Additionally, the implementation of multi-scale pooling operations enables the model to encompass an arbitrarily large range of receptive fields. This capability allows CanNet to consider a significantly broader context than is possible with the multiple receptive fields utilized in prior approaches.

3.2.2 Monocular Depth Estimation

The first CNN-based method for monocular depth estimation was presented by Eigen et al. [35]. They utilized global and local information in order to predict depth images from a single image. Further improvements of the pure CNN approaches focus on Laplacian pyramids [139], multi-scale convolutional fusion [155], structural information [75] or the exploitation of coplanar pixels [112] to improve the predicted depth. Recently, hybrids between CNN and Vision Transformer [32] based architectures improved the depth estimation process. Ranftl et al. [116] proposed to pass features from a CNN-based ResNet [48] extractor to a Vision Transformer to capture global information. Furthermore, the NeWCRFs [174] approach utilizes window-based Vision Transformers [90]. Fu et al. [41] introduced the depth prediction task as a classification–regression problem. Hereby, the classification part consists of predicting discretized bin centers, and the regression part utilizes the bin centers to produce high-quality depth maps. This approach was improved by Bhat et al. [7] by predicting adaptive bins. The PixelFormer architecture [2] combines transformer architectures with the bin center approach and adds skip connections modules to improve the feature flow between different encoder feature levels.

3.2.3 Revisiting PixelFormer

Similar to NeWCRFs [174], PixelFormer architecture utilizes a Swin transformer [90] as its backbone. Features extracted from the coarsest resolution are directed to a pixel query initializer (PQI) module, which is an adapted UperNet head [167]. The primary function of the PQI module is to aggregate global information through pyramid spatial pooling [49]. The resulting features are processed through two branches: one for predicting bin centers and the other for estimating the pixel-wise probability distribution over these bin centers. The branch responsible for predicting

bin centers comprises a bin center predictor (BCP) neural network, which employs global average pooling followed by a multilayer perceptron to generate the bin centers. These bin centers are represented as a one-dimensional vector, with the vector's length corresponding to the number of adaptive bins allocated per image. To compute the probability scores on a pixel-by-pixel basis, the aggregated features from the PQI are combined with the coarsest features from the backbone and passed to a skip attention module (SAM). The SAM operates similarly to a Swin transformer block [90], incorporating window cross-attention. However, unlike traditional configurations where the query, key, and value matrices originate from the same input feature map, the SAM shares the same input for the key and value matrices while utilizing a different input for the query matrix. Specifically, the first query matrix is derived from upsampled features generated by the PQI neural network, while the key and value matrices are based on the features from the coarsest level of the backbone. For subsequent SAM blocks, the query matrix is computed from the upsampled output of the preceding SAM block, and the second coarsest layer from the backbone is fed into the SAM. This process is iteratively repeated for each layer of the backbone. After the fourth SAM block, the output is processed through a convolutional layer followed by a softmax function. Ultimately, a linear combination of the resulting probability scores with the bin centers is used to produce depth maps.

3.3 CrowdFPN

This section delineates the foundational architecture of our proposed CrowdFPN presented in [98]. CrowdFPN is inspired by CanNet [88], integrating similar context-aware features. However, unlike CanNet, which utilizes the VGG architecture—now considered somewhat outdated—CrowdFPN adopts a more contemporary backbone structure, specifically leveraging ResNet. A significant innovation of CrowdFPN is its shift away from average pooling for generating feature maps across various scales. Instead, it employs the diverse feature maps produced by the backbone, which are processed through a FPN. This approach not only enhances the adaptability of the architecture but also ensures a more future-proof design capable of accommodating ongoing advancements in network architecture and feature extraction techniques.

3.3.1 Architecture Overview

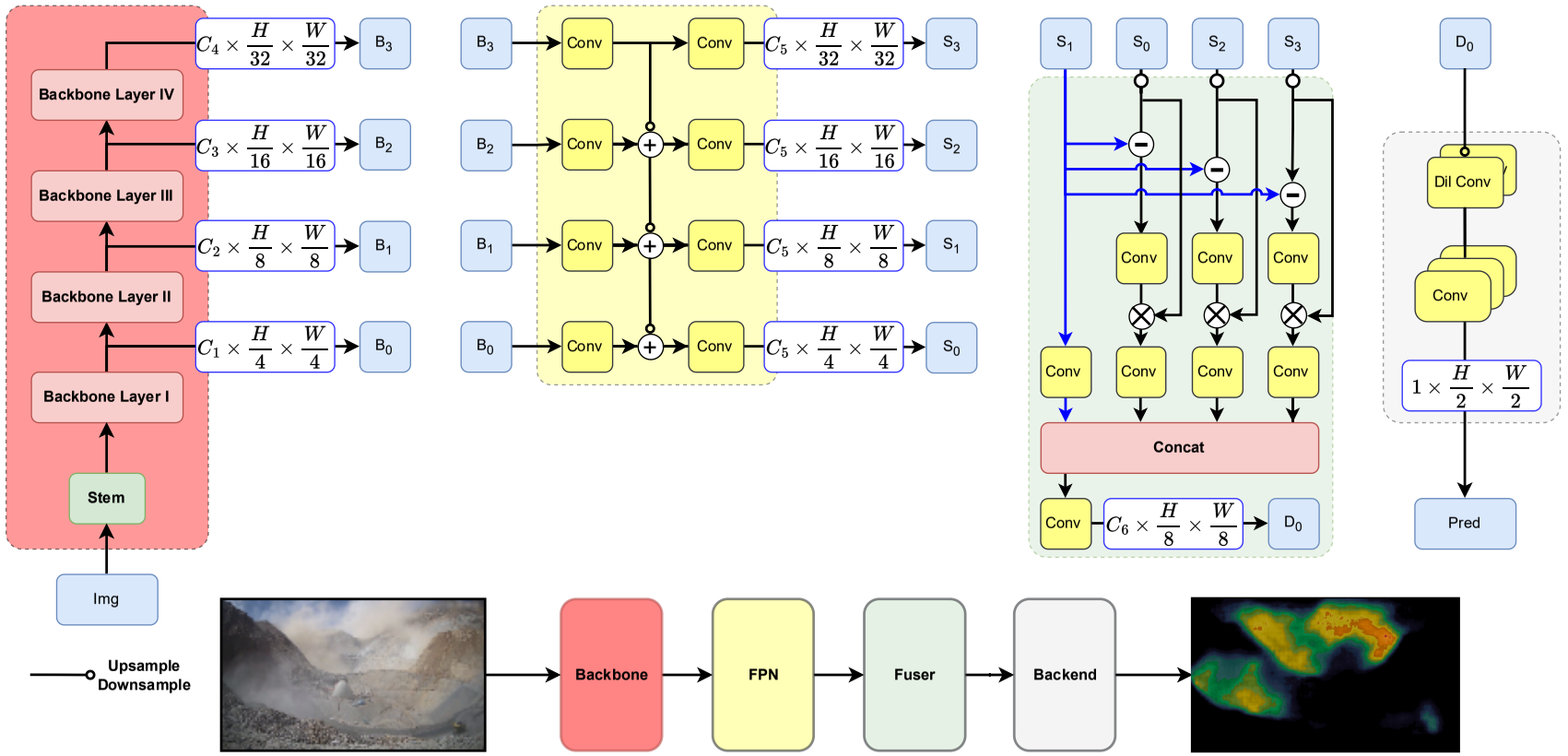


Figure 3.3: Architecture of CrowdFPN. The fundamental components of CrowdFPN consist of a backbone, a feature pyramid network (FPN), a fuser module, and a backend. These elements collaboratively facilitate the processing of large images to generate dust density maps.

CrowdFPN, as depicted in Figure 3.3, is meticulously designed to process large images, ultimately yielding continuous dust density maps. The initial phase involves the backbone architecture, which transforms these images into suitable embeddings tailored for the downstream task of dust density estimation. In the subsequent stage, the embeddings B_0, B_1, B_2, B_3 corresponding to image size of $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ are concurrently input into the FPN. This integration significantly enhances the flow of features between the layers, enabling the model to effectively leverage multi-scale information. The FPN generates four embeddings S_0, S_1, S_2, S_3 that maintain uniform spatial dimensions, facilitating seamless integration in the subsequent stages of the architecture. These embeddings are then directed to a fuser module, where they undergo merging and concatenation. This crucial step combines rich contextual information from various scales, leading to a more comprehensive representation of the input data. Ultimately, the backbone utilizes these combined embeddings to predict the dust density map. The final output demonstrates the model's capability to integrate and process features from different stages of the processing pipeline, resulting in an accurate and continuous representation of dust density across the analyzed images. The architecture's design underscores the importance of multi-scale feature integration, allowing CrowdFPN to effectively capture the complexities inherent in dust density estimation tasks.

3.3.2 CrowdFPN Modules

Fuser. The fuser module receives embeddings S_0, S_1, S_2, S_3 from the FPN, with S_1 serving as the reference embedding. The other branches are upscaled and downscaled to match the image size of S_1, corresponding to a size of $\frac{1}{8}$ of the original images. The reference embedding is subtracted from the remaining embeddings, which are then processed through a convolution module that consists of a 2D convolution layer, a batch normalization layer, and a sigmoid activation function. The resulting embeddings are element-wise multiplied by their respective residuals. The same convolution module processes the reference embedding, and finally, all feature maps are concatenated to form the embedding D_0.

Backend. The backend architecture includes a squeeze and excitation block, several conventional and dilated convolution modules, and concludes with a convolution layer that predicts the dust density map. The use of dilated CNN layers is justified to effectively enlarge the ERF. An overview of the various CNN layers can be seen in Figure 2.4. Each convolution module includes a sequence of a 2D convolution layer, batch normalization, and a ReLU activation function, ensuring effective feature extraction and representation throughout the model.

3.4 DeepDust

The core concept of our proposed algorithm, DeepDust presented in [104], is illustrated in Figure 3.4. DeepDust builds upon the foundation established by CrowdFPN, incorporating several architectural modifications. Notably, it replaces the Feature Pyramid Network (FPN) with a matcher structure that utilizes 2D transpose convolutional layers (Figure 2.4) and implements an alternative feature routing approach. Additionally, the number of branches has been reduced from four to three, while maintaining identical image resolutions across these branches. This reduction not only contributes to a smaller memory footprint but also facilitates the use of larger batch sizes during training. Furthermore, enhancements have been made to the backend module, optimizing its performance and contributing to the overall efficacy of the DeepDust architecture.

3.4.1 Architecture Overview

DeepDust utilizes high-resolution images to output continuous dust density maps. The images are first processed in the backbone to create suitable embeddings for the downstream task. In the next step, the embeddings of the different backbone blocks are simultaneously fed into the matcher. It improves the feature flow between the layers and outputs three embeddings of the same size. The features are then sent to the fuser and are merged and concatenated. Finally, the backbone predicts the density map using the formerly produced embeddings.

3.4.2 DeepDust Modules

Matcher. The matcher has an FPN-like [82] structure, enabling the features from highly semantic enriched features extracted from the upper backbone blocks to enrich low-level semantic features extracted from the lower backbone block. It is fed with embeddings from four backbone blocks with the resolution scales $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of the original images. Hereby, the embeddings extracted from the higher backbone blocks are upsampled by a transposed convolution module and respectively concatenated with the embeddings from the lower backbone block to three combined embeddings. The transposed convolution module consists of a sequence of a 2D transposed convolution layer, batch normalization, and ReLU activation function. Each embedding is then resized to a scale of $\frac{1}{8}$ of the original image and processed by a further convolution module. The convolution module includes a 2D convolution

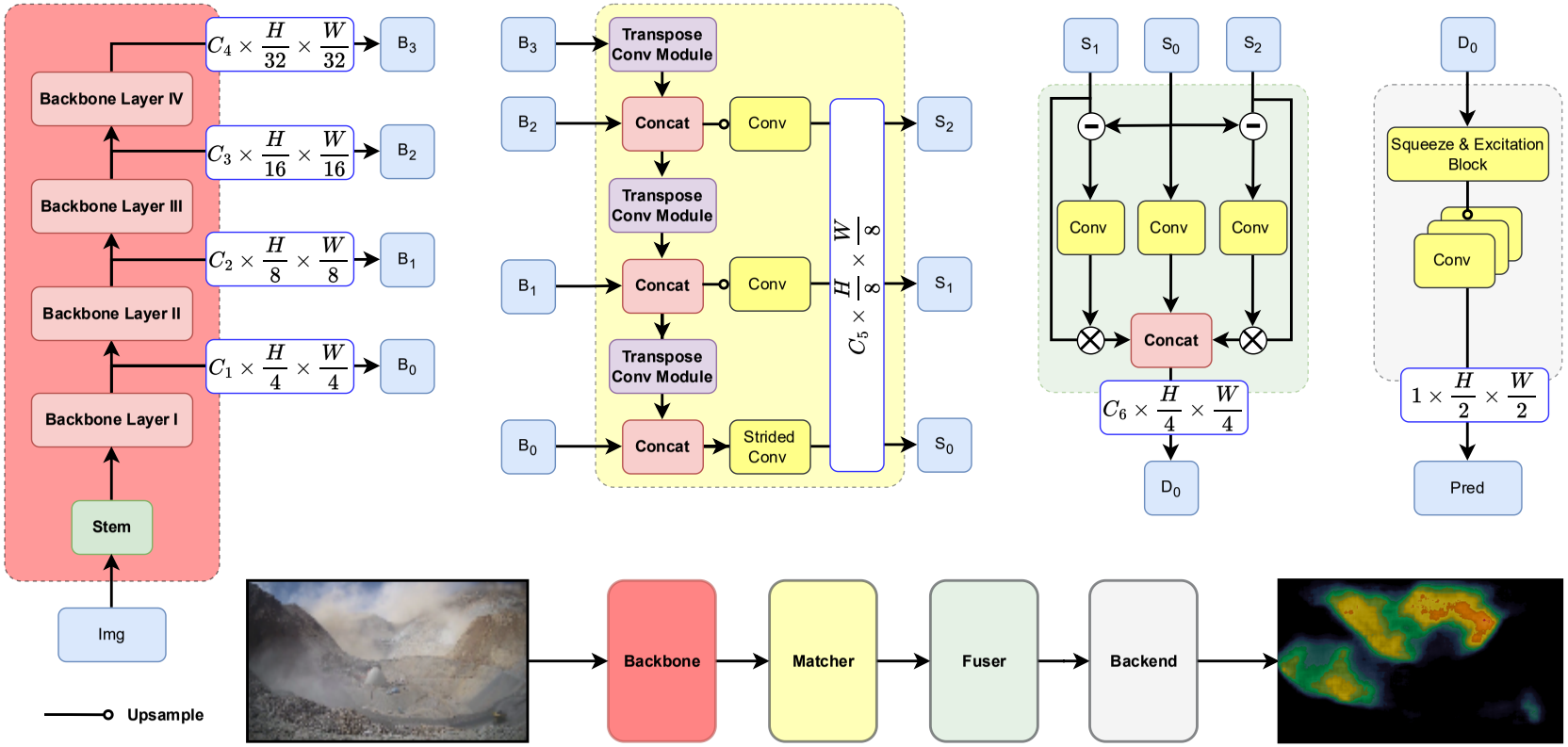


Figure 3.4: Detailed Overview of our proposed DeepDust. The backbone extracts features from a given image, which are then fed into the matcher. The task of the matcher is to improve the feature flow and match all embeddings to the same size. The resulting embeddings are merged in the fuser and finally processed to a dust density map in the backend.

layer, batch normalization, and ReLU activation function. After the resizing and the convolution module, each embedding S_0, S_1, S_2 possesses the same dimensions.

Fuser. The fuser utilizes the embedding S_0 as the reference. The reference is subtracted from the remaining embeddings and then processed by a convolution module consisting of a 2D convolution module, a batch normalization layer, and a sigmoid activation function. The resulting embeddings are then multiplied element-wise with their residuals. The same type of convolution module also processes the reference embedding. Then, all feature maps are concatenated to the embedding D_0.

Backend. The backend consists of a squeeze and excitation block, multiple convolution modules, and finally, a convolution layer that predicts the dust density map. The convolution module includes a sequence of a 2D convolution layer, batch normalization, and ReLU activation function.

3.5 DustNet

In the following, the proposed DustNet architecture [101] is introduced, as illustrated in Figure 3.5. After presenting the submodules, attention is directed towards the various temporal fusion approaches, depicted in Figure 3.6.

3.5.1 Overview of the Network Structure of DustNet

DustNet processes input image sequences X of the dimensions $T \times H \times W \times 3$ to a continuous dust density map of dimensions $\frac{H}{2} \times \frac{W}{2} \times 1$. T may consist of a maximum of three consecutive images, where the target y is assigned to the current image x_{t^0}. The images are fed into the backbone, which produces multiple feature maps with decreasing resolution and ascending information aggregation. The backbone features are passed to a Pyramid Pooling Module (PPM) [178] and the Attention Feature Pyramid Network (AFPN). Hereby, in order to reduce the computational complexity, only half of the channels of feature maps are transferred to the AFPN. The PPM head aggregates global information fed into the AFPN and the fuser module. The AFPN mixes the feature maps of different resolutions and information aggregation levels. The processed feature maps are transferred to the matcher module, accumulating the features maps into one high-resolution map. Then, the high-resolution features are merged with the global information features aggregated from the PPM head in the fuser module. Eventually, the combined

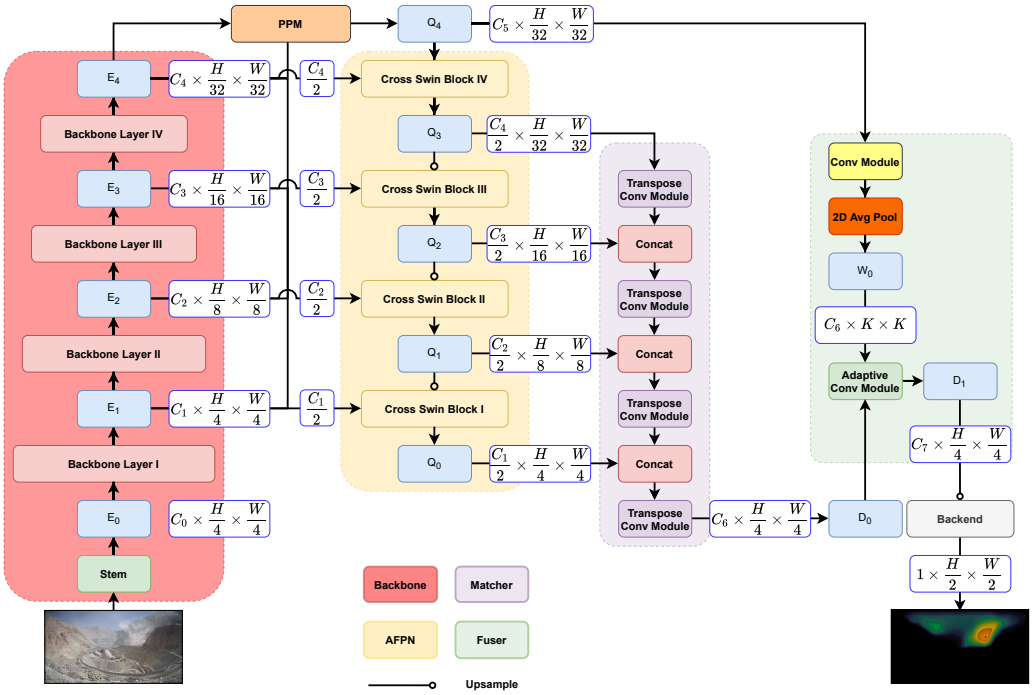


Figure 3.5: Overview of DustNet. The basic blocks are a backbone, the AFPN, the matcher, the PPM, the fuser, and the backend. Given an input image, a CNN-based encoder neural network extracts multiple feature maps in different scales. The features are fed into the PPM head to extract global features, and into the AFPN. The objective of the AFPN is to calculate the window cross-attention between the different feature maps. Then, the AFPN feature maps are passed to the matcher module, which combines the features by concatenating and upscaling the features by transpose convolutions. Thereafter, the features are fed into an adaptive convolutional layer in the fuser module. In order to mix local high-resolution features with global low-resolution features, features from the PPM head are pooled and serve as the kernel weights of the adaptive CNN. Finally, a backend consisting of multiple sequences of CNNs, batch normalization, and activation functions followed by a CNN predicts the dust density map.

features are processed by the backend, which consists of multiple sequences of CNNs, into a dust map.

3.5.2 DustNet Modules

Backbone. The backbone consists of a stem module and four blocks. We use this common backbone scheme in order to leverage pre-trained neural networks. We prefer a convolutional backbone like ResNet [48] instead of a transformer backbone due to the requirement to process high-resolution images. The backbone produces

multiple feature maps with the resolution scales $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of the original images with the number of channels C of $\{256, 512, 1024, 2048\}$.

Pyramid pooling module. We utilize a PPM head [178] like in [2, 174] to aggregate global information of the whole image. We use global average pooling of scales $\{1, 2, 3, 6\}$ to extract the information. After extracting the features, we concatenate them and process them by a convolutional layer to the feature map Q_4 with the dimension of $512 \times \frac{H}{32} \times \frac{W}{32}$.

Attention feature pyramid network. The AFPN mixes high-resolution features with low semantics with low-resolution high semantic features. Instead of a traditional FPN like [82] utilizing CNNs, we are inspired by [2] and use four Swin blocks with cross window attention to improve the feature flow between the feature map layers. However, instead of applying scaled dot attention, we utilize cosine attention similar to [89]. This modification enhances the attention mechanism's ability to scale effectively with large images. The key and value matrix inputs are derived from the backbone feature maps, but to reduce computational complexity, we transfer only half of the channels. The query matrix is filled by the output of the upsampled stage from the step before. The query matrix with the coarsest resolution originates from the global aggregated features of the PPM head.

Matcher. The matcher module also has an FPN-like architecture [82]. We upsample feature maps from the AFPN via a Transposed Convolution Module (TCM). It consists of a 2D transposed convolution with a stride and kernel size of two, followed by batch normalization [58] and a Sigmoid Linear Unit SiLU [36] activation function. Like [80] suggests, we apply only batch normalization without dropout [141]. The coarsest resolution feature map derived from the AFPN is fed to the first TCM block. The following AFPN feature maps are respectively concatenated to the output of the TCM block and processed via the next TCM block. The output of the matcher module has the dimension of $C_6 \times \frac{H}{4} \times \frac{W}{4}$.

Fuser. The fuser module processes the high-resolution features D_0 by leveraging the aggregated features Q_4 of the PPM head. Q_4 is fed into a 2D pointwise convolutional kernel, followed by a SiLU activation function, and pooled by global average pooling to a feature map W_0 of the dimension $C_6 \times K \times K$. W_0 serves as an adaptive kernel for the adaptive convolutional kernel layer [175], which enriches the feature map D_0 from the matcher with global information.

Backend. The backend consists of N blocks of a sequence of a 2D convolution layer, batch normalization, and SiLU activation functions that predict the dust density map. We split the features into two parallel blocks for each stage and accumulate

the outputs. Hereby, we choose a dilation of three for one branch to increase the receptive field. After four stages, a pointwise convolutional layer predicts the dust density maps.

3.5.3 Temporal Fusion

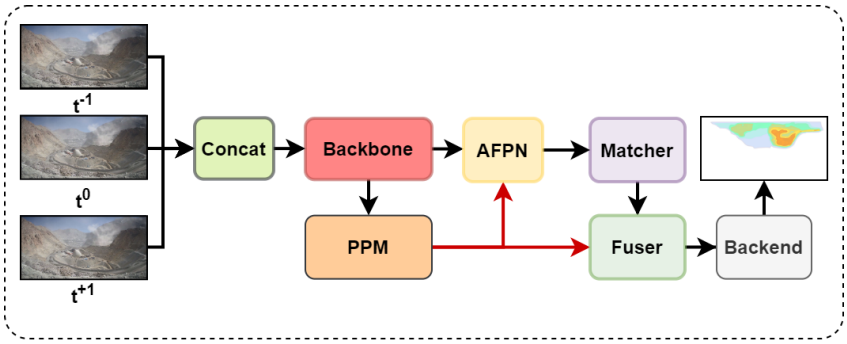
The preceding subsection focused exclusively on processing a single image, a variant referred to as DustNet S. To leverage the temporal information between consecutive images, several approaches were developed and investigated. Figure 3.6 illustrates the various fusion strategies, which range from early to late fusion.

Image Concatenate Image. An obvious way to concatenate the images to $D \times H \times W$, is where the product of the number of images T and the number of channels C is the new channel dimension D. This approach, called DustNet A, is illustrated in Figure 3.6a. Examples of this approach can be found in [20] and [91]. Hereby, the backbones are usually specifically adapted to 3D input.

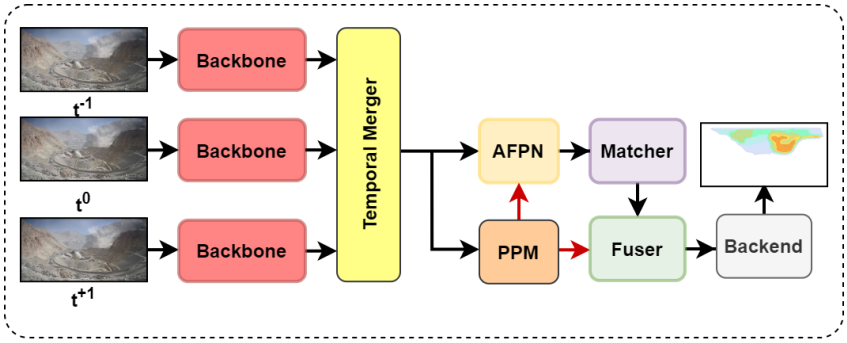
Early Multi-scale Feature Fusion. Figure 3.6b shows the methodology behind this fusion aspect. Features from three backbones, which share weights, are fed into the temporal merger (TM) neural network. TM subtracts the feature maps of images x_{t-1} and x_{t+1} respectively from x_{t^0} and multiplies the difference. We pass the new feature maps through a 2D pointwise convolutional layer and add skip connections from the feature maps of the image x_{t^0} to the output. This variant is described as DustNet B.

Late Multi-scale Feature Fusion. This approach, called DustNet C, represents a simple fusion of AFPN features (see Figure 3.6c). Backbone and AFPN weights are shared between the instances. The PPM head is only fed with the backbone features from image x_{t^0}. In order to reduce the computational complexity and avoid convergence problems, only two consecutive images may be used.

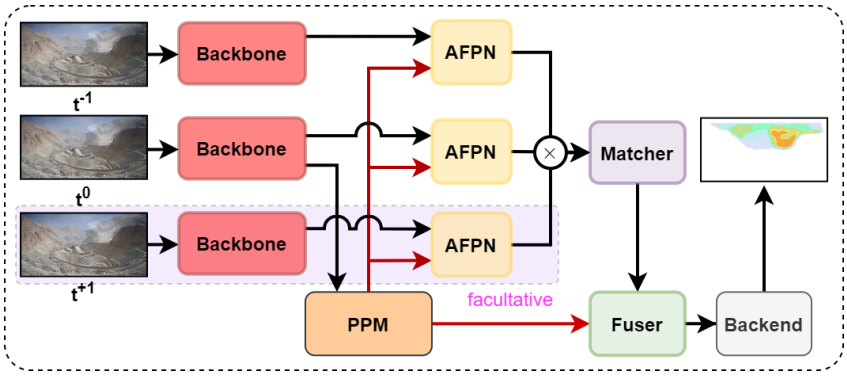
Adaptive Global Feature Fusion. The goal hereby is to calculate the global aggregated features from a PPM head for each image. Backbone and PPM weights are shared. The local feature branch is only fed with the multi-scale backbone features from image x_{t^0}. The fusion of the temporal information occurs in the fuser module. For each PPM head, an adaptive convolutional layer is added. This variant is called DustNet D.



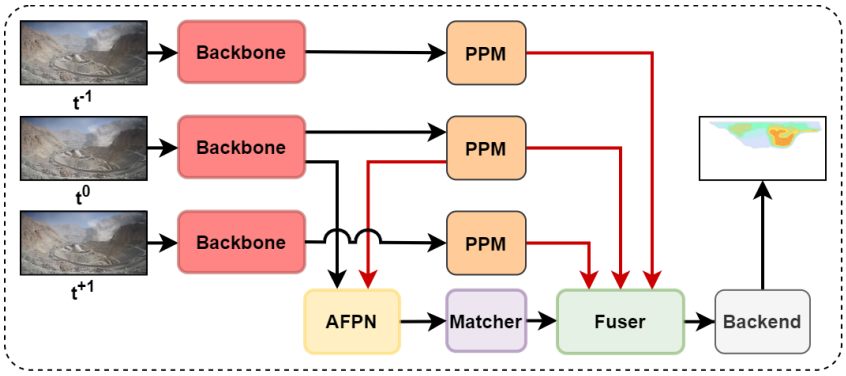
(a) Concatenation before processing.



(b) Early multi-scale feature fusion.



(c) Late multiscale feature fusion.



(d) Adaptive global feature fusion.

Figure 3.6: Different fusion approaches of DustNet: The fusion strategy spans from early (a) to late (d) feature merging of each consecutive image.

3.6 DustNet++

In the following, we introduce our proposed DustNet++ [100] illustrated in Figure 3.7. Our goal is to simplify the architecture of DustNet [102] while still facilitating the interaction between local and global features for high resolution images. In order to achieve this goal, inspired by MaxViT [149], we propose a Cross multi-axis Feature Pyramid Network (CmaxFPN). The basic blocks are a backbone, the CmaxFPN, a feature map matching module, and the backend.

3.6.1 From MaxViT to CmaxFPN

MaxViT [149] represents a scalable and efficient approach for computer vision applications by utilizing windowed local and dilated global attention. Let $X \in \mathbb{R}^{H \times W \times C}$ be an input feature map, which is processed by a stem consisting of a convolutional layer and multiple stages of sequences of MaxViT blocks. The basic MaxViT block consists of three modules: an MBConv [55] and a pair of self-attention blocks with each respectively using window and grid attention.

MBConv. The MBConv without downsampling can be described as follows:

$$X \leftarrow X + \text{Conv}_{\text{sh}}\left(\text{SE}\left(\text{DWConv}\left(\text{Conv}_{\text{ex}}\left(\text{Norm}(X)\right)\right)\right)\right), \tag{3.1}$$

where Norm represents Batch Normalization (BN) [58], Conv_{ex} denotes the expansion convolution layer in context of the numbers of channels with a kernel of 1×1, DWConv denotes a depthwise convolution layer with a kernel of 3×3, SE denotes the Squeeze Excitation Layer [56], and Conv_{sh} denotes the corresponding shrinking convolution layer with a kernel of 1×1. Each convolution layer is followed by BN and the Gaussian Error Linear Unit (GeLU) activation function [52]. For the first MBConv Block in every stage, the depthwise convolution layer has a stride of two and the skip connection is replaced with a 2D pooling layer.

Multi-Axis Attention. Multi-Axis Attention is based on relative attention [27, 133]. Relative attention adds a relative positional bias B to vanilla self-attention. B is a learned static location-aware matrix and influences the adaptive attention outputs. Consequently, relative attention offers several advantageous features for 2D vision

tasks such as input-adaptivity, translation equivariance, and global interactions. In the case of a single head, relative attention is formulated as

$$\text{RelAttn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V, \tag{3.2}$$

where $Q, K, V \in \mathbb{R}^{H \times W \times C}$ are the query, key and value matrices and d is the hidden dimension. In the case of MaxViT, Q, K, V are a linear projection with each weight respectively from the same input vector. Based on the convention for 1D input sequences for Eq. (3.2), the second-to-last dimension of an input $(..., L, C)$, represents the sequence length and can be defined as the spatial axis. MaxViT does not modify relative attention directly, but modifies the input for the spatial axis. In the case of block attention, MaxViT reshapes the input tensor into the shape

$$\begin{aligned}(H, W, C) &\to \left(\frac{H}{P} \times P, \frac{W}{P} \times P, C\right) \\ &\to \left(\frac{H}{P} \times \frac{W}{P}, P \times P, C\right)\end{aligned} \tag{3.3}$$

with a partition size of P. Therefore, instead of calculating self-attention globally, self-attention is only calculated locally within a partition for window attention. In order to enable sparse global interactions,

$$\begin{aligned}(H, W, C) &\to \left(\frac{H}{G} \times G, \frac{W}{G} \times G, C\right) \\ \to \underbrace{\left(G \times G, \frac{H}{G} \times \frac{W}{G}, C\right) \to \left(\frac{H}{G} \times \frac{W}{G}, G \times G, C\right)}_{\text{swapaxes}}\end{aligned} \tag{3.4}$$

the spatial dimensions denoted as H and W are partitioned by the grid size G. This division, followed by transposition and axis swapping, results in the spatial axis representing a fixed uniform grid. The application of self-attention to the reshaped tensor facilitates sparse global interactions. For both attention mechanisms, an inverse partitioning function is required.

Cross Multi-Axis Feature Pyramid Network (CmaxFPN). The CmaxFPN, a fusion between AFPN from Subsection 3.5.2 and MaxViT [149], not only facilitates local-global interactions among the feature maps, but also introduces connections between feature maps across different resolution scales and semantic levels. The reason for the global field of view is the grid attention introduced by MaxViT, which enables a global field of view without adding another branch like in DustNet.

Initially, the feature maps are uniformly interpolated to a square dimension. The reason for that is computational efficiency.

Subsequently, the CmaxFPN for the feature map with the highest semantic level and lowest resolution scale structurally resembles a MaxViT block, where each MBConv [55] is succeeded by two pairs of grid and window attention blocks. The motivation behind expanding the original MaxViT block is that the number of layers is similar to the following stages and to avoid architectural complexity. Cross- and self-attention alternate in the window and grid attention pairs in the following stages. Cross-attention is used in order to fuse the features between the the different feature maps:

$$Q_3 \leftarrow \text{CmaxFPN}_{s3}(E_3, E_3) \tag{3.5}$$

$$Q_2 \leftarrow \text{CmaxFPN}_{s2}(E_2, \text{Upsample}(Q_3)) \tag{3.6}$$

$$Q_1 \leftarrow \text{CmaxFPN}_{s1}(E_1, \text{Upsample}(Q_2)) \tag{3.7}$$

$$Q_0 \leftarrow \text{CmaxFPN}_{s0}(E_0, \text{Upsample}(Q_1)), \tag{3.8}$$

where $Q_i, i \in \{0, 1, 2, 3\}$ describes the output for each stage s_x of the CmaxFPN. The upsample operation can be described by the following term:

$$\hat{Q}_i \leftarrow \text{Conv}(\text{PixelShuffle}(Q_i)), \qquad i \in \{0, 1, 2, 3\} \tag{3.9}$$

Pixel Shuffle [134] is employed to upsample the output from the preceding CmaxFPN stage, subsequently aligning it with the channel number of the subsequent feature map E_i, for $i \in \{0, 1, 2, 3\}$. The selection of Pixel Shuffle over alternative methods such as interpolation functions or transpose convolutional layers is predicated on its demonstrated efficacy within the AFPN structure, as introduced in PixelFormer [2] and subsequently employed in DustNet [102]. Figure 3.8 delineates the fundamental architecture of a CmaxFPN block, where an MBConv module is succeeded by a pair of window and grid self-attention blocks, mirroring the structure of a MaxViT block. However, the subsequent pair is configured distinctively. Rather than utilizing a single input tensor for query, key, and value matrices, our CmaxFPN employs the upsampled output of the previous stage Q_{i+1} as the query matrix, while the output from the initial multi-axial attention block serves as the key and value matrices. This configuration fosters interactions between features from disparate resolution scales and semantic levels. The implementation of this cross-attention mechanism is a prominent feature of our methodology, distinguishing it from other multi-scale feature strategies that utilize MaxViT blocks, such as those in MaxViT-UNet [64], which typically concatenate features across various feature maps. Our approach distinctively leverages the cross-attention scheme to enhance feature integration.

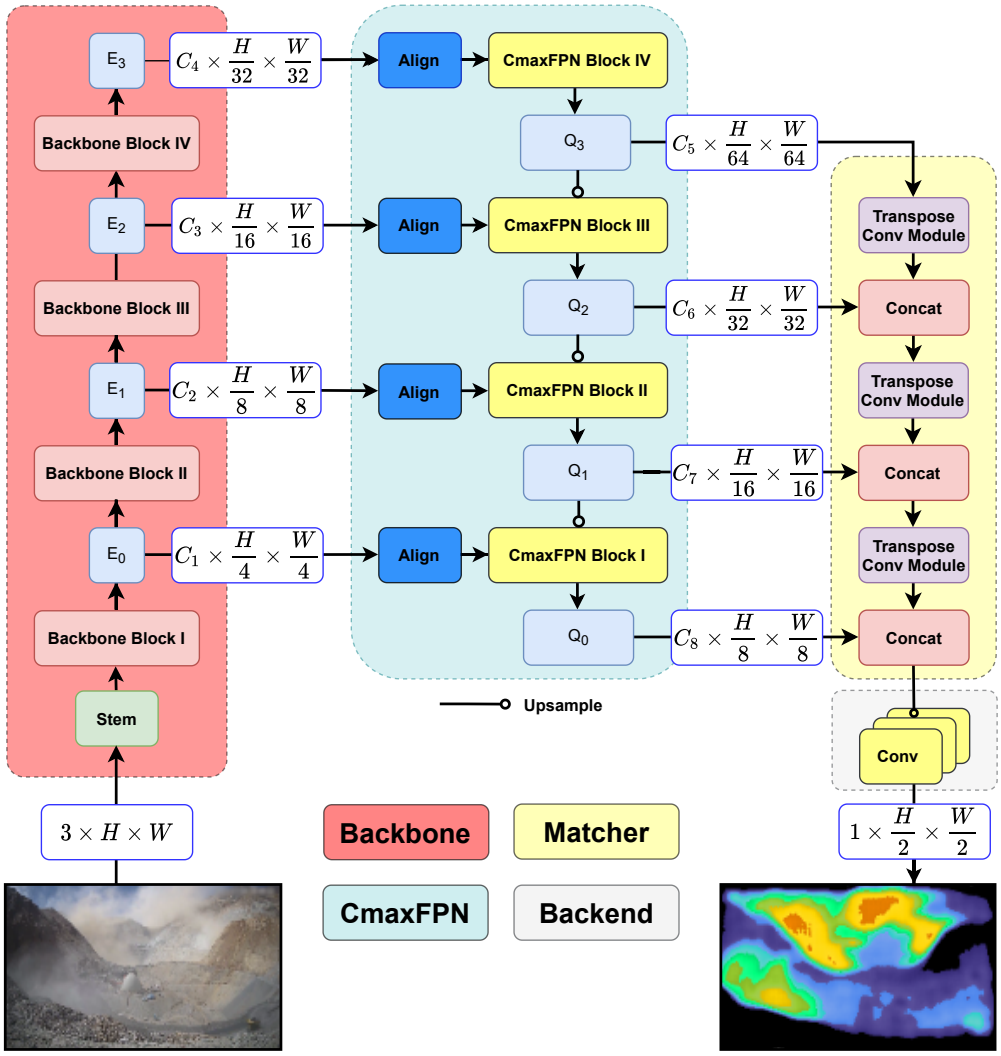


Figure 3.7: Architecture of DustNet++. The basic components of DustNet++ include the backbone, the CmaxFPN, the matcher, and the backend. These elements, except the Pyramid Pooling Module branch and its adaptive convolution layers, maintain a structural similarity to DustNet. The removal of the latter components has simplified the overall architecture. To facilitate global interactions, the AFPN has been substituted with CmaxFPN. Unlike AFPN, CmaxFPN employs cross-window attention and additionally integrates sparse grid cross-attention, thereby expanding the effective receptive field.

3.6.2 Components of DustNet++

In [102], the significance of integrating both global and local features was emphasized. Primarily, the synergistic combination of a broad field of view and high-resolution features contributes DustNet's superior performance. This synergy necessitates a relatively intricate architecture that incorporates an additional branch for high-level semantic features, albeit at a lower spatial resolution. Our proposed

subsequent version, DustNet++, strategically eliminates this branch. Nevertheless, in order to maintain a comprehensive field of view, MaxViT blocks are employed in place of the conventional cross Swin blocks. MaxViT blocks exploit grid attention to achieve global feature fusion.

Backbone. Considering an input image $I \in \mathbb{R}^{H \times W \times C}$, where C denotes the number of channels, W the width, and H the height. Analogous to DustNet, the image I is processed through a backbone network, which generates four feature maps $E_i \in \mathbb{R}^{H_i \times W_i \times C_i}$, for $i \in \{0, 1, 2, 3\}$, corresponding to the spatial resolution scales $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of I.

CmaxFPN. Each of the derived feature maps are then aligned by a channel mapper to a fixed number of channels. In contrast to DustNet, the already mentioned high-level semantic branch, which consists of the PPM and AFPN, has been replaced by the CmaxFPN, whose task is to fuse local and global features (cf. Section 3.6.1).

Matcher. The matcher module remains consistent with that of DustNet, processing the output features from the CmaxFPN. It has an FPN-like architecture [82]. We upsample the feature maps via a transpose convolution module (TCM). It consists of a 2D TCM with a stride and kernel size of two, followed by batch normalization [58] and a SiLU [36] activation function. As suggested in [80], we apply only batch normalization without dropout [141]. The coarsest resolution feature map derived from the CmaxFPN is fed to the first TCM block. The following CmaxFPN feature maps are respectively concatenated to the output of the TCM block and processed via the next TCM block.

Backend. The backend consists of N blocks of a sequence of a 2D convolution layer, batch normalization, and SiLU activation functions that predict the dust density map. We split the features into two parallel blocks for each stage and accumulate the outputs. Hereby, we choose a dilation of three for one branch to increase the receptive field. After four stages, a pointwise convolutional layer predicts the dust density maps.

3.6.3 DustNet++ Duo

DustNet++ Duo is heavily inspired by its predecessor, DustNet C. The goal of the design process for the fusion approach is to simplify the architecture, particularly to reduce the number of branches. Like DustNet C, DustNet++ Duo utilizes two consecutive images that are processed by a shared backbone. But in contrast to DustNet, DustNet++ utilizes grid attention. Grid attention enables DustNet++ a

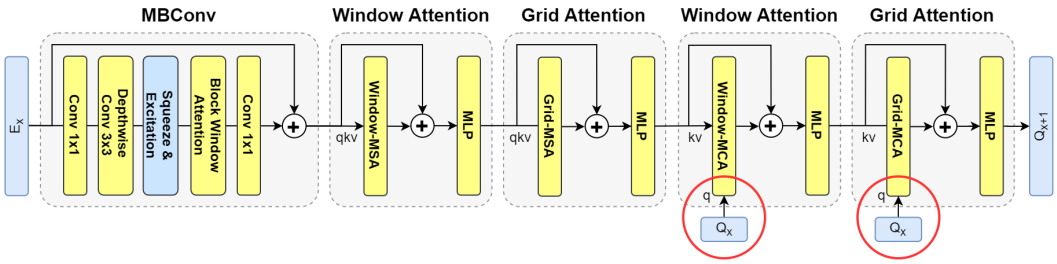


Figure 3.8: CmaxFPN Block. The CmaxFPN block enhances the MaxViT block by incorporating cross-attention into the basic blocks. Similar to MaxViT, an MBConv is followed by window and grid self-attention layers. In addition, a cross window and cross grid attention layer are included. In this block, the output of the previous feature map is utilized as a query matrix. This facilitates a feature pathway from higher semantic levels with low spatial resolution to features with a lower semantic level and higher spatial resolution.

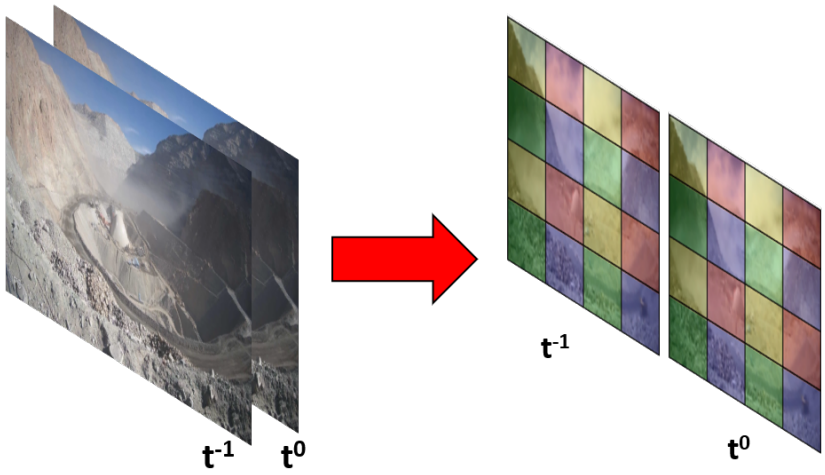


Figure 3.9: Temporal Fusion. The fundamental premise of the temporal fusion methodology implemented in DustNet++ Duo is to harness the potential of grid attention. This is not solely for the purpose of augmenting the field of view, but also to facilitate the feature flow among temporally sequential images. This is achieved by spatially concatenating the images along the minor dimension after processing through the backbone and then feeding them into the CmaxFPN.

simple and straightforward way to fuse the images. Before each image's feature maps are sent to CmaxFPN, both images are spatially concatenated along the axis with the smaller dimension. This can be seen in Figure 3.9. This configuration enables grid attention to orchestrate the interaction of information between consecutive images. This method presents a more streamlined approach to fusion compared to the techniques previously proposed by DustNet.

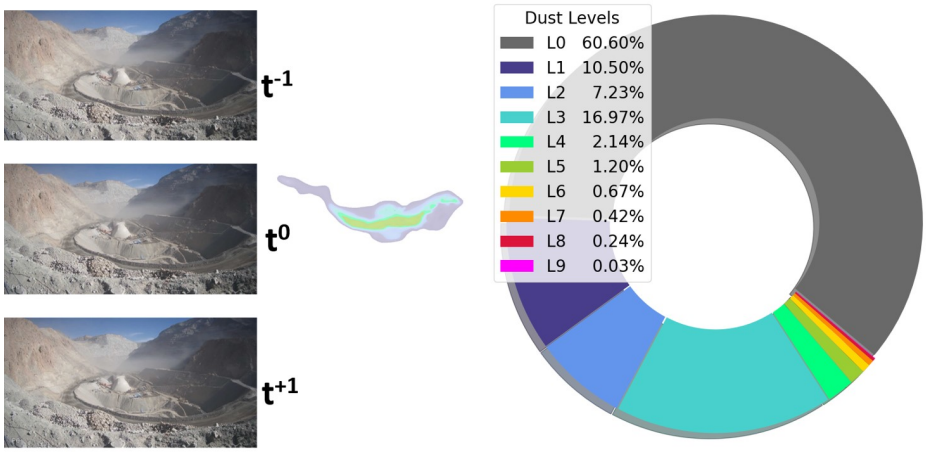


Figure 3.10: Meteodata Dust Dataset. The Meteodata dust dataset comprises a collection of sequential temporal images accompanied by their respective data distributions. Each data sample includes a triplet of consecutive images captured at varied temporal intervals, wherein the ground truth is associated with image at the time t^0. Each image possesses dimensions of 1000 $\times$ 1920 pixels, while the resolution of the ground truth is halved relative to that of the images. Adjacent to the dataset description, a chart visually represents the distribution of labels on a per-pixel basis. Notably, the dataset demonstrates a pronounced imbalance in label distribution, with high-intensity values, potentially indicative of explosive events, being remarkably rare.

3.7 Experimental Results

In this section, we outline the implementation specifics of our experiments, expound upon the results obtained, and provide an extensive discussion of our ablation study. Initially, we introduce the datasets employed in our research. Subsequently, we clarify the criteria for benchmark selection and address the intricate details of our experimental implementation. Following this, we engage in both qualitative and quantitative evaluations of the performance of our proposed approaches. Lastly, we elaborate on our ablation study.

3.7.1 Datasets

Our experiments are conducted using two datasets. The focus is set on visual regression for dust density estimation on the temporal Meteodata dust dataset [103], whereas the URDE dataset [29] is involved in reasoning about the generalization capabilities of the involved approaches.

Meteodata Dust Dataset. Meteodata, a spin-off company from the Department of Geophysics at the University of Chile, has collaborated with the Fraunhofer IOSB Institute to develop the Meteodata dust dataset. The fundamental objective of this dataset is not to represent a physically accurate depiction of dust but rather to simulate human perception of varying dust levels. This approach offers the advantage of eliminating the need for expensive measurement devices, allowing for the use of conventional industrial-grade RGB cameras instead. Human assessments of dust conditions are often sufficient for accurately identifying dust sources and events, thereby facilitating appropriate countermeasures, such as the watering of roads.

The dataset has been meticulously annotated over multiple iterations by a highly qualified team of experts specializing in airborne dust. It encompasses a diverse array of scenes from open-pit mines, featuring a wide spectrum of dust levels. Specifically, the dataset includes 2,298 RGB image triplets, each with dimensions of 1,000 $\times$ 1,920 pixels. Each triplet consists of three successive images, with the ground truth dust density map corresponding to the middle image. The mean temporal interval between consecutive images is approximately ten seconds. Dust density values in the ground truth are quantified using an 8-bit unsigned integer data type, with pixel values directly proportional to dust density levels. Notably, the ground truth represents only a quarter of the total pixels within each image, primarily due to the challenges associated with accurately labeling dust boundaries. Achieving pixel-perfect labeling is often impractical. Therefore, in light of the limitations regarding precise ground truth identification and the increased computational demands that would accompany higher resolution, a decision was made to employ a reduced resolution for this dataset.

Furthermore, the dataset is divided into training, validation, and testing subsets, containing 1,906, 144, and 248 image triplets, respectively. The primary challenges associated with this dataset include the significant computational demand arising from the large image sizes, the inherent difficulty in accurately estimating diverse dust levels due to the considerable variability in dust concentrations, and the pronounced imbalance in the frequency of various dust levels. Figure 3.10 illustrates the imbalance present in the ground truth data. As of the time of publication, the Meteodata dust dataset is not publicly accessible.

URDE Dataset. To assess our methodology's generalization capability, we conducted quantitative and qualitative evaluations on the publicly available URDE dataset [29]. The URDE dataset consists of 7000 images depicting unsealed road segments with ten distinct types of road surface materials under various conditions, designed specif-

ically for the assessment of vehicle-induced road dust emissions. The images were captured using a Canon EOS 200D single-lens reflex camera and were subsequently downsampled to a resolution of 1024 $\times$ 1024 pixels. The ground truth data, which required over 1500 hours of manual annotation, underpins the dataset's reliability. From the total, 897 images were selected to form the RandomDataset_897 dataset; this subset was divided into 800 images for training and 97 for validation purposes. The selection criteria aimed to minimize the presence of visually similar consecutive images within the samples and to ensure a substantial variation in dust patterns across the dataset.

The original training dataset from URDE RandomDataset_897 was employed solely to evaluate the generalization ability for testing purposes. It is important to note that no training was performed using URDE and was only used to assess the generalization capacity of the proposed method.

3.7.2 Evaluation Metrics

We focus on evaluating the localization and regression capabilities of our proposed models. To assess the localization performance, we modify the pixel values, setting all those below 30 to zero and converting the remaining pixels to one. This binary mapping allows us to utilize standard classification metrics such as Accuracy (Acc), Precision (Pre), and Recall (Rec) for performance evaluation. Additionally, to address the challenges posed by imbalanced data distributions commonly encountered in semantic segmentation, we employ the Intersection-over-Union (IoU) metric, which provides a robust measure of overlap between predicted and actual classifications. We use the same binary scheme as for the other classification metrics. Additionally, we utilize the Dice [31, 140] coefficient.

For the validation of regression quality, we utilize conventional metrics including the mean absolute error (MAE) and mean squared error (MSE). These metrics provide insights into the average magnitude of errors in the predictions. Moreover, to ensure a fair evaluation of performance across the tail values in imbalanced datasets, we incorporate the concept of balanced metrics. This approach adjusts the evaluation criteria to give proportional consideration to less frequent, yet significant, data points, thereby providing a more comprehensive assessment of model performance across diverse dataset characteristics [10]. Therefore, we bin our data into four bins:

0-29:	Zero dust density bin	(ZB)
30-99:	Low dust density bin	(LB)
100-169:	Medium dust density bin	(MB)
170-255:	High dust density bin	(HB)

For each bin, we calculate the MAE and MSE. Following [10, 118], we compute the mean across all bins and obtain the average binned mean absolute error (ØB-MAE) and the average binned mean squared error (ØB-MSE).

3.7.3 Benchmark Selection

Dust density estimation represents an emerging niche within the broader field of environmental monitoring. A review of the existing literature highlights a limited amount of research focused on this topic. This observed scarcity underscores the necessity for our benchmark to incorporate methodologies from other fields such as pixel-by-pixel visual regression, drawing insights from crowd counting, and Monocular Depth Estimation (MDE). Our selection criteria were confined to methodologies from different domains that concentrate on the analysis of individual images. This decision was based on preliminary observations indicating that temporal density estimation methods, commonly employed in crowd counting research [5], face convergence challenges in scenarios characterized by substantial irregular temporal intervals between large consecutive images. In our methodological framework, we juxtapose DeepDust and DustNet with CanNet [88], a context-aware, lightweight, fully convolutional network tailored for crowd counting. The rationale for selecting CanNet hinges on its foundational employment of the VGG16 architecture, which provides a basis for assessing the potential applicability of simpler network structures to temporal dust density estimation tasks. Furthermore, our exploration extends to cutting-edge MDE models, specifically NeWCRF [174] and PixelFormer [2], both of which incorporate the Swin Transformer as their backbone. These models are configured with a window size of twelve and are tested in both base and large variants. This selection is aimed at evaluating the feasibility of advanced MDE techniques in addressing the specific challenges associated with temporal dust density estimation.

Table 3.1: Model Settings. The table below outlines the configuration of the training process. The symbol #Img indicates the number of sequential images provided as input into the model. When two images are used, the ground truth corresponds to the second image or in the case of three images to the middle image. The time refers to the model's inference time for a 1000 $\times$ 1920 pixel image on a Nvidia A100 GPU, repeated 1000 times for a batch size of one. This configuration is also applied to determine the maximum memory usage. The abbreviations are as follows: Swin-B stands for Swin Transformer Base, Swin-L for Swin Transformer Large, CNV2-B for ConvNeXt V2 Base, and r101 for ResNet101.

Model	Backbone	Params	Time	MeM	#Img
CanNet	VGG16	18 M	28.53 ms	6.4 GB	1
CrowdFPN	r101	54 M	70.02 ms	24 GB	1
NeWCRF	Swin-B	140 M	80.36 ms	8.0 GB	1
NeWCRF	Swin-L	270 M	114.6 ms	9.6 GB	1
PixelFormer	Swin-B	128 M	66.7 ms	5.6 GB	1
PixelFormer	Swin-L	258 M	105.8 ms	7.2 GB	1
DeepDust	CNV2-B	101 M	149.0 ms	28.8 GB	1
DeepDust	Swin-L	207 M	205.9 ms	29.6 GB	1
DustNet S	r101	68 M	67.6 ms	12 GB	1
DustNet S	Swin-L	227 M	116.4 ms	12 GB	1
DustNet++	r101	82 M	123.6 ms	8 GB	1
DustNet++	Swin-L	235 M	181.3 ms	9.6 GB	1
DustNet A	r101	65 M	66.4 ms	10.4 GB	3
DustNet B	r101	67 M	131.1 ms	12.0 GB	3
DustNet D	r101	86 M	128.0 ms	11.2 GB	3
DustNet C	r101	68 M	107.3 ms	12.8 GB	2
DustNet C	Swin-L	227 M	203.4 ms	14.4 GB	2
DustNet++ Duo	r101	83 M	143.9 ms	9.6 GB	2
DustNet++ Duo	Swin-L	236 M	261.1 ms	12.0 GB	2

3.7.4 Implementation Details

The experimentation was facilitated using an array of four A100 Nvidia GPUs, each equipped with 80 GB of memory. The foundational benchmark [103, 102] employed the L_2 loss function alongside the AdamW optimizer [92] (with β values of 0.9 and 0.999) and a weight decay parameter set to 10^{-5}. Throughout the training phase, a learning rate of $\alpha = 3 \cdot 10^{-4}$ was maintained, and the models were subject to a training duration of 50 epochs with a non-trainable (frozen) backbone. Subsequent to this phase, the backbone was activated (unfrozen) and underwent further training for an additional 20 epochs. For all training steps, we utilized Gradient Accumulation with an accumlated batch size of 32. We normalize input images using the standard ImageNet values [69].

Alternative experimental configurations were explored in response to the observed suboptimal performance of transformer-based backbones as reported in [103, 102]. Initially, checkpoint wrappers (CPWs) were implemented to enhance memory efficiency during the training processes. Activation checkpointing, as described in [18, 61], involves the temporary removal of specific layer activations during backpropagation, which are then recomputed in the backward pass. This method significantly reduces memory consumption and may facilitate the use of larger batch sizes. The Fairscale [38] implementation of this checkpoint wrapper was utilized on the backbones, effectively allowing for a doubling of the real batch size in several instances.

Further adjustments were made to the learning strategy, extending the training period to 200 epochs with an active backbone, while reducing the learning rate for the backbone layers to ten percent of the initial rate. The properties of the employed models are presented in Table 3.1. Additionally, a comparative analysis was conducted between traditional backbones such as ResNet101 [48] and the Swin Large Transformer [90] to comprehensively evaluate the methodologies.

All models were trained on the Meteodata dust training dataset (See Figure 3.10) , ensuring consistency in the data used across different experimental setups.

3.7.5 Summarized Results on the Meteodata Dust Dataset

Table 3.2 outlines the comparative effectiveness of various density estimation methods applied to the Meteodata dust dataset. The column name #Img describes the number of images processed simultaneously. The tag CPW stands for Checkpoint Wrapper, and reduces the memory usage while training, and therefore increases the batch size. More detailed results are provided in Appendix 3.8.1.

Single-Image Analysis.

The single-image results are produced by models which process only one image. Firstly, it is noteworthy that models trained with a checkpoint wrapper yielded superior results for almost all cases. This result emphasizes the importance of batch size for performance. Notably, the Swin Transformer Large variants, which benefited from increased batch sizes, surpassed the performance of the ResNet101 variants. Despite this, both DustNet and DustNet++ significantly outperformed all other methods by a substantial margin. CanNet, which was not retrained, demonstrated early convergence prior to reaching 50 epochs in previous experiments, and its relatively compact size allowed for the use of large batch sizes without the need

Table 3.2: **Meteodata Dust Dataset Results.** Comparison of the best-performing density estimation methods on the Meteodata dust dataset. The highlighted values represent the best scores for using either one or more images simultaneously.

Model	Backbone	CPW	#Img	MAE	MSE	IoU
CanNet	VGG16	×	1	20.21	855.40	0.648
CrowdFPN	r101	×	1	19.95	848.18	0.659
NeWCRF	Swin-B	×	1	20.00	822.68	0.635
NeWCRF	Swin-B	✓	1	20.83	887.67	0.643
NewCRF	Swin-L	✓	1	19.27	740.54	0.654
PixelFormer	Swin-B	×	1	21.53	825.50	0.641
PixelFormer	Swin-B	✓	1	17.43	645.54	0.697
PixelFormer	Swin-L	✓	1	16.29	576.64	0.713
DeepDust	CNV2-B	×	1	19.60	749.36	0.687
DeepDust	CNV2-B	✓	1	17.11	561.89	0.702
DeepDust	Swin-L	✓	1	15.24	482.90	0.774
DustNet S	r101	×	1	19.27	705.47	0.660
DustNet S	r101	✓	1	14.10	458.96	0.756
DustNet S	Swin-L	✓	1	12.80	**402.20**	**0.793**
DustNet++	r101	✓	1	13.95	497.72	0.762
DustNet++	Swin-L	✓	1	**12.63**	422.33	0.784
DustNet A	r101	×	3	18.77	701.73	0.654
DustNet B	r101	×	3	26.60	1528.08	0.481
DustNet D	r101	×	3	17.44	639.10	0.677
DustNet C	r101	×	2	16.77	601.49	0.685
DustNet C	r101	✓	2	12.60	413.22	0.782
DustNet C	Swin-L	✓	2	12.52	414.92	**0.786**
DustNet++ Duo	r101	✓	2	12.20	427.72	0.766
DustNet++ Duo	Swin-L	✓	2	**11.73**	**396.10**	0.753

for CPW. However, CanNet exhibited limited regression capabilities at higher dust levels.

The method that least benefits from the new training scheme was NeWCRF, which showed a decrease in regression performance despite a slight improvement in localization for the Swin Transformer Base variant. In contrast, while NeWCRF experienced modest gains, PixelFormer saw significant enhancements in both regression and localization capabilities through the new training scheme. Although DeepDust still surpasses PixelFormer and CrowdFPN, the gap between the two methodologies has narrowed, possibly due to PixelFormer's adaptive bins.

The most successful approaches overall remain DustNet, closely followed by DustNet++. DustNet++ exceeds DustNet in MAE, closely aligns with its overall performance while offering a simpler architectural design.

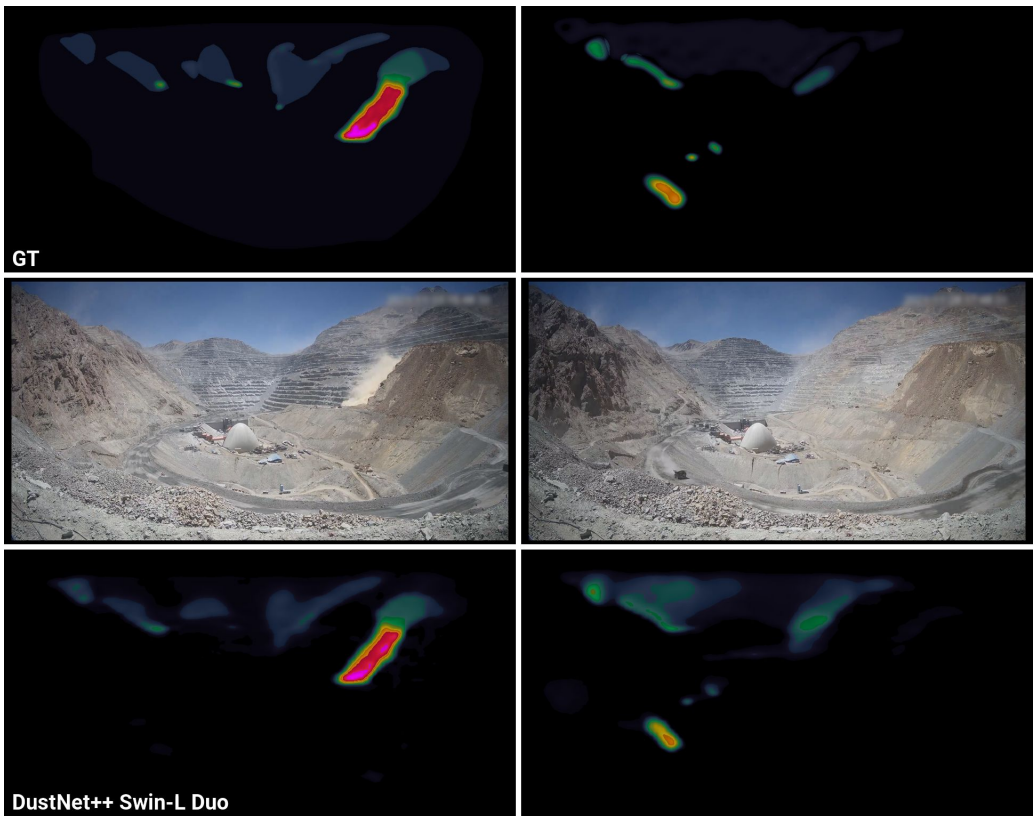


Figure 3.11: Results of DustNet++ Duo. The depicted outcomes encompass two scenes demonstrating the efficacy of DustNet++ Swin-L Duo with two consecutive images as input, which was trained utilizing the Meteodata dust dataset. The scene to the left illustrates the regression proficiency of DustNet++ Duo, whereas the scene on the right highlights its localization capabilities.

Multi-Image Analysis. The bottom part of Table 3.2 presents the outcomes on the temporal Meteodata dust dataset. Similar to the single-image scenario, the new training scheme significantly improved results, with the Swin Transformer Large variants outperforming the ResNet101 variants. Here in this case, DustNet++ demonstrated superior regression capabilities compared to DustNet C, albeit with a slight reduction in IoU. Incorporating temporal information consistently led to enhanced overall results.

Figure 3.11 illustrates the performance of DustNet++ equipped with a Swin Large Transformer backbone in two opencast mining scenes, showcasing the practical application and effectiveness of the model in real-world settings.

Table 3.3: **Results on the URDE Randomdataset_897 Dataset.** The following models are trained on the Meteodata dust dataset and then applied with a threshold of 35% on the original Randomdataset_897 training dataset without finetuning.

Model	Backbone	CPW	Acc	Pre	Rec	Dice	IoU
CanNet	VGG16	×	**0.972**	0.627	0.506	0.505	0.493
CrowdFPN	r101	×	0.947	0.660	0.835	0.697	0.604
NeWCRF	r101	✓	0.960	0.655	0.712	0.679	0.596
NeWCRF	Swin-L	✓	0.960	0.658	0.723	0.684	0.601
PixelFormer	r101	✓	0.934	0.627	0.867	0.679	0.588
PixelFormer	Swin-L	✓	0.947	0.651	**0.869**	0.708	0.615
DeepDust	CNV2-B	✓	0.951	0.655	0.840	0.708	0.617
DeepDust	Swin-L	✓	0.960	**0.677**	0.798	0.720	0.630
DustNet S	r101	✓	0.942	0.620	0.766	0.660	0.577
DustNet S	Swin-L	✓	0.957	0.663	0.791	0.707	0.618
DustNet++	r101	✓	0.852	0.544	0.714	0.545	0.472
DustNet++	Swin-L	✓	0.957	0.671	0.837	**0.723**	**0.631**

3.7.6 Results on the URDE Dust Dataset

In our analysis, we employed models trained on the Meteodata dust dataset to evaluate the generalization capability of our approach using binary segmentation on the URDE dataset. Given the binary nature of the labels in this dataset, our investigation was specifically confined to assessing the models' localization ability on the URDE RandomDataset_897 dataset.

Due to the different sensitivities to dust characteristics between the Meteodata dust dataset and the URDE dataset, our analysis primarily focused on the lower 35% of the prediction values generated by the models. This threshold was determined empirically, utilizing 5% intervals to ensure precision in our assessment. The results derived from the URDE RandomDataset_897 training dataset, which comprises 800 images, are detailed in Table 3.3. Notably, CanNet exhibits high accuracy and precision, yet it demonstrates low recall alongside suboptimal Dice and IoU scores. This suggests a low true-positive rate within a highly imbalanced dataset, rendering CanNet as the only model that did not achieve satisfactory outcomes on the URDE dataset.

Conversely, PixelFormer and NeWCRF delivered superior Dice and IoU scores, with PixelFormer exhibiting marginally higher results comparable to those observed in the Meteodata dust dataset. DeepDust transcends both models in terms of Dice and IoU scores, surpassing DustNet as well. Additionally, the impact of the backbone size on the performance of the DeepDust architecture is negligible when compared to other models.

Table 3.4: DustNet Ablation Study. This figure illustrates the results of the ablation study on the Meteodata dust dataset: The base model is DustNet C with two consecutive images as inputs. The modules following a ✗ are replaced. The base model outperforms the other variants except for replacing the matcher. Thus, localization accuracy slightly increases at the expense of decreased regression ability.

	Params	MAE	MSE	Acc	Pre	Rec
1x Img Input	64 M	19.27	705.47	0.80	0.81	0.80
2x Img Input	68 M	**16.77**	**601.49**	0.81	**0.83**	0.82
3x Img Input	68 M	17.83	675.34	0.81	0.82	0.82
✗ AFPN	**32 M**	19.85	805.74	0.79	0.80	0.79
✗ Fuser	68 M	19.63	754.25	0.80	0.81	0.80
✗ Matcher	64 M	17.08	651.39	**0.83**	**0.83**	**0.83**

DustNet++ is the top performer overall, though its advantage over DeepDust is not significantly substantial. Contrasting with the findings from the Meteodata dust dataset, the localization ability of DustNet++ surpasses that of DustNet, suggesting a potentially lesser degree of overfitting in DustNet++ relative to DustNet.

Figure 3.12 visually presents qualitative results of DustNet++ alongside DustNet S, both employing Swin Large Transformer backbones, demonstrating their practical application and effectiveness in handling the URDE dataset. Further results are provided in Appendix 3.8.2.

3.7.7 Ablation Study of DustNet

In order to show the efficacy of our proposed DustNet, we choose the best-performing proposed variant DustNet C, change the number of input images and replace several modules (see Table 3.4). We replace the AFPN, the matcher, and the fuser module, respectively.

Inputs. We compare the performance change of our proposed method by varying the number of consecutive images. Backbone and AFPN weights are shared in our experiment. The case of two consecutive images outperforms the other options.

AFPN. In the AFPN ablation experiment, the AFPN is removed, and the features from the backbone are passed directly to the matcher. Removing the AFPN nearly halves the number of parameters, but it causes a significant increase in MAE and MSE.

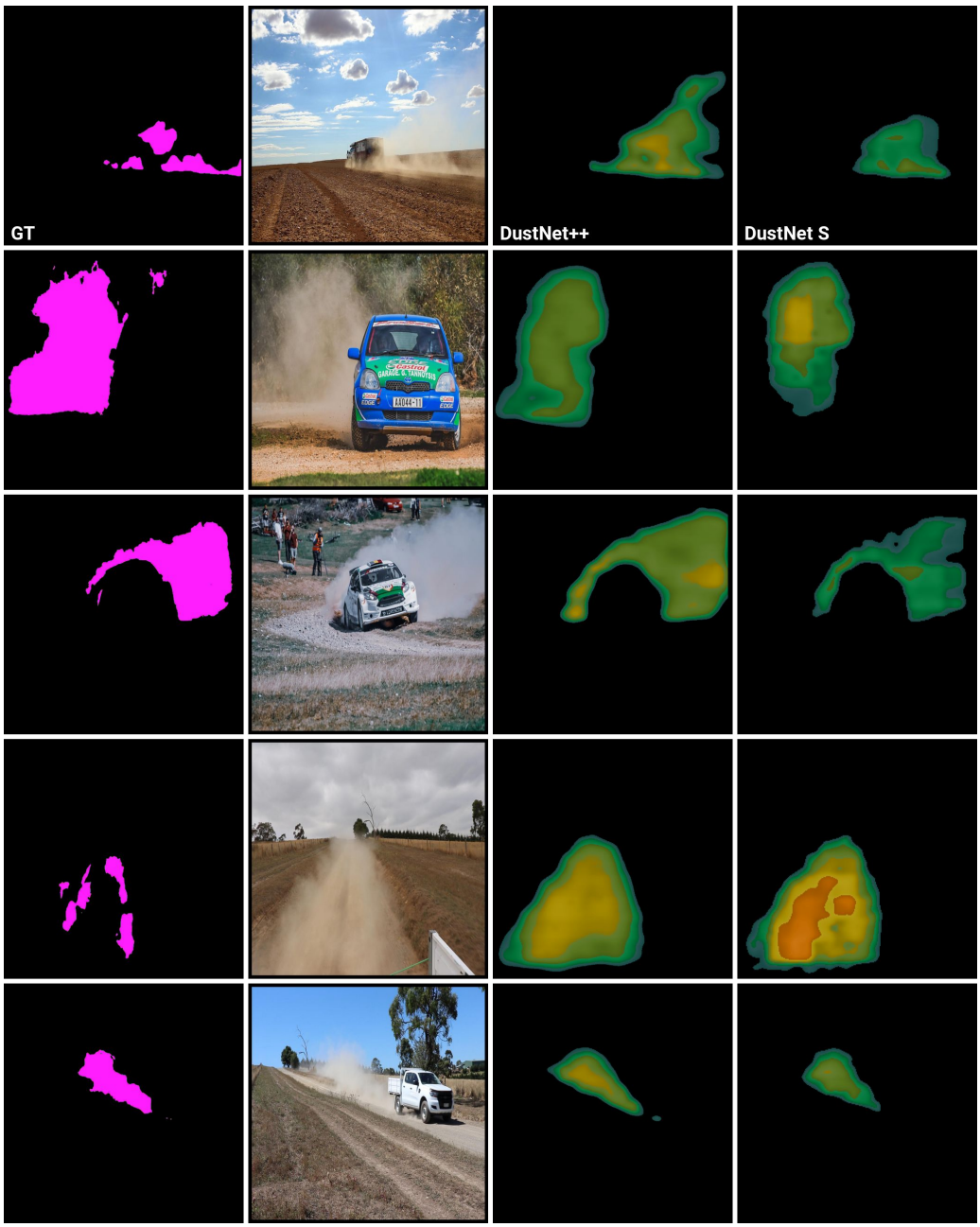


Figure 3.12: Display of the Generalization Ability of DustNet++. The shown results of DustNet++ and DustNet S employed with Swin Large Transformer backbones are produced on the URDE validation RandomDataset_897. Hereby, both methods are trained on the Meteodata dust dataset and applied to the URDE dataset without fine-tuning.

Fuser. In the matcher ablation experiment, a vanilla convolutional layer replaces the adaptive convolutional layer. Hence, the aggregated global information features cannot enrich the matcher's features. This leads to a significant increase in MAE and MSE.

Table 3.5: Results of the Ablation Study on DustNet++. All models utilize the Swin Large Transformer backbone and are trained with activation checkpoint wrappers for 200 epochs. The elimination of the MBConv necessitates the incorporation of a 2D max-pooling layer at the initial block of a stage and leads to a decrease in the number of parameters. In instances where a MSA variant was removed, it was substituted with the remaining MSA block variants to maintain a consistent overall parameter count.

MBConv	✓	×	✓	✓	✓	✓
Cross G-MSA	✓	✓	×	×	✓	✓
Cross W-MSA	✓	✓	×	✓	×	✓
G-MSA	✓	✓	✓	×	✓	×
W-MSA	✓	✓	✓	✓	×	×
MAE	**12.63**	13.74	15.46	15.62	17.94	18.93
MAE-ØB	**22.43**	24.66	22.94	28.35	27.04	36.07
MSE	**422.33**	493.87	512.89	633.24	686.77	813.20
MSE-ØB	1039.22	1371.80	**1001.67**	1503.64	1325.68	2285.55
IoU	**0.784**	0.776	0.761	0.677	0.678	0.634
Dice	**0.879**	0.874	0.864	0.807	0.808	0.775

Matcher. In the last ablation experiment, we replace the matcher with a simple convolutional layer followed by an activation function for channel adaptation. The feature map with the highest resolution from the AFPN is processed by the added convolutional layer and fed into the fuser module. The accuracy increases slightly for the tradeoff of a decreased regression ability. However, the main reason for keeping the matcher is the increased differentiation ability between dust and similar visual effects like clouds. Removing the matcher leads to a significant drop in in distinguishing capability.

3.7.8 Ablation Study of DustNet++

The outcomes of the ablation study on DustNet++ are presented in Table 3.5, wherein all models employ the Swin Large Transformer backbone and are trained with activation checkpoint wrappers for 200 epochs. In instances where the MSA block was removed, it was substituted with an MCA block to maintain a consistent overall parameter count.

Replacing MBConv. The removal of the MBConv mandates the integration of each 2D max-pooling layer at the first block of a stage to preserve comparable resolution. Consequently, this modification results in a reduction in model size. Overall, the omission of the MBConv results in a marginal decline in performance.

Window Attention vs. Grid Attention. The exclusion of window attention leads to a more pronounced degradation of overall performance. In contrast, the removal of the grid attention module impairs the model's capacity to discriminate between dust and visually analogous phenomena, such as clouds. Global interactions are critical in reducing the false positive rate associated with similar visual effects, and local interactions are indispensable for precise dust detection. Nevertheless, employing both variants concurrently results in superior outcomes compared to utilizing only one.

MCA vs. MSA. Utilizing solely the MCA as opposed to the vanilla MSA results in a greater decline in performance. This may be attributed to the fact that although the subsequent matcher module facilitates interactions between different feature maps from varying backbone stages, the absence of the Vanilla MSA could potentially lead to disruptions in the feature flow. Analogous to the comparison between window and grid attention, employing both MCA and Vanilla MSA enhances performance.

3.8 Additional Results from Models Trained on the Meteodata Dust Dataset

Figs. 3.13, 3.14, and 3.15 display additional results from models trained using the Meteodata dust dataset.

Figure 3.13 shows various mining events with noticeable differences in the size and density of the dust plumes across the images. CanNet generally performs well in terms of regression ability, but it fails to accurately delineate the boundaries of the dust plumes. Specifically, in the image on the left, the dust plume generated by the truck is not tracked correctly, indicating that CanNet's simpler architecture might be insufficient for this task. On the other hand, PixelFormer exhibits superior localization capabilities, although its regression performance is lacking. DeepDust shows enhanced outcomes but is outperformed by DustNet. DustNet S exhibits good localization capabilities, potentially better than DustNet++, but its regression performance is marginally inferior. However, DustNet++ Duo significantly outperforms the other methods in terms of regression ability in these scenarios.

Figure 3.14 focuses on scenarios characterized by significant cloud coverage. The data illustrate that DustNet++ effectively differentiates between dust particles and cloud formations. In contrast, the relatively simplistic CanNet model frequently

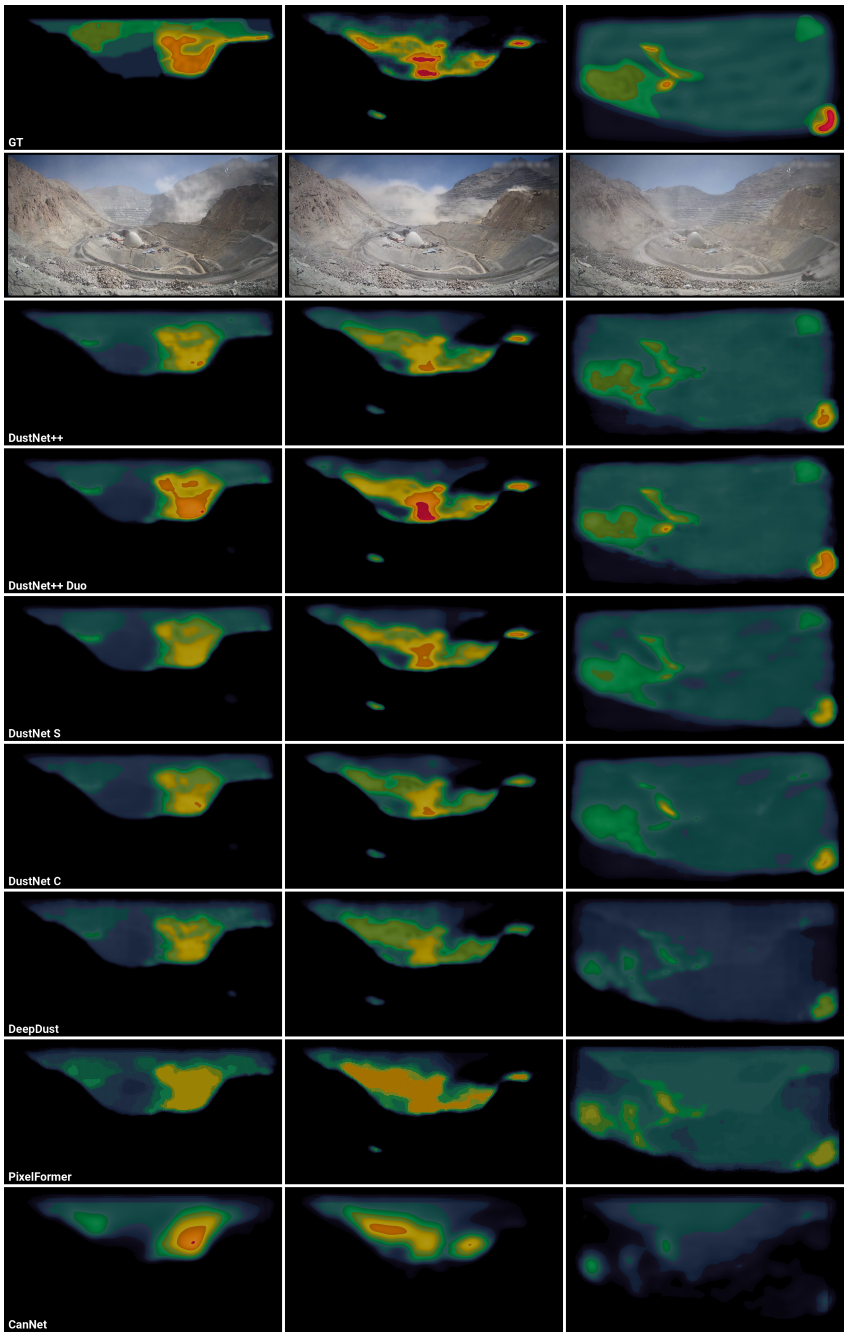


Figure 3.13: Different Mining Events. This figure shows various mining events and their associated dust estimation predictions. It is apparent that the considerably smaller CanNet has difficulties with accurate boundary localization, especially with the dust plume in the right images generated by the truck, whereas PixelFormer struggles with effective regression. DeepDust delivers improved outcomes but is readily surpassed by DustNet. Although DustNet S and DustNet++ show similar performance levels, DustNet++ Duo is clearly the best performer in terms of regression ability.

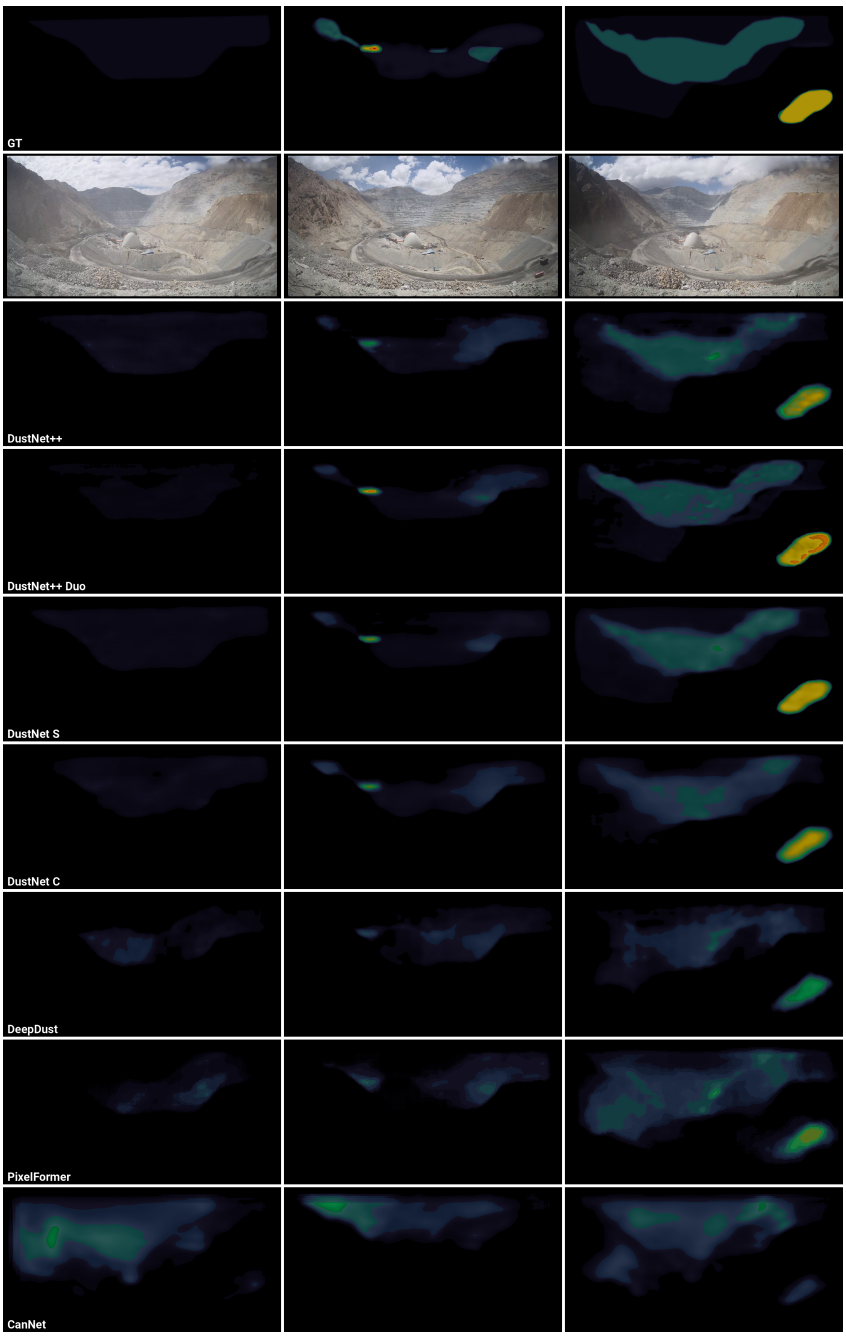


Figure 3.14: Cloudy Scenes. This figure presents the performance outcomes of models trained using the Meteodata dust dataset in scenarios involving opencast mining sites with substantial cloud cover. The results demonstrate that DustNet++ adeptly distinguishes between dust particles and cloud formations. In contrast, the rudimentary CanNet model frequently misclassifies clouds as dust. Specifically, in the image on the left, only DustNet++ and DustNet S correctly ascertain the absence of dust, while other models incorrectly detect its presence. In the central image, all models, aside from CanNet, exhibit reasonable performance. In the rightmost image, DustNet S achieves the most precise regression results in the right image, closely followed by DustNet++. The performance of the remaining models is markedly poorer by comparison.

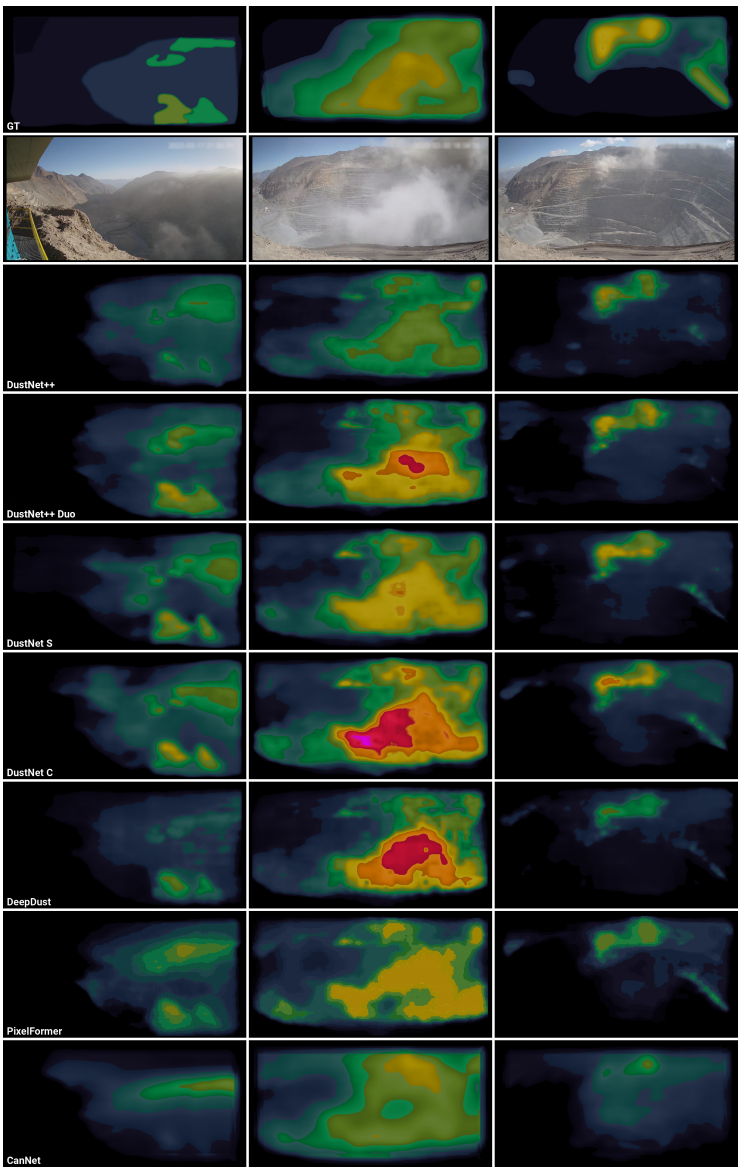


Figure 3.15: Assessment Of Model Performance On Unseen Mining Sites. This figure illustrates the outcomes from models trained using the Meteodata dust dataset on mine sites excluded from the training and validation datasets. The depicted scenes are characterized by their complexity and ambiguity. In the left image, discerning the most effective model is challenging, though it is evident that the basic CanNet model is the least effective. In the central image, the provided ground truth appears to underestimate dust concentrations, suggesting that DustNet++ may offer a more accurate assessment that exceeds these original estimates. Conversely, DeepDust tends to overestimate the dust levels. In the rightmost image, DustNet S outperforms the alternative methodologies.

misidentifies clouds as dust. In the left image, only DustNet++ and DustNet S accurately identify the absence of dust, whereas other models erroneously detect dust presence. The central image indicates that all models, except CanNet, perform

adequately. In the right image, DustNet S delivers the most accurate regression results, closely followed by DustNet++. The performance of the other models is notably inferior.

Figure 3.15 presents the performance of various models on mining sites that were not included in the training and validation datasets of the Meteodata dust project. These scenes are marked by their complexity and ambiguity. In the left image, it is challenging to discern the most effective model, though it is evident that the basic CanNet model performs the least effective. In the central image, the provided ground truth underestimates dust concentrations, suggesting that DustNet++ might offer a more accurate assessment surpassing these initial estimates. Conversely, DeepDust tends to overestimate the dust levels. In the rightmost image, DustNet S surpasses the performance of the other models.

3.8.1 Detailed Results on the Meteodata Dust Dataset

Tables 3.6 and 3.8 supplement Table 3.2 with more metrics: Accuracy, Precision, Recall, binned MSE and MAE. The findings remain very comparable to Table 3.2. In addition, Tables 3.7 and 3.9 provide detailed insights into individual MSE and MAE bins.

3.8.2 Results on the URDE Validation Dataset

Table 3.10 depicts the results from the binary dust segmentation URDE RandomDataset_897 validation dataset, consisting of 97 images. Analogous to the experiments conducted on the RandomDataset_897 training dataset, as referenced in Table 3.3, models were trained on the Meteodata dust dataset and subsequently tested with a threshold set at 35% of the value range to assess their generalization capabilities. Overall, the outcomes demonstrate a marked resemblance across both dataset variants. Furthermore, the localization proficiency of DustNet++ surpasses that of the competing methodologies.

3.8.3 Influence of the Different Dust Levels

DustNet++ utilizing a ResNet101 backbone is meticulously trained on each bin individually and on the entire dataset for 200 epochs, adhering to configurations comparable to those specified in Table 3.1. The models are trained using the

Table 3.6: **Single-Image Results.** Comparison of the best-performing density estimation methods on the Meteodata dust dataset for single images.

Model	Backbone	CPW	#Img	MAE	MSE	ØB-MAE	ØB-MSE	Acc	Pre	Rec	IoU
CanNet	VGG16	×	1	20.21	855.40	38.08	2917.52	0.787	0.798	0.790	0.648
CrowdFPN	r101	×	1	19.95	848.18	41.49	3068.95	0.786	0.797	0.790	0.659
NeWCRF	Swin-B	×	1	20.00	822.68	34.19	2254.98	0.779	0.799	0.783	0.635
NeWCRF	Swin-B	✓	1	20.83	887.67	40.61	3218.86	0.784	0.796	0.787	0.643
NeWCRF	Swin-L	✓	1	19.27	740.54	33.09	2019.21	0.792	0.809	0.796	0.654
PixelFormer	Swin-B	×	1	21.53	825.50	40.68	2864.61	0.783	0.805	0.787	0.641
PixelFormer	Swin-B	✓	1	17.43	645.54	27.41	1424.52	0.822	0.831	0.824	0.697
PixelFormer	Swin-L	✓	1	16.29	576.64	24.44	1141.12	0.833	0.841	0.835	0.713
DeepDust	CNV2-B	×	1	19.60	749.36	30.11	1640.22	0.782	0.796	0.786	0.687
DeepDust	CNV2-B	✓	1	17.11	561.89	28.28	1373.31	0.826	0.839	0.829	0.702
DeepDust	Swin-L	✓	1	15.24	482.90	27.26	1365.84	0.872	0.877	0.874	0.774
DustNet S	r101	×	1	19.27	705.47	31.35	1756.57	0.796	0.810	0.800	0.660
DustNet S	r101	✓	1	14.10	458.96	24.01	1134.22	0.861	0.868	0.864	0.756
DustNet S	Swin-L	✓	1	12.80	**402.20**	**21.19**	**941.10**	**0.885**	**0.888**	**0.886**	**0.793**
DustNet++	r101	✓	1	13.95	497.72	27.09	1395.00	0.865	0.870	0.867	0.762
DustNet++	Swin-L	✓	1	**12.63**	422.33	22.43	1039.23	0.879	0.883	0.881	0.784

Table 3.7: Detailed Single-Image Regression Results. Binned regression results of density estimation methods on the Meteodata dust dataset for single images: The values of the pixels are binned into zero dust (ZB), low dust (LB), medium dust (MB), and high dust (HB) density.

Model	Backbone	CPW	#Img	HB-MAE	HB-MSE	MB-MAE	MB-MSE	LB-MAE	LB-MSE
CanNet	VGG16	×	1	74.005	7896.82	42.655	2540.28	23.610	856.36
CrowdFPN	r101	×	1	75.311	7509.59	50.615	3229.95	30.338	1287.22
NeWCRF	Swin-B	×	1	59.809	5369.30	40.818	2371.23	25.242	954.47
NeWCRF	Swin-B	✓	1	82.638	8997.13	43.149	2622.07	24.043	877.14
NeWCRF	Swin-L	✓	1	58.703	4833.36	38.883	2085.80	23.655	845.02
PixelFormer	Swin-B	×	1	87.476	8325.33	36.052	1863.56	24.996	892.47
PixelFormer	Swin-B	✓	1	44.217	2990.18	33.335	1629.00	22.052	762.92
PixelFormer	Swin-L	✓	1	37.233	2203.08	30.220	1365.91	21.075	711.97
DeepDust	CNV2-B	×	1	47.280	3340.16	37.039	1979.88	23.933	873.30
DeepDust	CNV2-B	✓	1	46.871	2917.13	35.477	1694.30	20.061	629.20
DeepDust	Swin-L	✓	1	45.639	2982.74	37.135	1797.57	16.777	485.84
DustNet S	r101	×	1	51.202	3778.13	39.452	2152.64	22.345	754.67
DustNet S	r101	✓	1	40.524	2437.82	30.376	1379.73	16.881	514.35
DustNet S	Swin-L	✓	1	**36.064**	**2023.29**	**25.616**	1091.23	**14.631**	**428.98**
DustNet++	r101	✓	1	47.959	3081.32	36.763	1783.07	16.987	507.18
DustNet++	Swin-L	✓	1	41.204	2394.30	25.892	**1075.72**	15.552	463.08

Table 3.8: **Multi-Image Results.** Comparison of the best-performing density estimation methods on the temporal Meteodata dust dataset.

Model	Backbone	CPW	#Img	MAE	MSE	ØB-MAE	ØB-MSE	Acc	Pre	Rec	IoU
DustNet A	r101	×	3	18.77	701.73	32.21	1837.29	0.792	0.811	0.796	0.654
DustNet B	r101	×	3	26.60	1528.08	64.05	6948.25	0.667	0.754	0.677	0.481
DustNet D	r101	×	3	17.44	639.10	31.13	1767.09	0.809	0.832	0.813	0.677
DustNet C	r101	×	2	16.77	601.49	27.29	1361.71	0.814	0.829	0.818	0.685
DustNet C	r101	✓	2	12.60	413.22	20.08	872.33	0.878	0.881	0.879	0.782
DustNet C	Swin-L	✓	2	12.52	414.92	21.16	919.04	**0.880**	**0.885**	**0.882**	**0.786**
DustNet++ Duo	r101	✓	2	12.20	427.72	19.89	861.76	0.868	0.875	0.870	0.766
DustNet++ Duo	Swin-L	✓	2	**11.73**	**396.10**	**19.14**	**836.83**	0.860	0.870	0.863	0.753

Table 3.9: **Detailed Multi-Image Regression Results.** Binned regression results of density estimation methods on the temporal Meteodata dust dataset: The values of the pixels are binned into zero dust (ZB), low dust (LB), medium dust (MB), and high dust (HB) density.

Model	Backbone	CPW	#Img	HB-MAE	HB-MSE	MB-MAE	MB-MSE	LB-MAE	LB-MSE
DustNet A	r101	×	3	54.683	4076.94	40.610	2195.47	23.166	794.99
DustNet B	r101	×	3	135.613	19389.38	77.805	6698.77	34.851	1553.73
DustNet D	r101	×	3	52.328	3841.00	41.586	2295.11	22.360	742.86
DustNet C	r101	×	2	45.193	2871.32	33.294	1579.19	21.973	738.89
DustNet C	r101	✓	2	33.714	1849.49	23.348	**929.71**	**15.729**	**485.33**
DustNet C	Swin-L	✓	2	38.652	2026.50	**23.013**	942.24	15.777	494.49
DustNet++ Duo	r101	✓	2	31.861	**1662.78**	25.369	1049.18	17.278	555.68
DustNet++ Duo	Swin-L	✓	2	**31.785**	1676.82	23.262	999.39	15.970	504.80

Table 3.10: **URDE Randomdataset_897 Validation Dataset.** Following models are trained on the Meteodata dust dataset and then applied with a threshold of 35% on the Randomdataset_897 validation dataset

Model	Backbone	CPW	Acc	Pre	Rec	Dice	IoU
CanNet	VGG16	×	**0.974**	**0.700**	0.503	0.500	0.490
CrowdFPN	r101	×	0.948	0.660	0.834	0.699	0.606
NeWCRF	r101	✓	0.962	0.662	0.728	0.689	0.605
NeWCRF	Swin-L	✓	0.961	0.655	0.720	0.681	0.599
PixelFormer	r101	✓	0.933	0.625	**0.883**	0.677	0.587
PixelFormer	Swin-L	✓	0.945	0.645	0.879	0.702	0.610
DeepDust	CNV2-B	✓	0.951	0.652	0.845	0.706	0.615
DeepDust	Swin-L	✓	0.961	0.675	0.812	0.722	0.631
DustNet S	r101	✓	0.939	0.613	0.778	0.654	0.572
DustNet S	Swin-L	✓	0.959	0.662	0.786	0.705	0.616
DustNet++	r101	✓	0.851	0.543	0.722	0.543	0.470
DustNet++	Swin-L	✓	0.957	0.669	0.848	**0.724**	**0.632**

Table 3.11: **Training on Different Bins.** The pixel values of the Meteodata dust dataset are categorized into distinct dust density bins: zero dust (ZB), low dust (LB), medium dust (MB), and high dust (HB). DustNet++ is systematically trained on each of these bins using a ResNet101 backbone. It is observed that training on the comprehensive dataset generally yields superior outcomes compared to focusing on specific bins, with the exception of the ZB category.

	LB	MB	HB	All
ØB-MAE	77.45	70.53	69.84	**27.09**
ZB-MAE	10.35	0.67	**0.06**	6.63
LB-MAE	24.73	51.29	57.22	**16.99**
MB-MAE	92.55	87.25	115.03	**36.76**
HB-MAE	182.15	142.93	107.06	**47.96**
ØB-MSE	11178.723	8239.257	8212.195	**1394.996**
ZB-MSE	300.873	12.536	**3.010**	208.415
LB-MSE	971.591	3006.320	3642.044	**507.177**
MB-MSE	9635.054	8364.547	13997.481	**1783.074**
HB-MSE	33807.371	21573.625	15206.244	**3081.318**

Meteodata dust training dataset and evaluated on the validation dataset, with the outcomes presented in Table 3.11.

Training exclusively on bins with low dust (LB) values results in the least favorable mean binned MSE. While it outperforms models trained solely on medium (MB) and high dust (HB) values when evaluated on LB values, it is surpassed by the model trained on the holistic dataset. Consequently, training to encompass all bins demonstrates enhanced efficacy compared to an exclusive focus on the LB category. A similar pattern is observed with the models trained on MB and HB

values. These models excel within their respective bins but are surpassed by the holistic training approach. Thus, adopting a comprehensive training strategy across all bins consistently leads to more robust results than specializing in individual bins.

3.9 Discussion

In this section, we provide a comprehensive discussion of the chapter's main themes. We begin with a general overview, then examine the limitations of our study.

3.9.1 General Discussion

Our studies indicate a significant correlation between batch size and performance. Despite implementing Gradient Accumulation, it was not sufficient to significantly enhance the models' performance. This suggests that performance improvement may not be solely attributed to backpropagation with a larger batch size. Batch normalization could be a determinant factor in this context, particularly as the Meteodata dust dataset's ground truth often presents imprecise boundaries.

The impact of metrics in real-world applications, such as in an open-cast mine or construction site, is another essential aspect. An increase in regression ability, gauged by MAE and MSE, may be more pivotal than localization ability, supported by IoU, Accuracy, Recall, and Precision. The ambiguity of dust cloud boundaries makes minor variations in metrics like IoU less crucial. However, differences in regression ability could have a more profound effect. Intense dust events, for instance, those triggered by intentional explosions, are relatively rare but need to be accurately identified in terms of their intensity. All models struggle with such events, particularly those of medium intensity, underlining the importance of a strong regression ability.

In the single-image scenario, the DustNet variants are clearly superior to the full convolutional approaches CrowdFPN and DeepDust. DustNet++ outperformed DustNet on the original URDE training dataset (cf. Table 3.3) but slightly underperformed in terms of IoU and MSE on the Meteodata dust validation dataset. Since all methods are trained on the Meteodata dust dataset (cf. Table 3.2), the generalization ability of the method is more closely examined. In this regard, DustNet++ likely provides superior overall performance. While its regression ability in absolute terms is better, DustNet++ is more vulnerable to outliers than DustNet.

For the reasons outlined above, the evaluation can overlook the localization ability closely related to the model, making DustNet++ Duo the evident superior model.

In a real-world scenario, DustNet++'s higher inference time compared to DustNet does not pose a significant issue. Given that the processing time is well below one second (cf. Table 3.1), it is unlikely to exceed the one-second mark even on lower-power GPUs. The lower memory requirement, on the other hand, is a far superior trade-off as it allows for less powerful GPUs or potentially larger images.

3.9.2 Limitations

Our model demonstrates a tendency to overlook minor dust plumes, which we hypothesize is a trade-off for reducing false positives, ultimately leading to a lower loss during training. This trade-off, however, could be viewed as a negative byproduct of the training process. Additionally, we hypothesize that the use of the L_2 loss function may exacerbate the MSE in bins characterized by sparse occurrences. Notably, bins containing rare high dust values exhibit significantly poorer performance compared to those containing more frequently occurring lower values.

Moreover, the ground truth for the Meteodata dust dataset is not derived from physical measurements but rather relies on subjective annotations by human annotators. This method of data annotation can introduce a degree of subjectivity and variability in the ground truth. Furthermore, the boundaries of dust within this dataset are notably indistinct, leading to heightened levels of ambiguity compared to datasets utilized in classification or object detection tasks. Consequently, there are instances where our model may actually reflect the real dust conditions more accurately than the annotated ground truth.

Given these factors, a superior metric result within our testing framework does not necessarily translate to enhanced performance in practical, real-world applications. Although our model is also evaluated using the URDE binary segmentation dust dataset, a comprehensive comparison between density estimation and semantic segmentation approaches to dust remains crucial to fully understanding the capabilities and limitations of these methodologies in varying contexts.

3.10 Conclusion

In this chapter, we have proposed several techniques for estimating dust density, including CrowdFPN, DeepDust, DustNet, and DustNet++. The latter represents an advanced neural network specifically designed for dust density estimation. DustNet++ computes the dust density for each pixel within a given image by effectively harnessing and integrating local, global, and temporal information. This model not only excels in regressing various dust levels but also in differentiating dust from visually similar phenomena, such as clouds. Our proposed methodology demonstrates superior performance, surpassing all competing methods in regression capability on the Meteodata dust dataset and exhibiting improved localization abilities on the URDE dataset.

3.11 Acknowledgment

The images in the presented figures and those used for creating the Meteodata dust dataset are from the pit of Minera Los Pelambres, which collaborates with Meteodata in the advanced use of cameras for emission control strategies. The permission to use the images in this publication is kindly appreciated.

4 Enhancing Object Detection Training and Inference Techniques Across Diverse Object Densities and Scales

In the following, quotations to related publications are highlighted by colored bars placed on the outer border of the text, even though minor editorial changes have been made. The color encoding used is as follows:

- **Michel, A.**, Mispelhorn, J., Schenkel, F., Gross, W. & Middelmann, W. (2020, September). An approach to improve detection in scenes with varying object densities in remote sensing. In Image and Signal Processing for Remote Sensing XXVI (Vol. 11533, pp. 107-113). SPIE. Reprinted with permission. It is cited as [98].
- **Michel, A.**, Gross, W., Hinz, S. & Middelmann, W. (2022). ARM-NMS: Shape Based Non-Maximum Suppression For Instance Segmentation in Large Scale Imagery. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2, 291-298. Reprinted with permission. It is cited as [96].
- Kiefer, B., ..., **Michel, A.**, Gross, W. & Weinmann, M. (2024). 2nd Workshop on Maritime Computer Vision (MaCVi) 2024: Challenge Results. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 869-891). Reprinted with permission. It is cited as [66].

In this chapter, we delve into object detection techniques designed for contexts characterized by varying object densities and scales. Rather than focusing on the network architecture, like in Chapter 3, we concentrate on other critical factors: training schemes, post-processing techniques, and the proper handling of data. This process is motivated by the success of [153], which not only focuses on architecture optimization but also on the optimization of the training process. This is also true for

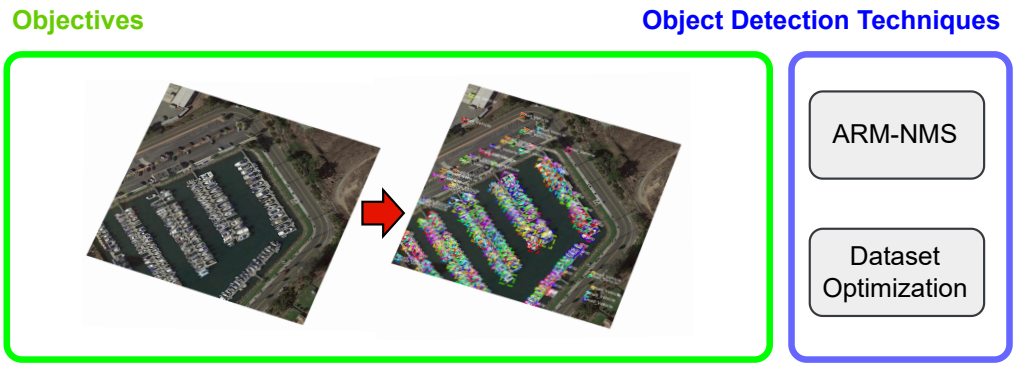


Figure 4.1: Topics of Chapter 4. In Chapter 4, we conduct an in-depth examination of methodologies for refining object detection techniques tailored to varying object densities and scales. The primary objective is to achieve robust monitoring capabilities in scenes characterized by these variations. This includes dataset optimization and the application of advanced Non-Maximum Suppression techniques.

the inference. Training and inference optimization techniques are particularly significant in the field of remote sensing, where precise identification and classification of objects are crucial for a multitude of applications, including environmental monitoring, urban planning, disaster management, and object counting. An overview of the chapter is illustrated in Figure 4.1.

We begin by enhancing object detection in the context of fluctuating object densities and scales using a post-processing technique called Area Rescoring Mask Non-Maximum Suppression (ARM-NMS) [96]. This approach does not require modifications to the underlying architecture but focuses on advanced post-processing techniques. We achieve superior detection outcomes by intelligently filtering detected objects based on their corresponding masks and incorporating size considerations into the decision-making process. This enhancement not only streamlines the post-processing phase but also ensures that the final detections are more robust and reliable, thereby advancing object detection in scenarios with varying object densities and scales.

Next, we demonstrate the importance of employing appropriate training schemes when working with a maritime dataset characterized by objects of varying densities and scales. Our technique enhances detection performance during inference by leveraging multiple representations of the input data, all without modifying the model's architecture.

With these two techniques, we want to demonstrate the importance of non-architectural optimization techniques, especially for different object densities and scales.

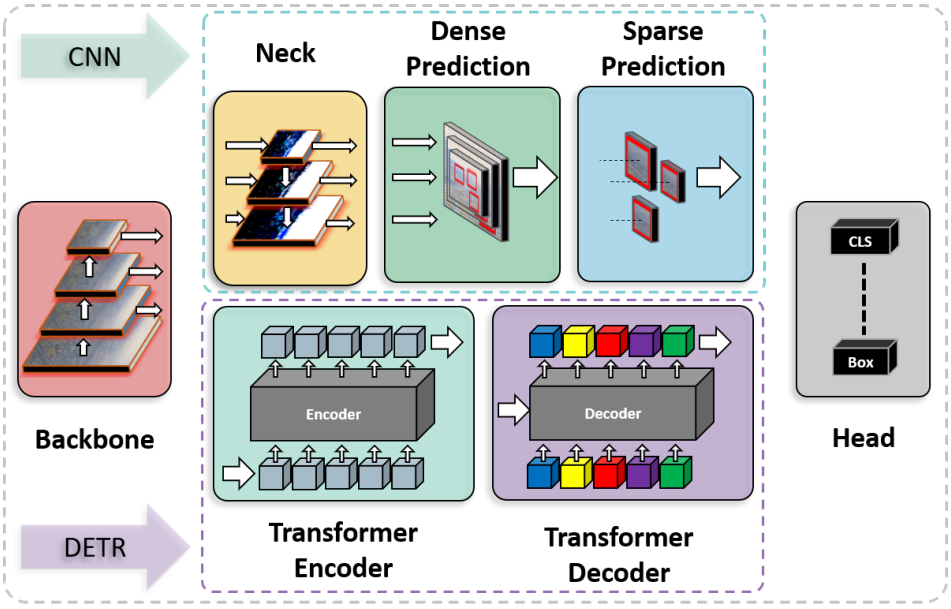


Figure 4.2: Object Detection Building Blocks. This figure depicts the fundamental components of deep learning-based object detection systems. It highlights the two primary variants: CNN-based architectures and DETR-based architectures, showcasing their distinctive structural elements and functionalities.

4.1 Related Work

This section outlines the foundational works in the field of object detection that inform the discussions in this chapter. Object detection can be divided into two primary subtasks: the localization of objects within an image and their classification. We will first examine the various object detectors, detailing their methodologies and advancements. Following this, we will explore filtering algorithms that are crucial for enhancing the effectiveness of object detection systems.

4.1.1 Object Detection

Object detection methods are generally categorized into one-stage and two-stage detectors. One-stage detectors [117, 87, 83, 8] employ a dense prediction head, which typically results in greater time efficiency. In contrast, two-stage detectors [45, 44, 119, 113] tend to achieve higher accuracy by utilizing an additional sparse prediction head. Contemporary object detection techniques are often class-aware, providing both localization information and estimated counts of objects.

Faster R-CNN [119] is one of the most widely used two-stage object detectors. Its architecture comprises a backbone network, a class-agnostic Region Proposal Network (RPN), and a sparse prediction head. Moreover, Faster R-CNN can be seamlessly upgraded to Mask R-CNN [47] by integrating a mask head into its structure, enabling the tasks of bounding box prediction and instance segmentation through the generation of object masks.

A critical component of various object detection algorithms is the use of anchor boxes. These anchors, which consist of windows of diverse shapes and scales, process feature maps through a sliding window approach to estimate bounding boxes for object detection. While some methods employ fixed anchor shapes [47], others allow for learned shapes [169]. Typically, anchors are distributed uniformly and densely, resulting in high computational complexity. However, methods such as the guided anchor [154] demonstrate superiority over the baseline RPN, achieving higher evaluation results with fewer generated anchors. In natural scenes, the variability in object scale and density often results in false positives and missed detections. For instance, objects such as ships can vary dramatically in size. Feature Pyramid Networks (FPNs) [82] can partially alleviate scaling issues. Moreover, urban environments tend to have a higher density of vehicles per square kilometer compared to rural areas, presenting significant challenges to object detectors, particularly in scenes with high object concentrations.

The introduction of the Detection Transformer (DETR) by Carion et al. [12] signifies a notable shift in object detection methodologies. DETR leverages the transformer model to streamline the traditional object detection pipeline, moving away from reliance on hand-designed components such as anchors and Non-Maximum Suppresion (NMS). Instead, it utilizes the self-attention mechanism inherent to transformers to directly predict object classes and bounding boxes, thereby eliminating the need for anchor generation and region proposals. Subsequent advancements, including a sparse attention mechanism [181], a contrastive approach for training denoising [176], and the integration of multiple parallel auxiliary heads [182], have further optimized the performance of DETR. An additional advancement involves integrating a mask branch into the DINO framework [79], aimed at enhancing the performance of instance segmentation tasks.

Figure 4.2 illustrates the fundamental building blocks of an object detector. It addresses the two main variants: CNN-based and DETR-based architectures. Both versions incorporate a multi-scale feature extractor and heads. The CNN typically employs FPN structures. Following this, prediction occurs along with a sparse prediction module for two-stage detectors. In contrast, the DETR variant utilizes a

full transformer. When the queries from the transformer are exclusively learned, it is described as a one-stage approach. However, if additional information from the encoder contributes to the generation of the queries, it is classified as a two-stage model.

4.1.2 Non-Maximum Suppression

This section surveys the most relevant works for NMS. NMS is an integral part of many algorithms for object detection. It is rooted in edge detection techniques [125], and further developments lead to the following algorithms.

Greedy-NMS. Greedy-NMS [28] remains a widely adopted method in state-of-the-art object detectors for filtering out unnecessary object proposals. The core concept of Greedy-NMS is that bounding boxes with high detection scores suppress overlapping boxes with lower scores. The algorithm begins by sorting the bounding box detections $\mathbf{b} \in \mathcal{B}$ according to their respective scores $s \in \mathcal{S}$ in descending order. In an iterative process, the bounding box $\mathbf{h}_b$ with the highest score $s_{\max}$ is moved to the list of final detections $\mathcal{F}$.

To determine if a bounding box is eligible for elimination, the Intersection over Union (IoU) scores between the selected box $\mathbf{h}_b$ and the remaining boxes are calculated. The new score $s_i \in \mathcal{S}$ for the i-th remaining detection $\mathbf{b}_i \in \mathcal{B}$ is defined as follows:

$$s_i \leftarrow \begin{cases} s_i, & \text{if IoU}(\mathbf{h}_b, \mathbf{b}_i) < n_t \\ 0, & \text{if IoU}(\mathbf{h}_b, \mathbf{b}_i) \geq n_t \end{cases} \tag{4.1}$$

where $n_t \in [0, 1]$ is the specified threshold. This procedure continues until all initial detections in $\mathcal{B}$ have been processed. While Greedy-NMS is efficient and popular, its reliance on a hard threshold n_t can introduce errors; a high n_t may retain many false-positive bounding boxes, while a low n_t may fail to eliminate them effectively.

Soft-NMS. Soft-NMS [9] was developed to overcome the limitations of Greedy-NMS. Instead of removing overlapping bounding boxes, Soft-NMS reduces the scores of lower-scored boxes that overlap with the bounding box $\mathbf{h}_b$ that has the maximum score. This approach leads to a more continuous penalty function and enhances the average precision of the detection results. However, because overlapping boxes are not removed from $\mathcal{B}$, the computational complexity of Soft-NMS is slightly higher

compared to Greedy-NMS. The Gaussian penalty function used in Soft-NMS can be expressed as:

$$s_i \leftarrow s_i e^{-\frac{\text{IoU}(\mathbf{h}_b, \mathbf{b}_i)^2}{\sigma}}, \quad \forall \mathbf{b}_i \notin \mathcal{F}, \tag{4.2}$$

where σ is a less sensitive threshold parameter compared to n_t. Similar to Greedy-NMS, this update rule is applied iteratively until all detections from $\mathcal{B}$ drop below a certain threshold or are transferred to $\mathcal{F}$.

Further Developments. Recent advancements such as Softer-NMS [50], Adaptive-NMS [86], and IoU-Net [62] modify the object detection model but require extensive retraining for marginal performance improvements. Methods like Cluster-NMS [179] aim to accelerate the NMS process while performing comparably to Greedy-NMS. Another noteworthy method is Weighted Boxes Fusion (WBF) [138], which merges overlapping boxes into new boxes instead of eliminating or rescoring them. WBF shows exceptional performance in test-time augmentation settings, although it underperforms compared to Soft-NMS in conventional scenarios.

As a result, Greedy-NMS and Soft-NMS remain among the most frequently used methods in object detection. However, all discussed methods primarily focus on bounding boxes rather than shapes. Recently, several approaches have emerged that utilize shape-based NMS, such as Mask-based NMS [158] or Matrix-NMS [159]. The objective of this chapter is to enhance the effectiveness of shape-based NMS methods.

4.2 Datasets

This section provides a comprehensive description of the datasets employed in this chapter. To demonstrate the generalization capabilities of the methods used, a selection of diverse datasets has been utilized. Firstly, we examine the maritime LaRS [183] object detection dataset, which serves as a crucial benchmark for evaluating obstacle detection algorithms in maritime environments. Subsequently, we discuss the VisDrone [33] dataset, a drone-based detection dataset that presents unique challenges associated with aerial imagery captured from unmanned aerial vehicles. Finally, we explore the iSAID [160] dataset, an aerial and satellite imagery dataset that extends the scope of our analysis to include airborne and spaceborne remote sensing data. By leveraging this variety of datasets, we aim to highlight the

robustness and adaptability of the proposed methods across different domains and acquisition conditions.

4.2.1 LaRS Dataset

LaRS (Lakes, Rivers, and Seas) is a object detection dataset [183] developed to advance research in maritime obstacle detection. The data were acquired employing a wide variety of both consumer-grade and industry-grade RGB cameras. It comprises over 4,000 per-pixel labeled key frames, each accompanied by nine preceding frames to enable the utilization of temporal texture, amounting to a total of over 40,000 frames. Each key frame is annotated with 8 'thing' classes, 3 'stuff' classes, and 19 global scene attributes. We present the results of applying 27 semantic and panoptic segmentation methods to this dataset, along with several performance insights and future research directions. LaRS boasts the largest diversity among related datasets in terms of recording locations, scene types, obstacle classes, and acquisition conditions.

4.2.2 VisDrone-Det 2019

The Vision Meets Drone Object Detection (VisDrone-DET2019) dataset [33] consists of 10,209 images with ten categories. The images are acquired from drone platforms without specific details provided. Object categories include classes such as 'person', 'car', and 'bicycle' from different viewing angles. In our experiment, we do not reduce the number of classes. Experiments conducted in this chapter rely on a training on the dedicated training dataset, which contains 6,471 images. The validation dataset contains 548 images and we evaluate the training progress on the test-dev dataset, consisting of 1,610 images.

4.2.3 iSAID Dataset

The Instance Segmentation in Aerial Images Dataset (iSAID) [160] is a large-scale aerial image dataset and is derived from the Object deTection in Aerial images (DOTA) [166] dataset. To mitigate sensor biases, images are collected from a variety of sensors and platforms, including Google Earth. iSAID contains 2,806 high-resolution images collected from multiple sensors and platforms, including 655,451 object instances of 15 categories. The object categories vary from mobile categories

like 'ship' or 'car' to static categories like 'storage tank' or 'bridge'. Every category consists of a large number of instances. Furthermore, the dataset exhibits a high object scale variation. The full list of classes can be seen in Figure **??**. For example, the ship class ranges from small boats to large aircraft carriers. The distribution of the objects in a given scene is often imbalanced and uneven to represent real-life conditions. Due to performance issues of the COCO API with a large number of objects, we select only images up to a maximum of 1500 ground-truth objects, which are 439 of 458 images from the iSAID validation dataset.

4.3 Effective Postprocessing Strategies for Enhanced Object Detection Inference

Detecting objects in aerial scenes is an essential and critical task in the field of remote sensing. However, state-of-the-art object detectors often produce correlated scores among neighboring detections, leading to an increased number of false positives. Moreover, performing detection on large-scale images typically necessitates a tiling scheme that employs overlapping windows, further exacerbating the issue of duplicate detections. To mitigate these challenges, a non-maximum suppression (NMS) approach is commonly integrated into the detection pipeline. NMS effectively suppresses overlapping detections based on their scores. Current NMS algorithms typically filter detections by assessing their corresponding bounding boxes. However, this method assumes that comparing bounding boxes to evaluate the overlap of non-rectangular objects entails a certain degree of inaccuracy. In response to this limitation, we propose Area Rescoring Mask-NMS (ARM-NMS), which utilizes object shapes for filtering. ARM-NMS leverages instance masks instead of conventional bounding boxes to eliminate redundant detections, thus enhancing the accuracy of the detection process without necessitating retraining of instance segmentation pipelines. Although ARM-NMS does not incorporate a density estimation approach, it is particularly well-suited for improving object detection results, especially in scenarios characterized by varying object densities and scales.

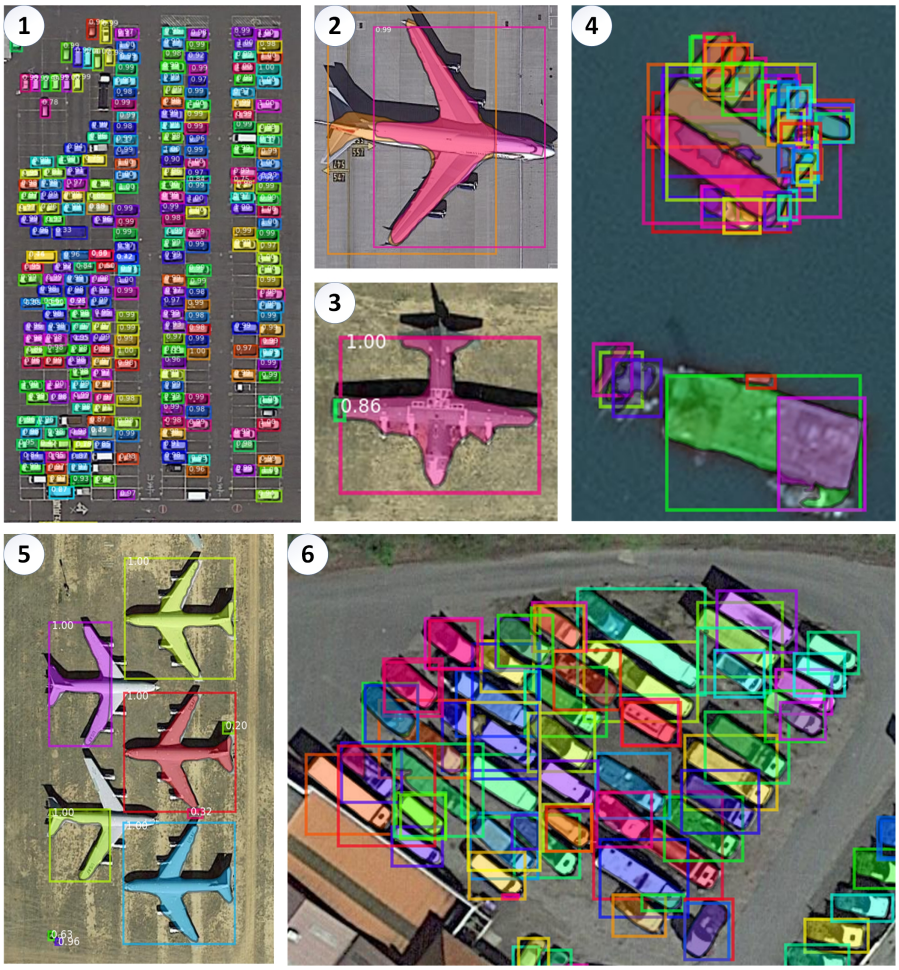


Figure 4.3: Challenges for NMS.
(1) Ideal case for box-based NMS. (2) Double detections. (3) Partial detections. (4) Cluster of objects. (5) Overlapping concave objects. (6) Diagonally-aligned objects.

4.3.1 Introduction

The expected input of object detectors is usually limited by pixel resolution [120, 117]. Input images with a higher resolution than the maximum resolution defined by detector architectures are usually resized. Resizing can lead to a loss of information or result in scaling issues. An alternative to resizing is to partition large-scale images into patches on which the detectors operate separately. Fusing the discrete detections into the original scenes leads to additional challenges. Utilizing non-overlapping patches can lead to missing detections along the borders. In contrast, overlapping patches result in duplicate detections, which the NMS can address.

In this work, we propose an improved approach to using bounding boxes by utilizing object shapes and their corresponding size. However, filtering overlapping detections comes with many difficulties. Figure 4.3 illustrates some of the main challenges for NMS:

- **The ideal case** for box-based NMS involves rectangular, non-overlapping objects that are aligned parallel to the borders of the image. In this scenario, the proposed bounding boxes can precisely encompass the present objects.
- **Overlapping detections** are the default-case addressed by NMS. Ideally, the better-aligned detection exhibits a higher confidence score and suppresses neighboring scores with lower confidence. However, if both detections have a similar score or the worse-aligned detection scores higher, the elimination process can lead to errors in the final detections. Also, most applied NMS algorithms filter each class separately, which leads to double detections with different labels being ignored by NMS.
- **Partial detections** are particularly hard for NMS. NMS utilizes intersection over union (IoU) as an overlap metric. IoU is size-independent, which is generally considered a positive characteristic. However, in the case of detections that differ strongly in size, the IoU between both is low, and thus partial detections are often overlooked by NMS.
- **Cluster of objects** is another issue for NMS algorithms. Many overlapping objects can lead to false detections due to inaccuracies in the alignment of the detection boxes.
- **Overlapping concave objects** are a significant issue of box-based NMS. The detection boxes are usually not well-aligned to the underlying objects, and the detection of neighboring objects can be suppressed.
- **Diagonally-aligned objects** in high object densities exhibit a substantial overlap of detected bounding boxes without actual overlapping objects. This can easily lead to wrongfully eliminated detections.

In order to address these cases and inspired by Mask-NMS [158], we propose the shape-based non-maximum suppression algorithm Area Rescoring Mask-NMS (ARM-NMS). ARM-NMS cannot be directly used in object detection algorithms, but object detection can be easily expanded to instance segmentation by predicting the shapes of the detected objects. Typical representatives of instance segmentation are Mask R-CNN [47] and DetectorRS [114]. Our proposed method does not require any retraining and can thus easily be implemented into existing instance segmentation pipelines. Additionally, we provide a solution to keep the computational complexity at an acceptable level, even with the more elaborate task of comparing shapes instead of boxes. Furthermore, we examine an additional overlap metric and further

modifications to NMS.

Contributions. (1) We propose an improved shaped-based NMS approach for instance segmentation, ARM-NMS. Our approach does not require any retraining and can be easily integrated into instance segmentation pipelines. (2) We propose an area-rescoring strategy which considers the size of an object in order to reduce partial detections. Area-rescoring can be easily integrated into ARM-NMS. (3) In order to display the effectiveness of ARM-NMS, we compare our method with box-based Greedy-NMS and Soft-NMS on the instance segmentation dataset iSAID.

4.3.2 Methodology

The basic idea behind ARM-NMS is quite intuitive: Use shapes instead of boxes to filter unnecessary detections. Shapes can align better with the underlying objects than just bounding boxes and lead to a more precise filtering process. However, comparing masks is a more computationally complex task than matching boxes. Hence, adjustments to the traditional NMS methods for an efficient workflow are required. Furthermore, we apply additional improvements to increase the filter performance. The pseudocode of our proposed algorithm can be seen in Figure 4.4.

ARM-NMS. ARM-NMS is computed on a list of detected instance masks $\mathcal{M} = \{\mathbf{m}_1, ..., \mathbf{m}_N\}$ with corresponding scores $\mathcal{S} = \{s_1, ..., s_N\}$ for a given scene with N detections. The first step is to adjust the scores after their relative pixel size $\mathcal{A} = \{a_1, ..., a_N\}$. We hypothesize the following: larger detections are probably more accurate than smaller ones with the same confidence score. We follow this hypothesis as larger detections are based on more pixels and, consequently, on more information. Ideally, area-rescoring leads to the suppression of partial detections and preserves complete detections. In the first step, the areas of the instance masks are calculated separately. Then, we compute the mean area $\overline{\mathcal{A}}$ of the instance masks. The area-rescoring function can be described as follows:

$$s_i \leftarrow \frac{s_i a_i}{\overline{\mathcal{A}}}^{w_a^{-1}}, \tag{4.3}$$

where the parameter w_a controls the impact of the corresponding area a_i. As a result, detections with a larger area are assigned higher corresponding scores and smaller areas result in a score reduction. After the area-rescoring, we iterate through the list of detections in a descending order regarding their scores. The detection $\mathbf{h}_m$ with the highest score s_{max} is removed from the list $\mathcal{M}$ and added to the list of final detections $\mathcal{F}$. Likewise, we utilize $\mathbf{h}_m$ to remove further detections from $\mathcal{M}$. Similar to box-based NMS approaches, we are looking for detections $\mathbf{m}_i$ which overlap with

$\mathbf{h}_m$. Yet comparing shapes is a more complex task, and therefore, we compare $\mathbf{h}_m$ only with detections $\mathbf{m}_i$ within a specified distance d_t. For these detections, we apply the Gaussian Soft-NMS penalty function to reduce the score of the detections with respect to their shape-based IoU. We repeat this step until $\mathcal{M}$ is empty. Finally, we return the final detection list $\mathcal{F}$ with their corresponding scores $\mathcal{S}'$.

4.3.3 Implementation Details

Our proposed method is flexible and can be easily adjusted. The pseudocode is illustrated in Figure 4.4. We have implemented the following variations:

- **Shapes:** Two-dimensional object shapes can be described as a binary mask, a run-length encoded mask, or as a polygon. State-of-the-art instance segmentation pipelines usually output a binary mask in the size of the original image for every detected object. This can easily lead to a computational bottleneck for large-scale images with a high object density. Consequently, a compression method is needed. Merging all binary shapes into a single one-hot encoded map is not feasible due to overlapping masks. In contrast, run-length encoded masks for every object can be an efficient approach to store and compare object shapes. For example, the MS COCO API [84] uses run-length encoding for evaluation purposes. Another procedure is to transform the masks into polygons. Polygons can be stored efficiently, are easier to visualize than run-length encoded masks, and can be comfortably implemented in other applications. As a result, we utilize polygons to store and calculate the overlap of object shapes.

- **Classes:** Our proposed method can be applied individually to each object category or jointly to all of them. Using our proposed approach jointly can improve detection accuracy if many double detections occur across different classes.

- **Distance:** In order to reduce the computational complexity, we compare only objects if their distance d_t is smaller than a certain radius. We use the Euclidean distance between the center points of the detections to calculate the distance. Other alternatives would be using only the first vertex of the polygon to save further computational steps or utilizing bounding boxes to determine

Input: $\mathcal{M} = \{\mathbf{m}_1, ..., \mathbf{m}_N\}$, d_t, w_a
$\mathcal{S} = \{s_1, ..., s_N\}$, σ, n_t
$\mathcal{M}$ is the list of initial detections masks
$\mathcal{S}$ contains corresponding detection scores
w_a influences the area rescoring
d_t is the distance threshold
n_t is the NMS threshold for Greedy-NMS
σ is the Gaussian parameter for Soft-NMS

begin
  $\mathcal{F} \leftarrow \{\}$;
  $\mathcal{A} \leftarrow area(\mathcal{M})$;
  $\overline{\mathcal{A}} \leftarrow mean(\mathcal{A})$;
  for s_i *in* $\mathcal{S}$ **do**
    $s_i \leftarrow rescore(s_i, a_i, \overline{\mathcal{A}}, w_a)$
  end
  Area-Rescoring
  while $\mathcal{M} \neq$ *empty* **do**
    $\mathbf{s}_{max} \leftarrow \text{argmax}\, S$;
    $\mathbf{h}_m \leftarrow \mathbf{m}_{s_{max}}$;
    $\mathcal{F} \leftarrow \mathcal{F} \bigcup \mathbf{h}_m$;
    $\mathcal{M} \leftarrow \mathcal{M} - \mathbf{h}_m$;
    for $\boldsymbol{m}_i$ *in* $\mathcal{M}$ **do**
      if $distance(\boldsymbol{h}_m, \boldsymbol{m}_i) < d_t$ **then**
        if $IoU(\boldsymbol{h}_m, \boldsymbol{m}_i) \geq n_t$ **then**
          $\mathcal{M} \leftarrow \mathcal{M} - \mathbf{m}_i$;
          $\mathcal{S} \leftarrow \mathcal{S} - s_i$
        end
        Greedy-NMS
      end
    end
  end
  return $\mathcal{F}, \mathcal{S}$
end

Figure 4.4: The Pseudocode. Our proposed approach can, as shown, be used in conjunction with Greedy-NMS, but it can also be replaced by Soft-NMS. The main difference is utilizing instance masks $\mathcal{M}$ instead of bounding boxes $\mathcal{B}$. The pseudocode in the blue box exhibits additional modifications. Area-rescoring leads to favoring larger areas in order to select better-aligned detections. The distance function decreases the computational complexity.

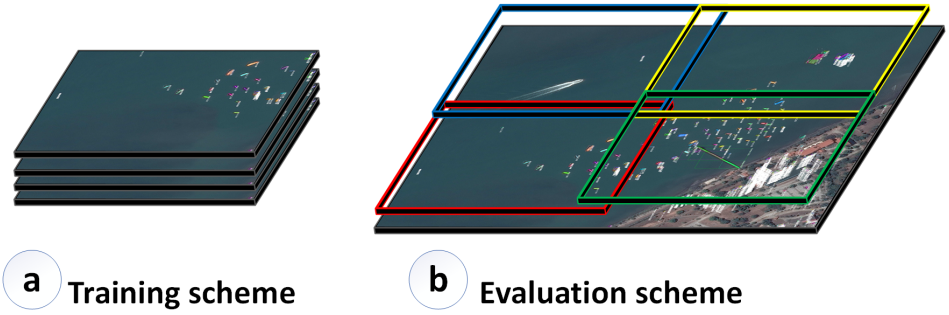


Figure 4.5: Tiling Schemes. We train our model on independent image patches, but for the evaluation process, we produce detection lists on overlapping patches.

proximity.

- **Overlap metric:** IoU is still the primary metric to compare the overlap between two shapes. IoU can be formulated as follows

$$\text{IoU} = \frac{\mathbf{m}_i \cap \mathbf{m}_j}{\mathbf{m}_i \cup \mathbf{m}_j}, \quad \mathbf{m}_{i,j} \in \mathcal{M}, \tag{4.4}$$

where $\mathbf{m}_i$ and $\mathbf{m}_j$ are two shapes, and we divide their intersection by their union. One of the most suitable characteristics of IoU is that it is scale-independent. This can lead to problems with partial detections. The IoU of a partial detection is usually low and may, therefore, be overlooked. For the case of an accumulation of partial detections, we implement an additional metric: IoMin. IoMin can be described as

$$\text{IoMin} = \frac{\mathbf{m}_i \cap \mathbf{m}_j}{min(a_i, a_j)}, \quad \mathbf{m}_{i,j} \in \mathcal{M}, \; a_{i,j} \in \mathcal{A}. \tag{4.5}$$

The difference between IoU and IoMin is that we divide the intersection of both shapes by the area of the smaller shape instead of the union.

4.3.4 Experiments

In order to demonstrate the effectiveness of our approach, we apply Mask R-CNN on the dataset iSAID to get lists of detections for large-scale images. We use these lists of detections to evaluate our method and compare them to box-based Greedy-NMS and Soft-NMS. For the evaluation, we use a considerable parameter grid. Furthermore, we show in the ablation study that our improvements to Mask-NMS are reasonable.

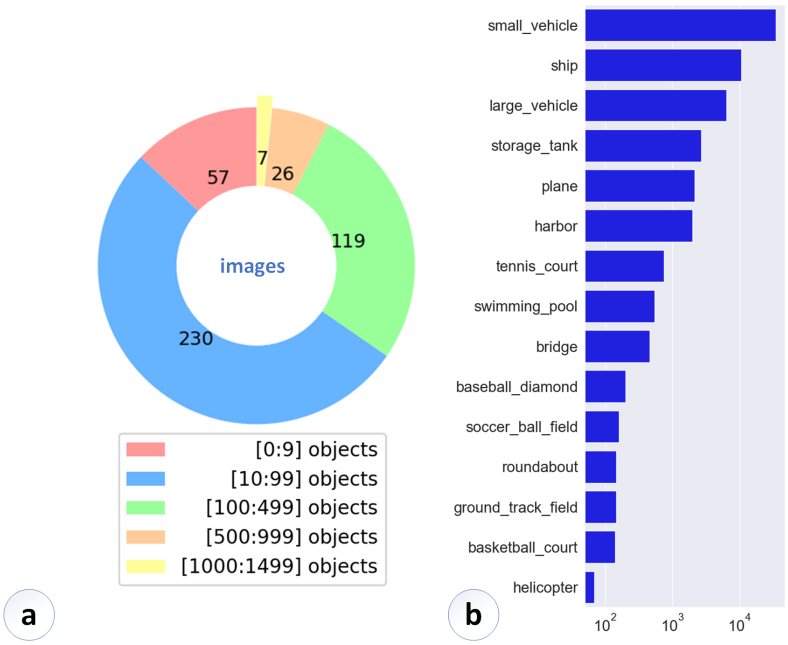


Figure 4.6: Density and Class Distribution. This analysis focuses on the iSAID validation dataset, selecting all images containing fewer than 1500 objects for evaluation. (a) The chart displays the number of images corresponding to specified object count intervals. (b) The bar plot illustrates the class distribution of the evaluation dataset, presented on a logarithmic scale for enhanced clarity.

Detection lists with shape information are required to comprehensively evaluate our proposed method. For this, we utilize the popular instance segmentation method Mask R-CNN [47]. For training the network, we crop the iSAID images into patches with a size of 800×800. We split the original iSAID training dataset randomly into a training dataset and a validation dataset. 95% of the images are used for training, the rest for validation. We train Mask R-CNN on all 15 categories of iSAID on two Nvidia V100 graphic processing units without pre-training except for the backbone. As a backbone, we use an ImageNet [127] pre-trained ResNet101 [48] to extract features for further processing steps. We utilize the default anchor configuration following [47]. Nevertheless, we modify hyperparameters that limit the amount of maximum detections per image. Hence, we significantly increase the number of possible detections per image to 1000. During training, we start with a learning rate of $\alpha = 3 \cdot 10^{-3}$ and uniformly reduce α to 10^{-3} over 50 epochs. We choose a batch size of 16 and utilize the AdamW optimizer [171] with a weight decay of 10^{-4}.

4.3.5 Evaluation Details

Table 4.1: Evaluation Parameter Grid. Overview of the key parameters used in our study.

method	[shape-based, box-based]	
penalty function	Soft-NMS	Greedy-NMS
IoU threshold	-	[0.01, 0.25, 0.5, 0.75]
σ	[0.1, 0.2, 0.3]	-
score threshold	[0.01, 0.1, 0.2, 0.5]	
unique labels	[True, False]	
overlap metric	[IoU, IoMin]	
area-rescoring	[True, False]	

To demonstrate the effectiveness of our method, we compare our approach to the widely used Greedy-NMS and Soft-NMS in large-scale images. We perform the experiments on the iSAID validation dataset. Instead of evaluating our method on images tiles, we test on the full images. One example is given in Figure 4.5. However, our used object detector is trained on image tiles with a size of 800×800 pixels. Consequently, we utilize the object detector in a sliding-window approach with a kernel size of 800×800 pixels and a stride of 600 pixels in each direction. The resulting overlap of 200 pixels is quite large, but it ensures that no object is overlooked or divided into two parts because it lies on the border between two tiles. The produced binary masks are transformed into polygons, and their local vertices are adjusted to their global location in the full image. For performance evaluation, we utilize the common metrics represented by mean Average Precision (mAP) and mean Average Recall (mAR), which are implemented by the COCO API [84].

4.3.6 Parameter Grid

NMS methods are parameter-sensitive approaches. Table 4.1 shows the used parameters and the performed method variations. Fields marked with an '-' are not considered for the evaluation. We implement Greedy-NMS and Soft-NMS box-based and also integrate them into our shape-based ARM-NMS. For Greedy-NMS, we apply different IoU thresholds n_t, which range from 0.001 to 0.75. In the case of Soft-NMS, we use a σ of 0.1, 0.2, and 0.3. Additionally, we apply different score thresholds, ranging from 0.01 to 0.5. Furthermore, we perform additional variations for the shape-based approaches. We filter each class separately and jointly. Additionally, we utilize IoU and IoMin as overlap metrics. Finally, we examine the effectiveness of the proposed area-rescoring. In addition, we set the distance threshold d_t to 250, and we filter for each unique label both separately and in conjunction.

Table 4.2: **NMS Results on the iSaid Dataset.** We compare our proposed shape-based ARM-NMS with box-based NMS.

metric	segmentation						box					
method	none		ARM-NMS		Box-NMS		none		ARM-NMS		Box-NMS	
penalty function	-	-	soft	greedy	soft	greedy	-	-	soft	greedy	soft	greedy
overlapping tiles	False	True	True	True	True	True	False	True	True	True	True	True
IoU threshold	-	-	-	0.5	-	0.5	-	-	-	0.25	0.1	0.5
sigma	-	-	0.1	-	0.1	-	-	-	0.1	-	-	-
score threshold	-	-	0.01	0.01	0.01	0.5	-	-	0.01	0.01	0.01	0.5
unique labels	-	-	True	True	True	True	-	-	True	True	True	True
overlap metric	-	-	IoU	IoMin	-	-	-	-	IoU	IoU	-	-
area-rescoring	-	-	True	True	-	-	-	-	True	True	-	-
AP [0.50:0.95]	24.6	20.7	**28.6**	28.2	25.3	25.2	28.4	24.2	**32.9**	32.4	29.4	29.3
AP 0.50	48.4	38.6	**53.5**	52.5	48.5	49.0	52.1	41.8	**57.1**	56.0	52.5	52.8
AP 0.75	22.2	19.3	**26.9**	26.5	23.1	22.6	27.5	25.2	**34.0**	33.6	29.6	29.1
AP[0.50:0.95] small	16.1	13.5	**18.9**	18.8	16.6	16.6	23.0	20.9	**25.7**	25.2	23.6	23.6
AP [0.50:0.95] medium	28.3	22.6	**32.0**	31.7	28.2	28.2	30.7	24.7	**35.6**	35.2	31.4	31.3
AP [0.50:0.95] large	31.5	28.5	**35.6**	34.9	33.5	33.6	29.5	25.0	**37.0**	36.5	32.7	32.6
AR [0.50:0.95] @100	30.0	31.3	**32.1**	31.1	31.6	31.4	33.4	35.2	**36.0**	35.1	35.3	35.0
AR [0.50:0.95] @1000	34.9	**37.9**	37.0	35.7	36.8	36.5	39.4	**43.5**	42.2	41.0	41.8	41.4
AR [0.50:0.95] @1500	35.0	**38.1**	37.1	35.7	36.9	36.6	39.5	**43.6**	42.2	41.1	41.8	41.5
AR [0.50:0.95] small	25.9	**27.7**	26.9	25.8	26.8	26.8	31.5	**33.7**	32.7	31.4	32.5	32.5
AR [0.50:0.95] medium	36.1	**38.7**	38.0	37.2	37.7	37.4	39.3	**43.7**	42.5	41.7	42.2	41.9
AR [0.50:0.95] large	39.9	**44.7**	43.8	42.1	43.4	43.0	42.6	**49.6**	48.1	46.8	47.4	46.9

4.3.7 Results

In this section, we evaluate the effectiveness of our shape-based NMS. First, we show the quantitative results of the box-based and shape-based ARM-NMS approaches. Then, we discuss some qualitative results. Finally, we display the ablation study.

Quantitative Results. In Table 4.2, we compare shape-based and box-based NMS approaches on detection lists produced by Mask R-CNN on the iSAID validation dataset. Fields with an '-' are considered irrelevant for the corresponding method. We apply the COCO-style mAP and mAR for segmentations and boxes. The unfiltered detection results are labeled 'none' under the method row and are used as a baseline. For each case, segmentation, and box, the average precision is generally higher for the unfiltered detections without overlapping tiles. In contrast, for the non-overlapping tiles, the detections indicate a higher average recall. This is reasonable because overlapping tiles lead to double detections, but they reduce the possibility of missing detections on the border of the tiles. On average, the mask-shaped approaches are 2.8 times slower than the box-shaped ones.

The results seen here are selected from experiments on a comprehensive parameter grid and for each method. We present the best results regarding overall mAP. Regarding the metric for the segmentations, we can see that our proposed shape-based approach with a Gaussian soft-max penalty function achieves not only a higher average precision than the unfiltered detections but can beat the box-based alternative by 3.3 points. The implementation with the Soft-NMS performs better in all metrics than the Greedy-NMS variant. Our approach has a slightly lower average recall for a max detection limit of 1500 instances than the unfiltered detections with overlapping tiles but obtains a higher average recall than those without overlapping tiles. However, the average precision improves by 7.9 points compared to the unfiltered detection with overlapping tiles and by 4.0 points without overlaps. Furthermore, ARM-NMS accomplishes a higher average recall than the box-based NMS methods. The best combination of parameters and variants for our method is using a Gaussian penalty Soft-NMS function, detections with overlapping tiles, a sigma of 0.1, a score threshold of 0.01, filtering for each label separately, IoU as a comparison metric and with area-rescoring. A similar result can be seen for the box metrics. Our method accomplishes the highest mean average precision while reaching, except for the unfiltered overlapping baseline, the highest mean average recall score.

Qualitative Results. We show a few qualitative results in Figure 4.7 and Figure 4.8. The detection results on entire images are filtered by ARM-NMS. Furthermore, we

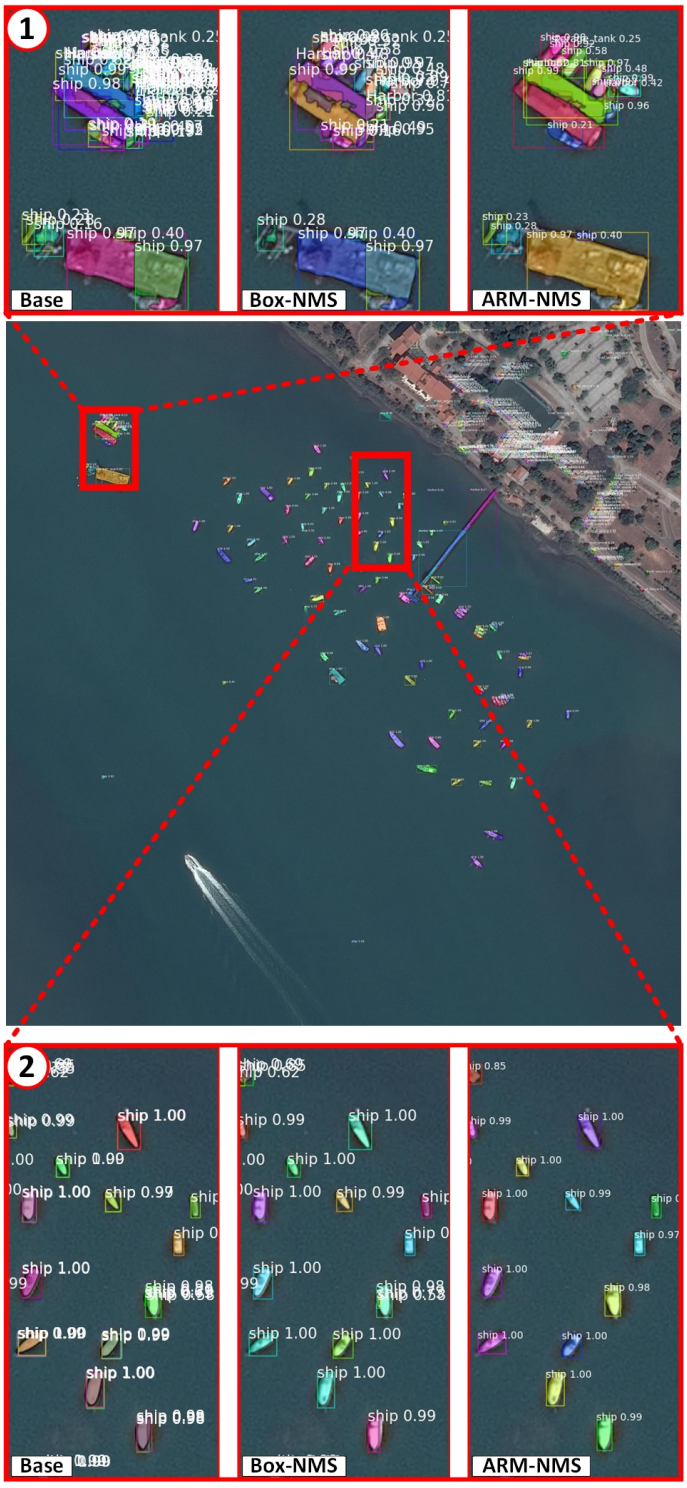


Figure 4.7: Visualization Results Achieved on the iSAID Dataset: Coast. The image sections show the unfiltered base detections, the box-based Soft-NMS, and our proposed shape-based ARM-NMS results. (1) The annotation density in this clustered image section can be used to indicate the detection density. Box-NMS can reduce the number of detections, but ARM-NMS outperforms its counterpart significantly. (2) We display a relatively sparse scene with double detections among the vessels.

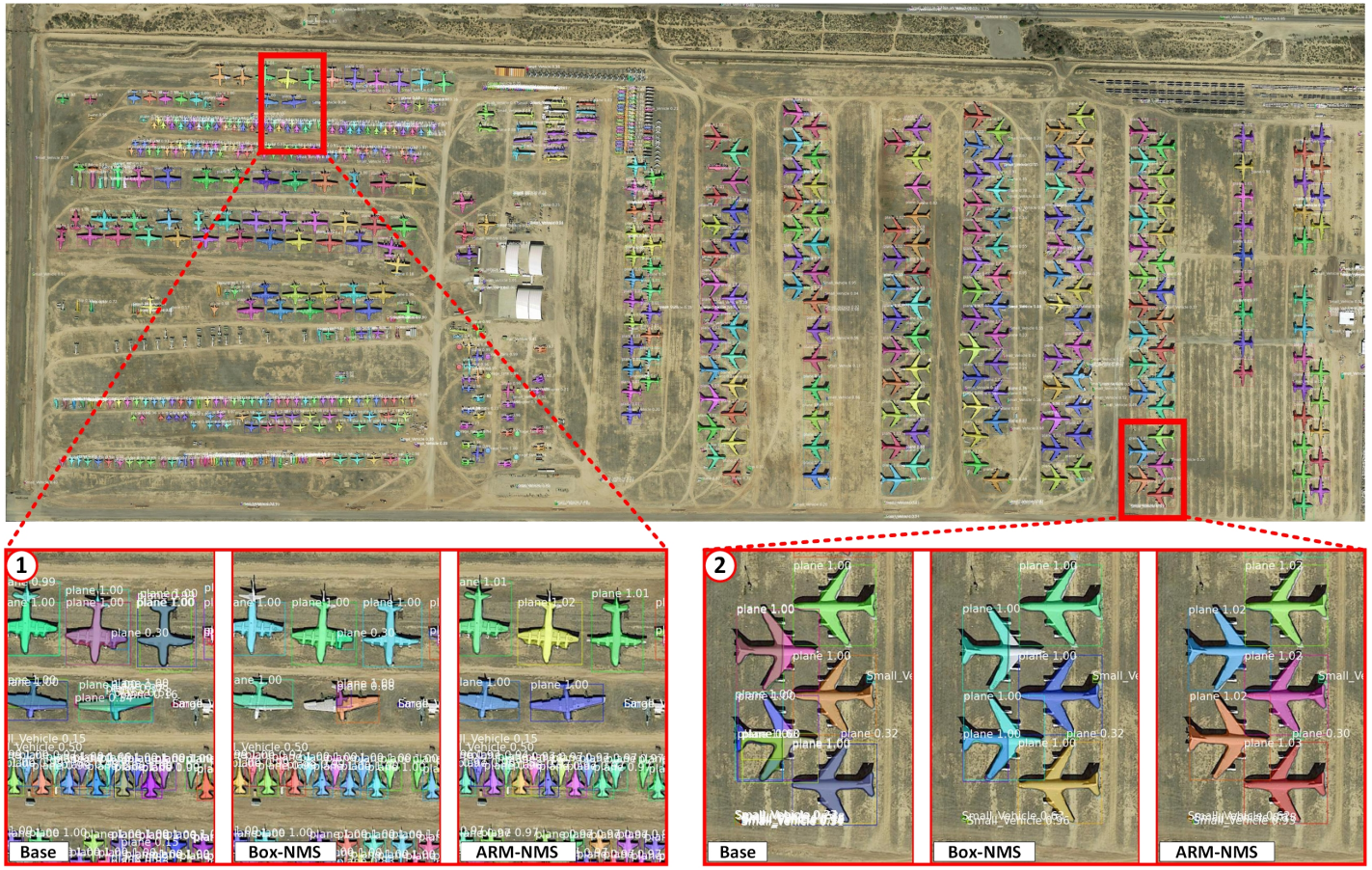


Figure 4.8: Visualization Results Achieved on the iSaid Dataset: Airplane Graveyard. The image subsets show the unfiltered box-based Soft-NMS and our proposed shape-based ARM-NMS results: (1) Example for multiple detections (2) Example for the intersection of two overlapping image patches.

Table 4.3: ARM-NMS Ablation Results. We present the best mAP and mAR results under the defined settings.

metric	segmentation			
unique labels	True	True	True	False
overlap metric	IoU	IoMin	IoU	IoU
area-rescoring	True	True	False	True
AP [0.50:0.95]	**28.6**	28.2	25.9	27.6
AP 0.50	**53.5**	52.7	50.7	52.0
AP 0.75	**26.9**	26.5	23.2	25.8
AP[0.50:0.95] small	**18.9**	18.9	17.2	18.3
AP [0.50:0.95] medium	**32.0**	31.8	28.9	30.9
AP [0.50:0.95] large	**35.6**	35.0	33.8	34.4
AR [0.50:0.95] @100	**32.1**	31.1	31.7	31.4
AR [0.50:0.95] @1000	**37.0**	35.7	36.8	36.2
AR [0.50:0.95] @1500	**37.1**	35.7	36.8	36.2
AR [0.50:0.95] small	**26.9**	25.8	26.8	26.4
AR [0.50:0.95] medium	**38.0**	37.2	37.8	36.9
AR [0.50:0.95] large	**43.8**	42.2	43.4	42.8

illustrate some image sections in more detail and display the unfiltered detections, detections filtered by ARM-NMS, and detections filtered by a box-shaped method. The settings of the NMS methods are based on the best results from Table 4.2. In Figure 4.7 we show a coastal scene with many vessels. In image section one, we can see how ARM-NMS outperforms Box-NMS significantly in a cluster of object detections. Likewise, as seen in image section two, ARM-NMS handles sparse double detections well. Figure 4.8 displays an airplane graveyard. Airplanes are concave objects with a high overlap potential for bounding boxes. Image section one shows how ARM-NMS excels in partial detections. The box-based approach performs worse in this scene. This is also the case for the second image section. The narrow concave objects cause an overlap of bounding boxes despite the underlying objects not actually overlapping. Furthermore, this image section is part of the intersection of two overlapping image patches. Therefore, high-scored partial detections are present. ARM-NMS excels in this case, while Box-NMS falls behind.

4.3.8 Ablation Studies

In order to determine the best composition for our approach and compare it to Mask-NMS, we evaluate the impact of the different variations on ARM-NMS. ARM-NMS with no area-rescoring, unique labels and IoU is identical to Mask-NMS. Table 4.3 shows the best results on the iSAID dataset for a specific parameter combination

regarding the COCO-style mAP metric.

- **Labels:** In the default case of NMS, each class is filtered separately. Therefore, double detections with different labels are ignored in the filtering process. Filtering all classes simultaneously may lead to better results in certain cases, but overall it decreases the average precision and the average recall score. Furthermore, it leads to an increased computing time due to the increase of objects compared to each other.

- **Overlap:** IoU is an integral part of most of the NMS methods. In this case, where the compared detections differ strongly in size, NMS does not suppress any detections. IoMin may perform better in some cases than IoU, but it slightly decreases average precision.

- **Area-rescoring:** Area-rescoring adjusts the detection scores favoring larger areas. Ideally, partial detections should receive a drop in their score and, thus, are less likely to suppress better-aligned detections. However, reducing the scores of all small objects by a large margin can lead to missing final detections. In general, area-rescoring leads to a significant increase in average precision in our experiments. A possible reason for this observation could be the use of the applied tiling scheme. Partial detections on the edges of a patch could still reach a high detection score and, therefore, suppress the overlapping patch's better-aligned detections. The partial detections are smaller than their better-aligned counterparts and, therefore, receive a higher penalty from area-rescoring. Consequently, the chance that partial detections suppress better-aligned detections is reduced.

4.4 Effective Data Optimization Strategies for Enhanced Object Detection Training

Maritime obstacle detection is crucial for enabling secure autonomous maritime vehicles and preventing collisions with wildlife. In maritime environments, objects can vary greatly in scale and density. In particular, wildlife may appear either densely clustered within scenes or sparsely distributed. To address this challenge, we implemented a three-stage training approach that utilizes Co-DETR [182]. Our

method achieved the second place in the MACVI 2024 USV Obstacle Detection Challenge [66].

4.4.1 Implementation

The used detection pipeline is based on the Co-DETR [182] object detector. Co-DETR exploits multiple parallel auxiliary heads in order to increase the learning effectiveness of the detection transformer. In our case, the underlying detector is DINO [176] with six encoder- and decoder-layers with sinusoidal positional encoding. We used the implementation provided in the MMDetection toolbox version 3.2 [17] to train our detector. As a backbone, we utilize an ImageNet-22K [30] pre-trained Swin-L [90] vision transformer. Our training is conducted on two NVIDIA A100 80GB graphics cards.

The detector is pre-trained on the Objects365 [132] and the MS COCO [84] datasets. In the next stage, we combine all images from MS COCO, including the category ship, with the LaRS [183] dataset and train for twelve epochs. The last training stage includes only the LaRS dataset with the dynamic obstacle class and is only two epochs long. While training, we apply data augmentation in the form of random vertical image flipping, random multiscale resizing, and random crop operations. AdamW [92] was used as an optimizer with an initial learning rate of $2 \cdot 10^{-4}$ and a weight decay of 10^{-4}. Furthermore, we apply gradient clipping and apply a 0.1 learning rate multiplier to the backbone. Testing was performed on an NVIDIA A100 80GB graphics card. The inference speed was about four FPS. Rather than utilizing the standard COCO evaluation metric we have adopted the Maritime Obstacle Detection Score (MODS) protocol as described by Cane et al. [11]. The MODS protocol employs the F1-score as its principal performance metric, which offers a balanced consideration of both precision and recall. The test pipeline includes resizing to a 2048 $\times$ 1280 scale but no Test Time Augmentation (TTA).

4.4.2 Experimental Results

Table 4.4 presents a comparative analysis of our proposed method against other approaches utilized in the MACVI 2024 USV Obstacle Detection Challenge [66]. Our method significantly outperforms the baseline YOLO-v7 model by a substantial margin, demonstrating its effectiveness in maritime obstacle detection tasks.

Table 4.4: **MACVI 2024 USV Obstacle Detection.** Overview of the three top submissions for the USV Obstacle Detection challenge, with results [66].

Place	Author	Model Name	F1 (%)
1	FER-LABUST	YOLOv8x pre-trained	51.50
2	Fraunhofer IOSB	Co-DETR + 3StageTrain(**OURS**)	43.56
3	DLR MI-SIT	ScatYOLOv8CBAM+SAHI	39.79
-	Baseline UL	YOLO v7 baseline	18.16

Table 4.5: **Ablation Study MACVI.** We conducted an ablation study on the LaRS test dataset to evaluate different training and inference schemes. The three-stage training process begins by utilizing pre-trained weights from ImageNet, Objects365, and COCO datasets. Subsequently, the detector is trained on specific COCO classes that are included in the LaRS dataset. Finally, we fine-tune the model on the LaRS dataset itself. In contrast, the two-stage approach skips the middle stage of training on specific COCO classes.

Method	Training Sceme	TTA	FPS	F1 (%)
Co-DETR	From Scratch	No	4	38.42
Co-DETR	3 Stage	Yes	0.9	40.80
Co-DETR	2 Stage	No	4	42.96
Co-DETR	3 Stage	No	4	43.56

Moreover, our approach also surpasses the performance of the YOLO-v8 model enhanced with scattering transform and attention mechanisms for maritime awareness [13]. This indicates that our method provides superior accuracy even when compared to more advanced models incorporating additional feature augmentation techniques.

Notably, the only method that exceeded our results was the YOLO-v8 [63], which was trained on a larger and more diverse dataset. The wider dataset likely contributed to its enhanced performance, highlighting the importance of extensive training data in improving model accuracy. Qualitative results can be found in Figure 4.9.

In our ablation study, illustrated in Table 4.5, we found that employing TTA with multi-scale resizing not only results in the expected substantial decrease of processing speed by approximately 3 FPS (frames per second) but also unexpectedly degrades detection performance. Furthermore, omitting the second training stage—training directly on the LaRS dataset for twelve epochs—leads to a reduction of over three points in the F1-score on the test dataset. These results indicate that the optimal approach is the three-stage training scheme without TTA.

Figure 4.9: Detection Results on the LaRS Dataset. This illustration presents the detection outcomes on the LaRS dataset using our proposed training scheme for the Co-DETR. The top image depicts the ground truth annotations, while the bottom image displays our model's predictions. The results demonstrate the effectiveness of our approach in accurately identifying maritime obstacles.

4.5 Discussion

This section explores the implications of ARM-NMS's superior performance over Box-NMS variants. The superiority of ARM-NMS in both bounding box detection and mask predictions suggests a significant advancement in non-maximum suppression techniques for object detection and segmentation tasks. This advancement is particularly noteworthy given the challenges associated with accurately detecting and delineating objects in complex visual scenes.

Our ablation experiments have further highlighted the efficacy of our approach compared to Mask-NMS [158]. While Matrix-NMS shares similarities with our method, particularly in handling multiple overlapping detections, the absence of area rescoring in Mask-NMS limits its performance. Area rescoring plays a pivotal role in refining detection scores based on the size of the detected objects, thereby enhancing the overall accuracy of the detection system.

However, the current conclusions are drawn from experiments conducted on a single dataset. This limitation underscores the necessity for conducting extensive experiments across multiple datasets to ensure the robustness and generalizability of the proposed method. Testing on diverse datasets will help understand how ARM-NMS performs under varying conditions and data distributions, which is essential for real-world applications. Furthermore, using masks instead of bounding boxes leads to a higher computational burden and requires a more elaborate ground truth annotation process. Additionally, the method could be compared with rotating bounding boxes for further evaluation.

The dependency of NMS on multiple parameters presents another layer of complexity. Optimizing these parameters can be both challenging and resource-intensive. The optimal parameter values may vary significantly with different datasets or in the presence of data shifts, such as changes in the underlying data distribution or the introduction of new object categories. This variability necessitates a flexible approach to parameter tuning, potentially involving automated or adaptive methods to reduce the burden of manual optimization.

While the introduction of area-rescoring enhances detection performance, it adds additional parameters to the model. This exacerbates the difficulty of the parameter search process. An extensive search may be required to identify the optimal combination of parameters that balance detection accuracy with computational efficiency.

Moreover, area-rescoring can introduce issues related to detection glitches. Glitched masks can emerge in certain situations, particularly with complex or noisy data. These masks may be anomalously large and carry low initial confidence scores. However, these low-confidence, large-area masks might receive an inflated score due to area-rescoring. This inflation can result in the erroneous elimination of correct masks with smaller areas but higher relevance. Such occurrences can adversely affect the overall performance of the detection system.

To mitigate these issues, exploring strategies that can detect and handle glitched masks effectively is essential. Potential solutions could involve implementing ad-

ditional validation checks or integrating robustness measures that account for the detected masks' size and quality.

The second study emphasizes a data-centric approach instead of the development of machine learning models. The findings reveal that conventional strategies for machine learning inference do not always lead to improved performance. One possible explanation is that augmentation techniques, such as multi-scale resizing, may introduce data shifts that adversely affect the model's accuracy. Additionally, the fusion process of results using filtering methods might introduce errors that compromise the overall performance. By concentrating on a data-centric methodology, an increase in performance was achieved. Notably, it has been demonstrated that a straightforward approach—modifying existing datasets for a new domain by selecting only the relevant classes—can significantly enhance performance. This underscores the critical role that data quality and relevance play in machine learning outcomes. Furthermore, the 2024 MACVI challenge demonstrated that detection algorithms often deemed superior, such as Co-DETR [182] in standard benchmarks like COCO, can be outperformed by models trained on diverse and relevant data samples, including YOLOv8 [63]. This finding implies that the importance of data selection and diversity may surpass that of the neural network architecture. In essence, optimizing the training dataset can have a more profound impact on model performance than relying solely on advanced neural network designs.

4.6 Conclusion

In this chapter, we have demonstrated how leveraging non-architectural methods can enhance the performance of object detection and its extension, instance segmentation, in scenes characterized by significant variability in scale and density.

First, we present ARM-NMS, a novel technique that utilizes shape information to filter out unnecessary object detections. ARM-NMS does not require any retraining, allowing for straightforward implementation in existing instance segmentation methods. To demonstrate the effectiveness of our approach, we generate detection lists using the well-regarded Mask R-CNN detector applied to the entire iSAID validation dataset. Our results indicate that ARM-NMS outperforms traditional box-shaped filtering algorithms by more than three points on the COCO-style (mAP) metric. Furthermore, we have confirmed our hypothesis that rescoring detections based on the shape and area of the objects enhances overall detection performance.

Future research will focus on refining final detections by exploring various strategies for merging overlapping detections.

Next, we present a methodology to improve maritime object detection through a novel three-stage training scheme. Our technique enhances detection performance during inference by leveraging multiple representations of the input data, all without modifying the model's architecture. This approach outperforms all existing methods except for one that benefits from using a superior dataset tailored to the task.

Overall, we demonstrate that object detection performance does not rely solely on architectural optimization. Training schemes, data handling, and post-processing techniques are equally important factors.

5 Fusion of Density Estimation and Object Detection Methods

In the following, quotations to related publications are highlighted by colored bars placed on the outer border of the text, even though minor editorial changes have been made. The color encoding used is as follows:

- **Michel, A.**, Gross, W., Schenkel, F. & Middelmann, W. (2022). Class-aware object counting. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 469-478). Reprinted with permission. It is cited as [97]. ■
- **Michel, A.**, Mispelhorn, J., Schenkel, F., Gross, W. & Middelmann, W. (2020, September). An approach to improve detection in scenes with varying object densities in remote sensing. In Image and Signal Processing for Remote Sensing XXVI (Vol. 11533, pp. 107-113). SPIE. Reprinted with permission. It is cited as [98]. ■

In this chapter, we combine the density estimation methods introduced in Chapter 3 with the object detection techniques discussed in Chapter 4. We explore this integration in three main ways: improving object counting, enhancing object detection performance, and refining dust density regression.

To achieve these objectives, we adapt and extend the CrowdFPN framework proposed in Section 3.3. Although originally developed for dust density estimation, CrowdFPN's versatility extends to object detection tasks. Traditional object detectors often face substantial limitations in environments with high object densities. To address these challenges, we propose a synergistic approach that combines object detection algorithms with density estimation techniques, thereby enhancing detection performance in densely populated contexts.

A notable application of this integrated methodology is in class-aware object counting. In this context, we enhance CrowdFPN to detect and count multiple object

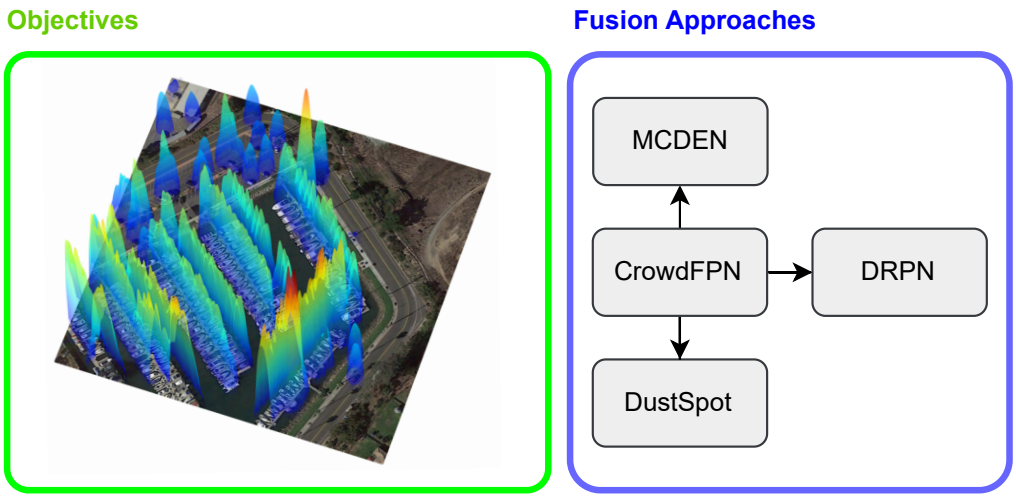


Figure 5.1: Topics of Chapter 5. In this chapter, we focus on methodologies that fuse density estimation with object detection methods to develop techniques for object counting, object detection, and dust density estimation. These methods explicitly account for varying object densities and integrate multiple data sources or models to enhance accuracy in complex scenes where object density significantly affects detection performance.

classes, significantly improving accuracy across varying densities. This advancement not only deepens our understanding of object distributions but also increases precision in applications such as monitoring fishing activities and analyzing urban traffic patterns.

Furthermore, we explore the direct application of CrowdFPN to enhance overall object detection performance. By refining the region proposal mechanisms—essential for identifying areas of interest in subsequent detection processes—we achieve notable improvements in both precision and recall rates, thus strengthening the efficacy of object detection tasks.

Finally, we address the limitations identified in Chapter 3, particularly regarding small dust events and high dust density occurrences. To overcome these challenges, we introduce DustSpot, a fusion approach that combines object detection and dust density estimation. By integrating these methodologies, we specifically target these shortcomings and improve regression capabilities, leading to more accurate detection and quantification of dust events across various scenarios.

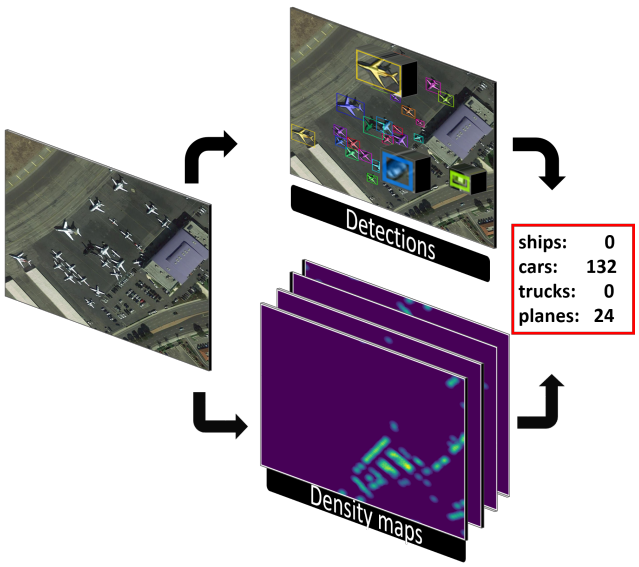


Figure 5.2: Class-aware Object Counting. The basic idea behind our approach is to detect relevant objects while simultaneously creating multi-class density maps. Results from both branches are fused to create a more precise object count for each class.

5.1 Fusion of Density Estimation and extended Object Detection for Improving Counting

Estimating the accurate number of objects within a given natural scene constitutes a significant challenge in the field of remote sensing. Natural environments typically encompass a plethora of object categories alongside varying object densities, complicating the counting process. While detection-based algorithms are adept at performing class-aware object counting, particularly in scenarios characterized by low object counts, they often exhibit diminished efficacy when confronted with high or fluctuating object numbers. To tackle this intricate problem, we propose a novel end-to-end methodology that augments an existing detection-based framework through the incorporation of a multi-class density estimation branch. This density estimation component is built on the CrowdFPN architecture (Section 3.3), as delineated in Chapter 3. The outputs from both the detection and density estimation branches are subsequently integrated within a successive count estimation network, which systematically estimates the object counts for each distinct category. Although the resulting counts generated by our approach lack explicit localization information, they serve as invaluable indicators for assessing the accuracy of the object detector's outputs and enhancing its overall counting performance. The fundamental concept behind our approach is visually represented in Figure 5.2. This integration of density

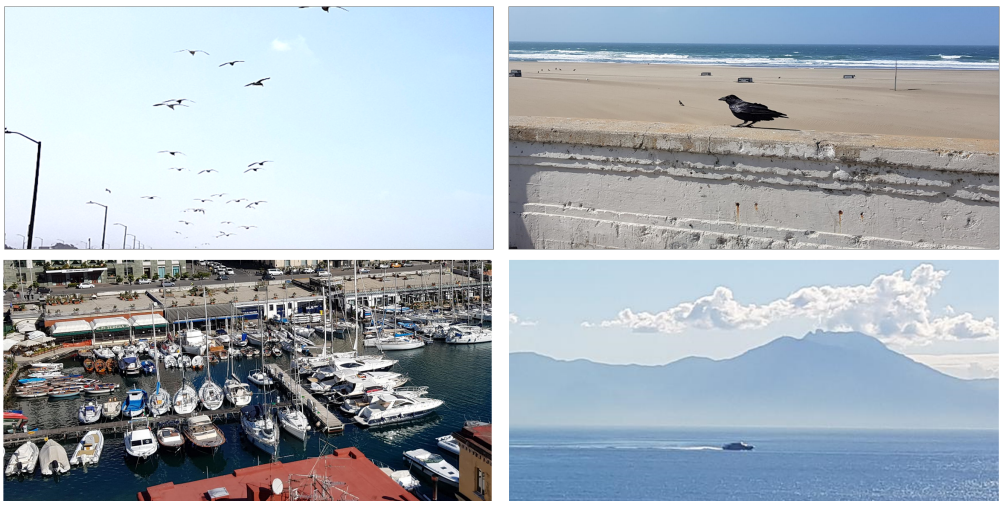


Figure 5.3: Overview of Real-world Scenes. We study the problem of class-aware counting in natural scenes, which pose many challenges for real-world applications. Leading counting approaches are usually focused on single-class counting, but realistic scenarios do not only vary strongly in object density and scale but usually include multiple relevant categories.

estimation with detection not only improves the count accuracy but also contributes to a more nuanced understanding of object distribution within complex scenes.

5.1.1 Introduction

Tasks, which are usually easy for a human, can be hard for a machine [105]. Counting objects and people is such a task. In real-world scenes, the challenges are diverse, including but not limited to occlusions, the presence of multiple object categories, and varying object densities [54]. Figure 5.3 shows examples of natural scenes. In particular, for the Coast Guard, not only the number of ships is relevant, but also the vessel type. Furthermore, counting has many valuable applications, such as traffic monitoring, surveillance, public safety, and urban planning [131]. Since manual object counting is a time-consuming task, it is not feasible on a large scale and with time-critical applications.

To address this issue, crowd counting provides a solution, as it can be generalized to object counting. Commonly, crowd counting approaches are based on detection, regression, or density estimation [43]. The main assumption of this work is that object detection methods excel in scenarios with low-density counts and low to moderate occlusion, whereas they underperform in scenes with high and diverse object counts. In contrast, direct regression methods [14, 129] are better suited for these challenging conditions; however, they lose information about the spatial

distribution of objects. This is a reason why density estimation methods have become popular in recent years [177, 131, 81, 88].

Instead of predicting a global count or localizing objects precisely, a density map of the relevant targets is estimated. The global count can be acquired by summation over the density map. Density estimation methods favor overestimating the number of sparse scenes and, consequently, underperforming in low object counts. Furthermore, nearly all recent density estimation methods are unfit for a class-aware approach. In real-world applications, this poses a significant disadvantage compared to detection-based methods. Only a few methods drive research towards multi-class density maps [23, 168]. Object counting in the context of a real-world application often requires predicting the number of objects in different categories, e.g., in traffic monitoring. In this scenario, it can be important to count all road users and simultaneously differentiate between cars and trucks. One of the main obstacles in this scenario is the strongly varying object density. Urban regions will usually contain more vehicles compared to rural areas. Detection-based methods will likely be contested on urban scenes, and density estimation-based methods will underperform on rural scenes with a sparse object density. Furthermore, diverse object densities are a common condition in air- and spaceborne images, e.g., given in the iSAID Dataset [160]. This can lead to a decrease in the reliability of the counting accuracy. One way to improve the counting accuracy is to combine different counting models. Liu et al. [85] proposed a fusion of detection-based and regression-based methods for crowd counting. Although DecideNet [85] achieves great performance, it is only developed for the single-class problem.

Class-aware object counting in varying object densities is a real-world problem, but common object detection datasets like COCO [84] have a strong bias towards one-digit object counts per image [15]. In contrast, common crowd counting datasets like ShanghaiTech [177] usually have diverse object densities yet include only one class. Yet single class solutions are not sufficient for many real-world applications. As a result, research in the field of class-aware object counting is still needed. Therefore, inspired by DecideNet, we propose a fusion of a modern object detector with a multi-class density estimation network. We utilize for our multi-class density estimation branch a backbone and a FPN. Furthermore, we utilize detection and density estimation results to predict accurate object counts for different categories. We see for our approach two use cases: The first use case is class-aware object counting. The second use case is to utilize it as a support structure for object detection. From the deviation between the predicted object count and the detection count, the presence of false positive or false negative detections can be inferred.

Contributions: (1) We propose a novel way to combine class-aware density esti-

mation with object detection. Our approach is well-suited to be integrated into the most common object detection and instance segmentation models and is end-to-end trainable. (2) Our approach includes a count estimation network optimized to predict object counts with different categories in scenes with diverse object densities. The predicted object counts can be used as an indicator to verify the detection results and, therefore, increase its reliability. (3) To display our approach's effectiveness, we evaluate our method on common object detection datasets.

5.1.2 Approach

Our proposed methodology combines a multi-class density estimation network (MCDEN) with a detection pipeline, serving as the precursor for the count estimation network (CEN). As previously articulated, natural scenes are characterized by the presence of multiple objects that exhibit significant variations in density and encompass diverse categories. To enhance counting performance, we propose a synergistic approach that fuses object detection with class-aware density mapping.

An overview of the architecture is presented in Figure 5.4. Initially, we employ a feature pyramid network (FPN) as the backbone to extract feature maps across varying receptive fields and semantic levels. The flow of these feature maps is separated into two distinct branches: the detection pipeline and the density estimation branch.

The detection pipeline comprises a region proposal network (RPN), a non-maximum suppression filter, a box head, and, optionally, a mask head. In parallel, the density estimation branch utilizes the MCDEN to generate class-aware density maps that encapsulate the distribution of objects across different categories. Subsequently, the outputs from both the detection pipeline and the density estimation branch are concatenated as inputs for the CEN. The primary objective of the CEN is to deliver accurate object counts for each category within the specified scene. By leveraging the strengths of both detection and density estimation, our approach aims to significantly enhance the performance of object counting in complex natural environments.

Multi-class Density Estimation Network. We consider a set of N training images $\{I_i\}_{1\leq i\leq N}$ with corresponding ground truth density maps $\{D^{\text{gt}}_{i,j}\}_{1\leq j\leq c}$, for c different object categories. We aim to learn a non-linear mapping $\mathcal{F}_j$ for each object category parameterized by θ to estimate c different density maps approximating $D^{\text{gt}}_{i,j}$ by minimizing the L_2 norm of the ground truth and the predictions.

$$D^{\text{est}}_{i,j}(I_i) = \mathcal{F}_j(I_i, \theta), \tag{5.1}$$

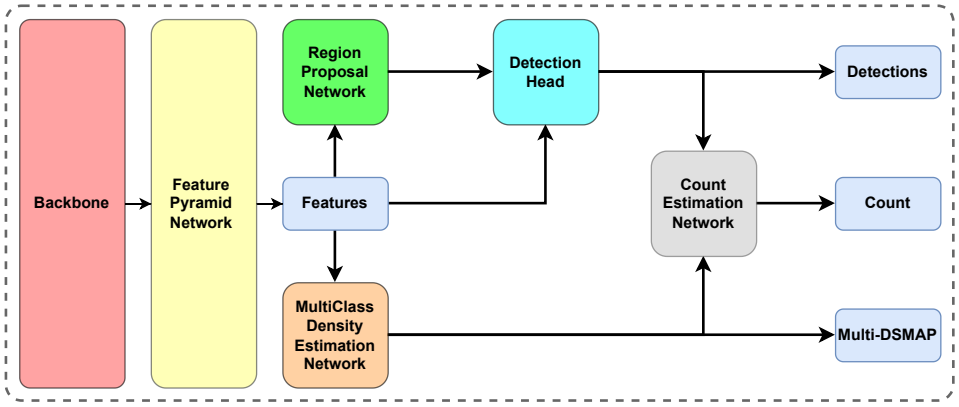


Figure 5.4: Network Architecture. A general overview of our proposed method. Features extracted from the backbone are processed in the RPN and in the MCDEN. Subsequently, detection results and density maps are sent to the CEN to refine counting results.

where θ are the learned parameters. Without loss of generality, we describe the proposed method for a single image I to facilitate the notation.
To start with, the FPN outputs four relevant feature maps S_i with $i \in \{0, 1, 2, 3\}$, one for each of the scaling factors $s \in \{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$. The feature map S_1 with the spatial size $\frac{1}{8}$-th of the original image size is used as the base feature and all other features are resized to this spatial size calculated by

$$R_l = \text{resize}(S_l) \qquad l \in \{0, 2, 3\}, \tag{5.2}$$

where R_l is the resized feature map. The upsampling is done by an bilinear interpolation, the downsampleing learned by applying a dilated convolutional layer with a stride of two. Then, we create scale-aware features C_l with $l \in \{0, 2, 3\}$ according to

$$C_l = S_1 - R_l \qquad l \in \{0, 2, 3\}. \tag{5.3}$$

In a next step, we calculate the individual contrast features W_l according to

$$W_l = \mathcal{F}_{sa}(\mathcal{F}_{sa}(C_l) \odot R_l) \qquad l \in \{0, 2, 3\}, \tag{5.4}$$

where $\mathcal{F}_{sa}$ is the output of a 1×1 convolution layer followed by a sigmoid activation function. The resulting features are multiplied element-wise by R_l, denoted by $\odot$. The contrast feature W_1 are calculated by

$$W_1 = \mathcal{F}_{sa}(R_1). \tag{5.5}$$

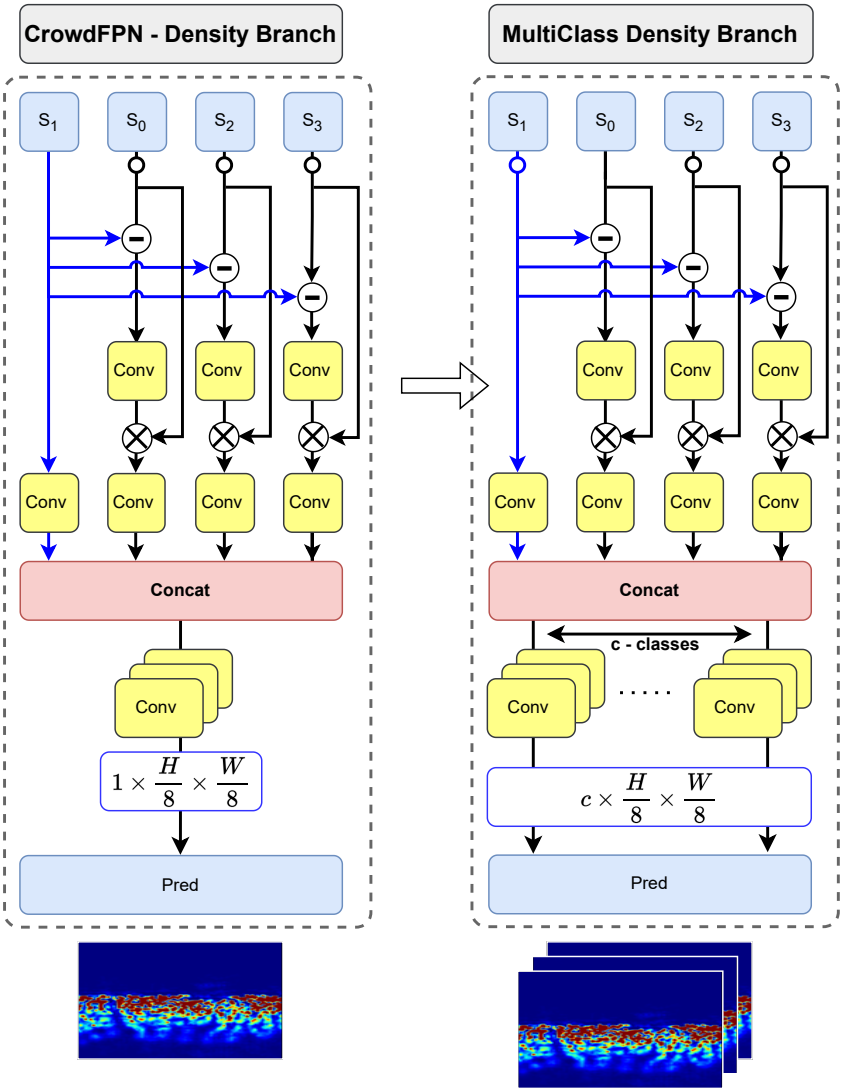


Figure 5.5: Multi-class Density Estimation Network. In this figure, we compare the multi-class density estimation network (MCDEN) with the CrowdFPN architecture. Unlike CrowdFPN, which typically focuses on generating a single density map for the entire scene, MCDEN aims to predict multiple density maps corresponding to each object category. This differentiation allows MCDEN to capture the nuanced distributions of various object classes more effectively, thereby enhancing the precision of density estimation in complex environments.

The contrast features W are then concatenated to F. After encoding the relevant information, particularly the scale of the original image I, into the feature block, we initiate the decoding process. Rather than using a single decoder as in [99], we employ a dedicated decoder network $\mathcal{F}_{\text{dc}}$ for each object category c

$$D_j^{est} = F_{dc,j}(F) \quad \forall j \in \{1, ..., c\}. \tag{5.6}$$

The decoder consists of three sequences of dilated convolutional layers followed by batch normalizations and ReLU activation functions. In contrast to CrowdFPN we reduce the number of weights significantly in the decoder block, as each additional category increases the number of required parameters to be learned.

Detection Pipeline. The detection pipeline is derived from Mask R-CNN. The essential parts of the detection pipeline are the RPN and the box head. Furthermore, we add a mask head, which is an optional step for our approach in order to create instance masks. The first step is the processing of the feature maps from the backbone in the RPN. The RPN proposes regions in which objects can be found. Then, the NMS filters the output region proposals. Afterwards, the filtered region proposals are sent to the box head, where the classification of the objects in the region are performed. The box head also refines the spatial properties of the region proposals. In an optional step, a mask head estimates the segmentation masks of the potential objects.

Count Estimation Network. Figure 5.6 shows an overview of the CEN. The CEN for a single object category $j \in c$ can be formulated as follows:

$$\text{CEN}_j = \mathcal{F}_{\text{stage2}}(n_j^{\text{det}} + \lambda_{res} \cdot \text{resize}(\mathcal{F}_{\text{stage1}}(D_j^{\text{est}}))). \tag{5.7}$$

Here, $\mathcal{F}_{\text{stage1}}$ and $\mathcal{F}_{\text{stage2}}$ are the operators describing the corresponding parts from Figure 5.6. $\mathcal{F}_{\text{stage1}}$ is a sequence of convolutional layers, $\mathcal{F}_{\text{stage2}}$ consists of multiple fully connected layers, and the resize function has a fixed output of the size $c \times \epsilon$. The constant ϵ influences the complexity of the first dense layer, which must be tuned during training, and c denotes the number of object categories. $\mathcal{D}^{est}$ and n^{det} and are the results from the density estimation and the detection pipeline. The rescaling factor $\lambda_{res,j} \in [0, 1]$ denotes the quotient between the size of the tensor before and after the resizing operation following the first stage of CEN. The multi-class density maps D_j^{est} and the number of detected objects n_j^{det} of the detection pipeline are fed forward to the CEN. In the first step of the CEN, the multi-class density maps are processed in a 2D-convolutional layer with a filter size of 32. The convolutional layer is followed by a batch normalization and a ReLU activation function. After a repetition of this sequence of operations, the resulting features are further processed by a 2D-convolutional layer. After applying another set of batch normalizations and ReLU activation functions, the results are interpolated to a matrix with the dimensions $c \times \epsilon$. Due to the different sizes of the input density maps, resizing the feature map can lead to wrong assumptions about the number of objects. Density maps are trained to approximate a normalized Gaussian distribution in which the sum of all pixels corresponds to the number of objects in a given scene. Applying

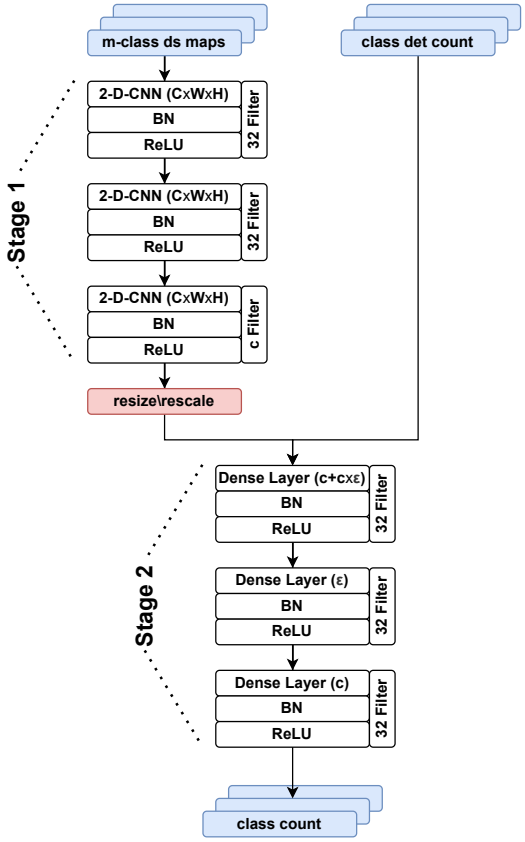


Figure 5.6: Count Estimation Network Architecture. The count estimation network (CEN) consists of two stages. Initially, the density maps are processed. Subsequently, the embeddings are concatenated with the class detection counts, and an estimation of the number of objects per class is performed through regression. This two-stage approach ensures a comprehensive integration of both density information and classification data, enhancing the accuracy of the final count estimations for each object category.

a simple resize operation to a fixed value can skew the distribution. Therefore, a resizing function has to be followed by the rescaling factor λ_{res}.

This featuremap and the previously predicted number of detections n_j^{det} are concatenated and fed into a sequence of two dense layers with a batch normalization layer, which is activated by ReLU functions. Finally, an additional dense layer outputs the estimated count result n_j^{cen} for each class.

Loss Functions. We employ a multi-loss function consisting of the detection loss L_{det}, the density loss L_{ds}, and the CEN loss L_{cen}. The overall loss function can be formulated as follows:

$$L = L_{\text{det}} + \sum_{j=1}^{c} \frac{1}{c}(L_{\text{ds},j} + L_{\text{cen,j}}), \tag{5.8}$$

where L_{det} is implemented according to [47]. Furthermore, $L_{\text{ds},j}$ is aggregated over all object categories c and calculated by the L^2-norm with a batch size $N' < N$ of the training images $I' \subset I$

$$L_{\text{ds,j}} = \frac{1}{2N'} \sum_{i=1}^{N'} \left\| D_{i,j}^{\text{gt}} - D_{i,j}^{\text{est}} \right\|_2 . \tag{5.9}$$

Finally, $L_{\text{cen,j}}$ per object category can be defined as

$$L_{\text{cen,j}} = \frac{\lambda_{\text{cen,j}}}{2N'} \sum_{i=1}^{N'} \frac{\left\| n_{i,j}^{\text{gt}} - n_{i,j}^{\text{cen}} \right\|_2}{n_{i,j}^{\text{gt}}} , \tag{5.10}$$

where n_j^{gt} is correct number of objects The L_{cen} also utilizes the L^2-norm. In order to reduce the effect of the L_{cen} on the weight update process, we apply an additional constant $\lambda_{\text{cen}} \in [0, 1]$ to L_{cen}. Additionally, we divide the loss term by the number of objects per category to create a relative loss function for balancing the loss between high- and low-density object counts.

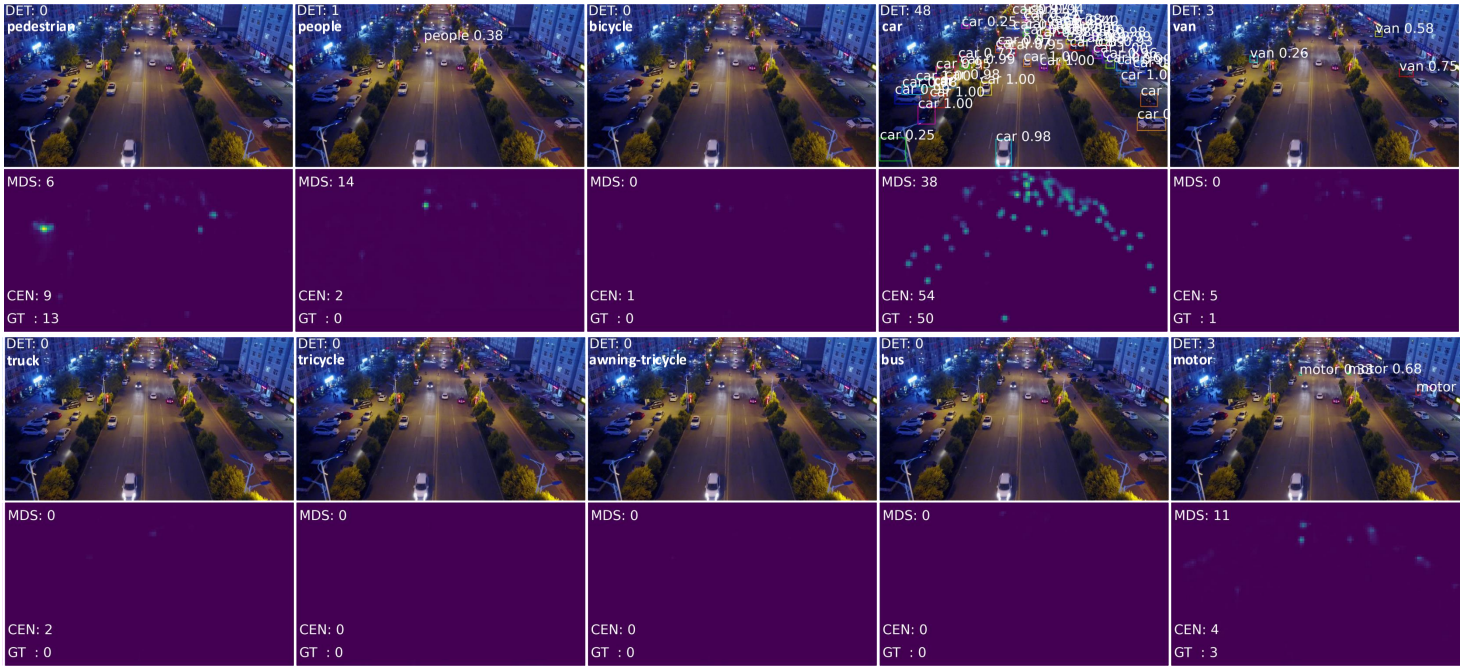


Figure 5.7: Sample of the VisDrone Dataset. For each target category, a paired representation of the density map and detection overlays is provided in the VisDrone dataset. This format facilitates clear differentiation between the respective density maps and their associated density levels. The top-left image pair illustrates the category "pedestrians," while the adjacent image to the right corresponds to the category "people." It is noteworthy that both the detection and density estimation branch exhibit limitations in effectively distinguishing between these two categories. However, the count estimation network (CEN) demonstrates commendable performance in this context, successfully discerning the object counts for each category despite the challenges faced by the preceding branches. This highlights the efficacy of the CEN in improving classification accuracy and count estimation in scenarios where object categories may overlap or be closely related.

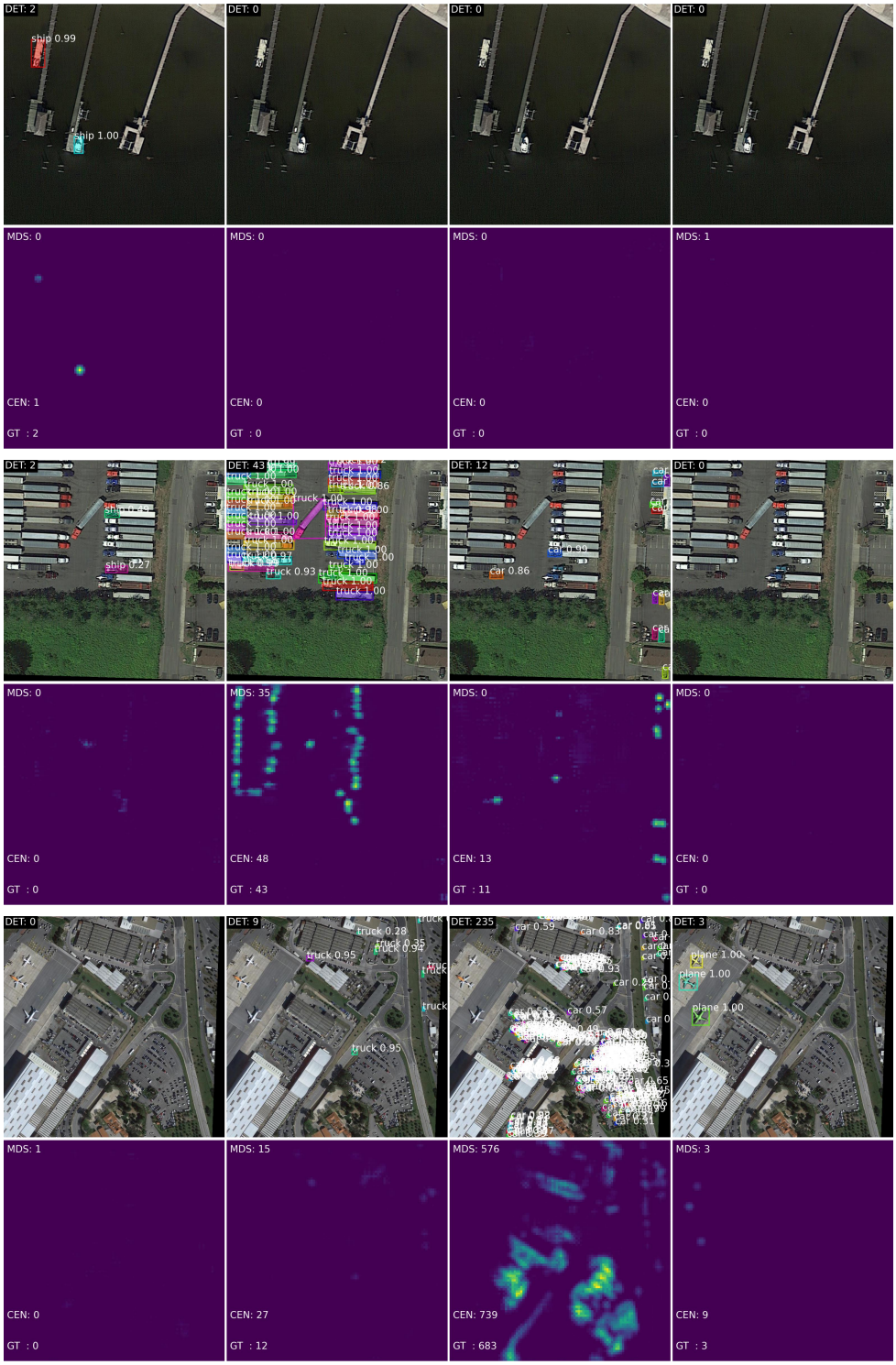


Figure 5.8: Visualization of Results achieved on the iSAID Dataset. From left to right column: ships, trucks, cars, and planes. DET is the number of detections, MDS is the sum over the density map, and CEN displays the number of the count estimation network, which combines the former results.

Table 5.1: Counting Results for the Visdrone Dataset. This figure illustrates the counting results on the VisDrone dataset.

Visdrone		[0-1000]	[0-10]	[11-50]	[51-100]	[101-1000]
Faster R-CNN	**MAE**	4.93	2.60	11.31	17.87	29.21
(*standalone*)	**RMSE**	13.45	3.98	13.56	33.34	64.88
Faster R-CNN	**MAE**	4.71	2.40	11.25	17.86	29.43
(*det-branch*)	**RMSE**	13.17	3.91	13.82	32.39	64.18
DetectorRS	**MAE**	4.77	2.35	14.71	18.35	22.86
(*standalone*)	**RMSE**	9.58	3.13	16.85	34.13	51.59
CenterNet	**MAE**	5.79	3.07	15.69	18.35	22.86
(*standalone*)	**RMSE**	10.92	4.19	18.23	36.65	54.25
MCDEN-OURS	**MAE**	6.53	4.29	14.39	14.81	26.92
	RMSE	11.50	5.74	16.00	25.78	53.88
CEN-OURS	**MAE**	3.76	2.07	10.49	13.84	23.55
	RMSE	9.56	3.25	12.52	25.07	51.11

5.1.3 Experimental Results

In this section, we evaluate the effectiveness of our proposed approach. First, we present details of our implementation. Then, we introduce the evaluation metrics and the benchmark datasets. After that, we discuss the counting results and compare them to state-of-the-art object detection methods. Finally, we conduct an ablation study.

Implementation Details. We use a ResNet50 [48] pre-trained on ImageNet [127] as our backbone. For the detection pipeline of our network, we use Faster R-CNN for object detection and Mask R-CNN as a baseline for instance segmentation. We apply the default anchor configuration following [47], but we increase the number of proposals to keep before and after applying the GreedyNMS filter to 2000 proposals. T Furthermore, we modify the number of possible detections per image to 1000. With this, we want to ensure that our detection pipeline is not limited in scenes with high object densities. In the training phase, the learning rate for the density estimation branch is set to one-tenth of that for the detection branch. This strategy facilitates a gradual increase in the learning rate of the density estimation branch with each epoch, leading to a more efficient training process overall. In the CEN, the constant loss reduction factor λ_{cen} and the interpolation constant ϵ are heuristically determined to be 0.2 and 128, respectively. During training, we start with a learning rate of $\alpha = 3 \cdot 10^{-3}$ and uniformly reduce α to 10^{-3} over 50 epochs. We choose a batch size of four and utilize the AdamW optimizer [171] with a weight decay of 10^{-4}. For training, ground truth density maps are created by a geometry adaptive Gaussian kernel [177] from the bounding box coordinates. We utilized the iSAID

[160] and VisDrone [33] object detection datasets to evaluate the effectiveness of our proposed approach. Specifically, for the iSAID dataset, we employed only a subset of the data, focusing on object instances from the movable object categories: 'ship', 'car', 'truck', and 'plane'. This selective use of categories allowed us to concentrate on objects that exhibit significant variability in both appearance and spatial distribution, thereby providing a robust assessment of our counting methodology.

Evaluation Metrics. Similar to previous works in object counting [23] and crowd counting [88, 177], the mean absolute error (MAE) and the root mean squared error (RMSE) are used as evaluation metrics. MAE_c per category for all N test images in the dataset can be defined as follows:

$$\mathrm{MAE}_j = \frac{1}{N}\sum_{i=1}^{N} |n_{i,j}^{est} - ni, j^{gt}|, \tag{5.11}$$

and RMSE_c can be calculated by:

$$\mathrm{RMSE}_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (n_{i,j}^{est} - n_{i,j}^{gt})^2}, \tag{5.12}$$

where N is the number of test images, $n_{i,j}^{est}$ is the estimated number of relevant targets in the i-th image for an object category j and $n_{i,j}^{gt}$ the corresponding ground truth. Furthermore, we define the mMAE as follows:

$$\mathrm{mMAE} = \frac{1}{c}\sum_{j=1}^{c} \mathrm{MAE}_j \tag{5.13}$$

and the mRMSE as:

$$\mathrm{mRMSE} = \frac{1}{c}\sum_{j=1}^{c} \mathrm{RMSE}_j \tag{5.14}$$

n^{est} is calculated for each branch individually. In the detection pipeline, $n^{est,det}$ is calculated by counting the number of detections. Meanwhile, $n^{est,ds}$ is derived by integrating over the pixels of the estimated density map in the density branch. Finally, the CEN outputs $n^{est,cen}$ directly.

Counting Results. Figure 5.7 presents a sample of the VisDrone test-dev dataset. The complete results are shown in Table 5.1, which displays the MAE and RMSE for all categories. In order to verify the effectiveness of our approach, we compare our method to the state-of-the-art object detectors DetectorRS [114], and CenterNet [34]. Both methods are trained for 50 epochs with hyperparameters similar to the standard settings described in [114, 34]. Even though they usually predict more

Table 5.2: Counting Results for the iSAID Dataset. This figure illustrates the counting results on a subset of the iSAID dataset.

iSAID		Mean	Ship	Truck	Car	Plane
Mask R-CNN	**MAE**	8.52	3.84	3.82	24.32	1.20
(*standalone*)	**RMSE**	50.68	11.11	9.94	99.15	3.73
Mask R-CNN	**MAE**	8.32	3.73	3.87	24.49	1.21
(*det-branch*)	**RMSE**	50.58	10.25	9.43	100.13	3.53
OURS-MCDEN	**MAE**	10.97	5.16	3.94	30.95	3.83
	RMSE	37.30	9.48	9.97	73.03	6.65
OURS-CEN	**MAE**	7.85	5.65	4.31	20.02	1.41
	RMSE	31.64	15.01	9.73	60.59	3.70

precise bounding boxes, their counting results are similar or worse in comparison to Faster R-CNN.

The detection branch outperforms the density branch in brackets with low object counts and vice versa for high object densities. This follows the expected performance discussed in Subsection 5.1.1. Both branches are outperformed in all brackets by our proposed CEN. Furthermore, the Faster R-CNN of our detection pipeline has a similar counting performance as the detection pipeline. Therefore, the multi-branch approach does not influence its detection part negatively in counting performance. Multiple images of the iSAID dataset are visualized in Figure 5.8 along with the corresponding CEN output. Similar to the VisDrone dataset, the density branch predicts fewer outliers than the detection branch, but it performed worse in the MAE for all categories (Table 5.2).

5.1.4 Ablation Study

The following section details an ablation study conducted to confirm the architectural decisions.

Multi-class Decoder. Our goal for the MCDEN is to estimate a density map for each category. The first approaches were challenged by crosstalk between each unique density map. In this case, the high-density level in one category influenced the density estimation of other categories. Figure 5.9 shows the qualitative results of our progress. At first, only a joint decoder for all categories is used. The number of trainable parameters is roughly the same as in the multi-decoder structure. Note that ships are the most numerous category in this example. This leads to strong crosstalk between the density map and the traces of the ship density map, which can be observed in Figure 5.9(a). Therefore, the MCDEN systematically overestimated

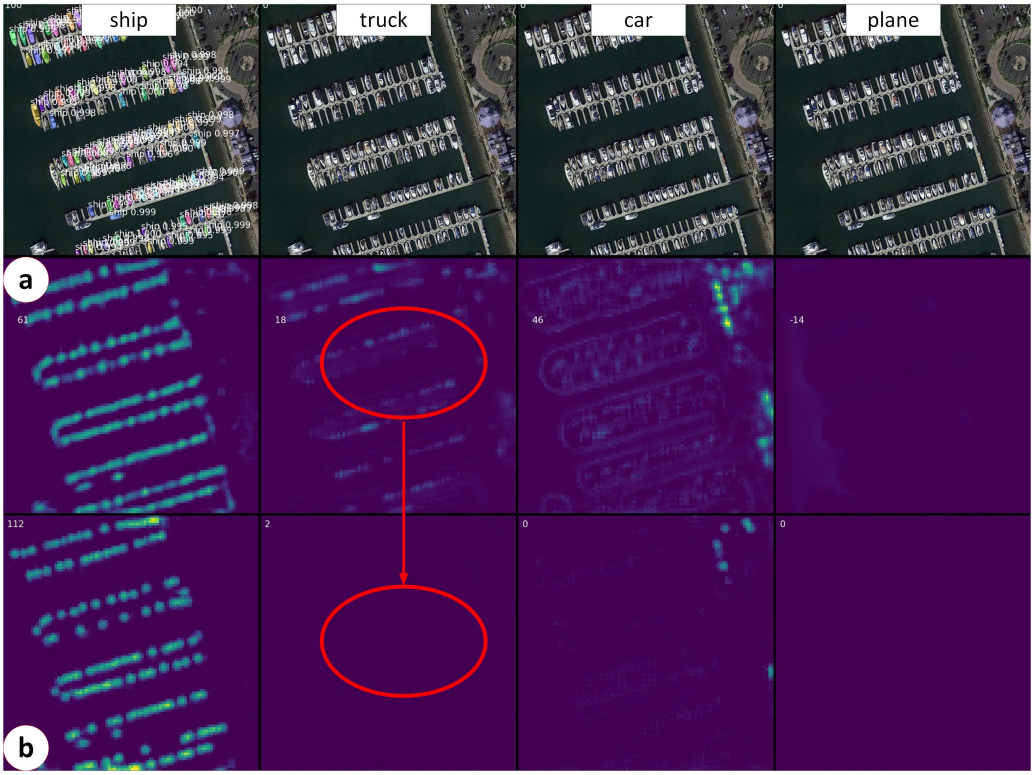


Figure 5.9: Comparison between Multi-Decoder and Single-Decoder Approach. This figure illustrates the density maps created from a single-decoder and a multi-decoder approach. Part (a) shows the single-decoder method with crosstalk between the density maps. In contrast, the density maps from the multi-decoder approach of Part (b) are almost crosstalk-free.

the predicted numbers of the other categories. By adding an additional decoder part for each category and simultaneously reducing the number of trainable parameters in each backbone, the crosstalk decreased dramatically. Figure 5.9(b) displays the results after applying these improvements. Simultaneously, due to the reductions in crosstalk, the number of predicted objects in the scene became more precise.

CEN Variants. The proposed CEN stands out for its efficient and versatile design. Table 5.3 shows the counting results of the CEN on the iSAID dataset over multiple intervals. The first row of Table 5.3 shows the CEN without the rescaling factor λ_{res}. A performance loss can be seen in comparison to the base implementation of the CEN. This effect becomes more substantial with the increasing variety of input image resolutions. The second row shows the counting results without dividing the L_2-loss by the number of objects. We expected a performance increase in the most common interval and a decrease in the performance of the remaining brackets. But surprisingly, by applying a relative loss term instead of the original L_2-term, the counting performance improved in all intervals.

Table 5.3: CEN Ablation Results. This table presents the results of multiple CEN variations on the iSAID dataset.

iSAID		0-10000	0-10	11-50	51-100	101-10000
CEN	**MAE**	9.82	3.02	9.80	25.29	66.90
(simple resize)	**RMSE**	35.29	11.34	22.38	34.71	112.02
CEN	**MAE**	8.40	2.56	8.49	18.41	57.28
(simple loss)	**RMSE**	34.73	10.46	20.133	28.19	109.31
CEN	**MAE**	7.85	2.50	6.51	17.86	50.31
(base)	**RMSE**	31.64	10.26	12.18	26.38	95.57

5.2 Fusion of Density Estimation and extended Object Detection for Improving Object Detection

Object detection methods frequently struggle with diminished performance in the presence of high object densities, a challenge that significantly impacts various real-world applications. To mitigate this issue, we introduce the Density Region Proposal Network (DRPN), a novel architectural component explicitly designed to enhance object detection capabilities in scenarios characterized by substantial object clustering. The DRPN leverages object density maps generated by the CrowdFPN, as detailed in Section 3.3. By employing these density maps, the DRPN excels at producing region proposals that are particularly well-suited for subsequent processing by two-stage object detectors. Furthermore, we advance the field by integrating our DRPN within the Mask R-CNN framework. This integration not only enhances object detection capabilities but also generates concurrent object density maps. These density maps can be effectively utilized alongside the detection results, enabling a comparative analysis that strengthens detection reliability. By combining detection data with corresponding density information, our approach enhances detection outcomes, effectively addressing the complexities of high-density environments.

5.2.1 Introduction

The ability to count, localize, and identify objects in aerial images yields valuable insights for researchers and decision-makers. However, the manual exploration and evaluation of vast tracts of land on a large scale remain impractical.

To address this challenge, the application of computer vision techniques is essential for extracting meaningful information from the vast quantities of visual data gener-

ated. A family of methodologies known as object detectors presents a viable solution for this task.

Another relevant technique in computer vision is density estimation, specifically designed for recognizing congested scenes. Algorithms like MCNN [177] estimate density maps that encapsulate the density of objects in specific areas, effectively determining the number of objects per section. In crowd counting applications, density estimation predicts the number of individuals in congested environments and allows delineating zones of high density.

The fusion of object detection and density estimation [85] allows for the synergistic exploitation of both methodologies. To that end, we propose a novel approach that simultaneously generates density maps and object detections, thereby enhancing the object detection process through the incorporation of density information into the region proposal mechanism. In summary, we present the following contributions:

1. We introduce an alternative scheme for region proposal in scenes exhibiting varying object density: the Density Region Proposal Network (DRPN). The DRPN leverages object density maps to enhance the anchor generation process.
2. We integrate the density estimation network CrowdFPN and the DRPN into the modern instance segmentation framework, Mask R-CNN.
3. To validate the effectiveness of our approach, we conduct tests of the DRPN on two distinct crowd counting datasets and evaluate its performance on a remote sensing instance segmentation dataset.

5.2.2 Approach

Following this introduction, we detail our proposed enhancements to object detection for datasets characterized by varying object densities. Our primary objective is to augment two-stage object detectors, such as Faster R-CNN or Mask R-CNN, by calculating a density map and dynamically generating region proposals tailored to specific scenes. To achieve this, we employ a density estimation network known as CrowdFPN, which is described in Section 3.3.

Basic Idea. Figure 5.10 illustrates the standard flow of Mask R-CNN, extended with a density estimation branch. Initially, features are extracted from a backbone network, which are subsequently used to propose regions of interest in the Region Proposal Network (RPN). The proposed regions undergo a filtering process that is heavily influenced by a set of hyperparameters. After filtering, the regions of

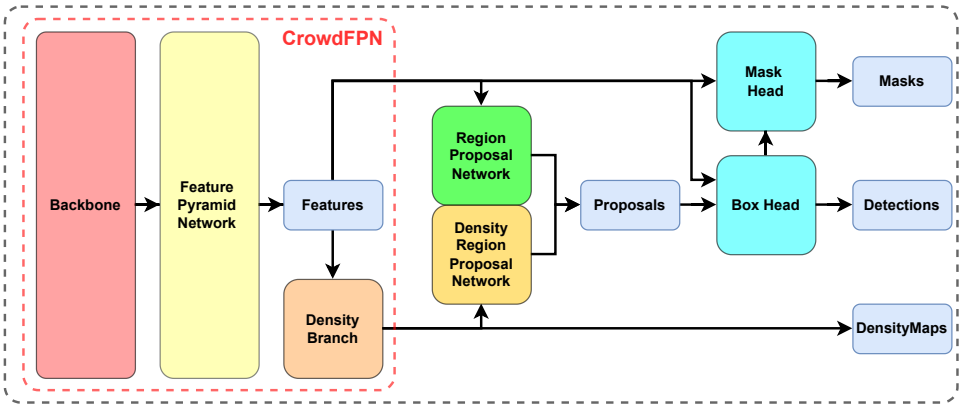


Figure 5.10: Integration of DRPN into Mask R-CNN. We incorporate CrowdFPN and the Density Region Proposal Network (DRPN) into the Mask R-CNN framework. CrowdFPN generates object density maps, which serve as input for the DRPN to create enhanced region proposals. These region proposals can be seamlessly integrated with the default proposals. Ultimately, the combined region proposals are input into the box head for further processing.

interest are aligned with the features to predict bounding boxes and corresponding class labels. Finally, masks for the detected objects are estimated. In addition to feature extraction, the extended backbone generates a density map, which is processed within the Density Region Proposal Network (DRPN) module to calculate region proposals for the Mask R-CNN workflow. Both the RPN and the DRPN can function independently or in conjunction. In the second case, the region proposals are concatenated for further processing steps.

Density Region Proposal Module. The first step within the DRPN module involves the generation of anchor points. This is achieved by applying a filtering process to the density map, where each pixel must exceed a predetermined density threshold ds_t to qualify as an anchor point. For each anchor point, we generate a region proposal in a predefined set of scales and shapes. These region proposals can be further processed in additional detection heads for fine-tuning and classification.

5.2.3 Experimental Results

We integrate our backbone into the standard workflow of Mask R-CNN and evaluate our proposed DRPN method on the instance segmentation dataset iSAID [160]. During the training process, we employed the Adam optimizer for a total of 200 epochs, with a batch size of 4, a learning rate of $1 \cdot 10^{-5}$, and a weight decay of $1 \cdot 10^{-5}$. Empirical evaluations indicate that the default Mask R-CNN hyperparameters, with

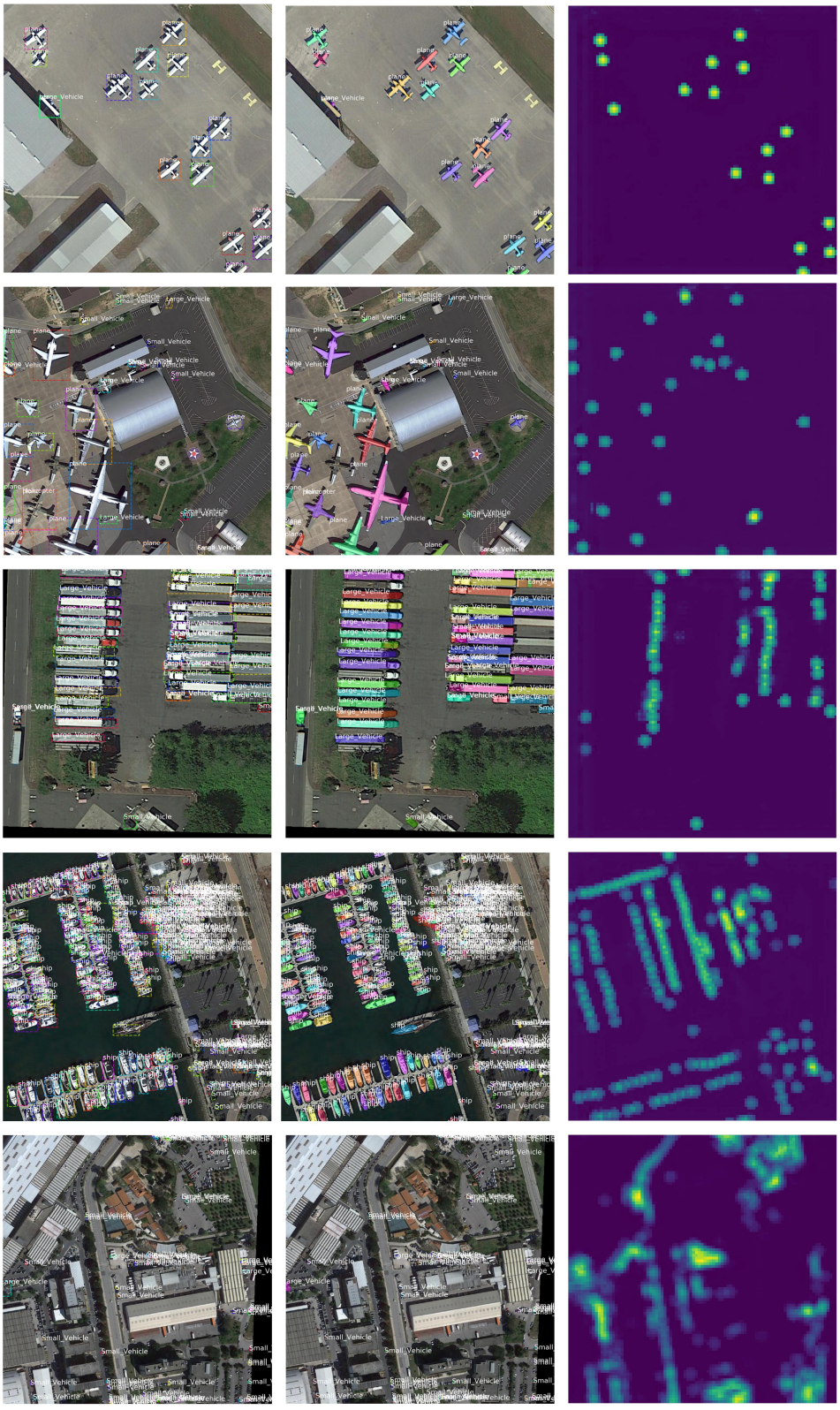


Figure 5.11: Quantitative Results of DRPN. This figure presents processed iSAID examples across varying levels of object density, highlighting the performance of the Density Region Proposal Network (DRPN).

Table 5.4: Results on iSAID. Results of DRPN, RPN and their combination on the instance segmentation dataset iSAID. Combining both approaches leads to a slight decrease in mAP, but a large gain in mAR.

settings			IoU metric: box			IoU metric: segm		
metric	**IoU**	**maxDets**	**RPN**	**Both**	**DRPN**	**RPN**	**Both**	**DRPN**
AP	0.50:0.95	10	**20.2**	20.1	15.7	**16.5**	16.4	1.34
AP	0.50	10	**28.3**	28.1	22.7	**27.6**	27.4	2.20
AP	0.75	10	**23.9**	23.8	18.4	**18.8**	18.7	1.54
AP	0.50:0.95	100	**35.8**	35.4	23.8	**30.8**	30.3	2.17
AP	0.50	100	**52.5**	51.7	36.3	**51.7**	50.7	3.63
AP	0.75	100	**41.8**	41.4	27.2	**34.8**	34.3	2.47
AP	0.50:0.95	500	**36.7**	36.5	23.9	**31.8**	31.5	2.19
AP	0.50	500	**54.7**	54.0	36.9	**53.9**	53.1	3.72
AP	0.75	500	**42.7**	42.4	27.2	**35.5**	35.3	2.48
AR	0.50:0.95	10	23.9	**24.8**	18.6	18.4	**18.8**	1.51
AR	0.50:0.95	100	41.9	**44.1**	29.7	35.0	**36.0**	2.68
AR	0.50:0.95	500	43.0	**46.0**	30.2	36.3	**38.1**	2.77

the exception of a box score threshold set to $1 \cdot 10^{-3}$, yielded the best results. Notably, the DRPN module was integrated into the workflow without the need for additional training and with a density threshold ds_t of 0.5.

The performance of the base Mask R-CNN workflow, the configuration utilizing DRPN in place of the RPN, and an alternative setup combining both DRPN and RPN is assessed using the Microsoft COCO [84] API (see Table 5.4). While the DRPN module demonstrates the capability to detect objects within scenes, its performance falls short compared to the standard Mask R-CNN flow. The combination of DRPN and RPN results in a slight decrease in mAP; however, it achieves a 3% improvement in mAR for densely populated scenes.

Figure 5.11 presents examples of the outputs generated by the mixed region proposal architecture. Furthermore, when focusing solely on the total count of objects present in a given scene, the density map provides a more accurate count in densely populated scenarios, underscoring the effectiveness of our approach in capturing object density metrics.

5.3 Fusion of Density Estimation and extended Object Detection for Improving Dust Density Estimation

Density estimation is essential for enhancing dust monitoring, as discussed in Chapter 3, and can improve the object detection techniques described in Sections 5.1 and 5.2. Moreover, advanced object detection methods that are extended by segmenting object shapes, known as instance segmentation, can further enhance the accuracy of density estimation. This section explains how instance segmentation can refine and elevate density estimation methods, highlighting the beneficial connection between these two domains.

5.3.1 Introduction

Airborne dust significantly influences both climate and human health. Comprehensive monitoring of dust can facilitate the identification of environmental hazards and the formulation of effective mitigation strategies. Nevertheless, traditional dust measurement instruments tend to be expensive and often fall short in capturing the spatial characteristics of dust. While widely available RGB camera systems present a promising alternative for assessing these spatial attributes, the automatic detection of airborne dust within such images remains an underexplored area of research. The development of algorithms for this automated detection faces numerous challenges, including the inherent opacity of dust, the diverse range of density levels, the visual similarities to phenomena such as smoke or clouds, and the indistinct boundaries that characterize airborne dust.

In Chapter 3, the development of several dust density estimation algorithms is outlined. Although the capabilities for dust monitoring are impressive, some challenges remain. One key limitation, as discussed in Subsection 3.9.2, is the insufficient regression ability during high dust density events, especially with very small dust plumes. To address this challenge, we propose integrating instance segmentation algorithms into our density estimation workflow. This approach specifically targets high dust density events, thereby improving accuracy in these situations. The combination of these two methodologies enhances dust monitoring performance, particularly in scenarios with elevated dust density.

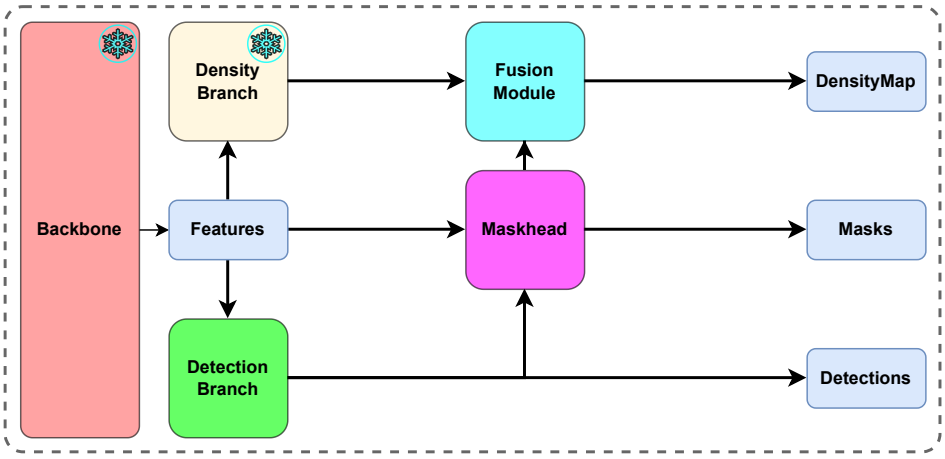


Figure 5.12: Overview of Our Proposed DustSpot. This is a comprehensive overview of our proposed DustSpot. Features extracted from the backbone are processed through two separate branches: the detection branch and the density branch. The object masks generated by the mask component of the object detection branch are subsequently integrated with the density maps, resulting in the creation of refined dust density maps.

5.3.2 Approach

The fundamental premise of our methodology is to integrate object detection techniques with density estimation approaches to enhance the accuracy of dust density estimation. An overview of our method is illustrated in Figure 5.12. We employ a standard backbone architecture, as outlined in [48, 90], consisting of four layers designed to extract feature maps at four different resolution scales: $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of the original image. This design allows for a straightforward replacement of feature extractors, enabling us to remain aligned with ongoing technical advancements in the field.

In the density estimation branch, we utilize algorithms such as DustNet and DustNet++, which were elaborated upon in Chapter 3. For instance, DustNet processes the feature maps extracted from the backbone to generate a dust density map D with dimensions $\frac{H}{2} \times \frac{W}{2} \times 1$. Building upon these methods, we introduce our new approaches, DustSpot and DustSpot++, which are derived from DustNet and DustNet++, respectively.

We adhere to the convention established by DustNet, where the ground truth corresponds to a quarter of the original image size in pixels.

In the detection branch, we can implement convolutional-based instance segmentation algorithms, such as Hybrid Task Cascade [16], as well as transformer-based detectors like MaskDINO [79]. This process yields N dust instance segmentation masks M. These instance segmentation masks correspond to the ground truth masks derived from the density estimation ground truth.

To create the density detection maps, we apply a threshold to the density maps, removing all values below a defined threshold d_{thr}. Following this thresholding step, we employ a connected-component labeling algorithm, as outlined in [40], utilizing 4-connectivity. For each unique segmentation that meets a specified size in pixels, we extract a binary mask and compute the corresponding bounding box. These segmentation masks subsequently serve as our ground truth, providing a reliable reference for training the detection branch.

After calculating the density map D and the instance segmentation masks M, both outputs are subsequently processed through a fusion module. In the initial step, the instance segmentation results M are downscaled to match the dimensions of D. Following this, all masks M are aggregated into a single mask D_{Det} that encompasses multiple instances. To ensure proper alignment with the ground truth, we restrict all values in D_{Det} to a maximum of one and multiply it by the corresponding threshold value d_{thr} used in the ground truth creation process. In the final step, we combine the two density maps, D and D_{Det}, by selecting the maximum value from each corresponding pixel position. This method ensures that if the density estimation process does not accurately regress a value, the resulting density map is raised to a lower bound in high-density regions, thereby enhancing the robustness of our density estimation.

5.3.3 Experimental Results

In this section, we summarize the implementation specifics of our experiments, present the quantitative results of our approach, and finally illustrate the qualitative outcomes.

Implementation Details. Initially, we trained DustNet S and DustNet++ following the scheme described in Section 3.7, utilizing a Swin Large backbone. We then froze the backbone weights and integrated them into MaskDINO. To construct the object detection dataset, we adhered to the methodology outlined in Section 5.3.2, deriving it from the Meteodata dust dataset with a threshold d_{thr} set at 70% of the maximum value. During training, we applied data augmentation techniques, including random

vertical image flipping, random multiscale resizing, and random cropping operations. We employed the AdamW optimizer [92] with an initial learning rate of $2 \cdot 10^{-3}$ and a weight decay of 10^{-3}. Additionally, we implemented gradient clipping and applied a learning rate multiplier of 0.1 to the backbone. After completing the training phase, we fused both networks for inference and conducted testing on the Meteodata dust test dataset.

The intention behind this approach was that training within a multi-learning scheme led to convergence problems in the neural network. Specifically, joint training on detection and density estimation tasks resulted in significantly poorer performance. Training sequentially—first on detection and then on density estimation—also did not yield satisfactory results. Therefore, we opted to train the networks separately and then fused them during inference, which proved to be more effective.

Quantitative Results. We conducted a comprehensive quantitative evaluation of our fusion approach, which integrates DustNet S and DustNet++ with an object detection component. The results are illustrated in Table 5.5.

When comparing DustNet S to DustSpot S, we observe that the combined approach exhibits a slight decrease in performance as measured by the MAE and MSE. This slight degradation is also apparent in the low-density brackets for both metrics. Notably, the classification accuracy between dust and no dust remains unchanged in these comparisons. Despite the marginal decrease in MAE and MSE, the integration of an object detector effectively addresses one of the primary weaknesses inherent in the regression approaches. Specifically, while the overall mean MAE and MSE across all brackets improved, the most significant enhancements are evident in scenarios characterized by high dust densities. The object detection component enhances the model's ability to accurately capture high-density dust events, leading to better performance in these challenging conditions. This improvement is also observed with DustNet++. In this case, the fusion approach yields a more substantial enhancement in the high-bracket MAE and high-bracket MSE, as well as improvements in the overall MAE and MSE. The greater degree of improvement with DustSpot++ suggests that the fusion approach synergizes effectively with models that are more sensitive to variations in dust density, thereby providing more accurate estimations in high-density scenarios.

Qualitative Results. Figure 5.13 presents three events occurring within a mining scene. In the leftmost image, the fusion approach elevates the predicted dust levels, although they remain somewhat lower than the ground truth measurements. A similar observation can be made in the middle image, where the fusion approach assists in detecting small, high-density dust events, a known weakness of the pure

Table 5.5: Quantitative Results. We comprehensively evaluated our combined approach, integrating DustNet S and DustNet++ with an object detection component. The experimental results on the Meteodata dust test dataset demonstrate that our fusion approach DustSpot enhances performance, particularly in scenarios characterized by high dust densities. In addition to the default metrics MAE, MSE, Acc, Pre, Rec, and IoU on the full images, the pixel values are binned into zero dust (ZB), low dust (LB), medium dust (MB), and high dust (HB) densities, while ØB-MAE and ØB-MSE represent the mean over all bins.

Model	DustNet S	DustSpot S	DustNet++	DustSpot++
Backbone	Swin-L	Swin-L	Swin-L	Swin-L
CPW	✓	✓	✓	✓
#Img	1	1	1	1
MAE	12.80	12.82	12.63	**12.61**
MSE	**402.20**	404.87	422.33	422.20
ØB-MAE	21.19	**20.88**	22.43	21.67
ØB-MSE	941.10	**926.47**	1039.23	993.78
Acc	**0.885**	**0.885**	0.879	0.879
Pre	**0.888**	**0.888**	0.883	0.883
Rec	**0.886**	**0.886**	0.881	0.881
IoU	**0.793**	**0.793**	0.784	0.784
HB-MAE	36.06	**34.37**	41.20	37.78
HB-MSE	2023.29	**1938.62**	2394.30	2189.27
MB-MAE	**25.62**	26.21	25.89	26.21
MB-MSE	1091.23	1093.88	**1075.72**	1093.88
LB-MAE	**14.63**	14.66	15.55	15.58
LB-MSE	**428.98**	432.56	463.08	466.69
ZB-MAE	8.47	8.47	**7.09**	**7.09**
ZB-MSE	**220.89**	222.47	223.80	225.27

regression approach. Finally, the rightmost scene also demonstrates improvement through the fusion method. Specifically, the integration of the object detection component with DustNet S and DustNet++ leads to more accurate dust density predictions. The fusion approach effectively identifies and quantifies areas with high dust concentrations, thereby enhancing the overall estimation accuracy compared to the base model without detection.

Figure 5.14 illustrates scenes with varying levels of cloud coverage. Notably, our detection approach demonstrates robustness against confounding factors such as clouds; it does not misinterpret cloud formations as dust. This capability indicates that our method effectively distinguishes between dust particles and clouds, ensuring reliable dust detection even under challenging atmospheric conditions. The ability to accurately discriminate between dust and clouds enhances the practical applicability of our approach in real-world mining operations where such conditions are prevalent. Moreover, the fusion approach is capable of detecting small-scale, high-density

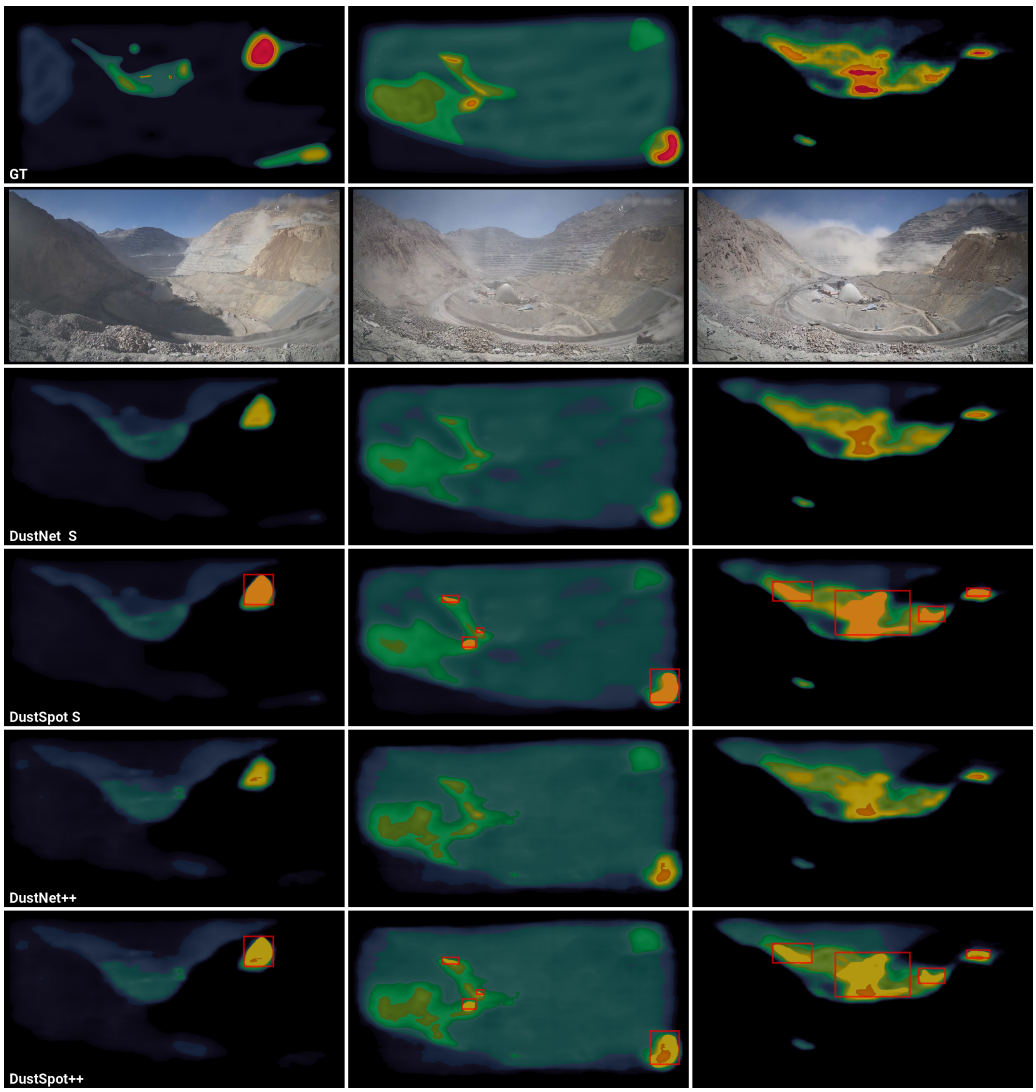


Figure 5.13: Mining Scene Under Varying Dust Levels. This figure illustrates the same mining scene under varying dust levels. In nearly every scenario, our fusion approach enhances the performance of the base density estimation method. Specifically, integrating the object detection component with DustNet S and DustNet++ leads to more accurate dust density predictions. The fusion approach effectively identifies and quantifies areas with high dust concentrations, thereby improving the overall estimation accuracy compared to the base model without detection.

dust events, which represents a significant advantage. However, it is essential to acknowledge that the detection component is limited to identifying dust at a single level of density. Despite this limitation, the combined approach constitutes an improvement over the pure regression model, as it enhances performance in

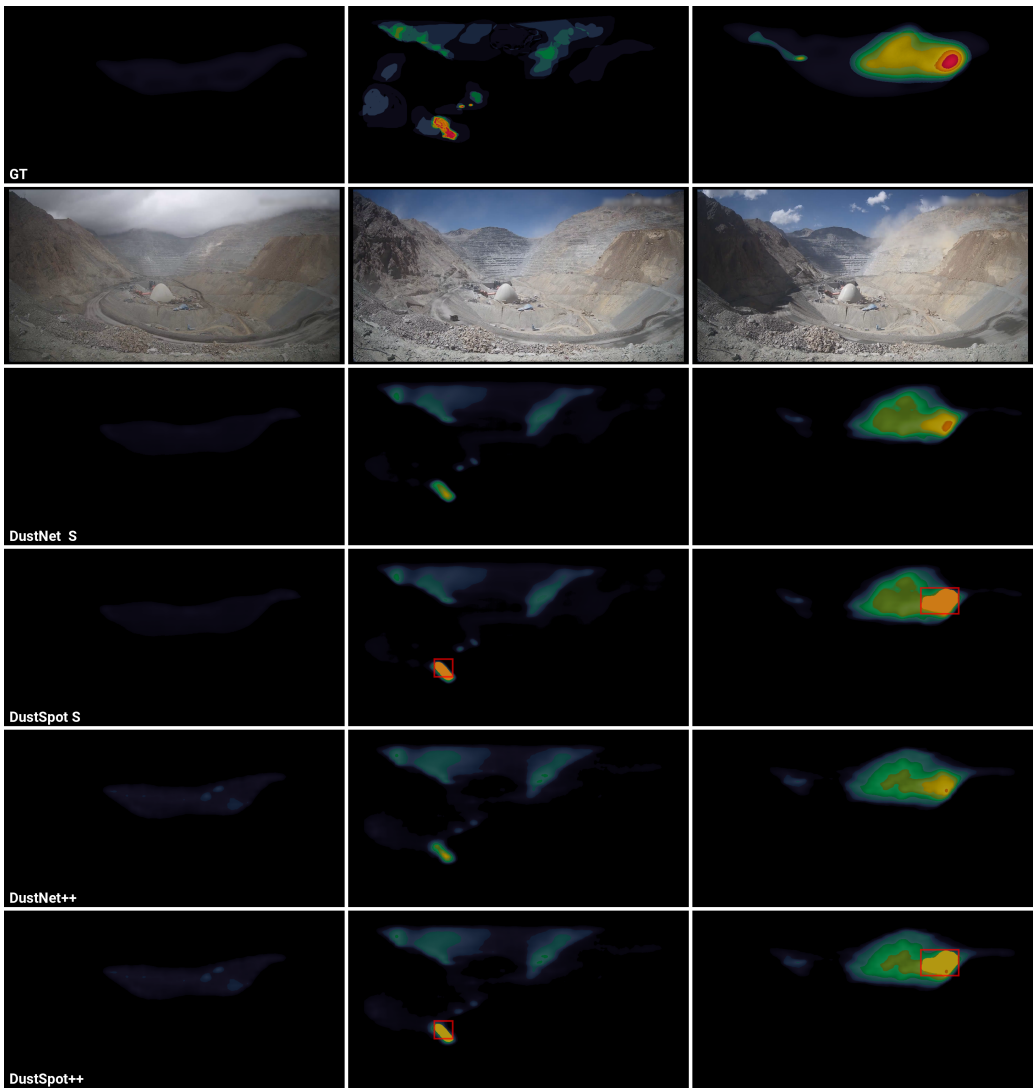


Figure 5.14: Mining Scene with Different Cloud Coverage. This figure presents various mining scenarios alongside their corresponding dust estimation predictions under different cloud coverage conditions. The fusion approach effectively distinguishes between dust and clouds, similar to the regression models. However, it also enhances performance in situations characterized by high dust densities, particularly in detecting small-scale dust events. By integrating the object detection component with DustNet S and DustNet++, the fusion approach not only maintains accurate differentiation between dust and cloud formations but also significantly improves the detection and quantification of dust in challenging conditions. This enhancement is especially notable for small dust events, which are often difficult to detect due to their subtle signatures and the presence of cloud cover. The improved performance underscores the robustness of our method in complex atmospheric scenarios commonly encountered in mining environments.

detecting and quantifying dust, especially in scenarios involving subtle or high-density dust occurrences.

5.4 Discussion

The first study presented in this chapter examined the integration of a multi-class density estimation network with an object detection branch to form a count estimation network. This combined approach significantly enhanced counting results on two widely used object detection datasets. By leveraging both, spatial distribution information and precise object localization, the model achieved more accurate and robust counting performance across a variety of scenarios.

Experimental results confirmed that object detection methods excel in situations involving low-density counts with minimal to moderate occlusion. In such environments, objects are sufficiently separated and visible, allowing detection algorithms to identify and localize each object effectively. However, these methods underperform in scenes characterized by high and diverse object counts due to challenges posed by occlusions and overlaps. Conversely, density estimation methods perform well in high-density scenes but suffer in performance in low-density scenarios.

Furthermore, it was demonstrated that training the proposed end-to-end approach in a multi-loss manner is feasible, simplifying the training process. Training all components simultaneously resulted in only a slight decrease in the performance of the detection branch. This minor reduction was anticipated because multi-loss learning typically requires a precise balancing of the loss terms to prevent any single task from dominating the training. Moreover, the results indicate that employing a multi-decoder approach is crucial to avoid crosstalk between different classes. Crosstalk occurs when signals or gradients associated with one class interfere with those of another, potentially degrading performance. Although using separate decoders mitigates this issue, it can be computationally expensive, especially when dealing with a large number of classes, since each additional decoder increases the computational load.

Additionally, the rescale factor in the CEN was found to be critical for the counting performance. This rescale factor adjusts the network's output to account for variations in image sizes, ensuring accurate counting regardless of input dimensions. Without this adjustment, the CEN cannot correctly estimate object counts for images of different sizes, leading to inconsistent results. Interestingly, applying a relative loss term instead of the traditional L_2 loss term improved counting performance across all intervals of object counts. It was initially expected that the relative loss would enhance performance in the most common interval while potentially decreasing performance in intervals with very low or high object counts. However, the

observed improvement across all intervals suggests that the relative loss provides a more balanced and effective optimization across different count ranges.

In the next study, we combined density estimation with object detection to enhance object detection performance. This approach resulted in a very slight decrease in mAP, but it yielded a substantial increase of 3 points in mAR compared to the pure RPN approach. The minor reduction in mAP can be explained by the fact that an increase in potential detections may lead to more false positives; however, this is more than offset by the significant gain in recall. Notably, using only the DRPN module led to a considerable decrease in performance, indicating that it cannot effectively function as a standalone indicative method. It should not be overlooked that deep learning-based object detection algorithms often include parameters that limit the number of objects they can detect. This limitation is particularly pronounced in DETR algorithms, which operate with a fixed number of queries. Consequently, when the actual number of objects exceeds this limit, these algorithms may fail to detect all objects. To address this issue, incorporating an additional density map can serve as an indicator of the validity of the results. Detecting high densities suggests that the threshold for the maximum number of detected objects has been surpassed. Recognizing this allows for necessary adjustments to be made, leading to more reliable outcomes.

Finally, in the concluding study of this chapter, the accuracy of dust density estimation is enhanced by combining it with instance segmentation. Utilizing instance segmentation results for high dust levels addresses the suboptimal performance of the dust density estimation algorithms presented in Chapter 3. This integration leads to substantial improvements: regression metrics in regions with high dust densities are enhanced, and smaller dust plumes are more readily detected. However, the current instance segmentation approach employs only a single dust level in the segmentation masks, leaving room for further research. Implementing a multilevel approach could be beneficial, potentially improving the model's ability to distinguish between varying dust densities.

Collectively, these findings underscore the importance of integrating multiple methodological approaches and carefully considering training strategies and loss functions in the development of effective counting models. Addressing challenges such as accurately counting objects in high-density scenes and managing computational complexity with multiple classes is essential for improving accuracy and efficiency in complex real-world environments.

5.5 Conclusion

In this chapter, we have demonstrated how to leverage fusion approaches between density estimation and object detection to enhance detection and counting performance in scenes characterized by significant variability in scale and density.

First, we propose a hybrid approach that integrates density estimation with object detection for multi-class object counting. Our findings indicate that the density estimation branch excels in scenarios with high object density, while the detection branch performs better in contexts with fewer objects. Additionally, we introduce a count estimation network that effectively fuses both branches, significantly improving prediction accuracy compared to relying solely on one branch. To validate the effectiveness of the proposed approach, evaluations were conducted on two widely used object detection datasets, and it was benchmarked against algorithms from the R-CNN family. As previously discussed, establishing a robust benchmark for multi-class object counting in environments with highly variable densities is essential for advancing research in this field.

Next, we propose a method that fuses object detection with density estimation. This approach generates density maps alongside object detections for a given scene. The density estimation branch employs multi-scale features to create contrastive features, thereby enhancing the density estimation process. Furthermore, non-uniform anchor proposals guided by the density branch improve the detection process. Our model, designated as DRPN, achieves a 3 points higher mAR when combined with a baseline RPN. Additionally, the overall number of detections can be validated through the density estimation branch.

Finally, we present a fusion approach called DustSpot to improve dust density estimation. The methods proposed in Chapter 3 are combined with an instance segmentation algorithm—in our case, the state-of-the-art method MaskDINO. While a multi-task learning approach was not successful, a specific training scheme led to a successful outcome. The DustSpot and DustSpot++ approaches address the weaknesses of existing dust density estimation methods, particularly in handling small and high dust density events. We demonstrate the capability of our approaches on the Meteodata dust density dataset.

6 Discussion

In the following, quotations to related publications are highlighted by colored bars placed on the outer border of the text, even though minor editorial changes have been made. The color encoding used is as follows:

- **Michel, A.**, Weinmann, M., Kuester, J., AlNasser, F. , Gomez, T., Falvey, M., Schmitz, R., Middelmann, W. & Hinz, S. (2025, February). DustNet++: Deep Learning-Based Visual Regression for Dust Density Estimation. In International Journal of Computer Vision (pp. 1-25). Springer. Reprinted with permission. It is cited as [100]. ■

This chapter provides a comprehensive discussion of the main themes of the work. Beginning with an impact analysis of dust density estimation research, it then addresses research conducted on object detection and object counting. Finally, it explores the broader societal implications of dust research.

6.1 Research Impact for Dust Density Research Impact

Our study has shown that throughout the evolution of the proposed neural network architectures in Chapter 3, a crucial aspect of dust density estimation is the simultaneous focus on the significance of integrating both global and local features. CrowdFPN is built on a strong FPN structure to process and combine high- and low-level semantic features. DeepDust improves this approach by simplifying the FPN structures and the fusion module. DustNet further enhances this by ingesting temporal information and adding a branch for high-level semantic features. By exploiting adaptive convolutional neural networks, the feature fusion process is improved. Finally, DustNet++ again utilizes only one branch but employs sparse global attention in combination with window attention to process high- and low-level semantic features. The ablation studies have shown that this is a critical aspect. The local features are essential for detecting dust, while the high-level features are vital to avoid false positive classifications with similar phenomena, such as clouds or smoke. Furthermore, temporal data and the use of advanced machine learning

training techniques like CPW to increase the batch size have led to impressive results on the Meteodata dust dataset and the binary segmentation URDE dataset. Notably, experiments on the latter datasets have demonstrated the excellent generalizability of the proposed methods.

However, key drawbacks were observed. The proposed models tends to overlook minor dust plumes, which is hypothesized to be a trade-off for reducing false positives, ultimately leading to a lower loss during training. Additionally, it is conjectured that using the L_2 loss function may exacerbate the regression ability in bins characterized by sparse occurrences. Notably, bins containing rare, high dust values exhibit significantly poorer performance than those with more frequently occurring, lower values.

Adding a detection branch, as described in Section 5.3, provides relief. The regression metrics in areas with high dust densities can be improved, and smaller dust plumes can also be detected more easily. However, the instance segmentation approach currently employs only a single dust level in the instance segmentation masks, which still leaves room for improvement. A multilevel approach could be utilized here.

The Meteodata dust dataset relies on subjective annotations by human annotators rather than physical measurements, introducing inherent subjectivity and variability. The indistinct dust boundaries within this dataset add further ambiguity compared to datasets used for classification or object detection tasks. Consequently, the model may occasionally reflect actual dust conditions more accurately than the annotated ground truth. Therefore, achieving superior test metric results does not necessarily correspond to enhanced performance in practical, real-world applications. Although the model is also evaluated using the URDE binary segmentation dust dataset, a comprehensive comparison between density estimation and semantic segmentation approaches to dust remains essential.

6.2 Research Impact for Object Detection

Detecting objects at varying scales and densities presents considerable challenges, often necessitating post-processing techniques to enhance accuracy. In this context, ARM-NMS demonstrates significant superiority over box NMS variants in both bounding box detection and mask predictions, marking an important advancement in non-maximum suppression techniques for object detection and segmentation tasks. Our ablation experiments show that ARM-NMS outperforms Mask-NMS due to the inclusion of area-rescoring, which refines detection scores based on object

size, enhancing overall detection accuracy. However, these conclusions are based on experiments with a single dataset, indicating the need for further testing across multiple datasets to ensure the method's robustness and generalizability. The dependency of NMS on multiple parameters significantly adds complexity to the optimization process, as optimal values may vary with different datasets or data shifts. Although area rescoring improves detection performance, it introduces additional parameters and can cause glitches, such as anomalously large, low-confidence masks receiving inflated scores and suppressing correct detections. Addressing these issues requires strategies to detect and manage glitched masks effectively.

This study underscores the importance of a data-centric approach in machine learning model development. It reveals that conventional inference strategies do not always enhance performance, potentially due to data shifts caused by augmentation techniques like multiscale resizing or errors arising from the fusion process with filtering methods. By focusing on modifying existing datasets—specifically by selecting only the relevant classes—an improvement in performance was achieved, highlighting the critical role of data quality and relevance. Furthermore, the 2024 MACVI challenge demonstrated that detection algorithms typically deemed superior in standard computer vision benchmarks, such as Co-DETR [182] compared to YOLOv8 [63], can be outperformed by models trained on a wide array of pertinent data samples. This finding implies that the selection and diversity of data may have a more significant impact on model performance than the neural network architecture itself, emphasizing the profound effect of dataset optimization over solely relying on advanced neural network designs.

Combining density estimation and object detection to improve object detection led, in this case, to a negligible decrease in mAP but resulted in a substantial increase of 3 points in mAR compared to the pure RPN approach. This can be explained by the fact that more potential detections may lead to more false positives, but this is more than compensated by the significant gain in recall. Nevertheless, using only the DRPN module leads to a significant decrease in performance, which is the reason why it cannot be used as a standalone indicative method. However, deep learning-based object detection algorithms typically have parameters that limit the number of detected objects, particularly in DETR algorithms with a fixed number of queries. An additional density map can serve as an indicator of the validity of the results because detecting genuinely high densities may suggest that the threshold has been surpassed. This leads to more reliable outcomes.

6.3 Research Impact for Object Counting

The study conducted in this work analyzed the fusion of a multi-class density estimation network with an object detection branch to establish a count estimation network (CEN). Experimental results confirmed the assumption that object detection methods excel in situations involving low-density counts with minimal to moderate occlusion, while density estimation methods perform better in high-density counts. Combining these approaches with the CEN led to the best counting results, especially in environments with varying object densities. Training all components simultaneously resulted in only a minor decrease in the detection branch's performance. This slight reduction was anticipated, as multi-loss learning generally requires a precise balancing of the loss terms to ensure that no single task dominates the training.

Surprisingly, applying a relative loss term instead of the traditional L_2 loss term improved counting performance across all intervals of object counts. It was initially expected that the relative loss would enhance performance in the most common interval while potentially decreasing performance in other intervals. Collectively, these findings demonstrate that the fusion approach enhances counting performance in scenarios with strongly varying densities.

6.4 Societal Impact

Mineral dust poses significant health risks to humans, serving as a medium for transporting biological components such as bacteria, endotoxins, and fungi through atmospheric pathways [106]. Although predominantly originating naturally, mineral dust can frequently result from anthropogenic activities. Notable examples include construction sites and stone-crushing plants, where dust adversely affects workers' health [76]. Additionally, eroded agricultural fields can impact nearby rural populations, especially due to the potential for carrying pesticides [107]. Strong winds can also facilitate the long-distance transport of dust from these sources to urban areas.

Despite its harmful effects, mineral dust remains comparatively under-regulated relative to industrial pollutants [106]. This is largely due to its primary impact on smaller, less-monitored populations, such as rural communities and workers. In the United States, the Environmental Protection Agency (EPA) predominantly monitors densely populated cities, leaving smaller communities under-monitored

[150]. Consequently, the limited monitoring efforts perpetuate a feedback loop that intensifies the issue.

To address this problem, the development of effective detection methods is essential. Enhanced detection and monitoring can help mitigate the health risks associated with mineral dust, ensuring better protection for affected populations. Although the total elimination of dust emissions is not feasible, it is possible to implement targeted mitigation strategies. Such measures could encompass but are not limited to, the hydration of untreated roads, the reduction of vehicle speeds, and the moderation of mining operations. However, the implementation of effective and cost-efficient monitoring systems is crucial to refine and optimize dust mitigation strategies.

We hope that DustNet++ and DustSpot, with their enhanced capabilities for detecting airborne dust, will make a substantial contribution to improving dust monitoring methodologies.

7 Conclusion and Outlook

This chapter concludes our work by summarizing the research conducted and provides an outlook based on the results produced, highlighting various aspects for future exploration.

7.1 Conclusion

This work focuses on the analyzing of objects and particles across a wide range of sizes and scales. The aim is to monitor small particles like dust, as well as larger objects such as cars, planes, and ships. To achieve this, neural network architectures for two-dimensional object detection, instance segmentation, and density estimation within the context of remote sensing are analyzed and adapted.

Firstly, several techniques for estimating dust density have been developed, including CrowdFPN, DeepDust, DustNet, and DustNet++ as presented in Chapter 3. Among these methods, DustNet++ emerges as an advanced neural network specifically designed for dust density estimation. DustNet++ computes the dust density for each pixel within a given image by effectively harnessing and integrating local, global, and temporal information. This comprehensive approach allows the model to capture fine-grained details at the pixel level, as well as broader contextual patterns across the image and over time, enhancing its overall predictive capabilities.

Notably, DustNet++ not only excels in regressing various dust levels but also demonstrates a high proficiency in differentiating dust from visually similar phenomena such as clouds. This distinction is crucial in remote sensing applications, where accurate identification of dust particles is essential for environmental monitoring and climate modeling. By accurately distinguishing between dust and clouds, DustNet++ reduces false positives and negatives, thereby improving the reliability of dust detection and analysis.

The proposed methodology demonstrates superior performance, surpassing all competing methods in regression capability on the Meteodata dust dataset and exhibiting improved localization abilities on the URDE dataset. These results underscore the

effectiveness of DustNet++ in accurately estimating dust density and highlight its potential for advancing research and applications in the field of remote sensing. The integration of advanced neural network architectures with comprehensive data processing strategies positions DustNet++ as a significant contribution to the ongoing efforts in environmental monitoring and atmospheric science.

Furthermore, the success of DustNet++ suggests that combining multiple scales of information—local, global, and temporal—can lead to substantial improvements in model performance for complex environmental phenomena. Overall, the advancements presented through DustNet++ provide valuable insights and tools for scientists and practitioners working on dust detection and density estimation, contributing to a better understanding of atmospheric dust and its impacts on climate and human health.

Hereafter, the focus shifts to larger objects, applying training and inference optimization techniques as discussed in Chapter 4, while still concentrating on scenes characterized by significant variability in density and scale.

A novel technique called ARM-NMS is introduced, which enhances object detection by utilizing shape information to filter out unnecessary detections without requiring retraining of existing instance segmentation methods. When applied to detection lists generated by Mask R-CNN on the iSAID validation dataset, ARM-NMS outperforms traditional box-shaped filtering algorithms by more than three points on the COCO-style (mAP) metric. This improvement indicates that rescoring detections based on the shape and area of objects enhances overall detection performance.

Additionally, a data-centric approach is utilized to enhance object detection, as demonstrated in the exemplary case of maritime object detection. This technique enhances detection performance during inference by leveraging multiple representations of the input data, all without modifying the model's architecture. Overall, the findings show that object detection performance depends not only on architectural optimization but also on training schemes, data handling, and post-processing techniques.

In Chapter 5, fusion approaches between density estimation and object detection are utilized to enhance detection and counting performance in scenes with significant variability in scale and density. Firstly, a hybrid method is proposed that integrates density estimation with object detection for multi-class aware object counting. This approach leverages the strengths of both branches: the density estimation branch excels in high-density scenarios, while the detection branch performs better with fewer objects. By effectively fusing these branches through a count estimation

network, prediction accuracy is significantly improved over relying on a individual method. This methodology is validated on two widely used object detection datasets and benchmarked against R-CNN algorithms.

Next, a method is introduced that fuses object detection with density estimation to simultaneously generate density maps and object detections. The density estimation branch employs multi-scale features to enhance the process, and non-uniform anchor proposals guided by the density branch improve detection performance. The resulting model, designated as DRPN, achieves a great increase in mAR compared to the baseline RPN, albeit at the cost of a slight reduction in mAP. Object detectors typically have, by design, an adjustable upper limit on the possible number of detections, whereas density maps do not possess such a limitation. Therefore, density estimation maps can serve as reliable indicators for validating the total number of detections produced by the object detection results.

Finally, the DustSpot approach is presented to improve dust density estimation by combining methods from Chapter 3 with an instance segmentation algorithm, specifically the state-of-the-art MaskDINO. Through a specific training scheme, a successful outcome is achieved. DustSpot addresses the weaknesses of existing dust density estimation methods, particularly in handling small and high dust density events, as demonstrated on the Meteodata dust density dataset.

In conclusion, this study successfully addresses the challenges associated with developing and training deep learning methods for analyzing objects and particles across a wide range of sizes and scales. It outlines the development process of dust density estimation methods, optimizes training and inference techniques for object detection, and ultimately combines object detection with density estimation to enhance the analyzing of remote sensing objects and particles at different scales.

Overall, this study contributes valuable insights to the field of remote sensing by effectively combining object detection and density estimation methods. The findings highlight the potential for advanced monitoring of various phenomena, offering promising directions for future research and applications in environmental monitoring, disaster management, and urban planning.

7.2 Outlook

Improving Dust Density Datasets. Developing precise dust density datasets presents significant challenges due to the inherently ambiguous nature of dust

boundaries and the difficulty in determining appropriate dust levels. To improve datasets for dust density estimation, it is crucial to enhance the creation of subjective ground truth data. An initial approach may involve utilizing predicted dust maps in a semi-automated manner to generate new, refined data samples or to revisit and augment existing datasets. Future developments should prioritize data-centric strategies that leverage advancements in machine learning. For instance, while the Segment Anything Model (SAM) [68] has limitations in handling ambiguous backgrounds, integrating self-supervised techniques such as DINOv2 [109] may yield significant improvements. Ultimately, generating more precise and extensive datasets is likely to substantially enhance the accuracy of dust density estimations.

Enhancing Dust Density Estimation Methods. Enhancing the datasets used for dust density estimation is crucial, but significant performance improvements can also be achieved by refining or replacing existing methodological approaches, especially when larger datasets are available. For instance, integrating a full-transformer architecture into the DustNet++ framework could offer substantial benefits. This architectural evolution is particularly promising in handling extensive datasets. Recent advancements such as Co-DETR [182] for object detection, MaskDINO [79] for instance segmentation, and Mask2Former [21] for semantic segmentation have demonstrated notable success. These methods leverage deformable attention mechanisms, which have proven effective in improving model performance across various tasks. By adopting these innovative methodologies, significant gains in the accuracy and reliability of dust density estimation may be achieved.

Moreover, it is essential to address the issue of imbalanced events, such as high-density dust within the data. Implementing a balanced loss function can mitigate this challenge, contributing to overall performance enhancement. Finally, incorporating additional dust classes into a fusion approach, similar to that employed by DustSpot, can further improve the model's ability to handle diverse scenarios and increase its robustness.

Instance Regression. The COCO [84] API includes a field named *is_crowd*, which denotes the presence of multiple objects within a single mask and is generally overlooked in most datasets. To address this oversight, an extension of instance segmentation by integrating regression masks is proposed. This approach involves specifically applying the regression process to areas where an instance is detected and is suitable for regression. Such a method represents a natural extension of the DustSpot technique presented in this work, which identifies instances but does not regress them. Instance regression could be utilized not only for the detection of dust but also for smoke, clouds, and other opaque objects. Additionally, its application in

Figure 7.1: Concept of a New Task Called Instance Regression. The fundamental idea behind instance regression is to extend instance segmentation, which is depicted in the left image, to also include regression masks. This enables the creation of more precise density masks and can be used in conjunction with instance masks.

crowd counting is also conceivable. Figure 7.1 illustrates the fundamental concept underlying the proposed task.

Evaluating Instance Segmentation versus Semantic Segmentation. In this work, density estimation was compared and combined with both instance segmentation and semantic segmentation. A natural continuation of this research would be to compare and integrate semantic segmentation and instance segmentation methods. For this purpose, the DustSCAN dataset [3] is particularly suitable. DustSCAN is a over a period of five-year, hourly dust plume dataset derived from the Spinning Enhanced Visible and InfraRed Imager (SEVIRI) on geostationary Meteosat satellites. The plumes are clustered into discrete entities using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [37], but still require human supervision to achieve precise results. Therefore, a more automated approach is needed.

Each plume in the dataset has an identifier, making it ideal for an instance segmentation approach. However, initial experiments have shown that although the instance masks can be very close to the ground truth, sometimes two masks appear instead of one large one, or vice versa, which disrupts the training and evaluation processes. This issue seems to be less problematic in semantic segmentation, but initial experiments indicate a lower recall compared to instance segmentation. Figure 7.2 illustrates a SEVIRI image from December 12, 2017, along with the corresponding results. Future work could focus on exploring how to utilize or combine both approaches for dust monitoring in order to reduce the need for human supervision.

Vision Language Integration. Finally, it is important to discuss the recent advancements in multimodal learning, which combine computer vision and natural language

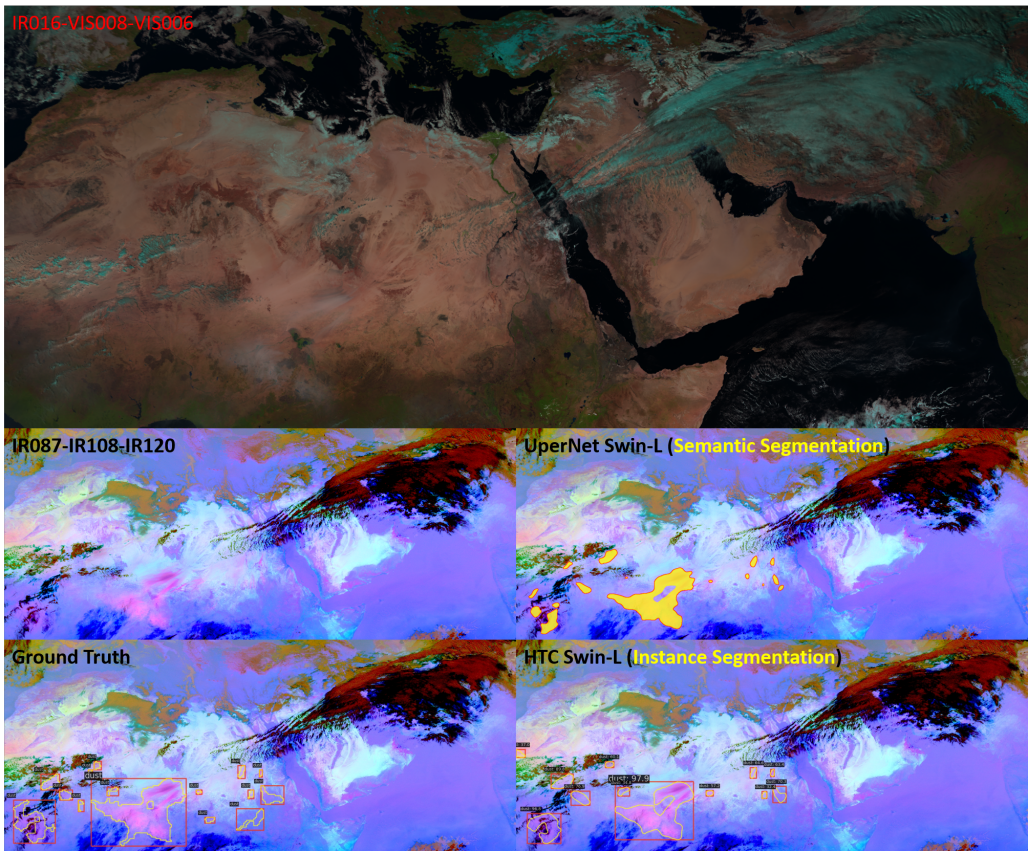


Figure 7.2: Dust Monitoring on the DustScan Dataset. The DustScan [3] Dataset is an hourly dust plume dataset derived from a geostationary-orbit satellite. We present results from both instance segmentation and semantic segmentation neural networks.

processing to interpret and generate images and text jointly. This field has seen significant progress due to the availability of powerful open-source large language models (LLM) such as LLAMA [148] and GWEN [6]. These models have lowered the barrier to entry, enabling researchers without extensive hardware resources to explore and contribute to the development of Vision Language Models (VLM)s. This technology has begun to make an impact in the field of remote sensing as well. A noteworthy example is GeoChat [70], which integrates vision and language models for geospatial applications. GeoChat leverages a CLIP [115] model, designed to learn visual concepts from natural language supervision to generate language embeddings from images. In other words, it translates visual data into a numerical representation that captures semantic information. These embeddings are then adapted to an LLM using a MLP, that maps input data to desired outputs.

As a potential direction for future work, it is proposed that not only should CLIP models and similar architectures be utilized, but traditional approaches such as

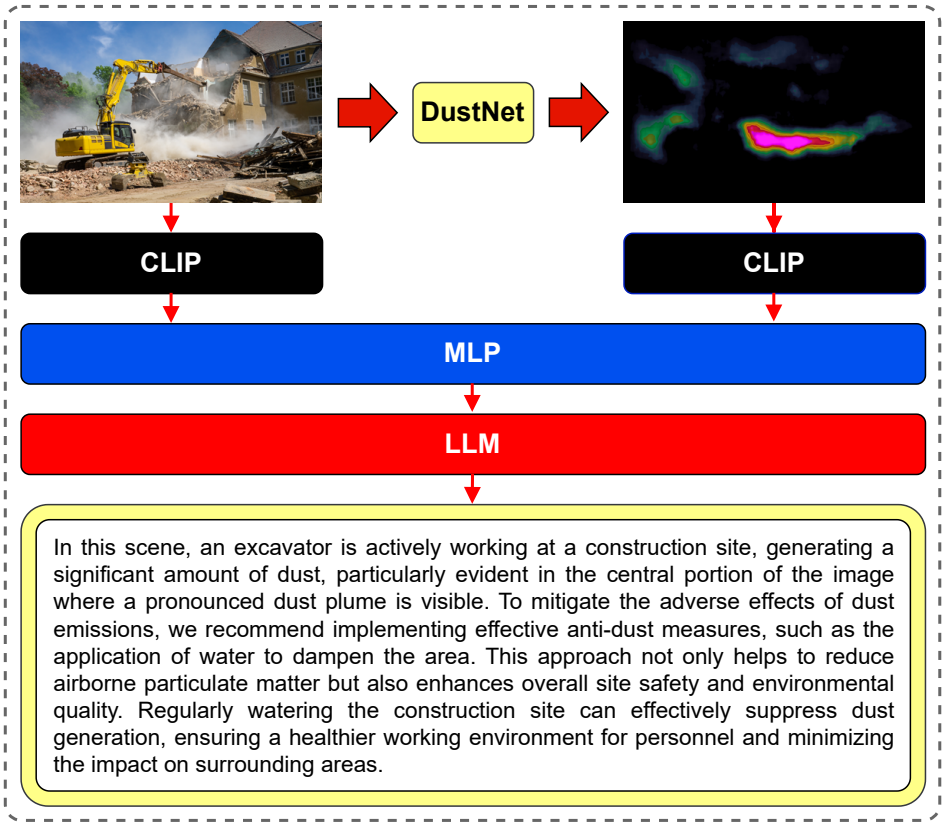


Figure 7.3: Concept of a Potential Method for Vision Language Model Specialized on Dust. In this figure, we present a concept for creating a Vision Language Model (VLM). For our approach, we want to exploit density maps created in order to improve the scene description or visual question answering (VQA).

DustNet should also be incorporated. DustNet is a specialized network tailored for dust detection and analysis in satellite imagery. By integrating specialized networks like DustNet with vision language models, the strengths of both approaches can be harnessed. This strategy enables the use of networks specifically designed for particular tasks, potentially improving accuracy and efficiency in those areas. Figure 7.3 provides a conceptual illustration of this idea, demonstrating how traditional specialized networks can be combined with modern VLMs. Particularly in the domain of remote sensing, which often involves processing multimodal data such as images, spectral information, and textual metadata, there is a significant advantage in leveraging pre-trained specialized networks. These networks have been trained on domain-specific data and can capture nuances that general-purpose models might overlook. By incorporating them into a vision-language framework, the model's ability to interpret complex geospatial information can be enhanced.

Overall, the integration of vision and language is considered an important future technological advancement, even within the specialized field of remote sensing. Combining these modalities allows for more comprehensive analysis and understanding of data, facilitating advancements in areas like environmental monitoring, disaster management, and resource exploration. As open-source models continue to improve and become more accessible, it is anticipated that researchers will develop increasingly sophisticated tools that leverage both vision and language to tackle complex challenges in remote sensing and beyond.

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023) (cit. on p. 168).

[2] Ashutosh Agarwal and Chetan Arora. “Attention Attention Everywhere: Monocular Depth Prediction with Skip Attention”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 5861–5870 (cit. on pp. 26, 27, 35, 40, 47).

[3] Faisal AlNasser and Dara Entekhabi. “DustSCAN: A five year (2018-2022) hourly dataset of dust plumes from SEVIRI”. In: *Scientific Data* 11.1 (2024), p. 607 (cit. on pp. 139, 140).

[4] Shunichi Amari. “A theory of adaptive pattern classifiers”. In: *IEEE Transactions on Electronic Computers* 3 (1967), pp. 299–307 (cit. on p. 7).

[5] Marco Avvenuti, Marco Bongiovanni, Luca Ciampi, et al. “A Spatio-Temporal Attentive Network for Video-Based Crowd Counting”. In: *Proceedings of the 2022 IEEE Symposium on Computers and Communications*. IEEE. 2022, pp. 1–6 (cit. on pp. 26, 47).

[6] Jinze Bai, Shuai Bai, Yunfei Chu, et al. “Qwen technical report”. In: *arXiv preprint arXiv:2309.16609* (2023) (cit. on p. 140).

[7] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. “Adabins: Depth estimation using adaptive bins”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4009–4018 (cit. on p. 27).

[8] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection”. In: *arXiv preprint arXiv:2004.10934* (2020) (cit. on p. 71).

[9] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. “Soft-NMS–improving object detection with one line of code”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5561–5569 (cit. on p. 73).

[10] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. “The balanced accuracy and its posterior distribution”. In: *Proceedings of the 2010 20th International Conference on Pattern Recognition*. IEEE. 2010, pp. 3121–3124 (cit. on pp. 46, 47).

[11] Tom Cane and James Ferryman. “Saliency-based detection for maritime object tracking”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2016, pp. 18–25 (cit. on p. 91).

[12]Nicolas Carion, Francisco Massa, Gabriel Synnaeve, et al. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer. 2020, pp. 213–229 (cit. on p. 72).

[13]Borja Carrillo-Perez, Angel Bueno Rodriguez, Sarah Barnes, and Maurice Stephan. “Improving yolov8 with scattering transform and attention for maritime awareness”. In: *2023 International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE. 2023, pp. 1–6 (cit. on p. 92).

[14]Antoni B Chan and Nuno Vasconcelos. “Bayesian poisson regression for crowd counting”. In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 545–551 (cit. on p. 100).

[15]Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R Selvaraju, Dhruv Batra, and Devi Parikh. “Counting everyday objects in everyday scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1135–1144 (cit. on p. 101).

[16]Kai Chen, Jiangmiao Pang, Jiaqi Wang, et al. “Hybrid task cascade for instance segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4974–4983 (cit. on p. 121).

[17]Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. “MMDetection: Open mmlab detection toolbox and benchmark”. In: *arXiv preprint arXiv:1906.07155* (2019) (cit. on pp. 91, 167).

[18]Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. “Training deep nets with sublinear memory cost”. In: *arXiv preprint arXiv:1604.06174* (2016) (cit. on p. 49).

[19]Yinpeng Chen, Xiyang Dai, Mengchen Liu, et al. “Dynamic convolution: Attention over convolution kernels”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11030–11039 (cit. on p. 17).

[20]Bowen Cheng, Anwesa Choudhuri, Ishan Misra, et al. “Mask2former for video instance segmentation”. In: *arXiv preprint arXiv:2112.10764* (2021) (cit. on p. 36).

[21]Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. “Masked-attention mask transformer for universal image segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 1290–1299 (cit. on p. 138).

[22]Zhi-Qi Cheng, Qi Dai, Hong Li, et al. “Rethinking spatial invariance of convolutional networks for object counting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19638–19648 (cit. on p. 26).

[23]Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. “Object counting and instance segmentation with image-level supervision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12397–12405 (cit. on pp. 101, 111).

[24] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014) (cit. on p. 9).

[25] MMPreTrain Contributors. *OpenMMLab's Pre-training Toolbox and Benchmark*. `https://github.com/open-mmlab/mmpretrain`. 2023 (cit. on p. 167).

[26] MMSegmentation Contributors. *MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark*. `https://github.com/open-mmlab/mmsegmentation`. 2020 (cit. on p. 168).

[27] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. "Coatnet: Marrying convolution and attention for all data sizes". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 3965–3977 (cit. on pp. 10, 38).

[28] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Vol. 1. Ieee. 2005, pp. 886–893 (cit. on p. 73).

[29] Asanka De Silva, Rajitha Ranasinghe, Arooran Sounthararajah, Hamed Haghighi, and Jayantha Kodikara. "A benchmark dataset for binary segmentation and quantification of dust emissions from unsealed roads". In: *Scientific Data* 10.1 (2023), p. 14 (cit. on pp. 3, 24, 44, 45).

[30] Jia Deng, Wei Dong, Richard Socher, et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255 (cit. on p. 91).

[31] Lee R Dice. "Measures of the amount of ecologic association between species". In: *Ecology* 26.3 (1945), pp. 297–302 (cit. on p. 46).

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. "An image is worth 16x16 words: Transattentions for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020) (cit. on pp. 2, 9, 17, 18, 27).

[33] Dawei Du, Pengfei Zhu, Longyin Wen, et al. "VisDrone-DET2019: The vision meets drone object detection in image challenge results". In: *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 2019 (cit. on pp. 74, 75, 111).

[34] Kaiwen Duan, Song Bai, Lingxi Xie, et al. "Centernet: Keypoint triplets for object detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6569–6578 (cit. on p. 111).

[35] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in Neural Information Processing Systems* 27 (2014) (cit. on p. 27).

[36] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning". In: *Neural Networks* 107 (2018), pp. 3–11 (cit. on pp. 35, 42).

[37]Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. “Density-based spatial clustering of applications with noise”. In: *Int. Conf. knowledge discovery and data mining*. Vol. 240. 6. 1996 (cit. on p. 139).

[38]FairScale authors. *FairScale: A general purpose modular PyTorch library for high performance and large scale training*. `https://github.com/facebookresearch/fairscale`. 2021 (cit. on p. 49).

[39]William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.4. Mar. 2019 (cit. on p. 167).

[40]Christophe Fiorio and Jens Gustedt. “Two linear time union-find strategies for image processing”. In: *Theoretical Computer Science* 154.2 (1996), pp. 165–181 (cit. on p. 121).

[41]Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. “Deep ordinal regression network for monocular depth estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2002–2011 (cit. on p. 27).

[42]gabort@AdobeStock. https://www.stock.adobe.com. 2023 (cit. on p. 23).

[43]Guangshuai Gao, Junyu Gao, Qingjie Liu, Qi Wang, and Yunhong Wang. “Cnn-based density estimation and crowd counting: A survey”. In: *arXiv preprint arXiv:2003.12783* (2020) (cit. on p. 100).

[44]Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448 (cit. on p. 71).

[45]Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587 (cit. on p. 71).

[46]Robert O Green, David R Thompson, EMIT Team, et al. “NASA’s Earth Surface Mineral Dust Source Investigation: An Earth Venture Imaging Spectrometer Science Mission”. In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE. 2021, pp. 119–122 (cit. on p. 2).

[47]Kaiming He, Georgia Gkioxafri, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969 (cit. on pp. 72, 78, 83, 107, 110).

[48]Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 9, 27, 34, 49, 83, 110, 120).

[49]Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9 (2015), pp. 1904–1916 (cit. on p. 27).

[50]Yihui He, Xiangyu Zhang, Marios Savvides, and Kris Kitani. “Softer-nms: Rethinking bounding box regression for accurate object detection”. In: *arXiv preprint arXiv:1809.08545* 2.3 (2018) (cit. on pp. 3, 74).

[51]Ali Al-Hemoud, Mane Al-Sudairawi, Subramanian Neelamanai, Adel Naseeb, and Weam Behbehani. “Socioeconomic effect of dust storms in Kuwait”. In: *Arabian Journal of Geosciences* 10 (2017), pp. 1–9 (cit. on p. 23).

[52]Dan Hendrycks and Kevin Gimpel. “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv:1606.08415* (2016) (cit. on p. 38).

[53]Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780 (cit. on p. 9).

[54]Mohammad Hossain, Mehrdad Hosseinzadeh, Omit Chanda, and Yang Wang. “Crowd counting using scale-aware attention networks”. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1280–1288 (cit. on p. 100).

[55]Andrew G Howard, Menglong Zhu, Bo Chen, et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017) (cit. on pp. 9, 17, 38, 40).

[56]Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141 (cit. on p. 38).

[57]Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708 (cit. on p. 9).

[58]Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the International Conference on Machine Learning*. 2015, pp. 448–456 (cit. on pp. 13, 14, 35, 38, 42).

[59]Alexey Grigorevich Ivakhnenko. “Polynomial theory of complex systems”. In: *IEEE transactions on Systems, Man, and Cybernetics* 4 (1971), pp. 364–378 (cit. on p. 7).

[60]Jitesh Jain, Jiachen Li, Mang Tik Chiu, et al. “Oneformer: One transformer to rule universal image segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2989–2998 (cit. on p. 2).

[61]Paras Jain, Ajay Jain, Aniruddha Nrusimha, et al. “Checkmate: Breaking the memory wall with optimal tensor rematerialization”. In: *Proceedings of Machine Learning and Systems* 2 (2020), pp. 497–511 (cit. on p. 49).

[62]Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. “Acquisition of localization confidence for accurate object detection”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 784–799 (cit. on p. 74).

[63]Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLOv8*. Version 8.0.0. 2023 (cit. on pp. 92, 95, 131).

[64]Abdul Rehman Khan and Asifullah Khan. “MaxViT-UNet: Multi-axis attention for medical image segmentation”. In: *arXiv preprint arXiv:2305.08396* (2023) (cit. on p. 40).

[65] Raihan K Khan and Mark A Strand. “Road dust and its effect on human health: a literature review”. In: *Epidemiology and health* 40 (2018) (cit. on p. 23).

[66] Benjamin Kiefer, Lojze Žust, Matej Kristan, et al. “2nd Workshop on Maritime Computer Vision (MaCVi) 2024: Challenge Results”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 869–891 (cit. on pp. 69, 91, 92).

[67] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. “Panoptic segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9404–9413 (cit. on p. 2).

[68] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. “Segment anything”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4015–4026 (cit. on p. 138).

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90 (cit. on pp. 1, 8, 24, 48).

[70] Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, et al. “Geochat: Grounded large vision-language model for remote sensing”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 27831–27840 (cit. on p. 140).

[71] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. “Fractalnet: Ultra-deep neural networks without residuals”. In: *arXiv preprint arXiv:1605.07648* (2016) (cit. on p. 13).

[72] Y. LeCun, B. Boser, J. S. Denker, et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551 (cit. on p. 7).

[73] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 8).

[74] Jangho Lee, Yingxi Rona Shi, Changjie Cai, et al. “Machine Learning Based Algorithms for Global Dust Aerosol Detection from Satellite Images: Inter-Comparisons and Evaluation”. In: *Remote Sensing* 13.3 (2021) (cit. on pp. 3, 24).

[75] Minhyeok Lee, Sangwon Hwang, Chaewon Park, and Sangyoun Lee. “Edgeconv with attention module for monocular depth estimation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 2858–2867 (cit. on p. 27).

[76] Saadullah Khan Leghari, Mudassir Asrar Zaidi, Muhammad Faheem Siddiqui, et al. “Dust exposure risk from stone crushing to workers and locally grown plant species in Quetta, Pakistan”. In: *Environmental Monitoring and Assessment* 191.12 (Nov. 2019) (cit. on p. 132).

[77] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *ArXiv e-prints* (2016), arXiv–1607 (cit. on p. 14).

[78] Clyde Zhengdao Li, Yiyu Zhao, and Xiaoxiao Xu. “Investigation of dust exposure and control practices in the construction industry: Implications for cleaner production”. In: *Journal of Cleaner Production* 227 (2019), pp. 810–824 (cit. on p. 23).

[79] Feng Li, Hao Zhang, Huaizhe Xu, et al. “Mask dino: Towards a unified transformer-based framework for object detection and segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3041–3050 (cit. on pp. 72, 121, 138).

[80] Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. “Understanding the disharmony between dropout and batch normalization by variance shift”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2682–2690 (cit. on pp. 35, 42).

[81] Yuhong Li, Xiaofan Zhang, and Deming Chen. “Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1091–1100 (cit. on pp. 26, 101).

[82] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2117–2125 (cit. on pp. 12, 31, 35, 42, 72).

[83] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988 (cit. on p. 71).

[84] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755 (cit. on pp. 80, 84, 91, 101, 118, 138).

[85] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. “Decidenet: Counting varying density crowds through attention guided detection and density estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5197–5206 (cit. on pp. 101, 115).

[86] Songtao Liu, Di Huang, and Yunhong Wang. “Adaptive nms: Refining pedestrian detection in a crowd”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6459–6468 (cit. on p. 74).

[87] Wei Liu, Dragomir Anguelov, Dumitru Erhan, et al. “Ssd: Single shot multibox detector”. In: *European conference on computer vision*. Springer. 2016, pp. 21–37 (cit. on p. 71).

[88] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. “Context-aware crowd counting”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5099–5108 (cit. on pp. 26–28, 47, 101, 111).

[89] Ze Liu, Han Hu, Yutong Lin, et al. “Swin transattention attention v2: Scaling up capacity and resolution”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12009–12019 (cit. on pp. 10, 35).

[90]Ze Liu, Yutong Lin, Yue Cao, et al. “Swin transattention attention: Hierarchical vision transattention attention using shifted windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022 (cit. on pp. 10, 18, 19, 27, 28, 49, 91, 120).

[91]Ze Liu, Jia Ning, Yue Cao, et al. “Video swin transattention attention”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3202–3211 (cit. on pp. 10, 36).

[92]Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017) (cit. on pp. 48, 91, 122).

[93]Ao Luo, Fan Yang, Xin Li, et al. “Hybrid graph neural networks for crowd counting”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 11693–11700 (cit. on p. 26).

[94]Yan Ma, Haiping Wu, Lizhe Wang, et al. “Remote sensing big data computing: Challenges and opportunities”. In: *Future Generation Computer Systems* 51 (2015), pp. 47–60 (cit. on p. 1).

[95]Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133 (cit. on p. 7).

[96]Andreas Michel, Wolfgang Gross, Stefan Hinz, and Wolfgang Middelmann. “ARM-NMS: Shape Based Non-Maximum Suppression For Instance Segmentation in Large Scale Imagery”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (2022), pp. 291–298 (cit. on pp. 69, 70).

[97]Andreas Michel, Wolfgang Gross, Fabian Schenkel, and Wolfgang Middelmann. “Class-aware Object Counting”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 469–478 (cit. on p. 97).

[98]Andreas Michel, Jonas Mispelhorn, Fabian Schenkel, Wolfgang Gross, and Wolfgang Middelmann. “An approach to improve detection in scenes with varying object densities in remote sensing”. In: *Image and Signal Processing for Remote Sensing XXVI*. Vol. 11533. SPIE. 2020, pp. 107–113 (cit. on pp. 21, 25, 26, 28, 69, 97).

[99]Andreas Michel, Jonas Mispelhorn, Fabian Schenkel, Wolfgang Gross, and Wolfgang Middelmann. “An approach to improve detection in scenes with varying object densities in remote sensing”. In: *Image and Signal Processing for Remote Sensing XXVI*. Ed. by Lorenzo Bruzzone, Francesca Bovolo, and Emanuele Santi. Vol. 11533. International Society for Optics and Photonics. SPIE, 2020, pp. 107–113 (cit. on p. 104).

[100]Andreas Michel, Martin Weinmann, Jannick Kuester, et al. “DustNet++: Deep Learning-Based Visual Regression for Dust Density Estimation”. In: *International Journal of Computer Vision* (2025), pp. 1–25 (cit. on pp. 21, 26, 38, 129).

[101]Andreas Michel, Martin Weinmann, Fabian Schenkel, et al. “DustNet: Attention to Dust”. In: *DAGM German Conference on Pattern Recognition*. Springer. 2023, pp. 211–226 (cit. on pp. 21, 25, 33).

[102]Andreas Michel, Martin Weinmann, Fabian Schenkel, et al. “DustNet: Attention to Dust”. In: Mar. 2024, pp. 211–226 (cit. on pp. 38, 40, 41, 48, 49).

[103]Andreas Michel, Martin Weinmann, Fabian Schenkel, et al. “TERRESTRIAL VISUAL DUST DENSITY ESTIMATION BASED ON DEEP LEARNING”. In: *Proceedings of the 2023 IEEE International Geoscience and Remote Sensing Symposium*. 2023 (cit. on pp. 25, 44, 48, 49).

[104]Andreas Michel, Martin Weinmann, Fabian Schenkel, et al. “Terrestrial visual dust density estimation based on deep learning”. In: *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2023, pp. 4923–4926 (cit. on pp. 21, 31).

[105]Marvin Minsky. *Society of mind*. Simon and Schuster, 1988 (cit. on p. 100).

[106]Suzette A Morman and Geoffrey S Plumlee. “Dust and human health”. In: *Mineral dust: A key player in the Earth system* (2014), pp. 385–409 (cit. on p. 132).

[107]Suzette A. Morman and Geoffrey S. Plumlee. “Dust and Human Health”. In: *Mineral Dust*. Springer Netherlands, 2014, pp. 385–409 (cit. on p. 132).

[108]NASA. *Mars Curiosity Image Gallery*. `https://www.nasa.gov/mission_pages/msl/images/index.html`, Last accessed on 2023-03-08. 2023 (cit. on p. 23).

[109]Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. “DINOv2: Learning Robust Visual Features without Supervision”. In: *Transactions on Machine Learning Research* () (cit. on p. 138).

[110]Jaeyoon Park, Jungsam Lee, Katherine Seto, et al. “Illuminating dark fishing fleets in North Korea”. In: *Science advances* 6.30 (2020), eabb1197 (cit. on p. 1).

[111]Adam Paszke, Sam Gross, Francisco Massa, et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019) (cit. on p. 167).

[112]Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. “P3depth: Monocular depth estimation with a piecewise planarity prior”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1610–1621 (cit. on p. 27).

[113]Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. “Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution”. In: *arXiv preprint arXiv:2006.02334* (2020) (cit. on p. 71).

[114]Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. “Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10213–10224 (cit. on pp. 78, 111).

[115]Alec Radford, Jong Wook Kim, Chris Hallacy, et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763 (cit. on p. 140).

[116]René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. "Vision transattentions for dense prediction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12179–12188 (cit. on p. 27).

[117]Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788 (cit. on pp. 71, 77).

[118]Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. "Balanced mse for imbalanced visual regression". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7926–7935 (cit. on p. 47).

[119]Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016), pp. 1137–1149 (cit. on pp. 71, 72).

[120]Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99 (cit. on pp. 1, 24, 77).

[121]Tianhe Ren, Shilong Liu, Feng Li, et al. *detrex: Benchmarking Detection Transformers*. 2023. arXiv: `2306.07265 [cs.CV]` (cit. on p. 168).

[122]Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241 (cit. on pp. 1, 2).

[123]Frank Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961 (cit. on p. 7).

[124]Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386 (cit. on p. 7).

[125]Azriel Rosenfeld and Mark Thurston. "Edge and curve detection for visual scene analysis". In: *IEEE Transactions on computers* 100.5 (1971), pp. 562–569 (cit. on p. 73).

[126]David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536 (cit. on pp. 7, 11).

[127]Olga Russakovsky, Jia Deng, Hao Su, et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252 (cit. on pp. 8, 83, 110).

[128]Syed AM Said, Ghassan Hassan, Husam M Walwil, and N Al-Aqeeli. "The effect of environmental factors and dust accumulation on photovoltaic modules and dust-accumulation mitigation strategies". In: *Renewable and Sustainable Energy Reviews* 82 (2018), pp. 743–760 (cit. on p. 2).

[129]Usman Sajid, Hasan Sajid, Hongcheng Wang, and Guanghui Wang. “Zoomcount: A zooming mechanism for crowd counting in static images”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.10 (2020), pp. 3499–3512 (cit. on p. 100).

[130]Tim Salimans and Durk P Kingma. “Weight normalization: A simple reparameterization to accelerate training of deep neural networks”. In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 15).

[131]Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. “Switching convolutional neural network for crowd counting”. In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2017, pp. 4031–4039 (cit. on pp. 26, 100, 101).

[132]Shuai Shao, Zeming Li, Tianyuan Zhang, et al. “Objects365: A large-scale, high-quality dataset for object detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 8430–8439 (cit. on p. 91).

[133]Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. “Self-attention with relative position representations”. In: *arXiv preprint arXiv:1803.02155* (2018) (cit. on pp. 18, 38).

[134]Wenzhe Shi, Jose Caballero, Ferenc Huszár, et al. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1874–1883 (cit. on p. 40).

[135]Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48 (cit. on p. 13).

[136]Roy W Simonson. “Airborne dust and its significance to soils”. In: *Geoderma* 65.1-2 (1995), pp. 1–43 (cit. on p. 23).

[137]Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on p. 8).

[138]Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. “Weighted boxes fusion: Ensembling boxes from different object detection models”. In: *Image and Vision Computing* (2021), pp. 1–6 (cit. on p. 74).

[139]Minsoo Song, Seokjae Lim, and Wonjun Kim. “Monocular depth estimation using laplacian pyramid-based depth residuals”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.11 (2021), pp. 4381–4393 (cit. on p. 27).

[140]Thorvald Sorensen. “A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons”. In: *Biologiske skrifter* 5 (1948), pp. 1–34 (cit. on p. 46).

[141]Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958 (cit. on pp. 13, 35, 42).

[142]Jianlin Su, Murtadha Ahmed, Yu Lu, et al. "Roformer: Enhanced transformer with rotary position embedding". In: *Neurocomputing* 568 (2024), p. 127063 (cit. on p. 18).

[143]Christian Szegedy, Wei Liu, Yangqing Jia, et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. on pp. 8, 12).

[144]Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114 (cit. on p. 9).

[145]Yi Tay, Mostafa Dehghani, Samira Abnar, et al. "Scaling Laws vs Model Architectures: How does Inductive Bias Influence Scaling?" In: *arXiv preprint arXiv:2207.10551* (2022) (cit. on p. 9).

[146]Daniel Q Tong, Thomas E Gill, William A Sprigg, et al. "Health and safety effects of airborne soil dust in the Americas and beyond". In: *Reviews of Geophysics* 61.2 (2023), e2021RG000763 (cit. on pp. 2, 23).

[147]Hugo Touvron, Matthieu Cord, Matthijs Douze, et al. "Training data-efficient image transformers & distillation through attention". In: *International conference on machine learning*. PMLR. 2021, pp. 10347–10357 (cit. on p. 10).

[148]Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023) (cit. on p. 140).

[149]Zhengzhong Tu, Hossein Talebi, Han Zhang, et al. "Maxvit: Multi-axis vision transformer". In: *European conference on computer vision*. Springer. 2022, pp. 459–479 (cit. on pp. 10, 18, 19, 38, 39).

[150]U.S. Environmental Protection Agency. *Particulate Matter (PM) 2022 Report*. Accessed: 2024-05-22. 2022 (cit. on p. 133).

[151]D Ulyanov. "Instance normalization: The missing ingredient for fast stylization". In: *arXiv preprint arXiv:1607.08022* (2016) (cit. on p. 15).

[152]Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on pp. 1, 9, 24).

[153]Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 7464–7475 (cit. on p. 69).

[154]Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. "Region proposal by guided anchoring". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2965–2974 (cit. on p. 72).

[155] Lijun Wang, Jianming Zhang, Yifan Wang, Huchuan Lu, and Xiang Ruan. "Cliffnet for monocular depth estimation with hierarchical embedding loss". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*. Springer. 2020, pp. 316–331 (cit. on p. 27).

[156] Mingpu Wang, Gang Yao, Yang Yang, et al. "Deep learning-based object detection for visible dust and prevention measures on construction sites". In: *Developments in the Built Environment* 16 (2023), p. 100245 (cit. on p. 3).

[157] Wenhai Wang, Enze Xie, Xiang Li, et al. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 568–578 (cit. on p. 10).

[158] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. "Solo: Segmenting objects by locations". In: *European Conference on Computer Vision*. Springer. 2020, pp. 649–665 (cit. on pp. 74, 78, 94).

[159] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. "Solov2: Dynamic and fast instance segmentation". In: *Advances in Neural information processing systems* 33 (2020), pp. 17721–17732 (cit. on p. 74).

[160] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, et al. "iSAID: A Large-scale Dataset for Instance Segmentation in Aerial Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 28–37 (cit. on pp. 74, 75, 101, 111, 116).

[161] Ross Wightman. *PyTorch Image Models*. Version 1.0.11 (cit. on p. 167).

[162] T Wolf. "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771* (2019) (cit. on p. 168).

[163] Haiping Wu, Bin Xiao, Noel Codella, et al. "Cvt: Introducing convolutions to vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 22–31 (cit. on p. 10).

[164] Yuxin Wu and Kaiming He. "Group normalization". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19 (cit. on p. 15).

[165] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. *Detectron2*. `https://github.com/facebookresearch/detectron2`. 2019 (cit. on p. 168).

[166] Gui-Song Xia, Xiang Bai, Jian Ding, et al. "DOTA: A large-scale dataset for object detection in aerial images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3974–3983 (cit. on p. 75).

[167] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. "Unified perceptual parsing for scene understanding". In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 418–434 (cit. on p. 27).

[168] Wei Xu, Dingkang Liang, Yixiao Zheng, and Zhanyu Ma. "Dilated-Scale-Aware Attention ConvNet For Multi-Class Object Counting". In: *arXiv preprint arXiv:2012.08149* (2020) (cit. on p. 101).

[169]Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. "Metaanchor: Learning to detect objects with customized anchors". In: *Advances in Neural Information Processing Systems*. 2018, pp. 320–330 (cit. on p. 72).

[170]Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. "On early stopping in gradient descent learning". In: *Constructive Approximation* 26.2 (2007), pp. 289–315 (cit. on p. 13).

[171]Yang You, Jing Li, Sashank Reddi, et al. "Large batch optimization for deep learning: Training bert in 76 minutes". In: *arXiv preprint arXiv:1904.00962* (2019) (cit. on pp. 83, 110).

[172]Yan Yu and Paul Ginoux. "Enhanced dust emission following large wildfires due to vegetation disturbance". In: *Nature Geoscience* 15.11 (2022), pp. 878–884 (cit. on p. 23).

[173]Feiniu Yuan, Lin Zhang, Xue Xia, Qinghua Huang, and Xuelong Li. "A Wave-Shaped Deep Neural Network for Smoke Density Estimation". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 2301–2313 (cit. on pp. 3, 24).

[174]Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. "Neural window fully-connected crfs for monocular depth estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3916–3925 (cit. on pp. 27, 35, 47).

[175]Julio Zamora Esquivel, Adan Cruz Vargas, Paulo Lopez Meyer, and Omesh Tickoo. "Adaptive convolutional kernels". In: *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 2019 (cit. on pp. 17, 35).

[176]Hao Zhang, Feng Li, Shilong Liu, et al. "Dino: Detr with improved denoising anchor boxes for end-to-end object detection". In: *arXiv preprint arXiv:2203.03605* (2022) (cit. on pp. 72, 91).

[177]Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. "Single-image crowd counting via multi-column convolutional neural network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 589–597 (cit. on pp. 1, 26, 101, 110, 111, 115).

[178]Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. "Pyramid scene parsing network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2881–2890 (cit. on pp. 33, 35).

[179]Zhaohui Zheng, Ping Wang, Dongwei Ren, et al. "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation". In: (2021) (cit. on p. 74).

[180]Xiao Xiang Zhu, Devis Tuia, Lichao Mou, et al. "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources". In: *IEEE Geoscience and Remote Sensing Magazine* 5.4 (2017), pp. 8–36 (cit. on pp. 1, 24).

[181]Xizhou Zhu, Weijie Su, Lewei Lu, et al. "Deformable detr: Deformable transformers for end-to-end object detection". In: *arXiv preprint arXiv:2010.04159* (2020) (cit. on pp. 18, 19, 72).

[182] Zhuofan Zong, Guanglu Song, and Yu Liu. “Detrs with collaborative hybrid assignments training”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 6748–6758 (cit. on pp. 72, 90, 91, 95, 131, 138).

[183] Lojze Žust, Janez Perš, and Matej Kristan. “Lars: A diverse panoptic maritime obstacle detection dataset and benchmark”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 20304–20314 (cit. on pp. 74, 75, 91).

List of Figures

List of Tables

List of Abbrevations

AFPN	Attention Feature Pyramid Network. 33–36, 39–42, 53, 55
ARM-NMS	Area Rescoring Mask Non-Maximum Suppresion. 70, 76, 79, 88, 93, 95, 130, 136
Acc	Accuracy. 46, 123
BN	Batch Normalization. 14, 15, 38
CEN	Count Estimation Network. 102, 103, 105, 106, 108–114, 126, 132
CNN	Convolutional Neural Network. 4, 7, 9–11, 14–17, 19, 30, 34, 35, 71, 72
COCO	Common Objects in Context. 76, 91, 95, 101, 118, 136
CPW	Checkpoint Wrapper. 49, 130
CV	Computer Vision. 1, 7–10
CmaxFPN	Cross multi-axis Feature Pyramid Network. 38–43
DETR	Detection Transformer. 19, 71, 72, 127, 131
DOTA	Object deTection in Aerial images. 75
DRPN	Density Region Proposal Network. 114–118, 127, 128, 131, 137
ERF	Effective Receptive Field. 8, 10, 16, 17, 19, 30
FPN	Feature Pyramid Network. 12, 25, 28–31, 35, 72, 101, 102, 129
GN	Group Normalization. 15
GPU	Graphic Processor Unit. 8
GRU	Gated Recurrent Units. 9
GeLU	Gaussian Error Linear Unit. 38
HB	High Dust Density Bin. 47, 62–64, 123
ILSVRC	ImageNet Large Scale Visual Recognition Challenge. 8
IN	Instance Normalization. 15
IoU	Intersection of Union. 46, 51, 65, 73, 78, 80, 82, 84, 86, 89, 90, 123

LB Low Dust Density Bin. 47, 62–64, 123
LLM Large Language Model. 140
LN Layer Normalization. 14, 15, 19
LSTM Long Short-Term Memory. 9
LaRS Lakes, Rivers, and Seas. 74, 75, 91–93
LiDAR Light Detection and Ranging. 2, 24
MAE Mean Absolute Error. 46, 47, 54, 60, 65, 111, 112, 122, 123
MB Medium Dust Density Bin. 47, 62–64, 123
MCA Multiheaded Cross-Attention. 9, 55, 56
MCDEN Multi-Class Density Estimation Network. 102–104, 112
MDE Monocular Depth Estimation. 25, 26, 47
MHSA Multi-Headed Self Attention. 19
MLP Multilayer Perceptron. 7, 19, 140
MODS Maritime Obstacle Detection Score. 91
MSA Multiheaded Self-Attention. 9, 55, 56
MSE Mean Squared Error. 46, 47, 54, 60, 64–66, 122, 123
NLP Natural Language Processing. 2, 9
NMS Non-Maximum Suppresion. 3, 72–74, 76–79, 81, 84–86, 89, 90, 94, 105, 110, 130, 131
PPM Pyramid Pooling Module. 33–36, 42
Pre Precision. 46, 123
RMSE Root Mean Squared Error. 111
RNN Recurrent Neural Network. 14
RPN Region Proposal Network. 72, 102, 103, 105, 115, 116, 118, 127, 128, 131, 137
ReLU REctified Linear Unit. 8, 30, 31, 33, 105, 106
Rec Recall. 46, 123
ResNet Residual Networks. 9
RoPE Rotary Positional Embeddings. 18
SiLU Sigmoid Linear Unit. 35, 42
TCM Transposed Convolution Module. 35
TTA Test Time Augmentation. 91, 92
URDE Unsealed Roads Dust Emissions. 24, 44–46, 66, 130
VGG Visual Geometry Group. 8, 28
VLM Vision Language Model. 140, 141

VQA	Visual Question Answering. 141
ViT	Vision Transformer. 2, 7, 9, 10, 17, 18
VisDrone	Vision Meets Drone Object Detection. 74, 75, 108, 110–112
WN	Weight Normalization. 15
ZB	Zero Dust Density Bin. 47, 62–64, 123
iSAID	Instance Segmentation in Aerial Images Dataset. 74–76, 79, 82–84, 86, 87, 89, 95, 101, 110–114, 136, 159
mAP	mean Average Precision. 84, 86, 89, 90, 95, 118, 127, 131, 136, 137
mAR	mean Average Recall. 84, 86, 89, 118, 127, 128, 131, 137

Declaration of Supportive Ressources

During the development and implementation of the algorithms presented in this work, as well as while composing this document, we utilized additional resources that were instrumental in advancing and refining our research. These resources provided valuable support in both the technical execution of our algorithms and the enhancement of the manuscript. The following sections offer a detailed account of these resources, explaining how each contributed to the successful completion and improvement of this project's technical and writing aspects.

List of Major Machine Learning Frameworks

1. **PyTorch** [111]: An open-source deep learning library developed by Facebook's AI Research lab. PyTorch offers dynamic computation graphs and efficient memory management, making it highly flexible for building and training neural networks.

2. **PyTorch-Lightning** [39]: A lightweight wrapper for PyTorch that simplifies the research process by organizing code and reducing boilerplate. It streamlines tasks like training loops and validation, allowing researchers to focus on developing models.

3. **TIMM** [161]: The PyTorch Image Models (TIMM) library provides a collection of image models and utilities. It includes a wide array of pre-trained models and building blocks for computer vision research and applications.

4. **MMPretrain** [25]: An open-source pre-training toolbox for computer vision tasks developed by OpenMMLab. MMPretrain offers implementations of various pre-training algorithms to improve model performance on downstream tasks.

5. **MMDetection** [17]: A comprehensive object detection toolbox based on PyTorch. MMDetection provides a modular framework with many detection algorithms, facilitating research and development in object detection.

6. **MMSegmentation** [26]: An open-source semantic segmentation toolbox by OpenMMLab. MMSegmentation offers a unified framework for training and evaluating segmentation models on various datasets.

7. **Detectron2** [165]: A next-generation library for object detection and segmentation developed by Facebook AI Research. Detectron2 is built on PyTorch and provides a flexible and extensible framework for computer vision tasks.

8. **Detrex** [121]: An open-source project focusing on Transformer-based object detection algorithms. Detrex aims to advance object detection by leveraging Transformer architectures for improved accuracy.

9. **Transformers** [162]: A library by Hugging Face that provides state-of-the-art pre-trained models for natural language processing. While primarily NLP-focused, it also supports models for computer vision and multi-modal tasks.

List of Tools Supporting the Writing Process:

1. **Grammarly**: This tool offered grammar and spelling checks, along with stylistic suggestions, to enhance the readability and clarity of the text.

2. **FhGenie, based on GPT [1]**: Large Language Models were used to provide grammar and spelling suggestions, as well as stylistic improvements. Only self-authored texts were input into these tools, and their suggestions were evaluated and selectively incorporated to augment the readability and comprehension of the manuscript.

Colophon

This thesis was typeset with LaTeX 2_ε. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.