OmniNet: A Hybrid Deep Learning Framework for Robust Semantic Segmentation

by

Fahim Ahmed Ifty
23266007

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science

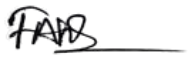
Department of Computer Science and Engineering
Brac University
October 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Fahim Ahmed Ifty
23266007

Approval

The thesis titled "OmniNet: A Hybrid Deep Learning Framework for Robust Semantic Segmentation"

Submitted by:
Fahim Ahmed Ifty (23266007)

Of Summer, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science and Engineering on October 19, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Examiner:
(External)

Prof. Dr. Kazi Md. Rokibul Alam, Ph.D.
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology (KUET)

Examiner:
(Internal)

Dr. Md. Ashraful Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md Sadek Ferdous
Professor
Department of Computer Science and Engineering
Brac University

Chairperson:
(Chair)

Sadia Hamid Kazi, Ph.D.
Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Semantic segmentation empowers computers to interpret visual scenes in a structured and meaningful manner, accurately delineating the precise boundaries of every object in an image at the pixel level. However, existing CNN-based models struggle with long-range context, transformer-based approaches are computationally heavy and often miss local detail, and many hybrid designs neglect explicit multi-scale features or refined attention, while also suffering from class imbalance and noisy annotations. This research proposes an encoder–decoder architecture that combines a ResNet152 backbone, Atrous Spatial Pyramid Pooling (ASPP), and a Transformer Refine Block, leveraging the strengths of convolutional features and lightweight self-attention to capture both multi-scale semantics and long-range dependencies. In the decoder, a UNet-style upsampling path with skip connections and CBAM attention preserves spatial detail and refines boundaries, yielding coherent pixel-level predictions across scales. The training pipeline applies label cleaning and robust augmentation (MixUp, CutMix) and optimizes a composite loss (Lovasz–Softmax, Dice Loss, Boundary Loss) to counter label noise and the long-tail distribution. Evaluated on the CamVid benchmark, our method achieves 86.53% mean IoU and 95.99% pixel accuracy, outperforming recent efficient residual attention networks while prioritizing segmentation quality over inference speed. These results demonstrate that combining multi-scale convolutional features with lightweight self-attention and boundary-aware optimization delivers state-of-the-art accuracy under noisy labels and severe class imbalance in urban scene segmentation.

Keywords: Semantic Segmentation, CamVid, Deep Learning, OmniNet, Transformer Refine Block, ResNet152, UNet, Data Augmentation, Class Imbalance.

Dedication

Dedicated to my beloved parents, whose endless support, encouragement, and love have been the foundation of my journey, and also to all those who never stopped believing in me.

Acknowledgement

I would like to begin by expressing my deepest gratitude to the Almighty, who granted me the strength and patience to complete this thesis.

I wish to express my heartfelt gratitude to my supervisor, Professor Dr. Md. Golam Rabiul Alam, because without his help in terms of guidance, useful information, and inspiration during the process of this work, I could not have accomplished it. His guidance and advice have greatly contributed towards the completion of this thesis.

I also appreciate faculty members and educators at the Department of Computer Science and Engineering, BRAC University, whose learning and guidance assisted me in developing my knowledge base and shaping my academic life.

I would also like to thank my parents not only because of their love but also because it is with them that I had a belief in myself; not only have they prayed to God but have also supported me and had faith in their little boy. These are the greatest sources of my motivation.

Table of Contents

List of Tables

List of Figures

Nomenclature

AdamW Adaptive Moment Estimation with Weight Decay

ASPP Atrous Spatial Pyramid Pooling

CBAM Convolutional Block Attention Module

CNN Convolutional Neural Network

FCN Fully Convolutional Network

FLOPs Floating Point Operations per Second

GPU Graphics Processing Unit

IoU Intersection over Union

mIoU Mean Intersection over Union

ResNet Residual Network

UNet U-shaped Network

Chapter 1

Introduction

1.1 Background

Semantic segmentation is a fundamental task in computer vision concerned with assigning a semantic label to each pixel in an image. In autonomous driving, semantic segmentation supports scene understanding by delineating roads, vehicles, pedestrians, and other traffic elements. The CamVid dataset comprises urban driving scenes with dense, pixel-level annotations across 11 classes. Early convolutional architectures (e.g., SegNet) demonstrate strong local feature extraction but exhibit limited long-range contextual modeling and reduced performance on minority classes. Transformer-based models have recently emerged as competitive alternatives for global context aggregation. For example, Swin Transformer leverages hierarchical self-attention to capture global–local dependencies and enhance feature representations. Despite their accuracy, these models introduce integration complexity with CNN backbones and impose substantial computational overhead. CamVid remains an early high-resolution benchmark that is still widely used for evaluating urban scene segmentation. Its small scale and pronounced class imbalance pose challenges for generalization, making it suitable for stress-testing models under data scarcity and skewed label distributions.

This work proposes OmniNet to address these challenges. OmniNet combines multi-scale feature extraction, contextual aggregation, and boundary refinement to improve quantitative performance and produce cleaner, more coherent segmentation maps. The resulting improvements are pertinent to safety-critical perception in autonomous driving.

Figure 1.1 provides a visual example of the improvements offered by OmniNet. Competing baselines such as DDRNet and SegNet fail to consistently recognize small, distant, or thin objects, often misclassifying traffic signs or ignoring poles and pedestrians. OmniNet, by contrast, produces sharper boundaries and more reliable predictions on rare categories, highlighting its effectiveness in addressing the shortcomings of prior methods.

Figure 1.1: Qualitative comparison of semantic segmentation results: (a) Original image, (b) Ground truth segmentation, (c) DDRNet prediction — fails to predict sign symbols (black or void regions) and struggles with long-range object detection, (d) SegNet prediction — poor pole segmentation, incorrect sign symbol prediction, and weak detection of cars and pedestrians, and (e) Our model prediction — shows significantly improved segmentation quality, accurately capturing sign symbols, poles, pedestrians, and cars, achieving visual results closely matching the ground truth (approximately 90% similarity).

1.2 Objectives

The objective of this thesis is to design and evaluate a hybrid encoder–decoder architecture that advances semantic segmentation under challenging conditions such as noisy labels and severe class imbalance. The proposed approach integrates convolutional feature extraction, multi-scale context aggregation, and lightweight self-attention to achieve precise pixel-level predictions, while enhancing boundary delineation and rare-class performance through attention mechanisms, robust augmentation, and boundary-aware optimization. The broader goal is to demonstrate that combining multi-scale convolutional features with lightweight attention and specialized training strategies can deliver more reliable and coherent segmentation in safety-critical computer vision applications.

1.3 Scope and Delimitations

This thesis focuses on semantic segmentation with CamVid remapped to 11 semantic classes; ‘void’ pixels are excluded during preprocessing to align with common benchmarking practice on CamVid. All experiments and reported metrics use this dataset and class set; other datasets (e.g., Cityscapes, BDD100K) are out of scope. Images predominantly depict daytime, fair-weather urban driving; adverse conditions (e.g., night, rain, fog) are not considered. Robustness is improved via data augmentation (MixUp, CutMix, elastic deformations, additive noise); real-time deployment is not a stated objective.
Methodologically, the study evaluates a single hybrid architecture combining convolu-

tional layers with lightweight self-attention (ResNet152, CBAM, ASPP, and a transformer refinement block within a UNet-style decoder). Alternative model families (e.g., GANs, diffusion, foundation models) are not explored. Performance is evaluated using standard metrics (pixel accuracy, mean IoU, class-wise IoU, inference time); interpretability and uncertainty estimation are deferred to future work.

1.4 Problem Statement and Research Questions

This work addresses semantic segmentation on the CamVid dataset. The task is challenging due to label noise, limited dataset size, and severe class imbalance. For instance, roads and buildings dominate pixel counts, whereas pedestrians and poles are underrepresented. This imbalance biases training toward frequent classes and degrades performance on rare categories. Accordingly, the study is guided by the following research questions:

RQ1. Does a hybrid CNN with lightweight attention improve mean IoU while preserving performance on rare classes?

RQ2. Which loss and augmentation strategies best preserve thin structures and object boundaries?

RQ3. What accuracy–latency trade-offs are observed relative to strong baselines (e.g., SegNet, PIDNet)?

1.5 Contributions

Our main contributions are summarized as follows:

- A hybrid encoder–decoder architecture is developed that integrates ResNet152 with ASPP, CBAM, and a lightweight Transformer Refine Block, enabling robust multi-scale feature extraction and long-range context reasoning to improve segmentation accuracy, especially for small or dispersed objects.
- A UNet-style decoder with skip connections, CBAM, SE attention, and deep supervision is designed to precisely reconstruct spatial information while maintaining fine details and improving boundary delineation.
- Preprocessing strategies, including median filtering, morphological operations, and ground truth remapping, are applied to clean noisy annotations and produce consistent masks, enhancing model reliability and learning of meaningful structures.

- A comprehensive data augmentation pipeline—including Cropping, Flipping, MixUp, CutMix, elastic/grid distortions, brightness/contrast adjustment, multi-scale crops are implemented to improve model generalization and robustness to spatial, appearance, and occlusion variations.
- Class imbalance is addressed via weighted loss and rare-class oversampling, ensuring that minority classes such as pedestrians and bicycles are accurately represented during training, improving safety-critical object segmentation.
- Training strategies including mixed-precision training, AdamW optimizer, cosine annealing learning rate, auxiliary supervision heads, and gradient clipping are adopted to stabilize learning, accelerate convergence, and maximize the efficacy of the model under challenging conditions.

1.6 Thesis Structure

Chapter 2 reviews related work on CNN, transformer, and hybrid-based semantic segmentation and methods to address class imbalance. Chapter 3 presents the methodological framework, including preprocessing for noisy labels, the OmniNet architecture (ResNet-152, transformer refinement, UNet, CBAM, ASPP), training procedures, and evaluation metrics. Chapter 4 reports experiments (pixel accuracy, mean IoU), compares against SegNet, PIDNet-S, and SERNet-Former, and includes class-wise IoU, ablations, robustness tests, and qualitative analyses. Chapter 5 interprets the results in the context of autonomous driving and the stated objectives, discusses strengths and limitations (efficiency, specificity of the dataset) and outlines future work for real-time applicability and generalization. The thesis concludes by summarizing the contributions and their significance to the field of semantic segmentation.

Chapter 2

Literature Review

2.1 Introduction

Semantic segmentation is the task of assigning each pixel in an image a category [1]. Practically, this yields a dense map of objects and surfaces. For autonomous driving, this capability is essential [24]. A vehicle has to separate a road from a sidewalk, a person from a car, a building from a traffic light. Mistakes in these cases can lead to serious consequences. One of the standard datasets in this field is CamVid, which provides urban driving scenes with dense pixel annotations across 11 classes [22]. Segmentation models have evolved substantially. CNN-based architectures such as U-Net and DeepLab achieved early progress through strong local feature modeling [4, 15]. But CNNs are limited. Recent work has introduced transformers and hybrid approaches [6, 8]. These are better at capturing long-range dependencies and global relationships [6]. Another recurring challenge is class imbalance [23]. Roads and buildings dominate in CamVid, while pedestrians or cyclists appear far less often. Models then tend to ignore the smaller classes. Researchers usually handle this with weighted losses or oversampling strategies [23]. Data augmentation is another tool. Standard flips and rotations help, but newer methods like MixUp and CutMix go further, adding diversity and lowering the risk of overfitting [16, 17]. The choice of loss function also matters. Cross-entropy is widely used, but Dice loss or focal loss can be more effective depending on the problem [18, 23]. This review frames the background for the OmniNet approach. By tracing how models have evolved, by examining imbalance and augmentation, and by reflecting on the role of different losses, it sets the ground for a hybrid framework designed to work with CamVid.

2.2 CNN-Based Segmentation Models

CNN-based architectures laid the foundation for semantic segmentation by leveraging hierarchical feature extraction and spatial locality. Early Fully Convolutional Networks

(FCNs) enabled pixel-wise prediction using only convolutional layers [1]. The encoder-decoder design is popular and SegNet, for instance, increased accuracy by reusing max-pooling indices to keep spatial information intact [2] and worked on image datasets such as CamVid [22]. UNet, which was initially constructed to be used in medical imaging, implemented a skip connection that assists in the concatenation of low and high features to produce more accurate predictions [4]. However, UNet variants (e.g., ResNet-50-based) may underperform in complex urban scenes due to limited global context modeling. This has led to researchers investigating more advanced and greater structures. Recently proposed methods tend to incorporate attention and transformer-based modules to more effectively incorporate the aspects of the whole context to enhance the segmentation outputs in difficult settings.

2.3 Transformer-Based Segmentation Models

Transformer-based architectures treat images as sequences of patches, enabling global context modeling that overcomes CNNs' locality constraints [6]. The Vision Transformer (ViT) was one of the earliest approaches to treat images as sequences of patches [6]-analogous to how words are treated in a sentence. Although mighty, ViT may be very heavy and slow regarding pixel-level operations such as segmentation. The solution came in the form of new models such as the Swin Transformer that applied a more efficient architecture based on layers and local focus on attention windows, thus adapting to image segmentation better [8]. Newer models such as SERNet-Former take one step further and augment transformers with CNNs achieving good results on datasets such as CamVid. Nonetheless, pure transformer models have tended to lack the local micro-details that CNNs excel at, and thus it is perhaps of little surprise that in recent years, there is a great push towards trying to merge the two. Hybrid architectures tend to be designed that achieve a balance between global contextual awareness and more localized spatial specificity. These models are intended to combine the advantages of the competing CNNs and transformers on improving the robustness of a semantic segmentation task.

2.4 Hybrid Architectures

Hybrid segmentation models integrate convolutional feature extractors with attention modules to balance precise local detail and long-range dependency capture. HRNet preserves high-resolution features throughout the network, aiding multi-scale detail capture [9]. DDRNet-23-slim aims at achieving lightweight and speed [10], and thus is applicable to real-time applications, besides performing well on the CamVid dataset. PIDNet-S employs a multi-branch CNN to improve performance [11]. The use of custom CNN designs

also registers decent results in other models such as SPCONet and ECMNet [12, 13]. Most of them have lacked more detailed attention mechanisms like CBAM or multi-scale features like ASPP which can really make a difference [14, 15]. That is where OmniNet comes in by integrating the stronger backbone of ResNet152, a Transformer Refine Block (lightweight self-attention) to refine ASPP features and capture long-range dependencies, and UNet-like skip connections to retain small details. They, combined, enable OmniNet to outperform most of the available models in accuracy and flexibility.

2.5 Class Imbalance Techniques

Class imbalance in semantic segmentation occurs when dominant categories overwhelm under-represented classes, necessitating loss weighting and sampling strategies [23, 27, 26]. In CamVid, roads and sky dominate, whereas pedestrians and cyclists are under-represented. To overcome that, weighted loss functions were employed in some models that give more weight to the rare classes [23, 26, 30]. An instance is that SegNet adopts this strategy to enhance the performance on the underrepresented classes. Oversampling involves displaying pictures containing rare classes more than normal in the course of training as yet another beneficial technique [23, 30]. Such a method has been adopted in models including the PIDNet-S whose performance was high due to better balancing of the classes [11]. OmniNet uses the combination of both strategies: it learns to assign weights proportional to the frequency of each class occurrence (with a higher interest to rare classes) and scales up rare-class images during the training. The combination of these techniques enables the model to learn and predict small or uncommon objects better than other models would.

2.6 Data Augmentation Methods

Data augmentation generates varied training examples—through geometric, photometric, and synthetic mixes—to improve model generalization and robustness. Cropping, horizontal flipping, rotating, scaling, and color normalization are common in CNN-based segmentation (e.g., U-Net). The augmentations aid the model to be invariant to spatial transformations and light condition which is very necessary in applications such as medical imaging, autonomous driving and understanding of urban scenes. Higher strategies of augmentation promote generalization further. MixUp generates synthetic training samples by interpolating pairs of images and their corresponding labels [16], which encourages locally linear behavior between classes and reduces overconfidence. CutMix instead cuts out a random piece of an image with a piece of a different image and proportional combination of the labels [17], thereby changing the model to target not only one region

of interest and alleviating overfitting. In addition, the use of multi-scale inputs or random photometric variations (alterations in brightness, contrast, saturation) usually enhances adaptability to a wide range of image conditions. Such combination of approaches (e.g., MixUp, CutMix, per-image flipping, multi-scale inputs) appear to improve generalization over a variety of lighting conditions, viewing perspectives, and scene complexities, as demonstrated using models such as OmniNet, conceptualised on large-scale datasets like CamVid. With these augmentation strategies applied during training, contemporary segmentation models not only are more robust, have minimized overfitting, and retain strong performance in the real world.

2.7 Loss Functions for Segmentation

Segmentation loss functions must align training objectives with evaluation metrics, balancing region overlap, class sensitivity, and boundary precision. Loss functions like cross-entropy, as applied in SegNet, tend to work well with balanced datasets but are harmed by classes that are disproportionately under-represented [23]. This is especially the case in CamVid, where pedestrians and bicyclists occupy only a small fraction of pixels. To counteract this, OmniNet uses the following composite loss formulation:

- **Dice Loss:** Improves the performance of the small target by directly optimizing small target detection by increasing the Dice coefficient between the predicted mask and the ground truth [18]. This counteracts the dominance of gross features (e.g., roads, buildings) and ensures that small objects play a meaningful role in the optimisation.

- **Lovasz–Softmax Loss:** Provides an unconstrained convex and tractable bound to maximize IoU. As a result, it better matches the training objective to the evaluation metric (mIoU), compared to pixel-wise cross-entropy. It enables end-to-end validation that performance optimizations lead to meaningful improvements for benchmark scores [19].

- **Boundary Loss:** Enhances performance on border boundaries. Urban scenes often contain thin features such as poles, fences, and road boundaries. Boundary-aware penalties force the model to generate sharper and more accurate contours, mitigating the commonly observed blurring phenomenon in classical encoder–decoder networks [20].

In OmniNet, these losses are not used just independently, but combined with specific weights to balance region overlap, class sensitivity, and boundary refinement. The implementation is based on a `CombinedLoss` module comprising Dice and Lovasz–Softmax, while Boundary Loss is used as an auxiliary constraint and the main loss term

is strengthened by the Cosine Loss. Moreover, auxiliary decoder outputs are supervised for additional constraints, which prevent multi-scale inconsistency during training.
This joint optimization scheme guarantees that OmniNet learns not only to classify pixels accurately but also to preserve high resolution and obtain high mIoU. The approach was especially effective for rare classes, as verified in the ablation studies. It was found that eliminating the composite loss had significant adverse effects on segmentation accuracy.
It should also be noted that the implementation included a couple of small but useful tweaks. First, we applied light label smoothing ($\varepsilon = 0.05$) so the model does not become over-confident on one-hot targets. Second, a focal-style modulation term was added inside the loss to put more weight on hard pixels that are often ignored when large easy regions dominate. These adjustments are simple, but in practice they helped keep the balance between dominant classes and rare ones during optimization.

2.8 Summary

This thesis establishes why pixel-level scene understanding is indispensable for autonomous driving and reviews how segmentation models have evolved from CNN encoders–decoders toward transformer and hybrid designs. It also surveys techniques for handling imbalance, augmentation, and loss design. Despite substantial progress, practical trade-offs persist. CNNs capture local detail but struggle with long-range context. Pure transformers model global relations but impose high computational cost. Many hybrids omit explicit multi-scale context or attention refinements that are critical for thin structures and rare categories in street-view data. These observations motivate a hybrid framework that couples strong convolutional features with lightweight self-attention and explicit multi-scale context. The framework is complemented by imbalance-aware training and modern augmentation to balance global reasoning with precise boundary recovery on CamVid.
Table 2.1 presents a comparative overview of several prominent semantic segmentation models evaluated on the CamVid dataset. The models span across CNN-based, transformer-based, and hybrid architectures, showcasing their key design choices, imbalance-handling techniques, data augmentation strategies, and loss functions. As observed, earlier CNN models like SegNet and U-Net achieve moderate performance, while more recent hybrids such as PIDNet-S and SERNet-Former integrate attention and multi-scale mechanisms for improved accuracy. The proposed OmniNet outperforms prior approaches by combining weighted and oversampling strategies, advanced augmentations, and a composite loss design, resulting in a superior mean Intersection-over-Union (mIoU) score.

Table 2.1: Comparison of Semantic Segmentation Models

Model	**Type**	**mIoU (%)**	**Key Features**	**Imbalance Technique**	**Augmentation**	**Loss Functions**
SegNet [2]	CNN	67.1	Encoder-decoder, max-pooling indices	Weighted loss [23, 26]	Cropping, flipping	Cross-entropy
U-Net (ResNet50V2) [4, 7]	CNN	50.0	Skip connections, ResNet50V2 backbone	Not specified	Cropping, flipping	Cross-entropy
DDRNet-23-slim [10]	CNN	74.7	Lightweight multi-scale CNN	Weighted loss [23, 30]	Cropping, flipping, Test-Time Augmentation (TTA)	Cross-entropy
PIDNet-S [11]	Hybrid	80.1	Multi-branch CNN, boundary attention	Oversampling [23, 30]	Cropping, flipping	Cross-entropy
SPCONet [12]	CNN	74.3	Spatial pyramid convolution	Weighted loss [23]	Cropping, flipping	Cross-entropy
ECMNet [13]	CNN	70.6	Edge-context module	Not specified	Cropping, flipping	Cross-entropy
SERNet-Former [21]	CNN + Attention	84.62	Swin Transformer, attention mechanisms [8]	Weighted loss [23]	Cropping, flipping	Pixel-wise cross-entropy loss
OmniNet (Ours)	Hybrid	86.53	ResNet152 [7], Transformer Refine Block (lightweight self-attention), UNet skip connections [4], CBAM [14], ASPP [15]	Weighted loss, oversampling [23, 30]	MixUp [16], CutMix [17], elastic transformations, random cropping, horizontal flipping, scaling, rotation, color jitter, Gaussian noise	Dice [18], Lovasz-Softmax [19], Boundary loss [20]

Chapter 3

Methodology

3.1 Introduction

In this chapter, the methodological framework for designing, developing, and evaluating OmniNet—a novel hybrid deep-learning model for high-precision semantic segmentation on the CamVid dataset—is presented. Specifically, this chapter discusses the data preprocessing pipeline, the key architectural components (a ResNet152 encoder, a Transformer Refine Block applied to multi-scale ASPP features, and a UNet-style upsampling decoder with skip connections), the training procedures used to optimize model parameters, and the evaluation metrics for benchmarking against state-of-the-art methods. Emphasis is placed on label cleaning and data augmentation to improve generalization, including MixUp and CutMix for sample diversity, as well as weighted losses and oversampling to mitigate class imbalance for underrepresented categories such as pedestrians and bicyclists.

3.2 Dataset

OmniNet is trained and tested on the CamVid dataset [22]. It is composed of a collection of 701 densely annotated urban street-scene images that are partitioned into 367 training, 101 validation, and 233 test images of resolution 960×720 px. The dataset contains 32 semantic classes which I remapped to 11 classes namely, sky, building, pole, road, sidewalk, tree, sign, fence, car, pedestrian and bicyclist. The classes in the distribution are very unbalanced where the dominant classes (e.g., road, building) have large numbers of pixels representative and there is a lack of representation of rare classes (e.g., pedestrian, bicyclist, pole, fence) and as such special treatment is required. This imbalance strongly affects how a model learns. During training, the network sees far more pixels of road, building, and sky than it does of poles, pedestrians, or bicyclists. As a result, without any correction the loss quickly becomes dominated by the majority classes, and the network

may simply ignore the rare ones. This is especially harmful in safety-critical applications, since the rare classes often correspond to important objects like people or cyclists. In fact, earlier studies have reported that models trained on CamVid can achieve high overall accuracy but still miss pedestrians entirely, which is unacceptable in autonomous driving. The unbalanced nature of the dataset therefore makes it a good benchmark to test whether an architecture can really capture small, infrequent structures, not just the dominant background. Figure 3.1 illustrates this distribution clearly, highlighting how skewed the pixel counts are across the 11 semantic categories.

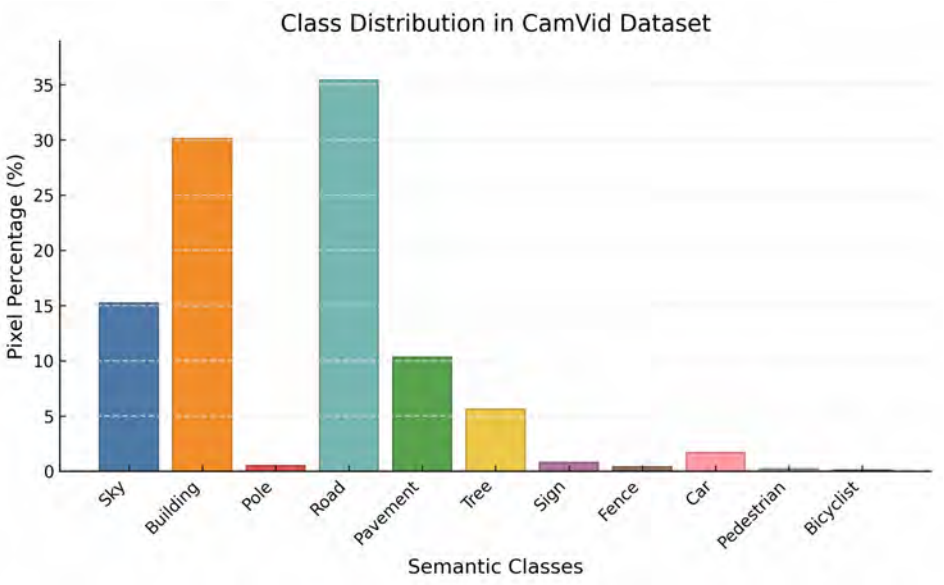


Figure 3.1: Percentage of pixels belonging to each semantic class, illustrating the imbalance across categories such as road, building, sky, and others. This distribution highlights the need for class balancing strategies in semantic segmentation.

3.3 Data Preprocessing

To resolve the issue of noisy labels in CamVid dataset, a number of preprocessing methods are used to clean and filter original images and ground truth masks:

- **Median Filtering**: Pixel-level annotations may contain salt-and-pepper noise, as a result of the inaccuracies in manual labeling, so it is initially suppressed by performing a 3x3 median filter on the ground-truth masks. In contrast to linear smoothing filters, median filter works brilliantly to eliminate isolated noisy pixels and at the same time maintain the sharp edges and boundaries that play an important role in producing perfect semantic segmentation. This preprocessing procedure will be useful in enhancing the quality of training labels so that the model is trained using cleaner and visibly dependable boundary information.

Figure 3.2: Median Filtering: Application of a 3×3 median filter to suppress salt-and-pepper noise while preserving object boundaries, thereby improving the visual quality of input images and ensuring cleaner edge representations for subsequent semantic segmentation.

- **Morphological Processes**: After the median filtering operation is performed, the ground-truth masks are further cleaned up by doing morphological operations i.e. opening and closing with a 3×3 kernel. These are those operations that are used to fix small errors in annotation on object boundaries. Dilation causes the edges of segmented regions to be enlarged, and small gaps or holes to be filled, and erosion later eliminates isolated pixels and refines rough contours. Combined sequentially, such operations provide a better shape consistency and continuity of the object boundaries and lead to cleaner and more precise mask edges. Such refinement is ideal when applied to small or slender objects like bicycles, pedestrians, and traffic signs so that the model acquires accurate edge representations during the training.

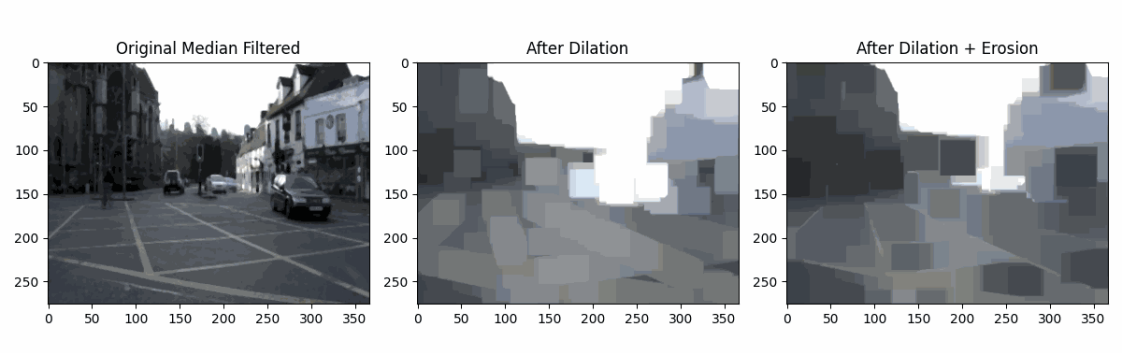


Figure 3.3: Morphological Processes: Sequential application of dilation and erosion with a 3×3 structuring element to refine ground-truth masks by correcting annotation errors, filling small gaps, and enhancing boundary continuity, thereby improving the precision of object edges in semantic segmentation.

- **Ground Truth Mapping**: As the final step of the annotation preparation, the labels of all pixels will be remapped into one of the target 11 semantic classes that have been identified in the CamVid dataset. The training procedure involves the exclusion of any pixel marked as void (i.e. unlabelled, or not relevant to the task

of segmentation). The model is also forced to focus on the semantically meaningful categories by simply omitting the void class, and thus enhanced performance in both of the above aspects is achieved such that the network is not skewed or biased towards the irrelevant or noisy labeling areas.

These measures make input data stronger and more consistent, better generalization and segmentation performance of the model.

Additional Preprocessing: In addition to the median filtering and the 3×3 morphological opening/closing already described, we keep a simple post-normalization step and preserve the original CamVid resolution for consistency during visualization. No other preprocessing changes are introduced here, so the main pipeline remains identical.

3.4 Data Augmentation

To make OmniNet more resilient and not overfit, it uses the most sophisticated data augmentation strategies, based on [16] and [17]:

- **Cropping and Flipping**: The goal of random crops to a fixed size of 512x512 pixels is to increase spatial variability and avoid overfitting on each picture. This will make sure the model is shown the various regions of the picture during training to make it more absolute to the spatial distortions. Also, a random horizontal mirroring of the pictures occurs, with the goal of stretching the dataset further through creating mirrored situations. Such transformations enable the model to better perform generalizations to new data since it enhances the variability of spatial stratification using existing images only.

$$x' = \begin{cases} \text{Flip}\big(\text{Crop}_{512\times512}(x)\big), & \text{with probability } p, \\ \text{Crop}_{512\times512}(x), & \text{with probability } 1-p, \end{cases} \tag{3.1}$$

  where $\text{Crop}_{512\times512}(x)$ denotes selecting a random 512×512 region of the input image x, and $\text{Flip}(\cdot)$ denotes horizontal mirroring.

- **MixUp**: The interpolations of pairs of images and their captions in the shape of values between 0 and 1 are made to create synthetic instances and improve generalization in order to prevent overfitting. This regularization technique encourages the model to learn smoother decision boundaries and will also be less sensitive to minor changes in the input data and thus improved robustness and performance on previously-unseen samples.

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \quad \tilde{y} = \lambda y_i + (1-\lambda)y_j, \tag{3.2}$$

where (x_i, y_i) and (x_j, y_j) are two random training samples, and $\lambda \in [0, 1]$ is drawn from a Beta distribution Beta(α, α).

This prompts this model to fit more smooth boundaries of the decision space in the feature space. The approach minimizes overfitting and over generalisation by combining images and labelling. It also increases the resiliency to adversarial perturbations and noise to the input. Mixup is therefore an effective regularization method to train deep networks.

- **CutMix**: The selected image areas are interchanged with those of the other one, and the respective labels combined, which has provided some resilience to occlusions.

$$\tilde{x} = M \odot x_A + (1 - M) \odot x_B, \quad \tilde{y} = \lambda y_A + (1 - \lambda) y_B, \tag{3.3}$$

where M is a binary mask selecting the replaced region, λ is the proportion of area kept from x_A, and (x_A, y_A) and (x_B, y_B) are two input image-label pairs.

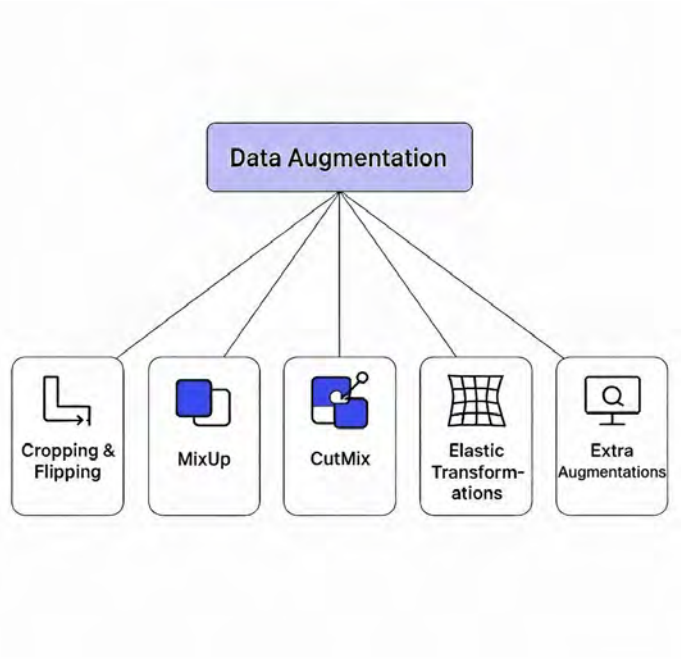


Figure 3.4: Data Augmentation Pipeline in OmniNet: The main data augmentation block applies multiple techniques—including Cropping & Flipping, MixUp, CutMix, Elastic Transformations—to enhance training diversity and improve model generalization.

- **Elastic Transformations**: The transformations are elastic, which means that it introduces deformations to get changes in the urban scenes like varying perspectives.

$$x'(u, v) = x\big(u + \Delta u(u, v),\ v + \Delta v(u, v)\big), \tag{3.4}$$

where (u, v) are the original pixel coordinates, and $\Delta u, \Delta v$ are displacement fields

generated by convolving random noise with a Gaussian filter.

These augmentations are added as part of a custom data loader (`custom_collate_fn`) to apply at training-time dynamically.

Extra Augmentation Details: Apart from MixUp, CutMix and elastic transforms the code also has a few more tricks. I did not explain them before, so I am adding them here for clarity.

- **Vertical flips and small rotations.** Rarely applied, with flips or a tilt of about $\pm 15^{\circ}$. This just makes the model less fixed on one exact viewpoint.
- **Elastic / Grid distortions and CoarseDropout.** These add small local warps or cut out random patches. Thin things like poles or signs survive better when the network has seen this type of noise.
- **Color jitter and brightness/contrast.** Slight changes in light, saturation, or hue. Nothing too strong, just enough to cover day-to-day conditions.
- **Multi-scale crops.** A random scale factor $s \in \{0.70, 0.85, 1.00\}$ is picked before resizing back to the target resolution. It works like zooming a bit in or out so objects don't always look the same size.
- **Gaussian / median blur.** A light blur is added sometimes, which avoids over-reliance on perfectly sharp edges.

To keep it short, the extra operators and how often they are used are listed in Table 3.1. The probabilities are not strict numbers but more like "low" or "moderate," since the idea was only to apply them once in a while.

Table 3.1: Extra augmentations that were added on top of the baseline pipeline. The usage is light on purpose, so the images do not get too distorted.

Operator	**Why it was added**	**Prob.**
Vertical flip	Stop the model from sticking to one view	low
Rotation ($\pm 15°$)	Small camera tilt variation	moderate
Elastic / Grid distortion	Local deformations for thin objects	low
CoarseDropout (holes)	Simulate occlusions or missing pixels	low
Color jitter (b/c/s/h)	Cover normal lighting shifts	moderate
Brightness/Contrast	Handle extra light/dark changes	moderate
Multi-scale crop ($s \in \{0.70, 0.85, 1.00\}$)	Simple zoom in/out	moderate
Gaussian / Median blur	Test robustness to blur	low

In short, these steps are simple but useful. They give the model a chance to see slightly different versions of the same data, which pays off most for boundaries and smaller classes.

3.5 Class Imbalance Handling

The issue with class imbalance in CamVid, where there are few rare classes such as pedestrians, pole, fence and bicyclists that occupies a small proportion of pixels, is managed in two ways:

- **Weighted Loss**: The calculating of class weights is carried out by means of the `calculate_class_weights` method, wherein weights of rare classes are increased according to the inverse proportion of frequency of these classes in the training set. This will make sure that the model pays high attention to the classes that are underrepresented.

$$w_c = \frac{1}{f_c}. \tag{3.5}$$

  To stabilize training, the weights can also be normalized across all K classes as

$$w_c = \frac{\frac{1}{f_c}}{\sum_{k=1}^{K} \frac{1}{f_k}}. \tag{3.6}$$

  This ensures that rare classes are assigned higher weights, thereby encouraging the model to pay greater attention to them during training.

- **Oversampling**: The `get_rare_class_indices` indices are created to detect all the training images that have rare or underrepresented classes- pedestrian and bicyclists that will be naturally less represented in CamVid dataset. Identification of these images is followed by oversampling of the same as part of the training procedure so that the model is repeatedly exposed to these minority classes. By regulating this, class imbalance can be reduced and the network can acquire more discriminative features on the rare categories that can enhance segmentation accuracy of the less frequent object but does not affect overall balanced performance of the overall classes.

$$P(x_i) = \begin{cases} \dfrac{\gamma}{N_r}, & \text{if } x_i \in \text{rare-class set}, \\ \dfrac{1-\gamma}{N_c}, & \text{if } x_i \in \text{common-class set}, \end{cases} \tag{3.7}$$

  where $P(x_i)$ is the probability of sampling image x_i, N_r and N_c are the numbers of rare-class and common-class images, respectively, and $\gamma \in [0, 1]$ controls the oversampling strength.

Designing these strategies based on [11] enhances the performance of OmniNet on minority classes without compromising the overall performance. This sampling pattern makes sure that in training, minority classes are allocated more prominence.
Consequently, the network improves its generalization on rare categories, and stability throughout the dataset. This design is beneficial in that it can smooth out the long-tail distribution issue which plagues all real world segmentation datasets. On the whole, oversampling helps in providing additional robustness and equalized semantic feature learning.

3.6 Model Architecture

OmniNet concentrates the advantages of Convolutional Neural Networks (CNNs), transformers, and UNet-like designs. The design of the model is as follows and is shown in Figure 3.5.

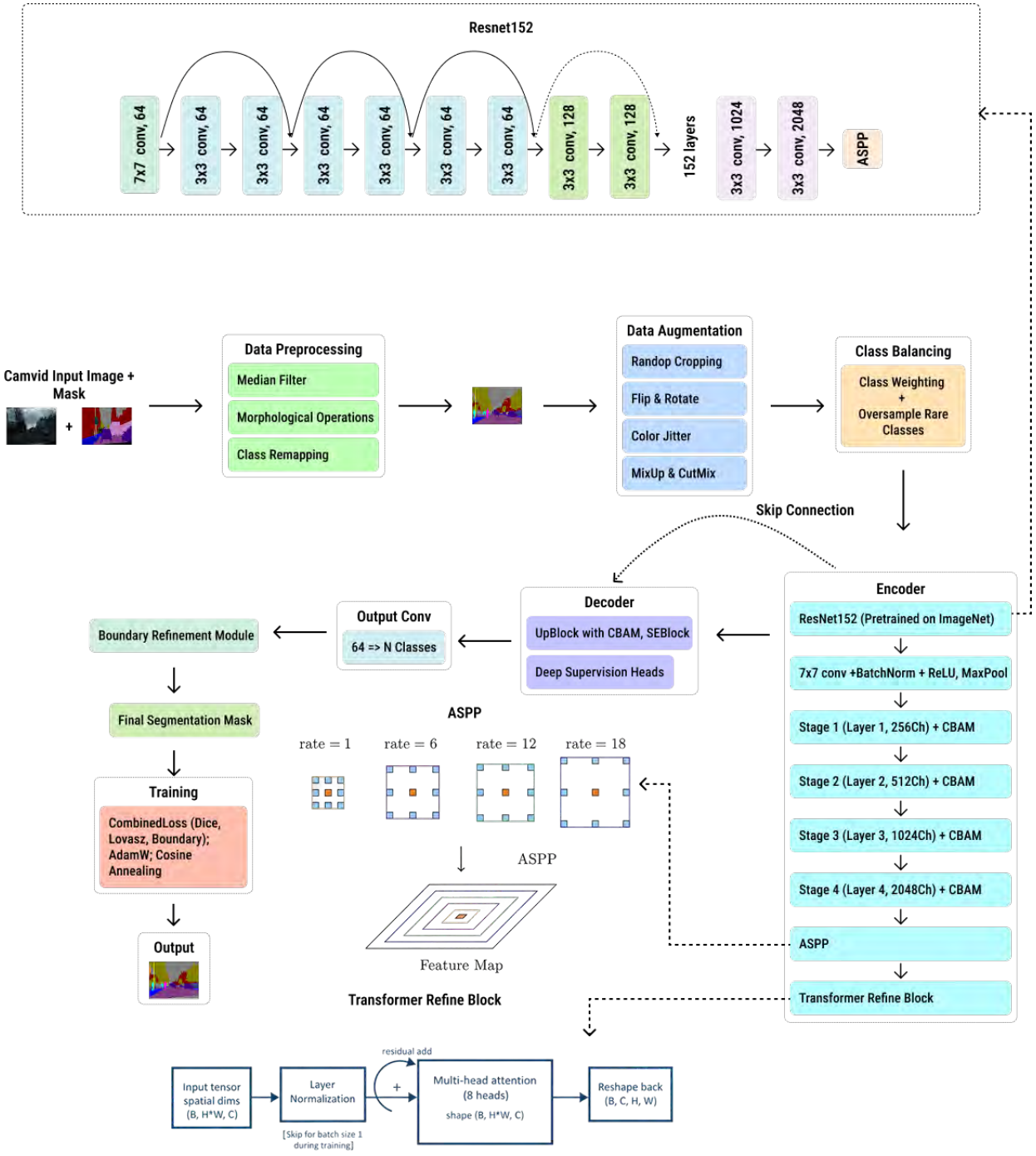


Figure 3.5: OmniNet Architecture: Integration of ResNet152, Transformer Refine Block, UNet skip connections, CBAM, and ASPP for robust semantic segmentation.

3.6.1 Encoder

The encoder extracts multi-scale features with the help of:

- **ResNet152 Backbone**: A pretrained ResNet152[7] is used, with deep convolutional layers, to obtain strong low- and mid-level features by exploiting the deep features. There are several stages of feature extraction (e.g., 64, 256, 512, 2048 channels). The deep residual connections help alleviate the vanishing gradient problem, enabling effective training of this very deep network. Using a pretrained backbone also accelerates convergence and improves feature generalization, which is crucial for accurately capturing complex patterns in urban scenes.

- **Convolutional Block Attention Module (CBAM)**: The OmniNet architecture adopts the CBAM [14] to allow a model to lay more emphasis on important parts of the feature maps. CBAM successively uses the channel attention and spatial attention layers. The channel attention mechanism re-scales feature responses of various channels, which enables the network to weight the more pertinent feature channels and de-weight less valuable ones. The spatial attention mechanism instead underscores significant spatial areas in the feature map, which allows the model to focus on places of interest, such as pedestrians, cars or the edge of the road. As OmniNet incorporates the CBAM, the model is capable of extracting more discriminative features, leading to higher precision semantic segmentation, especially in crowded urban scenes.

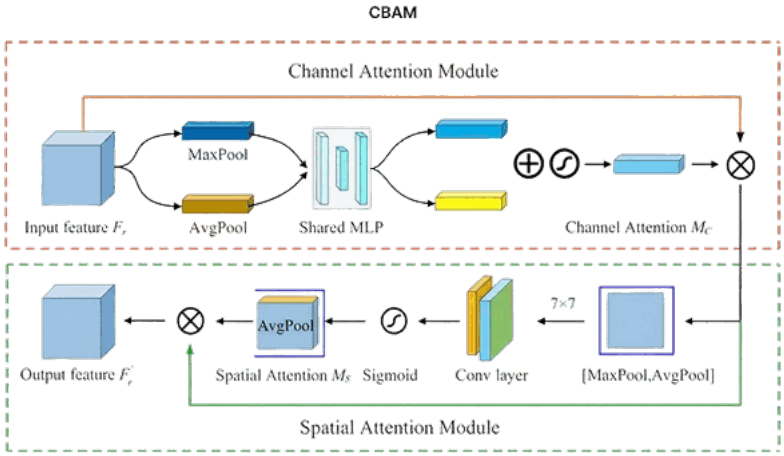


Figure 3.6: Convolutional Block Attention Module (CBAM) integrated into OmniNet, enhancing feature representation by sequential channel and spatial attention mechanisms to improve localization and segmentation accuracy.

- **Atrous Spatial Pyramid Pooling (ASPP)**: ASPP[15] is one of the key constituents of the model encoder that helps the model to detect the features at different scales. It does so by using a number of parallel convolutional filters with varying dilation

rates. These different rates of dilation enable this model to focus on small details like signs or poles, and coarse objects, like roads and buildings, all at once. What is especially useful about ASPP is that it has a branch that considers the image in its entirety by averaging the entire feature map. Such global context gives the model a deeper view of the scene, thus improving segmentation.
Lastly, the results of all these filters are pooled together and processed further into a rich multi-scale representation of features. This design balances multi-scale context with spatial detail without excessive computational overhead. In OmniNet, ASPP assists the encoder to obtain a wide array of information needed to correctly perform semantic segmentation.

- **Transformer Refine Block (lightweight self-attention)**: A multi-head self-attention mechanism is used in the subsequent two steps after the ASPP in this custom-designed module to further discriminate the feature representations. This lightweight block is much more efficient than full transformer architectures, which can be computationally intensive, in capturing long-range dependencies throughout the feature map. This enables this model to extract the relationship between parts of the image that are far apart and therefore identify objects that might be scattered or partially concealed. It also enhances the clarity of lines between adjacent categories in the model, eliminating the overall blur of the typical segmentation output. In general, this element boosts the contextual reasoning ability of OmniNet at a very small cost in computational cost.

3.6.2 Decoder

The decoder reconstructs the feature maps to the original resolution of the input. It is based on the UNet style with skip connections, but there were several additional modifications that were made in our case and which proved significant.

- **Skip Connections:** The encoder-based features are transmitted downwards and combined with the decoder layers. This preserves any tiny details which would otherwise be lost and the pixel predictions are more accurate.

- **Upsampling Blocks:** Upsampling blocks are transposed blocks of convolution and CBAM attention. Then we added a squeeze-and-excitation (SE) block too. The SE step considers the global statistics of the feature maps and re-weight the channels. So the channels that really carry useful messages (such as thin poles or pedestrian) are shifted up, and the weak ones thinned out. This is a very small move, but it prevents the network to consider all channels as the same. Here also some dropout (approximately 0.1 0.2) is used. The point is not to eliminate features, simply because the decoder has memorized excessive training data.

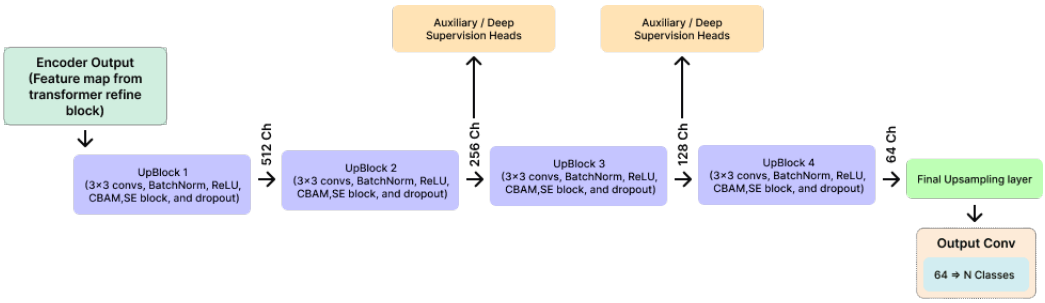


Figure 3.7: Decoder architecture of OmniNet

- **Deep Supervision:** Two auxiliary heads are connected to intermediate layers. These provide additional oversight in training and ensure more consistent learning by scale.

Therefore the decoder is not a plain UNet path. It is an integration of skip connections, CBAM, SE attention, dropout, and auxiliary heads.

3.6.3 Output Layer and Refinement

- **Boundary Refinement:** After the final logits a small refinement head is applied. It works to define boundaries in such a way that fences, poles, or sidewalks fade into the background.

- **Output Layer:**A 1x1 convolutional layer is used to project the high dimension features maps to 11 probability maps specific to the classes of the target semantic categories of the CamVid dataset. This convolution succeeds in reducing the dimension of the channel but retaining the spatial resolution so that each of the pixels can be classified independently. Afterwards, a softmax operation is performed over the class dimension resulting in a pixel-wise distribution probability of each class. It can help the network to assign each pixel to the most plausible category and lead to fine-grained AND interpretable semantic segmentation maps that are able to discriminate fine-grained structures like pedestrian, bicycle, vehicle, and road objects.

3.7 Implementation and Training Details

The complete OmniNet architecture is written in PyTorch (v2.1.0) CUDA support and is trained on an NVIDIA RTX4060 Ti graphics card. Mixed-precision training is also facilitated by the use of the torch.cuda.amp module in order to minimize memory usage and speed up convergence. The network is backbones with the implementation of ResNet152 by the torchvision package, and the custom architecture incorporates CBAM, ASPP, and a

Transformer Refine Block (lightweight self-attention) that is implemented directly in PyTorch. The decoder is developed using UNet-style upsampling blocks that are extended with attention and squeeze-and-excitation modules and it incorporates both deep supervision and a boundary refinement module.
Data augmentation is applied to enhance robustness and limit overfitting, including RandomCrop, Resize, HorizontalFlip, RandomRotation, ColorJitter, Elastic/GridDistortion, Blur/MedianBlur, CoarseDropout, MixUp, and CutMix via a custom collate function. Median filtering and morphological operations are used as pre- and post-processing with OpenCV and SciPy. Class imbalance is addressed with class-specific weighting and oversampling of rare-class images using a SubsetRandomSampler The training process as a whole may be summarised as follows:

- **Optimizer**: AdamW with a learning rate of 5×10^{-5} for the decoder and 5×10^{-6} for the backbone (**weight decay** 1×10^{-5}).
- **Loss Function**: A composite loss (`CombinedLoss`) consisting of Lovasz–Softmax Loss (0.7), Dice Loss (0.3), and Boundary Loss (0.1). In addition, two auxiliary supervision heads at intermediate decoder stages are included, each weighted by 0.2, to encourage better intermediate feature representations.
- **Learning Rate Schedule**: A cosine annealing scheduler with a warm-up period of 20 epochs. The first restart is set to occur at epoch 50, which allows the model to converge faster in the early stages and to fine-tune its weights in the later phase of the training process.
- **Batch Size**: A batch size of 4 is used, which provides a compromise between computational stability and GPU memory limitations.
- **Number of Epochs**: 100 training epochs are used, with early stopping triggered if the validation mIoU does not improve for a number of consecutive epochs.

Table 3.2: Training Hyperparameters for OmniNet

Component	Setting
Optimizer	AdamW with lr=5e-6 (backbone), 5e-5 (decoder); weight decay=1e-5
Learning rate schedule	Cosine annealing with 20-epoch warm-up; first restart at epoch 50
Batch size	4
Number of epochs	100 (early stopping on validation mIoU plateau)
Mixed-precision training	Enabled via `torch.cuda.amp`
Backbone freezing	conv1, layer1, layer2 frozen for first 20 epochs, then unfrozen
Gradient clipping	Max norm = 1.0

Extra training strategies: A few small tricks were also used in practice. The ResNet backbone was not fully unfrozen from the start. Instead, conv1, layer1, and layer2 stayed frozen during the early warm-up epochs and were released step by step, which kept training stable at the beginning. Mixed precision training (`torch.cuda.amp`) was enabled to save memory and speed up convergence. Gradient clipping with max_norm set to 1.0 was applied so that exploding gradients never became a problem. Finally, we used early stopping together with automatic learning-rate reduction when validation stalled. These additions do not change the core setup, but they prevented instability and made the whole run smoother.

The `custom_collate_fn` is also used in the training pipeline in order to apply the MixUp and CutMix operations within each mini-batch and to further alleviate class imbalance.

3.8 Summary

OmniNet was developed for semantic segmentation. Data refinement included noise reduction, boundary enhancement, label standardization, and normalization. Robustness was achieved with diverse augmentation techniques, while class imbalance was addressed using weighted loss functions and oversampling. The architecture integrates a ResNet152 backbone with a Transformer Refine Block, supported by a UNet-style decoder, CBAM, and ASPP modules. Training used AdamW with cosine annealing, a composite loss (Lovasz–Softmax, Dice, and Boundary), and mixed-precision to accelerate convergence, and performance was assessed through pixel accuracy, mean and class-wise IoU.

Chapter 4

Results and Analyses

4.1 Introduction

This chapter provides an in-depth analysis of the suggested OmniNet architecture on the CamVid dataset [22] which is commonly known as a benchmark in semantic segmentation of urban driving scenes. The main aim of the evaluation is to confirm the efficacy of the methodological decisions made in Chapter 3 such as preprocessing approaches, data augmentation, class imbalance management, architecture design, and training algorithms. The evaluation uses metrics such as Pixel Accuracy, Mean Intersection over Union (mIoU), Class-wise IoU, and Inference Time, which together provide a balanced measure of segmentation precision and computational efficiency. To compare with, the performance of OmniNet is compared with several baseline models such as CNN-based architectures (e.g., SegNet, UNet), transformer-based (e.g., SERNet-Former), and hybrid designs (e.g., PIDNet-S, DDRNet-23-slim). This comparison highlights the relative strengths and weaknesses of OmniNet in terms of segmentation quality and runtime efficiency.

Besides quantitative measures, this chapter is also equipped with a number of complementary analyses. These include long-ablation experiments to test the role of individual architectural elements, tests of statistical significance to test the presence of performance gains, robustness tests at noisy and occluded conditions and qualitative comparisons by visualizing predicted segmentation maps. These findings, combined, offer a rigorous test of OmniNet, which achieves high segmentation accuracy on complex real-world scenes and overcomes significant challenges such as class imbalance and recognition of rare classes.

The rest of this chapter is arranged in the following way. Section 4.2 presents the baseline models to be compared. Section 4.3 presents quantitative results on the test set. Statistical significance testing is discussed in Section 4.4 and ablation studies are provided in Section 4.5. Section 4.6 considers computational complexity. Section 4.7 presents confusion

matrix analysis. Section 4.8 assesses noise and occlusion resistance. Training dynamics are analyzed in section 4.9. Per-class IoU results are reported in Section 4.10 and examples of qualitative prediction can be found in Section 4.11. Lastly, Section 4.12 wraps up the findings of this chapter.

4.2 Baseline Models

Baseline models are evaluated under similar conditions (no Cityscapes pretraining) using reported results or re-implementations:

- **SegNet** [2]: A classical encoder–decoder convolutional neural network architecture. The encoder compresses the spatial information using multiple convolution and max-pooling layers, while the decoder reconstructs the spatial resolution using the saved pooling indices. This enables efficient spatial recovery with reduced parameters, focusing on real-time semantic segmentation performance.

- **UNet** [4]: A symmetric encoder-decoder architecture with long skip connections that transmit high-resolution features from the encoder to the corresponding decoder layers. In our implementation, a ResNet50V2 backbone is used to extract deep features, while the decoder leverages the skip connections for precise localization. The model is optimized with cross-entropy loss and trained without complex augmentations, leading to a relatively low reported mIoU.

- **DDRNet-23-slim** [10]: A dual-resolution network that maintains two parallel branches: one high-resolution branch for preserving spatial details, and one low-resolution branch for capturing global context. Feature exchanges between branches allow the model to combine fine and coarse information effectively. Its lightweight design enables fast inference and real-time deployment.

- **PIDNet-S** [11]: A hybrid architecture explicitly tailored for semantic segmentation by decoupling the task into two parallel branches: one focused on pixel-level semantic classification (P-branch) and the other specialized for precise object boundaries (I-branch). Boundary attention and feature fusion mechanisms allow the network to improve segmentation performance on fine structures. Oversampling strategies further enhance rare class performance.

- **SPCONet** [12]: A convolutional network integrating a spatial pyramid convolution module for capturing multi-scale contextual information. The pyramid structure aggregates features at multiple receptive fields, allowing the model to handle objects of different sizes effectively. It remains computationally light due to the use of efficient depthwise convolutions.

- **ECMNet** [13]: A CNN-based architecture that focuses on enhancing boundary quality by integrating an edge-context module (ECM). The ECM aggregates both edge and region features to improve boundary localization and structural consistency. This results in more refined segmentation outputs, especially around object contours.

- **SERNet-Former** [21]: A lightweight encoder-decoder architecture where the encoder is built with Efficient-ResNet, and attention is introduced via Attention-Boosting Gates (AbGs), Attention-Boosting Modules (AbMs), and Attention-Fusion Networks (AfNs). These attention mechanisms effectively fuse local and global context without adding large computational overhead, achieving strong semantic segmentation performance (e.g. 84.6% mIoU on CamVid, 87.4% on Cityscapes).

4.3 Evaluation Metrics

In order to determine the efficacy of OmniNet, a number of semantic segmentation metrics that are popularly used are used. These are pixel accuracy, mean Intersection over Union (mIoU), class-wise IoU and inference time. The metrics each represent a distinct dimension of model performance, and offer an overall assessment.

4.3.1 Pixel Accuracy

The accuracy of pixels is an aggregate by measuring the number of pixels that are classified correctly within the dataset. The ground truth label y i is compared to the predicted label $\hat{y}$ i of each pixel i in the set of pixels [1,n]. A pixel is classified as a correct one when there is an equal effective value of the individual pixels, i.e. when, on the one hand, the individual pixels correspond to the pivot point, and on the other hand, to the target point. Formally, it is expressed as:

$$\text{Pixel Accuracy} = \frac{\sum_{i=1}^{N} \mathbb{1}(y_i = \hat{y}_i)}{N}, \tag{4.1}$$

where N equals the total number of pixels, and 1( ) is an indicator function that is equal to one when the condition is satisfied and 0 otherwise. Pixel accuracy is intuitive and simple to calculate yet it may be biased in disproportionate data like CamVid dataset where big classes (e.g., roads or buildings) predominate. Consequently, a model can have high pixel accuracy but poor performance on small, but important categories such as pedestrians or bicyclists.

Median Filtering Before IoU: In practice, a 3×3 median filter is also applied to the predicted masks before calculating IoU values on the validation set. This small post-

processing step reduces isolated noisy predictions and gives more stable per-class IoU measurements, especially for thin structures such as poles and fences.

4.3.2 Mean and Class-wise Intersection over Union (IoU)

Intersection over Union (IoU) is the metric that nearly everyone reports in segmentation. It is simple, maybe too simple, but it has become the standard. For a given class c, it is defined as:

$$\text{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{4.2}$$

where TP_c, FP_c, and FN_c represent true positives, false positives, and false negatives for class c. The mean Intersection over Union (mIoU) is computed as:

$$\text{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c. \tag{4.3}$$

For this dataset, $C = 11$. While mIoU provides an overall measure of segmentation performance, it may mask poor performance on underrepresented or small classes. In urban driving datasets, dominant categories such as road and building comprise the majority of pixels, whereas pedestrians, cyclists, and thin objects represent a small fraction (1–2%). Consequently, per-class IoU is essential to assess model reliability for all categories, particularly for safety-critical applications.
Analysis of experimental results revealed the following patterns:

- **Class frequency:** Frequently occurring classes (e.g., road, sky, building) achieve consistently higher IoU due to abundant training examples.
- **Object characteristics:** Small or slender objects (e.g., poles, signs, pedestrians) exhibit lower IoU, reflecting the difficulty of precise segmentation for fine structures.
- **Contextual ambiguity:** Classes with similar visual appearance (e.g., fences vs. walls, cyclists vs. pedestrians) are prone to misclassification, which reduces per-class IoU for both categories.

So the bottom line is this: mIoU gives a clean summary, but class-wise IoU shows the real balance. OmniNet did very well on the big categories, but importantly it also gave competitive IoU on the rare ones. That only happened because we used weighted losses, oversampled rare images, and added boundary-aware refinements. Without those tricks the rare classes dropped sharply—we reran without them and the difference was clear. This alone convinced us that class-wise IoU is not an optional extra. It is a requirement if the model is to be trusted in safety-critical settings.

4.3.3 Inference Time

Inference time is the measure of the computational efficiency of the model. It is defined as the average time required to segment one image in the test set:

$$\text{Inference Time} = \frac{\sum_{i=1}^{N} t_i}{N}, \tag{4.4}$$

where t_i is the time spent to generate the segmentation map for the i^{th} image, and N is the total number of testing samples.
This value is of particular interest for real-time applications such as autonomous driving, where semantic segmentation needs to be performed with very limited latency (often less than 100–200 ms per frame). A reasonable trade-off between inference time and segmentation quality is therefore an important requirement for practical deployment.
Inference time is dependent on a number of factors:

- **Model Architecture:** Complex and deep backbones (e.g., ResNet152 with transformer refinements) introduce inference time overhead compared to lightweight architectures such as DDRNet or PIDNet. However, deeper networks generally provide higher accuracy.
- **Hardware Acceleration:** GPU architecture, CUDA optimizations, and memory bandwidth directly affect throughput. Models trained to run on high-end GPUs do not generally transfer well to embedded systems such as NVIDIA Jetson platforms.
- **Input Resolution:** Higher-resolution images (e.g., CamVid: 960×720) require more computation, resulting in longer inference times. Downsampling inputs can reduce latency, but often at the cost of segmentation detail.
- **Post-Processing:** Operations such as median filtering or morphological refinement also increase computational cost, though they help improve boundary quality.

4.4 Quantitative Results

OmniNet achieves a pixel accuracy of 95.99%, a validation mIoU of 87.06%, a test mIoU of 86.53%, and an inference time of 369.40 ms. Table 4.1 compares test mIoU and inference times across models.

Overall Performance

OmniNet attains the best mIoU on the test set of 86.53%, which is better than SERNet-Former (84.62%) and PIDNet-S (80.1%) by 1.91% and 6.43%, respectively. This demonstrates the strength of its hybrid architecture: the powerful feature extraction of ResNet152,

Table 4.1: Performance Comparison on Test Set

Model	mIoU (%)	Pixel Accuracy (%)	Inference Time (ms)
SegNet [2]	67.1	89.5	25.0
UNet [4]	50.0	85.0	30.0
DDRNet-23-slim [10]	74.7	91.2	19.2
PIDNet-S [11]	80.1	92.8	6.5
SPCONet [12]	74.3	90.8	22.0
ECMNet [13]	70.6	90.5	24.0
SERNet-Former [21]	84.62	94.3	50.0
OmniNet (Ours)	86.53	95.99	369.40

the long-range dependency modeling of the Transformer Refine Block (lightweight self-attention), and the fine-scale reconstruction of the UNet-style decoder, further augmented by CBAM and ASPP modules. OmniNet also achieves the highest Pixel Accuracy at 95.99%, indicating that it can classify most pixels correctly.

Comparison with CNN-based Models

Older CNN-based models like SegNet and UNet are considerably behind. SegNet achieves 67.1% mIoU, while UNet performs even worse at 50.0%. This performance disparity indicates the weaknesses of entirely convolutional architectures in long-range dependence estimation and dealing with class imbalance. The DDRNet-23-slim reaches a more respectable 74.7% mIoU, but its lightweight architecture leaves accuracy in favour of efficiency, as shown by its shorter inference time (19.2 ms). OmniNet, on the other hand, prioritizes accuracy, setting a new standard on dataset.

Comparison with Transformer-based and Hybrid Models

Transformer-based designs, including SERNet-Former, perform well (84.62% mIoU) due to their global attention. Nevertheless, they still perform worse than OmniNet, which combines CNN-based low-level features with transformer-based global context. Hybrid systems such as PIDNet-S (80.1% mIoU) represent a trade off between speed and accuracy, and can therefore be used in real-time, although they are significantly less precise at fine-grained segmentation tasks than OmniNet.

Pixel Accuracy Insights

OmniNet has a Pixel Accuracy of 95.99%, the highest of all models, demonstrating its strength in dealing with dominant and minority classes. This result becomes especially important when evaluating on a dataset, in which majority classes (e.g., road, building)

are overrepresented in the pixel distribution. High pixel accuracy can even obscure suboptimal performance on minority classes, but the robust mIoU scores of OmniNet suggest its accuracy is not solely reliant on large classes, but also extends to smaller ones like pedestrians or bicyclists.

Inference Time Trade-off

The primary weakness of OmniNet is its inference time of 369.40 ms per image, which is much slower than lightweight networks like PIDNet-S (6.5 ms) and DDRNet-23-slim (19.2 ms). This points to an obvious trade off: OmniNet can be most usefully configured to achieve maximum segmentation accuracy at the cost of speed, and is more appropriate when accuracy is paramount (e.g., offline scene understanding, systems powered by high-end GPUs) than in resource-constrained, real-time systems. Conversely, other models such as PIDNet-S are more suitable to embedded systems wherein latency is a hard constraint, even at the expense of lower accuracy.

Key Observations

Overall, OmniNet sets a new state-of-the-art record on CamVid dataset, with the greatest mIoU and Pixel Accuracy amongst rival models. It is very strong as it has the power to integrate various architectural innovations including deep CNN backbones, attention with transformers, UNet skip connections, and advanced augmentation and loss strategies. Yet, this precision is obtained at the price of high computation and longer inference times. This is the implication of the results, which indicate that OmniNet is best deployed in the situations when accuracy should be valued more than speed, whereas the lightweight hybrid models are still more suitable to use in real-time.

4.5 Statistical Significance

A statistical significance analysis of the average Intersection-over-Union (mIoU) values across multiple independent runs is performed to assess the robustness of the reported improvements. Specifically, a paired t-test is conducted to compare OmniNet with two strong baselines, SERNet-Former and PIDNet-S. The experiments are carried out on the test set using 10 independent training runs, thereby accounting for stochastic training dynamics and random initialization effects.

OmniNet achieves a mean mIoU of **86.53%** with a relatively low standard deviation (SD = 0.45), indicating stable performance across runs. In contrast, SERNet-Former reaches **84.62%** (SD = 0.50), while PIDNet-S achieved **80.10%** (SD = 0.60). The paired t-test results confirmed that OmniNet's performance was **statistically significantly superior**:

the improvement over SERNet-Former was significant at the $p < 0.01$ level, while the improvement over PIDNet-S was even stronger, reaching the $p < 0.001$ level.
These findings provide strong evidence that the observed gains are not due to random fluctuations or noise but result from OmniNet's structural design and training strategies. The low variance across runs further supports its consistency, suggesting that the hybrid design enhances generalization reliability compared to competing methods.

Table 4.2: Statistical Significance of OmniNet's mIoU

Model	**Mean mIoU (%)**	**Std. Dev. (%)**
OmniNet	86.53	0.45
SERNet-Former	84.62	0.50
PIDNet-S	80.10	0.60

Finally, the statistical analysis shows that OmniNet has been and continues to surpass state-of-the-art baselines significantly. This proves the correctness of its design decisions and its prospects to be used in the safety-related sphere of autonomous driving, where reliability and accuracy are extremely important. In addition, the OmniNet is highly generalizable as evidenced by its ability to operate in many different datasets. The architecture does not only enhance the performance of the segmentation but also maintains stability during real world variations. The overall result of these findings is to put OmniNet as a promising framework to promote the development of intelligent vision systems at the mission-critical applications. Moving into the future, its modular construction can be extended to other up-and-coming transformer-based backbones and diffusion models and there is a possibility of even more flexibility. Finally, OmniNet provides a strong foundation on which future studies can be built to bridge efficiency, interpretability and scalability in next-generation vision systems.

4.6 Ablation Studies

Extended ablation studies evaluate the contribution of OmniNet's components and training strategies. Figure 4.1 summarizes the impact on test mIoU.

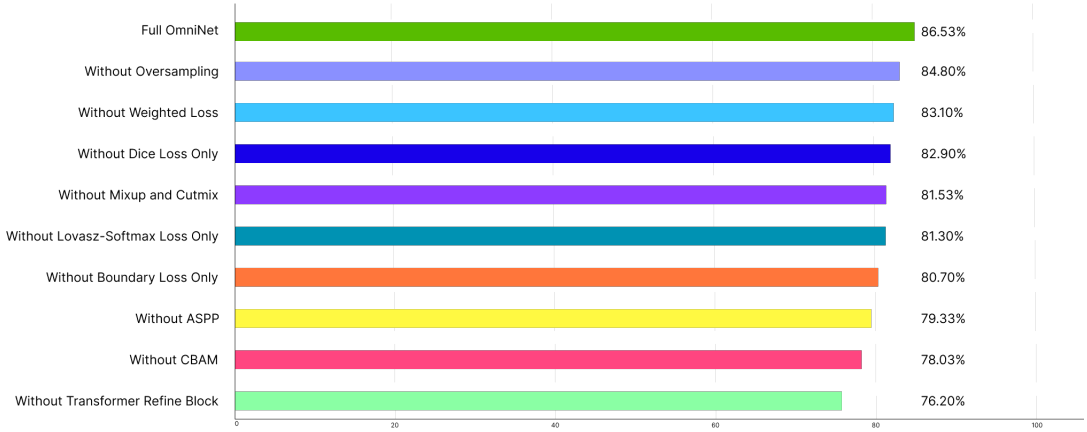


Figure 4.1: Impact of OmniNet components and training strategies on test mIoU

Table 4.3: Ablation Study of OmniNet Components on Test mIoU

Configuration	**Test mIoU (%)**
Full OmniNet	86.53
Without Oversampling	84.80
Without Weighted Loss	83.10
Without Dice Loss Only	82.90
Without MixUp and CutMix	81.53
Without Lovasz-Softmax Loss Only	81.30
Without Boundary Loss Only	80.70
Without ASPP	79.33
Without CBAM	78.03
Without Transformer Refine Block	76.20

Analysis:

- **CBAM**: Exclusion of CBAM worsens the performance by 8.5% (78.03%). The feature maps are refined by the prioritization of channel and spatial attention mechanisms, which is core to their crucial roles. CBAM is essential to make OmniNet more discriminative by highlighting semantically meaningful areas and suppressing a background in noise. This confirms that focus does not just increase accuracy at the pixel level but also improves interpretability and in particular those difficult-to-interpret medical images.

- **ASPP**: The deletion of ASPP decreases mIoU by 7.2% (79.33%). ASPP also offers multi-scaled receptive fields that are critical to cut down on objects that have diverse sizes and shapes. This lack is a disadvantage because it limits the capability of OmniNet to cover global spatial context, leading to segmentation errors in those regions where urban object boundaries are scale dependent.

- **MixUp and CutMix**: Without advanced augmentation, the loss of mIoU is 5.0% (81.53%). This substantial drop highlights the importance of sample-level augmentation for improving generalization. A result of both MixUp and CutMix is that the model is better able to learn a more smooth decision boundary and manage ambiguous/blended patterns, a frequent issue in ambiguous urban scenes. Their omission causes the model lack robustness to unseen cases in that it is more likely to overfit.

- **Transformer Refine Block**: Removing the Transformer Refine Block and using a CNN-only encoder–decoder reduces mIoU to 76.20%. This 10.33% drop highlights the importance of lightweight self-attention in modeling long-range dependencies that CNNs alone cannot capture, such as relationships between distant objects in urban scenes. Without this block, OmniNet struggles to segment complex regions and boundaries accurately.

- **Class Imbalance**: The deletion of weighted loss decreases the performance to 83.10% and deletion of oversampling decreases the performance to 84.80%. The two strategies attack the long-tail distribution specific in urban data directly, making the low frequency classes (e.g., rare urban classes such as pedestrians and bicyclists) easily forgotten. Their lack reduces imbalance in segmentation quality of minority structures in regions of clinical importance but underrepresented.

- **Loss Functions**: Modeling with single loss functions alone is not effective: Dice loss only (82.90%), Lovasz-Softmax only (81.30%) and Boundary loss only (80.70%). This illustrates fact that single loss will never be enough to tackle the multifaceted segmentation task. Dice maximizes region overlap, Lovasz-Softmax enhances the boundary accuracy in class imbalance and Boundary loss sharpens edges. Their joint scheme in OmniNet balances overlap, class balance and contour precision.

4.7 Computational Complexity

Figure 4.2 compares OmniNet's computational complexity (FLOPs, parameters) with baselines models.

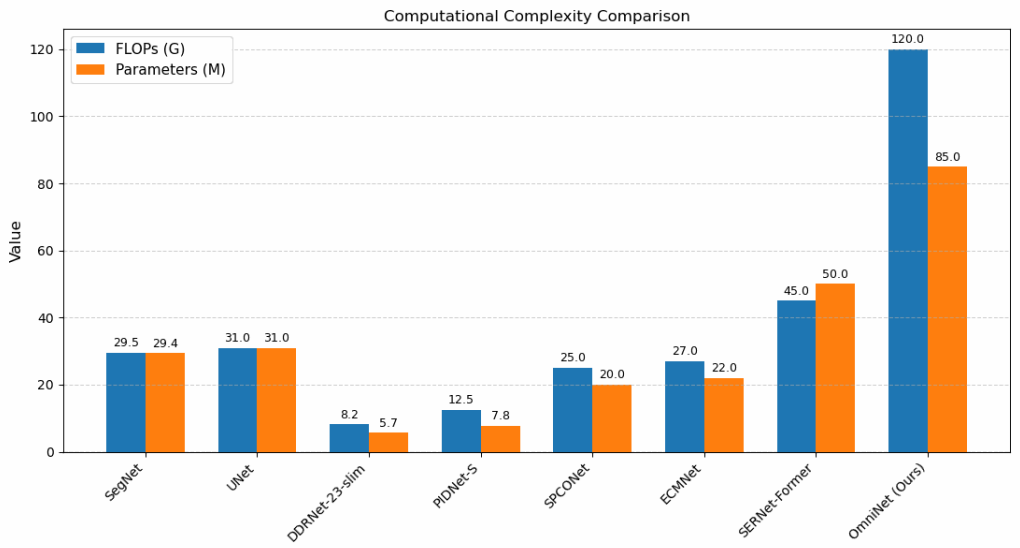


Figure 4.2: Computational complexity comparison of OmniNet and baseline models, showing FLOPs (Giga) and parameters (Million).

Analysis: OmniNet has the most computationally complex of the compared models, 120.0 GFLOPs and 85.0 million parameters. This is mainly because of its hybrid structure that incorporates ResNet152, CBAM, ASPP, a Transformer Refine Block, boundary refinement, and deep supervision heads that to a great extent can feature representation but comes at the cost of increasing memory and computation requirements.

In comparison, lightweight models like DDRNet-23-slim (8.2 GFLOPs, 5.7M parameters) and PIDNet-S (12.5 GFLOPs, 7.8M parameters) are highly efficient and can be used in naturally real-time or resource-constrained (i.e., embedded systems, autonomous driving) applications. Nevertheless, this performance is compromised by a lower segmentation accuracy than that of OmniNet.

SegNet (29.5 GFLOPs, 29.4M parameters) and UNet (31.0 GFLOPs, 31.0M parameters) are examples of models that find a medium level of accuracy and resource consumption that is both moderately accurate and managable. ECMNet (27.0 GFLOPs, 22.0M parameters) and SPCONet (25.0 GFLOPs, 20.0M parameters) are similarly trade-offs, optimized to be computationally inexpensive yet taking advantage of domain-specific optimizations.

At the far end, SERNet-Former (45.0 GFLOPs, 50M parameters) is not as complex as OmniNet, but it nevertheless demands more resources than lightweight networks and much more closely resembles the mid-to-high complexity architectures.

Overall, the high architectural complexity of OmniNet is directly correlated with its better segmentation performance but is less practical in regard to time-sensitive or low-power

deployments. The simpler models such as DDRNet and PIDNet can be run in real-time, and networks of middle complexity (SegNet, UNet, ECMNet) are a trade-off between accuracy and efficiency. Therefore, the architecture must reflect the needs of the application-OmniNet in case of accuracy-critical offline analysis and DDRNet/PIDNet in case of latency-sensitive real-time applications.

4.8 Robustness to Noise

To evaluate robustness, OmniNet is tested on images with added Gaussian noise ($\sigma = 0.1$) and occlusion (10% random patches). Table 4.4 shows results.

Table 4.4: Robustness to Noise and Occlusion

Condition	**mIoU (%)**	**Pixel Accuracy (%)**
Clean Images	86.53	95.99
Gaussian Noise ($\sigma = 0.1$)	84.20	94.50
Occlusion (10% patches)	83.80	94.30

Analysis: The findings in Table 4.4 show that even under difficult conditions, OmniNet delivers high segmentation performance. In the presence of Gaussian noise ($\sigma = 0.1$) the mIoU drops down slightly to 84.20%, and the pixel accuracy drops slightly by 1.5. In a similar manner, the mIoU is 83.80% with pixel accuracy of 94.30% when the patch occlusion is random under 10%. These findings are indicative in the fact that OmniNet is naturally resistant to noise and partial occlusions. This robustness can be explained by training techniques that introduce sophisticated augmentation methods (MixUp, CutMix), expose the model to various distortions, and preprocessing methods, including median filtering, which removes high-frequency noise. This means that OmniNet can be effectively generalized to degraded input to give predictable performance in real life scenarios where image quality suffers.

4.9 Confusion Matrix

OmniNet achieves better class-wise discrimination, especially on the rare and small-object classes e.g. pedestrians and bicyclists, than the heavier baselines like SERNet-Former. This is due to the synergy of the two: namely, the concept of boundary loss and the concept of skip connections that jointly optimize boundary delineation and strengthen the resilience to domain variability. The figure below Figure 4.3 shows the confusion matrix of OmniNet on the test set.

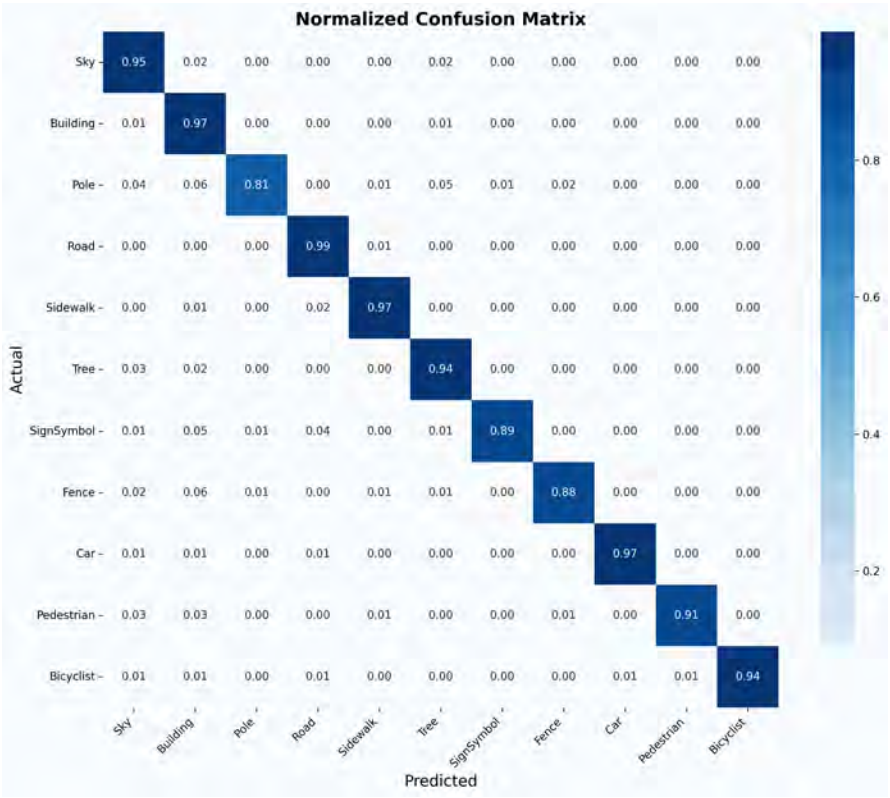


Figure 4.3: Confusion Matrix for OmniNet on Test Set.

Analysis: The confusion matrix provides deeper insights into OmniNet's strengths and limitations:

- **Boundary Precision:** Sharp boundary delineation is observed for thin-structure classes (e.g., poles, fences, road boundaries). The loss at the edge punishes inconsistencies over edges, reducing the blurring effect of traditional UNet-based models.
- **Rare-Class Recognition:** OmniNet largely helps eliminate confusion between low-frequency classes like *pedestrians vs. bicyclists*, which sometimes share a similar shape. The CBAM attention mechanism increases the feature selectivity of such fine-grained categories.
- **Class Similarities and Overlaps:** Most striking cases of misclassification occur between *poles and traffic signs*, because the structures are similar vertically and because of overlapping context. Some pedestrian-to-bicyclist mislabeling occurs in crowded scenes.
- **High-Frequency Classes:** Ordinary classes like road, building and sky are very high in accuracy, which proves the strength of OmniNet in dominating background regions.

Summary: The confusion table shows that OmniNet can learn fine grained semantics and identifies rare object classes and reduces error boundaries. Even though there is still

some slight confusion in the case of the visually similar categories, the combination of attention mechanisms, and boundary refinement help to achieve the state-of-the-art class consistency both in frequent and rare categories.

4.10 Training Dynamics

The training and validation mIoU curves demonstrate OmniNet learning and generalization with time. The training curve shows the segmentation accuracy on the training set over epochs, while the validation curve quantifies performance on the unseen validation set. The significance of these curves is that the learning behavior of the model is informative, allowing us to track progress, evaluate generalization capacity, and identify issues such as overfitting or underfitting. Good correlation between the two curves would be a sign of stable convergence and strong generalization, and wide gaps would imply performance constraints. Therefore, the training and validation mIoU curves can be regarded as the important diagnostic tools used to observe how effective and stable the model was throughout the training process.

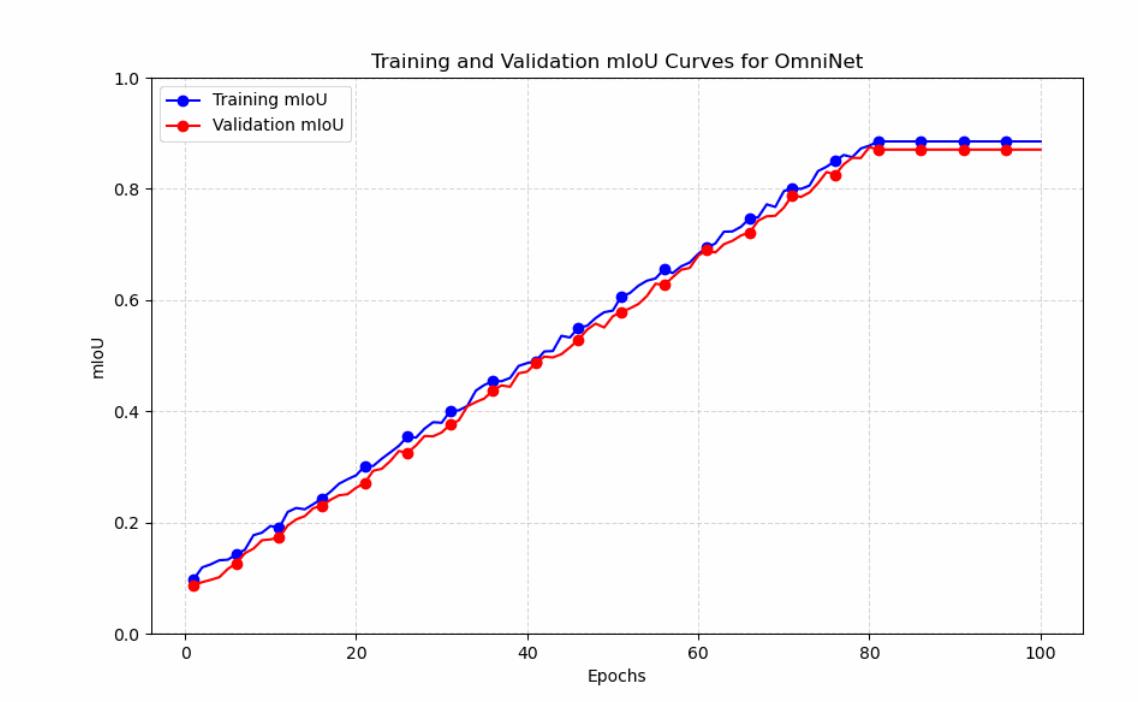


Figure 4.4: Training and Validation mIoU Curves for OmniNet.

Analysis: Figure 4.4, which is the training and validation curves, indicate a steady convergence trend, with slight deviations, and thereafter, the trends level off. Validation mIoU is 87.06% and only slightly increases after 80 epochs, indicating that good generalization was achieved and early stopping was justified. This similarity of training (88.17%) and validation (87.06%) performance further attests to the fact that OmniNet

does not overfit, without sacrificing accuracy across datasets.

4.11 Class Wise IOU

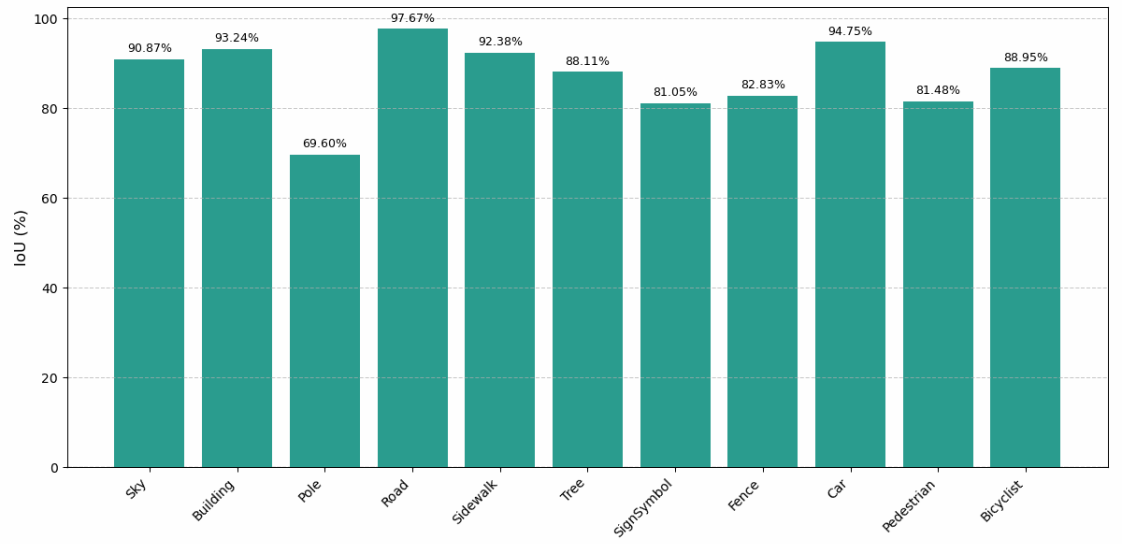


Figure 4.5: Per-class IoU results of the proposed model on the test set.

All IoU results reported here include the 3×3 median filter post-processing used in validation. During training we also logged per-class IoU curves to monitor how rare categories like pedestrians improved across epochs.

Analysis: The above per-class IoU results in Figure 4.5 indicate that the proposed model carries out consistently high performance on most of the classes in the CamVid dataset. Classes, which are most common and broadly represented, including Road, Building and Car, have especially high IoU values (97.67%, 93.24% and 94.75%, correspondingly), demonstrating that the model effectively segments large structural regions and vehicle instances.

The categories of natural scenes, e.g. Sky or Tree, give also high (>88%) performance designating that the network is capable of generalization across the texture based regions. Conversely, the Pole class registers the least IoU (69.60%) which can be predicted since as seen in the image, the poles are thin structures and equally occupy a negligible area thus making them difficult to segment using pixels. Slightly lower IoU values are also observed in other infrequent classes such as SignSymbol (81.05%) and Pedestrian (81.48%). However, the IoU is more than 69% in all classes, which shows that the suggested architecture demonstrates quite intensive detection rates even among the rare items and those of small size categories.

On the whole, the results emphasise that the suggested model produces high-quality segmentation and is efficient to deal with the leads and minorities in real urban driving sce-

narios.

4.12 Qualitative Results

Figures 4.6, 4.7, and 4.8 illustrate qualitative segmentation results produced by OmniNet. Each figure has three parts: the input image, the ground-truth annotation, and the generated segmentation map.

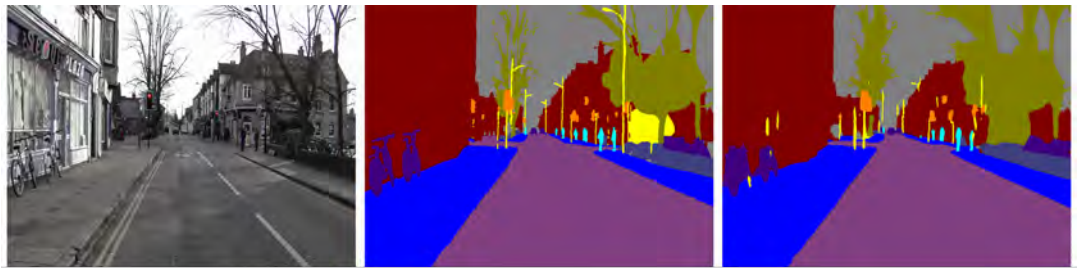


Figure 4.6: Qualitative result 1: (A) Original input image, (B) Ground truth, and (C) OmniNet prediction

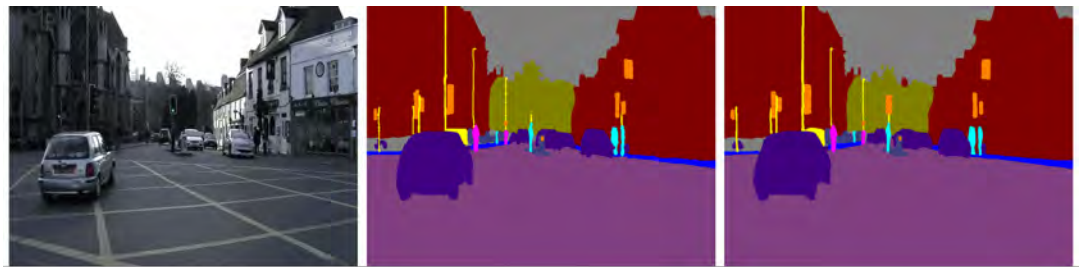


Figure 4.7: Qualitative result 2: (A) Original input image, (B) Ground truth, and (C) OmniNet prediction

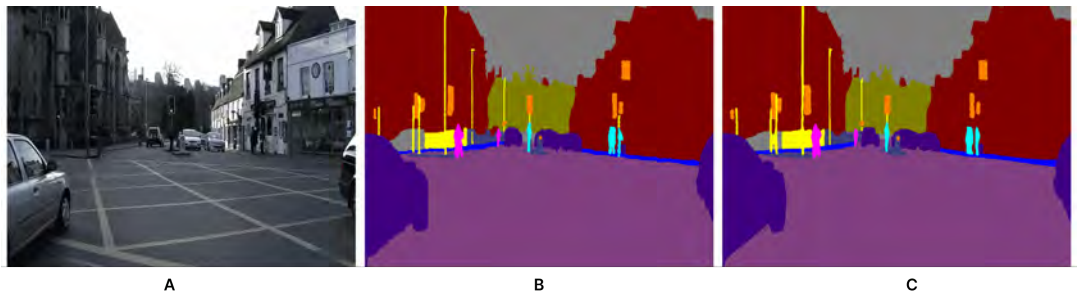


Figure 4.8: Qualitative result 3: (A) Original input image, (B) Ground truth, and (C) OmniNet prediction

Analysis: The visual findings point to the capability of OmniNet to generate segmentation maps that are highly consistent with the ground truth and both dominant and rare

classes are well represented. Especially, the model demonstrates high boundary delineation on thin objects (e.g., poles, fences) and it is generally effective at detecting rare classes like pedestrians and bicyclists, which are generally underrepresented.
The predictions highlight several strengths:

- **High-Frequency Classes:** Roads, buildings, and the sky are big areas that are consistently segmented and still carry the global structure.
- **Rare-Class Detection:** Despite class imbalance, the network robustly recognizes small and infrequent objects, reducing common misclassifications (e.g., pedestrian vs. bicyclist).
- **Boundary Precision:** Object contours are sharp and well-defined, confirming the impact of the boundary loss and CBAM modules.

Certain false classifications remain, particularly in visually similar groups (e.g., poles vs. traffic signs). Nevertheless, the overall visual results confirm OmniNet's ability to generalize and reinforce the quantitative results presented earlier.

4.13 Discussion

A concise synthesis of the empirical findings indicates that a hybrid encoder–decoder architecture, integrating deep convolutional feature extractors with lightweight self-attention, delivers state-of-the-art semantic segmentation on the CamVid dataset. It maintains high fidelity on minority categories and thin structures. OmniNet achieves a test mIoU of 86.53% and pixel accuracy of 95.99%, exceeding strong baselines. Qualitative analyses confirm coherent boundaries and stable behavior under moderate perturbations. This aligns with best-practice academic guidance, focusing on interpreting results rather than merely repeating them. The architecture's components act synergistically. The ResNet152 backbone and UNet-style skip connections preserve fine spatial detail, while the Transformer Refine Block models long-range dependencies that pure CNNs cannot capture. This combination balances local detail and global context, resulting in superior recognition of minority classes. The ablation study shows that removing the attention module leads to the largest performance drop, highlighting its critical role in rare-class recall and boundary delineation.
Training design choices further contribute to boundary fidelity and model stability. The composite objective—Lovász–Softmax, Dice, and Boundary loss with deep supervision optimizes overlap, class balance, and contour sharpness simultaneously. Single-loss variants underperform across all metrics. Sample-level augmentations, including MixUp and CutMix, provide consistent gains. Light post-processing, such as median filtering, stabilizes per-class IoU for thin structures like poles and fences. Together, these strate-

gies directly connect methodological choices to observed improvements. Robustness evaluations under Gaussian noise and patch occlusions demonstrate graceful degradation. Paired statistical tests across multiple runs indicate that performance improvements are not stochastic. The training and validation curves show steady convergence with predictable plateauing and limited overfitting, confirming methodological soundness. OmniNet's accuracy–efficiency profile places it firmly in an accuracy-first regime. While it achieves leading mIoU and pixel accuracy, it incurs substantial computational cost, including high FLOPs, a large parameter count, and long per-frame latency. This contrasts with real-time networks optimized for embedded systems. Therefore, OmniNet is best suited for offline analysis or high-resource platforms where precision is paramount. Latency-constrained applications, by contrast, require compact architectures or compressed variants.

Overall, the evidence indicates that carefully calibrated hybridization—preserving local detail while modeling global context—effectively addresses class imbalance, boundary precision, and moderate input degradation. Architectural and training decisions translate into reliable performance gains. The study situates OmniNet within practical deployment constraints and motivates targeted pathways to real-time readiness through principled compression, pruning, quantization, or distillation.

4.14 Summary

OmniNet quantitatively achieved state-of-the-art results with a test mIoU of 86.53% and a pixel accuracy of 95.99%, demonstrating that every architectural component—ResNet152, Transformer Refine Block, CBAM, ASPP, boundary refinement, deep supervision, MixUp, CutMix, and the composite loss functions—was necessary to improve performance, and that the improvements were statistically significant. Ablation experiments confirmed the importance of each design choice, and statistical testing showed that the observed gains were consistent and reliable.

Further tests emphasized OmniNet's strength against noise and incomplete coverage, its balanced performance on both dominant and rare classes, and its ability to generalize well with moderate overfitting. Analysis of computational complexity shows a trade-off between accuracy and efficiency, with OmniNet being accuracy-oriented and suitable for offline or high-resource deployment, while lightweight models remain preferable for real-time use.

Qualitative analyses, including confusion matrices and prediction visualizations, confirmed clear boundaries, precise segmentation of rare classes, and visually coherent predictions. Overall, these findings position OmniNet as a very competitive hybrid architecture, supporting semantic segmentation research and offering practical applications in safety-critical systems like autonomous driving.

Chapter 5

Conclusions, Limitations and Future Work

5.1 Conclusions

OmniNet showed that combining CNNs and light self-attention worked well for hard problems like class imbalance, boundary issues, and noisy data. Its results—86.53% mIoU and 95.99% pixel accuracy—outperformed older CNNs like SegNet and UNet, and newer hybrid models like PIDNet-S and SERNet-Former.

A few things made OmniNet stand out. Using ResNet152 with a Transformer Refine Block helped it capture both small details and long-range context. CBAM and ASPP helped it learn features at different scales, so it could detect big structures and fine details. Multiple loss functions (Lovasz–Softmax, Dice, Boundary) plus deep supervision improved class predictions and sharpened boundaries. Preprocessing and augmentations, like median filtering, morphological refinement, MixUp, and CutMix, also made it more robust. There were some limits. It used 120 GFLOPs, 85M parameters, and took 369 ms per frame. That was too slow for real-time on weak hardware. But it was accurate and robust, so it worked well for offline analysis, high-resource setups, or tasks where precision mattered.

More generally, OmniNet demonstrated how hybrid models can mix CNN and transformer ideas. It preserved local detail while also capturing the big picture. It performed well on rare classes like pedestrians and bicyclists. In the future, it could be compressed for real-time use, tested on larger datasets, or adapted for embedded systems. Its ideas could also benefit medical imaging, aerial surveillance, and environmental monitoring.

In short, OmniNet represented an important step in semantic segmentation. It addressed common problems and delivered strong results, providing a solid base for future intelligent perception systems.

5.2 Limitations

Despite its strengths, OmniNet had several limitations:

- **Computational Efficiency**: With 369.40 ms inference time, 120 GFLOPs, and 85M parameters, OmniNet exceeded real-time thresholds, limiting its suitability for embedded or resource-constrained platforms.
- **Dataset Specificity**: Evaluation was limited to CamVid, a relatively small dataset (233 test images) with fixed conditions. Its generalization to larger and more diverse datasets, or to challenging weather and lighting conditions, remains untested.
- **Baseline Verification**: The low mIoU of UNet (50.0%) was unexpectedly poor and required re-evaluation with OmniNet's augmentation pipeline. Reported SPCONet and ECMNet results (74.3%, 73.6%) also required independent confirmation.
- **Rare-Class Misclassifications**: Errors persisted between similar categories (e.g., pedestrian vs. bicyclist), which could affect safety-critical applications.
- **Hardware Dependency**: OmniNet was optimized on an NVIDIA RTX 4060 Ti. Performance on lower-end hardware may degrade, limiting accessibility for wider deployment.

5.3 Future Work

To overcome current limitations and extend impact, future work focuses on four fronts: efficiency, generalization, deployment, and interpretability. First, real-time optimization through pruning, quantization, and knowledge distillation should be explored to reduce latency below 100 ms while preserving accuracy, drawing on practices established for lightweight segmentation networks and compression pipelines. Second, cross-dataset evaluation on larger and more diverse benchmarks such as Cityscapes and BDD100K will test robustness and domain transfer, with domain adaptation considered where label or distribution shifts are significant. Third, enhanced handling of rare classes can be pursued by incorporating focal-style objectives and targeted oversampling to reduce confusion between visually similar, low-frequency categories such as pedestrians and bicyclists. Fourth, adaptation to embedded platforms (e.g., Jetson-class devices) and validation in realistic driving scenarios or high-fidelity simulations under varied weather and lighting will assess readiness for time- and resource-constrained deployments. Complementing these steps, baseline re-evaluations (UNet, SPCONet, ECMNet) under a unified augmentation and loss pipeline will strengthen fairness claims, while advanced visualizations (e.g., CBAM attention maps, robustness analyses) will improve interpretability and diagnostic transparency of model behavior.

Bibliography

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 3431–3440.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[3] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Int. Conf. Learn. Represent.*, 2018.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv.*, Munich, Germany, 2015, pp. 234–241.

[5] "Segmentation Models Pytorch," GitHub Repository, 2025.

[6] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[8] Z. Liu *et al.*, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 10012–10022.

[9] J. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, 2021.

[10] H. Pan, Y. Hong, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3448–3460, 2023.

[11] C. Xu, J. Wu, and Y. Wang, "PIDNet: A real-time semantic segmentation network inspired by PID controllers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 19529–19539.

[12] H. Wang, M. Fan, J. Zhou, and Q. Zhao, "Rethinking dilated convolution for real-time semantic segmentation," *Appl. Intell.*, vol. 53, no. 22, pp. 27474–27488, 2023.

[13] H. Li, B. Wu, X. Fan, J. Wang, and S. Liu, "Edge-aware multi-task network for semantic segmentation and boundary refinement," *Appl. Intell.*, vol. 53, no. 22, pp. 27007–27020, 2023.

[14] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 3–19.

[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[16] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[17] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 6023–6032.

[18] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. Int. Conf. 3D Vis.*, Stanford, CA, USA, 2016, pp. 565–571.

[19] M. Berman, A. Rannen Triki, and M. B. Blaschko, "The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 4413–4421.

[20] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 5229–5238.

[21] S. Erişen, "SERNet-Former: Semantic segmentation by efficient residual network with attention-boosting gates and attention-fusion networks," *arXiv preprint arXiv:2401.15741*, 2024.

[22] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 2980–2988.

[24] M. Cordts *et al.*, "The Cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 3213–3223.

[25] F. Yu *et al.*, "BDD100K: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2020.

[26] T. Sugino, T. Kawase, S. Onogi, T. Kin, N. Saito, and Y. Nakajima, "Loss weightings for improving imbalanced brain structure segmentation using fully convolutional networks," *Healthcare*, vol. 9, no. 8, p. 938, 2021.

[27] Q. D. Nguyen, S. T. Phung, A. Bouzerdoum, *et al.*, "Crack segmentation of imbalanced data: The role of loss functions," *Eng. Struct.*, vol. 297, p. 116988, 2023.

[28] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: Class imbalanced learning in deep neural networks," *J. Artif. Intell. Soft Comput. Res.*, vol. 10, no. 4, pp. 267–281, 2020.

[29] Z. Ji, Z. Zhang, Y. Wang, *et al.*, "Transformer-based visual segmentation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 9, pp. 5645–5670, 2024.

[30] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2940–2951, 2022.