Enhancing Semantic Segmentation

Architectural Innovations and Strategies for Label-efficient Learning

Tharrengini Suresh

Electrical and Computer Engineering

Lakehead University, Thunder Bay, Ontario

A thesis submitted to Lakehead University in partial fulfillment
of the requirements for the Master of Science degree
in Electrical and Computer Engineering

Thesis Committee Members

The members listed below served on the Examining Committee for this thesis:

Supervisor:	Dr. Thangarajah Akilan Department of Software Engineering.
Committee Members:	Dr. Abdulsalam Yassine Department of Software Engineering.
	Dr. Saad Bin Ahmed Department of Computer Science.
Session Chair:	Dr. Yushi Zhou Department of Electrical and Computer Engineering.

Declaration of Co-Authorship / Publications

I. Co-Authorship Declaration

I hereby declare that this dissertation includes material resulting from research publications completed under the supervision of Dr. Thangarajah Akilan (Chapters 3 and 4).

In all other parts of this dissertation, I am the primary author, having undertaken the main responsibilities, including idea generation, experimental design, data analysis, interpretation, and writing. The contributions of the co-authors in these instances were limited to proofreading and technical guidance.

I am fully aware of the Lakehead University Policy on Authorship and affirm that I have properly acknowledged the contributions of other researchers to this dissertation. Additionally, I have obtained permission from each co-author of the respective conference publications mentioned in Section II on page iii to include relevant content in this dissertation.

With these clarifications, I certify that this dissertation and the research it encompasses are my original work.

II. Declaration of Previous Publications

This thesis incorporates the content of two original research papers, which have either been previously published or awaiting acceptance in academic conferences or journals. The details are as follows:

Chapter	Publication title/full citation	Status
Chapter 3	**T. Suresh** *et al.*, "ECASeg: Enhancing Semantic Segmentation with Edge Context and Attention Strategy," in *2024 Springer Nature RTIP2R: Recent Trends in Image Processing and Pattern Recognition*, 2024.	In press
Chapter 4	**T. Suresh** *et al.*, "SS-DeepSeg: An Efficient DeepLab with Smart Scaling for Robust Semantic Segmentation," in *2025 IEEE International Symposium on Industrial Electronics*, 2025.	Accepted

III. General

I declare that, to the best of my knowledge, this thesis does not infringe on any copyrights or violate proprietary rights. All ideas, techniques, quotations, or other materials derived from the work of others, whether published or unpublished, are fully acknowledged in accordance with standard referencing practices. Additionally, where copyrighted material exceeds the limits of fair dealing as defined by the Canada Copyright Act, permission has been obtained. This is a true copy of my thesis, including all final revisions as approved by my thesis committee and the Graduate Studies office. This thesis has not been submitted for a higher degree at any other university or institution.

Tharrengini Suresh *14/05/2025*

Acknowledgements

I wish to express my sincere appreciation and gratitude to Dr. Thangarajah Akilan for his unwavering support, encouragement, and invaluable advice throughout my academic journey. His dedication to impactful research and his enduring optimism have been a constant source of inspiration, providing both the guidance and motivation that shaped this work. His expertise and thoughtful insights were instrumental in refining and elevating this thesis.

I am grateful to my colleagues at the Vision Intelligence and Data Lab for their insightful discussions that greatly aided me in overcoming key research challenges.

Sincere thanks to my thesis committee members: Dr. Abdulsalam Yassine, Dr. Yong Deng, and Dr. Yushi Zhou, for their invaluable feedback and guidance, which were instrumental to the successful completion of this research.

I am also deeply grateful to my family for their unwavering love and support. Their encouragement sustained me through every challenge.

Lastly, I acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), DGECR/416-2022. This research was partially enabled by the support of the computing infrastructure provided by the Digital Research Alliance of Canada (https://alliancecan.ca/en).

Dedication

This thesis is dedicated to my family. Every milestone I achieve is a reflection of their encouragement, sacrifice, and belief in me.

Abstract

Semantic segmentation is a fundamental component of modern computer vision applications. Although supervised learning models have achieved state-of-the-art performance in this domain, they rely heavily on large volumes of labeled data, which is an expensive and time-consuming requirement. Thus, this research aims to develop enhanced supervised semantic segmentation models that balance accuracy and data efficiency for visual perception tasks in autonomous driving environments. To achieve this, the thesis is organized into two distinct phases. The first phase investigates a dual-network architecture, in which an auxiliary boundary detection network is incorporated into the primary segmentation framework to mitigate pixelation artifacts at object boundaries in multi-class segmentation of complex scenes. The experimental findings demonstrate the importance of designing unified segmentation models that take advantage of architectural enhancements capable of extracting richer feature representations for improved performance. The second phase leverages insights from the previous stage and focuses on the development of an efficient deep learning model with attention mechanisms and multi-scale feature refinement. The proposed method introduces a novel depth-wise, point-wise feature pyramid module that extracts information-rich spatio-semantic context from early and deep feature representations, improving model efficacy. Exhaustive experimental studies conducted on widely used benchmark datasets validate the effectiveness of the proposed models, which achieve competitive performance while offering improved computational efficiency relative to baseline approaches. The findings highlight that strategically balancing resource utilization with architectural innovation can yield strong performance while minimizing annotation demands and environmental impact. This research sets a valuable precedent for building competitive, resource-aware vision systems suited to constrained application settings.

Table of Contents

Thesis Committee Members i

Declaration of Co-Authorship / Publications ii

Acknowledgements iv

Dedication v

Abstract vi

List of Figures xi

List of Tables xiii

List of Key Acronyms xiv

1 Introduction **1**

1.1 Thesis Overview 1

1.2 Motivation 2

1.2.1 Quest for High Efficiency and Performance in Semantic Segmentation . . . 2

1.2.2 Limitations in the Existing Research 2

1.3 Taxonomy of Semantic Segmentation Methods 3

1.4 Technical Approach 4

1.5 Machine Learning for Computer Vision Automation 5

1.5.1 The Foundation of ML Automation 5

1.5.2 The Three Basic Computer Vision Tasks 6

1.6 Deep Learning-driven Computer Vision 8

1.6.1 Unsupervised Learning 9

1.6.2 Self Supervised Learning 9

1.6.3 Semi and Weakly Supervised Learning 10

1.6.4 Supervised Learning 10

1.7 Supervised DL Architectures for Semantic Segmentation 10

1.7.1 2D Convolutional Architectures 11

1.7.2 Encoder-Decoder Architectures 11

1.7.3 Transformer Architectures 12

1.8 Loss Functions and Evaluation Metrics 12

1.8.1 Loss Functions 13

1.8.2 Evaluation Metrics 14

1.9 Thesis Contribution 15

2 Literature Review **17**

2.1 Chapter Overview 17

2.1.1 The Shift from Conventional to Deep Learning Models 18

2.1.2 Image Segmentation Methods: A Dive into Semantic Segmentation 19

2.1.3 The Developments Trends in Semantic Segmentation 21

2.2 Chapter Summary 27

3 Developing a Dual-Network Multi-Class Segmentation Model **28**

3.1 Overview 28

3.2 The Challenge in Urban Scene Segmentation 29

3.2.1 Edge Context in Semantic Segmentation 30

3.2.2 Attention Mechanisms in Semantic Segmentation 31

3.2.3 Atrous Spatial Pyramid Pooling (ASPP) 32

3.3 Methodology 32

3.3.1 Model Architectures 34

3.3.2 Network Justification 41

3.3.3 Environment 43

3.3.4 Dataset 43

3.3.5 Quantitative Analysis 44

3.3.6 Qualitative Analysis 45

3.4 Chapter Summary and Findings 45

4 An Efficient Network with Smart Scaling 48

4.1 Overview 48

4.2 Feature Pyramid Networks 49

4.3 Advanced Attention Mechanisms 50

4.4 Methodology 52

4.4.1 The Backbone Network 52

4.4.2 The Improved ASPP Network 53

4.4.3 The Depth-wise Point-wise Feature Pyramid (DPFP) Module 53

4.4.4 A Smart Scaling for Multi-level Feature Learning 54

4.4.5 Environment 55

4.4.6 Datasets 55

4.4.7 Data Preprocessing: An Online Data Augmentation Strategy 57

4.4.8 Model Training 57

4.4.9 Comparative Study 58

4.4.10 Qualitative Analysis 62

4.5 Chapter Summary 65

5 Conclusion 66

Appendix 78

A Permission to Reprint 78

A.1 Elsevier Permission to Reprint 78

A.2 IEEE Permission to Reprint 78

B Source Code 79

List of Figures

1.1 A categorization of semantic segmentation: traditional vs deep learning approaches. 3

1.2 An overview of basic computer vision tasks with different objectives. 7

2.1 Image segmentation types: (a) Instance segmentation w/t per-object mask and class label; (c) Semantic segmentation w/t per-pixel class labels, and (b) Panoptic segmentation w/t per-pixel class and instance-level labels. Image credit: https://ieeexplore.ieee.org/document/8953237. 20

2.2 (a) - (b) Comparison of standard and dilated 3×3 convolutions on a 5×5 feature map. Both use 9 sampling points, but dilation increases the receptive field. (c) Residual block. 22

2.3 Examples of data augmentations applied to an image. 25

3.1 Proposed architecture and network details describing the interactions between the networks, as well as the overall pipeline from input to segmentation maps. Top: SNet, Segmentation Network. Bottom: ENet, Edge Network. 33

3.2 Training progress of the ENet for 100 epochs. 35

3.3 Training progress of the SNet for 150 epochs. 39

3.4 The feature fusion and skip connection locations employed between the two subnets. 41

3.5 Prediction samples of the model on CamVid test. Row # 1 - 3: Input images, ground truths, and predicted segmentation maps. The image IDs are provided for reproducibility. 46

4.1 An architectural overview of the proposed model. 52

4.2 Design specifications of the Depth-wise Point-wise Feature Pyramid (DPFP) module. 53

4.3 A sample input image and its corresponding ground truth mask across datasets used in this work. 56

4.4 Training progress plots of the ablation experiments on Cityscapes miniset. 58

4.5 Training progress plot of the proposed podel for 100 epochs on Cityscapes. 59

4.6 Sample of predictions on Cityscapes validation set. Col. # 1 - 4: Input images, ground truths, baseline predictions, and proposed model predictions, respectively. . 62

4.7 Predictions on the CamVid Test set, IDs provided for reproducibility. 63

4.8 Proposed method's predictions on the LoveDA validation set. Image IDs provided for reproducibility. 64

List of Tables

1.1 The thesis road map consisting of two progressive stages 1

1.2 Key application domain of machine learning and their characteristics 6

2.1 A summary of key CNN-based models adapted for urban scene segmentation . . . 23

2.2 A summary of key Transformer-based models in urban scene segmentation 24

2.3 Literature summary of semi-supervised and unsupervised learning methods on Cityscapes 26

3.1 ENet: Encoding Sub-Network Architecture 34

3.2 Training Hyperparameter of the ENet 35

3.3 Layer-wise Connectivity Pattern of the Decoding Pathway of the ENet 36

3.4 SNet: ASPP Module Architecture 37

3.5 SNet: Encoding Sub-Network Architecture 38

3.6 Training Hyperparameter of the SNet 39

3.7 Layer-wise Connectivity Pattern of Decoding Pathway of the SNet 40

3.8 Sanity analysis of the feature fusion strategies used to build the proposed model. % change is w.r.t the baseline U-Net 42

3.9 Ablation study of modifications made to achieve the final proposed model, ECASeg 43

3.10 Summary of CamVid Database [1] 44

3.11 Quantitative analysis summary. The % change is calculated w.r.t test mIoU between the proposed model and the respective model in comparison 45

4.1 Train Hyperparameters of the Proposed Method on Cityscapes 57

4.2 Ablation Study on Cityscapes Miniset 59

4.3 Class-wise performance comparison in Cityscapes Validation Set 60

4.4 Quantitative analysis on Cityscapes. Change is calculated w.r.t mIoU of the baseline 60

4.5 Performance of various models on CamVid Test 61

4.6 Performance of various models on LoveDA Test 61

List of Key Acronyms

Acronym with Full Form	**Synopsis**
ADAM: Adaptive Moment Estimation	A gradient-based optimization algorithm for training deep learning models, combining the advantages of momentum and RMSProp to adapt learning rates for each parameter.
ASPP: Atrous Spatial Pyramid Pooling	A convolutional neural network module that captures multi-scale contextual information by applying parallel dilated (atrous) convolutions with different dilation rates and global average pooling, followed by feature fusion. Used in models (e.g. DeepLabv2, DeepLabv3, DeepLabv3+) to improve receptive field without increasing resolution or computational cost excessively.
BCE: Binary Cross-Entropy	A loss function for binary classification problems, minimizing the difference between predicted probabilities and actual binary label.
CCE: Categorical Cross-Entropy	A loss function for multi-class classification problems, minimizing the difference between the predicted probability distribution (typically from a softmax output) and the true one-hot encoded label.
CNN: Convolutional Neural Network	A specialized deep neural network architecture for processing structured data (e.g., images) that leverages convolutional layers to automatically learn spatial features and patterns at different levels of abstraction.

Acronym with Full Form	Synopsis
DeepLab:	A CNN architecture designed for semantic image segmentation that utilizes pre-trained CNN backbones and atrous (or dilated) convolutions for multi-scale feature extraction and a fully connected Conditional Random Field to refine the results. There are several variants (e.g., DeepLabv2, DeepLabv3, DeepLabv3+, etc.), each building upon their predecessors to overcome challenges in segmentation.
DS Conv: Depth-Wise Separable Convolution	An efficient alternative to standard convolution that combines depth-wise convolution and point-wise convolution in series. This factorization reduces computation and parameters compared to regular convolutions, making it ideal for lightweight models (e.g., MobileNet), as the depth-wise step preserves channel independence. In contrast, the point-wise step merges features, maintaining expressiveness with lower computational cost.
DL: Deep Learning	A subset of artificial intelligence that utilizes neural networks with multiple layers to analyze data and extract meaningful patterns, often applied in tasks such as computer vision, natural language processing, and time-series analysis.
DNN: Deep Neural Network	A type of neural network consisting of multiple hidden layers of interconnected neurons (deep network) that enable hierarchical feature transformation through successive nonlinear operations on sample data points.
GFLOPS: Giga Floating Point Operations Per Second	A unit that measures computing speed, quantifying a system's capability to perform floating-point arithmetic operations per second.
GPU: Graphics Processing Unit	A specialized processor optimized for parallel computation, widely used in deep learning and image processing tasks.
mIoU: Mean Intersection over Union	A commonly used metric for evaluating the performance of semantic segmentation models across multiple samples and object classes.

Acronym with Full Form	**Synopsis**
ML: Machine Learning	A subset of artificial intelligence that trains models to identify patterns and make predictions from data.
MLP: Multi-Layer Perceptron	A fully connected neural network architecture commonly used in supervised learning tasks.
ReLU: Rectified Linear Unit	A non-linear activation function commonly used in neural networks.
RGB: Red Green Blue	A color model used for representing visual data, where images are composed of three color channels (red, green, and blue).
U-Net:	A type of CNN originally developed for medical image segmentation tasks. Its symmetric U-shaped architecture encapsulates an encoder subnetwork that captures semantic information and a decoder subnetwork that enables precise localization, making it effective for pixel-wise segmentation in any domain.

Chapter 1

Introduction

1.1 Thesis Overview

Table 1.1: The thesis road map consisting of two progressive stages

Research Phase	Research Aim	Research Outcome
A Dual-Network Model	To explore the feasibility of two networks working in tandem to ameliorate performance, demonstrating how multi-model feature extraction can improve performance and emphasizing the need for a more robust lightweight approach.	Thesis Chapter 3, Publication #1 – Springer Nature RTIP2R 2024
An Efficient and Robust Model	To leverage insights from Phase 1 and advanced strategies to refine the supervised segmentation approach, achieving competitive performance without the added complexity of dual-network methods.	Thesis Chapter 4, Publication #2 – IEEE ISIE 2025

Table 1.1 outlines the research phases of the thesis. The phases are carried out pragmatically, first establishing a foundational understanding of supervised deep learning for semantic segmentation before progressing toward achieving the thesis's core objective: the development of an optimized lightweight model. Given the inherent complexity of semantic segmentation in computer vision, this phase examines existing models to elucidate key theoretical and practical aspects of the field. Building on the insights of the initial phase, the second phase introduces an efficient and sustainable segmentation framework designed to reinforce deep learning-based image segmentation principles, enhancing model efficiency through rigorously validated methodologies, thereby contributing robust and scalable solutions.

1.2 Motivation

1.2.1 Quest for High Efficiency and Performance in Semantic Segmentation

Semantic segmentation is a cornerstone in computer vision, enabling pixel-level understanding in critical applications, viz., medical imaging, autonomous driving, and satellite image analysis. Although supervised deep learning remains the gold standard for achieving high accuracy, its reliance on large-scale, meticulously labeled datasets poses significant challenges. In medical diagnostics, for instance, expert annotations of pathologies or anatomical structures are costly and time-consuming. Similarly, autonomous driving systems demand pixel-perfect labeling of diverse road elements—a process that is labor-intensive and prone to human error. As real-world applications increasingly require real-time processing and scalability, there is a pressing need to improve model efficiency without sacrificing segmentation quality. Current architectures often trade off computational complexity for marginal accuracy gains, limiting their deployment in resource-constrained environments. This motivates the exploration of supervised methods that optimize both structure (e.g., through lightweight yet expressive networks) and training efficiency (e.g., via improved data utilization or learning strategies). By refining these aspects, we can push the boundaries of what supervised segmentation can achieve, particularly in scenarios where labeled data is limited but high precision remains non-negotiable.

1.2.2 Limitations in the Existing Research

This thesis explores supervised learning approaches that strike a balance between computational efficiency and segmentation performance. By investigating architectural innovations and training methodologies tailored specifically for semantic segmentation, the aim is to advance models that maintain high accuracy but remain computationally tractable for real-world deployment. Such advancements are particularly crucial for applications like real-time segmentation, where both precision and speed are critical, or large-scale satellite imagery analysis, where processing efficiency directly impacts environmental monitoring capabilities.

This research is driven by the recognition that, despite the growing interest in semi-supervised and self-supervised methods, supervised learning remains a fundamental baseline. It provides reproducibility, well-delineated segmentation boundaries, and more direct optimization pathways. By exploring the potential of supervised approaches with limited resources, this study aims to establish new efficiency-performance trade-offs that can benefit both academic research and industrial applications.

1.3 Taxonomy of Semantic Segmentation Methods

Semantic segmentation algorithms are broadly divided into two categories: traditional feature-based classification and clustering approaches, and machine/deep learning (ML/DL) methods, as depicted in Fig. 1.1. Traditional methods are limited by the reliance on hand-engineered feature extraction methods that require extensive expertise and can be biased by the distribution of data. Moreover, complex scenes may contain objects with limited representations. As a result, deep learning-based methods have been popularized as they exhibit strong adaptability and generalization capabilities, allowing them to adapt to different scenarios and data characteristics. Moreover, in DL, the feature extraction process is automated and does not require domain expertise for optimization.

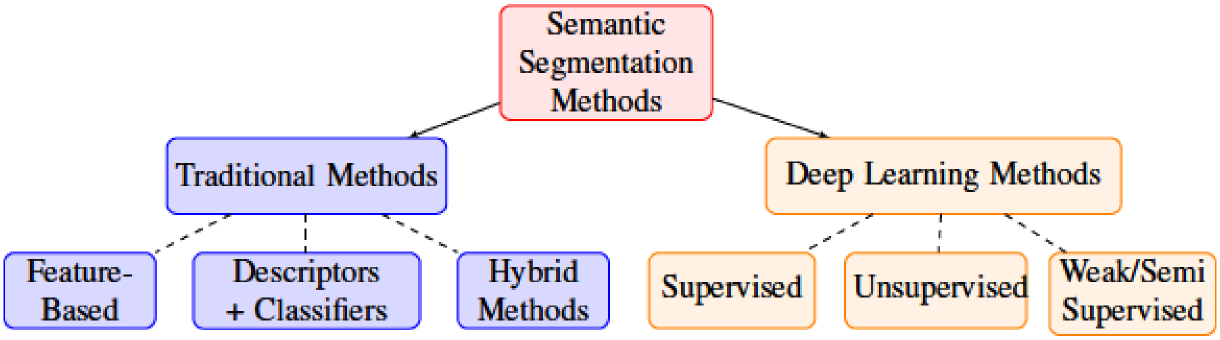


Figure 1.1: A categorization of semantic segmentation: traditional vs deep learning approaches.

1.4 Technical Approach

This thesis advances supervised semantic segmentation research by leveraging deep neural networks with attention mechanisms to develop practical and lightweight solutions for automated scene segmentation. The research is structured into two distinct phases, each building upon the insights and outcomes of the previous phase:

- **Phase 1: Exploring Dual-Network Architectures:** The first phase investigates the feasibility of a feature enrichment strategy via a dual-network structure. While the multi-model approach is not included in the final system, this exploration provided critical insights into addressing challenges associated with small and under-represented objects in complex environments. The findings from this phase directly informed the development of the optimized supervised learning framework presented in Chapter 4. This phase served as a methodological probe to assess and understand the challenges associated with multi-class semantic segmentation in complex environments like urban scene segmentation (cf. Chapter 3).

- **Phase 2: Refining and Evaluating a Lightweight Model:** The second phase builds upon the insights from Phase 1 that revealed explicit edge detail at multiple levels improved feature representations and recovery but at the cost of increased computational overhead due to the auxiliary network. This raised the question of whether multi-scale feature refinement could preserve and propagate edge detail effectively within a unified framework, driving the transition to a single, lightweight model. To this affect, this phase focuses on optimizing a supervised single-model architecture by integrating advanced techniques such as attention mechanisms, dilated separable convolutions, atrous spatial pyramid pooling, and feature pyramids. The refined model is rigorously tested on three benchmark datasets, CamVid, Cityscapes, and LoveDA, achieving competitive performance and better efficiency than current state-of-the-art methods. This phase demonstrates the feasibility of a streamlined approach, providing a practical and deployable solution tailored to real-world conditions while highlighting the potential of sustainable approaches (Chapter 4).

This structured approach tackles the shortcomings of current systems while establishing a strong base for future progress. By focusing on real-world applicability and scalability, it connects academic research with practical implementation. The following section outlines foundational concepts in machine learning, computer vision, and deep learning to contextualize the solutions proposed in this thesis.

1.5 Machine Learning for Computer Vision Automation

★ **Scope Clarification:** This thesis builds upon intersectional concepts in machine learning, computer vision, and deep learning. To maintain focus on the core contributions, condensed overviews of these foundational elements are presented rather than comprehensive explanations to offer the necessary context for understanding the work presented in this thesis.

1.5.1 The Foundation of ML Automation

Automation refers to the process of using systems, often powered by algorithms and data, to perform tasks with minimal or no human intervention. At the core of modern automation is ML, a scientific discipline that focuses on developing algorithms capable of learning patterns from data to solve specific problems or extract actionable insights. ML has been foundational for many automated systems across diverse fields, as it enables machines to improve performance adaptively based on data-driven feedback. ML tasks are typically categorized on the basis of the type of reasoning needed to understand and interpret input data. For instance, classification involves predicting discrete labels or categories based on image/text data, such as determining whether a news article is fake or not. A regression problem involves predicting continuous values based on learned relationships, like ascertaining blood sugar levels based on diet, exercise, age, and body mass index. Clustering focuses on grouping data points into 'clusters' based on similarity, which is useful in applications where a thematic grouping of items or objects is required. Association tasks involve mining frequent patterns by extracting statistically significant correlations, or simply put, identifying relationships between variables in datasets. For example, it is useful in market basket analysis to discover items frequently purchased together. Machine learning has evolved into a versatile toolset, and based on the nature of the input data and the specific end objectives, ML has given rise to several specialized subfields–each addressing distinct challenges, data modalities, and domain-specific nuances. These paradigms have matured into specialized research areas, tailored to process and learn from particular types of data for targeted outcomes. Table 1.2 summarizes major ML domains and their unique characteristics.

Table 1.2: Key application domain of machine learning and their characteristics

Domain	Type of Data	Process	Application
Natural Language Processing (NLP)	Text	Linguistic analysis and synthesis	Sentiment analysis, language translation
Computer Vision	Images, Videos	Visual data interpretation	Object detection, facial recognition, image segmentation
Speech Recognition and Processing	Audio / Speech	Spoken language interpretation	Voice-controlled systems, transcription, accessibility tools

This thesis addresses a specific task within computer vision (CV). To better understand its role and significance, an overview of various CV tasks is discussed here. These tasks differ primarily in their objectives: classifying, localizing, or segmenting the contents of an image.

1.5.2 The Three Basic Computer Vision Tasks

The computer vision paradigm has evolved from basic image classification to more complex tasks such as object detection and semantic segmentation, as illustrated in Fig. 1.2.

- **Image Classification:** It categorizes an entire image into predefined classes, for instance, determining whether an image contains a cat or a dog.

- **Object Detection:** It aims to identify objects within an image and localizes them using bounding boxes, like locating where a cat is present in an image.

- **Semantic Segmentation:** An extension of a classification and object detection problem, which involves identifying pixel regions within an image through pixel-wise classification. For example, segmenting the region of pixels corresponding to a cat in a picture.

Since image segmentation is the central focus of this thesis, it necessitates a deeper exploration of the intersection between machine learning and computer vision. To develop effective solutions for these tasks, various learning paradigms have been introduced—-each tailored to different levels of data availability, task complexity, and computational demands. Deep learning, a specialized branch of machine learning, plays a central role in this context. It enables automatic feature extraction and end-to-end prediction, significantly reducing the need for manual intervention and domain-specific feature engineering. The following section outlines the primary deep learning paradigms used in this domain, ranging from label-independent unsupervised approaches to label-dependent fully supervised methods, with intermediate strategies like semi-supervised and self-supervised learning offering a balance between the two.

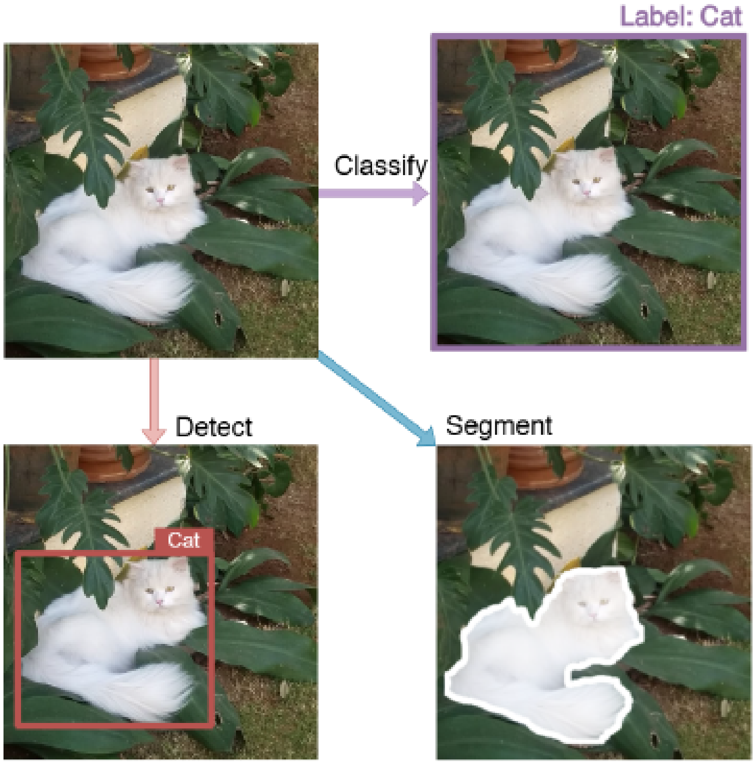


Figure 1.2: An overview of basic computer vision tasks with different objectives.

1.6 Deep Learning-driven Computer Vision

Deep Learning (DL) is a subset of machine learning that employs neural networks with multiple layers to model complex patterns in large datasets. Neural networks are inspired by the structure and functionality of brains, and these networks work on the principle of transforming raw sensor inputs (e.g., LiDAR, cameras) into progressively abstract representations, eliminating the need for manual feature engineering [2–4]. Segmentation models, a critical application of DL, are used for pixel- or point-wise scene parsing, which in the context of autonomous vehicles, involves partitioning road scenes into drivable areas, pedestrians, vehicles, and obstacles to inform decision-making for unmanned vehicles [5]. In remote sensing, it involves analyzing aerial imagery to categorize regions based on land cover classes such as buildings, vegetation, water bodies, barren land, etc., supporting tasks like urban planning, environmental monitoring, and disaster management [6, 7]. A key strength of DL in such applications is its ability to generalize patterns from data, even when correlations are non-intuitive to human designers. However, performance hinges on the quality of large, diverse datasets representative of real-world conditions. Insufficient data risks overfitting, where models fail in unseen scenarios (e.g., adverse weather, rare edge cases). For instance, a model trained primarily on urban daytime data may underperform in rural nighttime settings or fail to detect obscured pedestrians. This underscores the need for datasets encompassing varied geographies, lighting, and occlusion scenarios.

DL models are trained by minimizing loss functions that penalize misclassifications (e.g., false negatives). Through backpropagation and gradient descent, parameters are optimized iteratively. To mitigate overfitting, datasets are split into training and validation subsets, with the latter assessing generalization. However, robust validation performance alone cannot guarantee real-world safety without exposure to corner cases (e.g., jaywalking pedestrians, construction zones, dense cloud cover). With comprehensive datasets, DL architectures achieve state-of-the-art results in semantic segmentation. Their scalability, combined with techniques like hyperparameter tuning and multi-sensor fusion (e.g., camera-LiDAR), enables precise environmental perception—critical for the safety and reliability of autonomous systems [8, 9].

Despite these capabilities, the effectiveness of DL models remains closely tied to the availability and quality of annotated training data. Consequently, deep learning approaches are broadly categorized based on their dependence on labeled data. The following subsections outline these paradigms, ranging from fully label-independent to fully label-dependent, highlighting their differing strategies for tackling computer vision tasks.

1.6.1 Unsupervised Learning

Unsupervised learning involves training models on unlabeled datasets, allowing them to discover patterns, structures, or relationships within the data without the need for explicit labels [10, 11]. These methods leverage techniques such as clustering based on similarities, generative models, or domain adaptation to discover meaningful pixel-level representations. While unsupervised methods reduce annotation dependency, they typically achieve lower accuracy than supervised counterparts and face challenges in handling fine-grained class distinctions. In addition, the absence of labeled data poses challenges in evaluating unsupervised models, as there is no explicit ground truth to measure performance. However, they are still able to establish some results where only unlabeled data is available to uncover insights that guide further analysis.

1.6.2 Self Supervised Learning

Self-supervised learning paradigm aims to reduce dependency on manual annotations by leveraging unlabeled data to learn meaningful representations through pretext tasks [12]. Common approaches include contrastive learning [13], where models discriminate between similar and dissimilar image regions, and generative methods that reconstruct masked or perturbed input patches. Techniques like clustering-based pseudo-labeling and vision transformer-based pretraining further enhance feature learning. Recent advancements explore cross-modal supervision (e.g., text-image alignment [14] and dynamic pretext tasks to improve transferability. While self-supervised methods achieve competitive performance with limited labeled data, challenges remain in closing the gap with fully supervised models, particularly in fine-grained segmentation tasks.

1.6.3 Semi and Weakly Supervised Learning

Semi-supervised and weakly supervised approaches in semantic segmentation aim to reduce the reliance on large-scale pixel-level annotations [5, 15], by leveraging alternative forms of supervision, such as image-level labels, bounding boxes, or sparse annotations. These methods often employ techniques like consistency regularization, pseudo-labeling, or attention mechanisms to propagate supervision signals from limited labeled data to unlabeled or weakly labeled samples. While significantly reducing annotation costs, such approaches typically trade off some accuracy compared to fully supervised methods. Recent advancements explore self-training, contrastive learning, and multi-task frameworks to bridge this performance gap.

1.6.4 Supervised Learning

Supervised learning is fully reliant on labeled data, where models are trained on input samples, each paired with a known output. This approach is commonly used in segmentation systems to classify pixels in a scene based on labeled examples. The model learns patterns by minimizing the error between its predictions and the provided labels, enabling it to generalize to new, unseen data [3]. A significant challenge in supervised learning is the need for large, diverse, and high-quality labeled datasets. Annotating such datasets is labor-intensive and costly, particularly where expert knowledge is often required, like medical image segmentation [16–18]. Despite these challenges, supervised models excel in scenarios where comprehensive datasets are available, achieving high accuracy by associating new inputs with patterns learned during training. The following section highlights prominent supervised models for segmentation-related tasks.

1.7 Supervised DL Architectures for Semantic Segmentation

DL models can be used to address various domain-specific challenges. The following section highlights the most common architectures, particularly relevant to semantic segmentation solutions.

1.7.1 2D Convolutional Architectures

2D Convolutional Neural Networks (2D CNNs) are a specialized type of supervised deep learning models designed to analyze spatial information in images or video frames for tasks like semantic segmentation (e.g., FCN [2]). Unlike 3D CNNs, which process volumetric or temporal data, 2D CNNS focus on extracting hierarchical spatial information, making them effective for pixel-level classification tasks where contextual understanding within a single frame is critical. The core operation in a 2D CNN is the 2D convolution, where filters slide across the height and width of an image to generate feature maps. These features are progressively refined through successive convolution layers, pooling, and non-linear activations like ReLU, successively reducing spatial dimensionality while preserving discriminative information for precise pixel-wise predictions. Early layers encode low-level features like local patterns (e.g., edges, textures, etc.), while deeper layers aggregate global context and semantic features. Broadly, 2D CNNS–especially when adapted for segmentation–are considered the gold standard for spatial feature extraction, outperforming traditional computer vision methods in tasks requiring fine-grained localization.

1.7.2 Encoder-Decoder Architectures

Building on the strengths of 2D CNNs for hierarchical feature extraction, the encoder-decoder architecture enhances spatial recovery by adding a precise localization pathway. The process of hierarchical feature extraction inherently loses spatial resolution as the network progresses to deeper layers, which can be problematic for segmentation tasks that rely on precise localization at the pixel-level. Encoder-Decoder architectures solve this by pairing the 2D CNN encoder with a learnable upsampling decoder that employs transpose convolutions or interpolation to reconstruct spatial details from the abstract representations extracted by the encoder. One such model, is the U-Net [3] architecture. Originally developed for medical image segmentation, it has demonstrated versatility across domains, leading to its widespread success. Ongoing research continues to refine its architecture, some of which are discussed later in this work.

1.7.3 Transformer Architectures

Transformers are a deep learning architecture originally developed for natural language processing (NLP) [19] that have since been adapted to tasks in computer vision and time-series analysis. Transformers rely on attention mechanisms to process entire sequences simultaneously. This design enables Transformers to effectively capture long-range dependencies and dynamic relationships within data [20, 21]. By capturing scene-wide spatial and contextual dependencies, Transformers excel in demanding semantic segmentation tasks. Furthermore, their attention mechanisms can help with selective focus on relevant objects, such as pedestrians and vehicles, and their spatial interactions to improve segmentation accuracy for critical road elements. However, their computational complexity and large parameter count can pose challenges for real-time deployment despite achieving great performance. Although not directly employed in this thesis, the attention mechanisms at the core of transformers offer valuable insights and techniques that can be exploited to improve other DL models. Attention mechanisms allow models to dynamically prioritize critical features, improving robustness against occlusions or background distractions. In summary, among various models, 2D CNNs and encoder-decoder architectures show strong promise for semantic segmentation and remain the most practical choice, offering an efficient accuracy trade-off. With appropriate enhancements, they effectively parse complex scenes without the heavy computational cost of transformers.

1.8 Loss Functions and Evaluation Metrics

Loss functions define the training objective by quantifying the discrepancy between model predictions and ground truth, guiding parameter updates through iterative refinement to minimize prediction errors. On the other hand, evaluation metrics assess model performance by quantifying model outcomes based on the specific task. The chosen loss functions and evaluation metrics were selected for their simplicity, ease of implementation, and widespread use in CNN-based learning for semantic segmentation and edge detection. Categorical cross-entropy loss and binary cross-entropy are well-established choices that effectively handle multi-class segmentation and binary edge segmentation without unnecessary complexity in interpreting training dynamics and optimization behavior. Similarly, mIoU, accuracy, and recall provide straightforward measures of performance. They are the most commonly reported metrics for segmentation and edge detection tasks, making them the obvious choice for effective comparative analysis. The efficiency metric used in this work normalizes performance gains relative to a baseline, aligning with the thesis focus on improving existing architectures rather than developing a novel model. Furthermore, the choice of CCE maintains consistency with mIoU, as both inherently treat classes equally. While weighted loss functions may marginally improve rare class predictions, the overall mIoU is still calculated as the average of per-class IoUs without weighting based on class frequency, potentially diminishing the practical value of incorporating more complex loss functions. Additionally, while weighted CCE or focal loss penalize rare classes more heavily, they do not alter the underlying data distribution and may over-penalize classes that become more prevalent due to the applied augmentation strategies. Since the augmentations used in this work alter both geometric and photometric properties, the model's perception of class rarity may shift, complicating the calibration of class weights. Thus, the simple, unweighted CCE was chosen to maintain consistency across training and evaluation, avoiding unnecessary hyperparameters that could obscure the focus on architectural contributions. Exploring more complex loss functions and their effects on training behavior, as well as studying their interaction with augmentation strategies, was considered beyond the scope of this work. The loss function and metrics used in this work are outlined below.

1.8.1 Loss Functions

Categorical Cross-Entropy (CCE)

An objective function that is used in multi-class classification problems. It is computed as:

$$\mathcal{L}_{\text{CCE}} = -\frac{1}{n}\sum_{j}\sum_{i} y_{ij} \cdot \log(\hat{y}_{ij}), \tag{1.1}$$

where n represents the batch size, y_{ij} denotes the actual probability of class i and $\hat{y}_{ij}$ is the predicted probability of class i for sample j.

Binary Cross-Entropy (BCE)

An objective function for a single class problem such as edge detection. Here, a pixel classified as positive represents an edge pixel. It is defined as:

$$\mathcal{L}_{\text{BCE}}(y, \hat{y}) = -\Big[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})\Big], \tag{1.2}$$

where $y \in \{0, 1\}$ is the true binary label, $\hat{y} \in [0, 1]$ is the predicted probability of the positive class. Similarly, for a batch of N samples, the averaged BCE loss becomes:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)\Big], \tag{1.3}$$

where y_i and $\hat{y}_i$ are the ground truth label and the predicted probability for the i^{th} pixel.

1.8.2 Evaluation Metrics

Accuracy (Acc)

Accuracy computes the ratio between the number of correctly segmented pixels and the total number of pixels in the input image as given in Equation (1.4).

$$\text{Acc} = \frac{\text{Sum of correctly segmented pixels}}{\text{Total number of pixels}} \times 100. \tag{1.4}$$

However, Acc can be unreliable in the presence of class imbalance.

Mean Intersection over Union (mIoU)

It measures the degree of overlap between the prediction and the ground truth, as defined in Equation (1.5).

$$\text{mIoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad = \frac{1}{N} \sum_{i=1}^{N} \frac{|P_i \cap G_i|}{|P_i \cup G_i|}, \tag{1.5}$$

where N, P_i, and G_i, represent the total number of semantic segmentation classes, the predicted segmentation mask for class i, and the ground truth segmentation mask for class i, respectively.

Recall

Binary segmentation of boundaries/edges is evaluated using recall, which is computed as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} \times 100, \tag{1.6}$$

where true positives refer to correct edge pixel predictions, and false negatives refer to missed edge pixel predictions, effectively measuring the completeness of edge pixel prediction.

Efficiency

To quantitatively assess the trade-off between segmentation performance and model complexity, a model efficiency metric is introduced:

$$\text{Efficiency} = \frac{\Delta\text{mIoU}}{log_{10}(\text{Params}) \times \text{GFLOPS}} \times 100\%, \tag{1.7}$$

where ΔmIoU is the gain in mIoU between the respective model and the baseline, GFLOPS is giga floating point operations per second, and Params is the number of trainable parameters of the model. This formulation enables a fair comparison by normalizing performance improvements against the increase in both computational complexity and model size.

1.9 Thesis Contribution

This thesis advances supervised semantic segmentation through the development of a robust, efficient lightweight architecture that maximizes neural capacity via multiscale feature mixing and targeted architectural innovations. Unlike methods relying on semi-supervised learning or synthetic data, this framework demonstrates how carefully designed supervised models can achieve competitive performance while remaining deployable in resource-constrained environments. The work bridges critical gaps between computational efficiency and segmentation accuracy through these key contributions:

- **Insights from a Multi-Model Learning Approach:** This thesis investigates a dual-network architecture that introduces feature enrichment via a supplementary network. While this approach does not feature in the final framework, its exploration provided invaluable insights into addressing limitations imposed by large architectures. These findings informed the development of a streamlined and efficient approach that enables robust semantic segmentation. The dual-network model also serves as a foundational strategy for future researchers to improve feature representation with constrained datasets.

- **Comprehensive Data Augmentation Techniques:** This thesis provides an in-depth exploration of data preprocessing methods, including strategies for improving data variation, normalizing data, and introducing perturbations to mirror real-world diverse scenarios.

- **Optimizing Framework Performance:** Through extensive experimentation and sanity analysis of architectural components, the study minimizes architectural redundancies while optimizing performance. The final proposed framework significantly improves the efficiency and accuracy of image segmentation tasks, showcasing notable advancements with reduced network complexity.

- **Robustness Across Diverse Datasets and Critical Comparative Analysis:** Several benchmark datasets were used for model evaluation to validate the performance. Furthermore, the effectiveness of the proposed framework is rigorously evaluated through comparative analyses, highlighting the framework's improvements over existing methods.

These contributions seek to advance research in this domain by establishing robust methodologies and frameworks that optimize neural network capacity, enhancing performance and efficiency. In addition, they provide scalable and precise modeling strategies to support the development of sustainable and resilient deep learning systems.

Chapter 2

Literature Review

2.1 Chapter Overview

This chapter provides a comprehensive review of the evolution of semantic segmentation methodologies, emphasizing their application to urban scene understanding and autonomous driving. The discussion traces the field's progression from classical techniques reliant on handcrafted features (e.g., HOG, CRFs) to modern deep learning methods, highlighting pivotal architectural innovations such as FCNs, U-Nets, and transformer-based models (e.g., SegFormer, KMaX-DeepLab). Critical challenges were identified, including trade-offs in feature extraction, computational inefficiency in resource-intensive architectures that hinder real-time deployment, and data dependency of state-of-the-art models that often require large labeled datasets or synthetic data augmentation to achieve optimal performance gains.

The review further examined auxiliary strategies like data augmentation and semi-supervised learning, underscoring their role in improving generalization while acknowledging their limitations in fully replacing supervised training. Thus, by identifying critical gaps in current research, this chapter lays the groundwork for the novel contributions of this study.

2.1.1 The Shift from Conventional to Deep Learning Models

Prior to the advent of deep learning, conventional image segmentation relied on hand-engineered feature descriptors and classical machine learning techniques. Approaches such as Histogram of Oriented Gradients (HOG) [22], textons [23], and dominant color vectors [24] were employed to encode local texture and appearance characteristics, while classifiers, viz., Support Vector Machines (SVM) [22], Conditional Random Fields (CRF) [25], and Random Forests [26] provided structured pixel- or region-wise labeling. Unsupervised methods, including K-means clustering and mean-shift segmentation, grouped pixels based on low-level feature similarity [27]. Hybrid pipelines often combined these techniques—for instance, CRFs refined SVM outputs using spatial smoothness constraints or clustering initialized region-based classifiers. However, these methods faced fundamental limitations. Handcrafted features lacked the representational power to capture high-level semantic concepts, struggling with complex textures and intra-class variations. Classifier performance plateaued due to shallow architectures and linear decision boundaries, and was biased by data distribution. Clustering often produced over-segmented or under-segmented regions without semantic meaning. The entire pipeline from feature descriptor engineering to segmented maps required meticulous, task-specific tuning of hyperparameters (e.g., cluster counts, CRF edge potentials), limiting generalizability across diverse datasets.

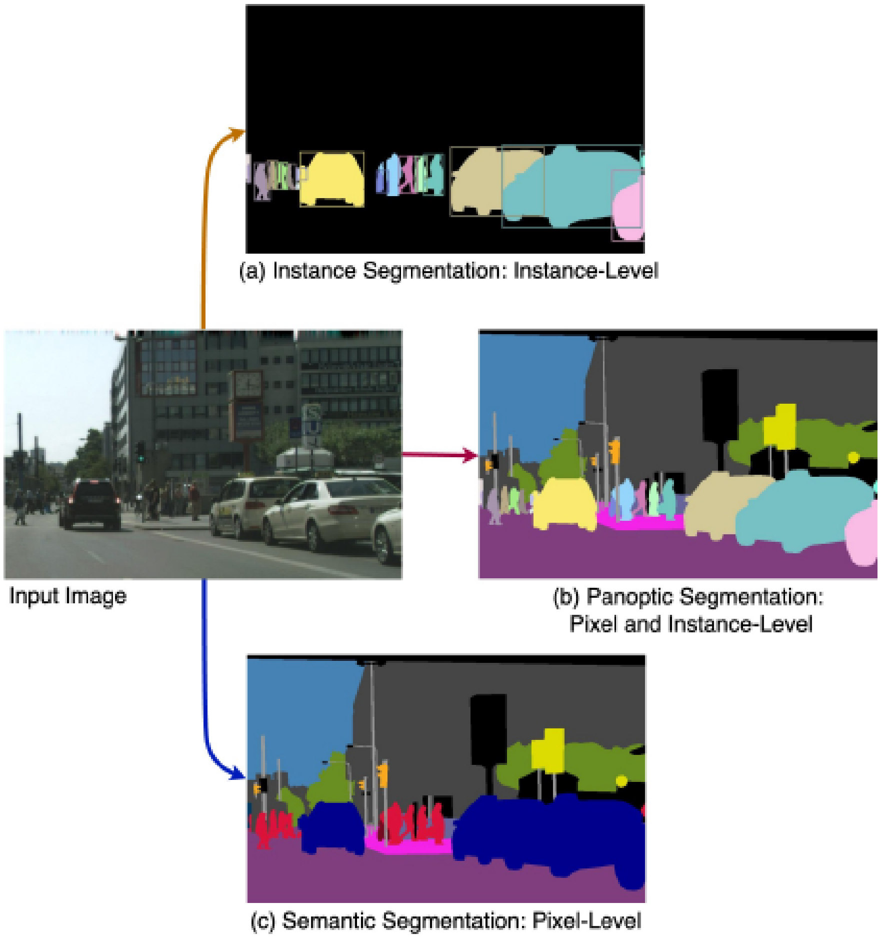
Moreover, traditional segmentation pipelines were further constrained in complex scenarios characterized by high density of objects, varying object scales, and substantial variability in object appearances. Handcrafted features like HOG or textons, were effective for simpler segmentation tasks but exhibited performance degradation when confronted with urban scenes due to the limited feature representational capacity of the various objects and their appearances in such scenes. For example, in an urban scene, vehicles under different lighting and weather conditions, pedestrians with varying attire, etc., often exceed the descriptive power of traditional feature descriptors. As a result, segmentation performance suffers as these methods fail to distinguish semantically similar but visually distinct objects. These constraints necessitated the shift toward data-driven deep learning paradigms capable of hierarchical feature learning.

2.1.2 Image Segmentation Methods: A Dive into Semantic Segmentation

Image segmentation represents a computationally intensive task that extends beyond conventional image classification by generating pixel-level annotations to partition an image into semantically meaningful regions. This fine-grained approach enables the extraction of intricate spatial details, rendering it indispensable for advanced computer vision applications where precise localization is paramount. Supervised image segmentation remains the gold standard for achieving state-of-the-art performance despite the growing interest in alternative learning paradigms. While the challenges of annotation cost and scalability are well documented, the bottom line remains that when sufficient high-quality labeled data is available, supervised methods consistently outperform other methods in terms of accuracy and reliability. The key challenge, therefore, shifts to optimizing performance by extracting the maximum predictive value from the available labeled data through intelligent architectural design and training strategies.

Modern approaches leverage several critical insights from deep learning research to achieve this efficiency, including architectural innovations, advanced regularization techniques to prevent overfitting to limited training samples, and label-efficient architectures that minimize information loss. Instead of abandoning supervised techniques due to data constraints, the path forward lies in making them more annotation-efficient through smart strategies that maximize information extraction to minimize reliance on large amounts of labeled data. The right strategies depend on the type of image segmentation task as well, which are broadly categorized into three types: instance, semantic, and panoptic segmentation.

Figure 2.1 illustrates the difference between the types of segmentation. Instance segmentation focuses on segmenting individual object instances within an image, i.e., segmenting the different instances of the same class [28–30]. Semantic segmentation involves the classification of pixels into different object categories such that the image is segmented into its constituent elements that make up the scene [5, 31, 32]. Panoptic segmentation combines semantic and instance segmentation to simultaneously classify each pixel into a semantic category while also detecting instances of different objects within the scene [29, 30, 33], thus providing a comprehensive description of object identities and their spatial occurrences. Semantic segmentation has gained significant momentum



(a) Instance Segmentation: Instance-Level

(b) Panoptic Segmentation: Pixel and Instance-Level

(c) Semantic Segmentation: Pixel-Level

Figure 2.1: Image segmentation types: (a) Instance segmentation w/t per-object mask and class label; (c) Semantic segmentation w/t per-pixel class labels, and (b) Panoptic segmentation w/t per-pixel class and instance-level labels. Image credit: https://ieeexplore.ieee.org/document/8953237.

in many domains, including precision agriculture and autonomous transportation [5]. Semantic segmentation in the context of driving and urban scenes is explored in depth within the scope of this thesis, with the following section delving into relevant literature in the field.

2.1.3 The Developments Trends in Semantic Segmentation

Advancements with Convolutional Neural Networks

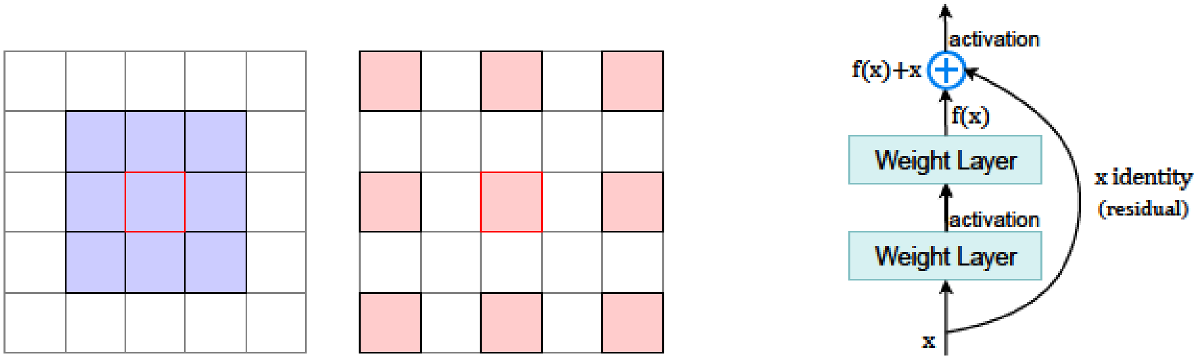
Deep learning methods automated the feature extraction process and reduced reliance on traditional feature engineering. The evolution of semantic segmentation architectures for autonomous driving has been fundamentally shaped by progressively sophisticated approaches to hierarchical feature extraction. The field began with fully convolutional networks (FCNs) [2], which established the paradigm of end-to-end learnable segmentation systems by replacing dense layers with convolutional operations. While groundbreaking, FCNs exhibited significant limitations in autonomous driving applications due to their reliance on single-scale feature maps and naive upsampling, resulting in poor boundary delineation, a critical shortcoming for tasks like drivable area segmentation and obstacle detection. The core challenge in hierarchical feature learning lies in balancing spatial precision against levels of abstraction.

The subsequent development of U-Net [3] introduced a symmetric encoder-decoder architecture with skip connections, enabling precise localization through multi-scale feature fusion. Given its success in medical image segmentation, many modifications have been made to the U-Net for driving scene parsing, resulting in its successful adaptation to Cityscapes [34, 35]. However, U-Net still struggles with satisfactory dilineation of rare object classes. SegNet [36] addressed this through its innovative use of max-pooling indices for decoder upsampling, reducing memory consumption compared to U-Net while maintaining competitive accuracy on road scene datasets.

Meanwhile, models like the Mask R-CNN [37, 38], which extends the Faster R-CNN framework to perform image segmentation, combines object detection and semantic segmentation into one unified task framework on the basis that both tasks share common feature dependencies to enable a synergistic improvement for both task outcomes. However, the arrangement requires significant computational resources due to its two-stage architecture (region proposal network + mask prediction), making it slower and more memory-consuming compared to other models and a less practical solution for real-time deployment. The field witnessed transformative progress with the integration of powerful backbone networks into segmentation frameworks. The introduction of HRNet [39] achieved significant improvement by maintaining high-resolution representations throughout the network rather than recovering them in the decoder. This approach proved particularly valuable for autonomous driving, where preserving fine details in complex urban scenes is crucial. HRNet's parallel multi-resolution subnetworks and repeated information exchange achieved state-of-the-art performance on the Cityscapes and PASCALContext datasets at the time, with particular improvements in small object segmentation.

Similarly, the Deeplab series [4, 40–42] introduced several innovations critical for autonomous driving, including Atrous Spatial Pyramid Pooling (ASPP), which enabled multi-scale feature extraction without resolution reduction or computational overhead by using dilated convolutions [43] (cf. Fig. 2.2a-b). The ASPP module improved feature extraction efficiency by aggregating contextual information from backbone features. This was particularly useful for handling objects at varying distances in driving scenes.

Meanwhile, powerful backbones like Residual Networks (ResNet) [44], introduced residual skip connections (cf. Fig. 2.2c) to address the issue of vanishing gradients. Gradients can flow through these connections as a shortcut, which makes it possible to train incredibly deep networks. Several versions of ResNet exist, varying primarily in depth through the number of stacked residual blocks. Key architectures include ResNet-50, ResNet-101, and ResNet-152, etc., where the suffix denotes the number of layers.



(a) Standard 3×3 Conv (b) Dilated 3×3 Conv (rate=2) (c) Residual block

Figure 2.2: (a) - (b) Comparison of standard and dilated 3×3 convolutions on a 5×5 feature map. Both use 9 sampling points, but dilation increases the receptive field. (c) Residual block.

The adoption of ResNet [44] backbones allowed for more complex feature hierarchies that proved particularly beneficial for fine-grained segmentation as it allowed for simultaneous recognition of diverse object classes at multiple scales. However, despite their success, its sequential bottleneck design creates depth-induced latency, as each additional block linearly increases its floating operations per second while offering diminishing returns in accuracy. As a result, many emerging alternatives address the issues of popular but bulky backbones by introducing dynamic routing, i.e., switchable networks and neural architecture search [45, 46], and attention-based feature selection [21]. A summary of key CNN-based models and backbone-optimized models developed for urban scene segmentation is provided in Table 2.1, along with their performance in the Cityscapes [47] benchmark dataset for comparison.

Table 2.1: A summary of key CNN-based models adapted for urban scene segmentation

Ref.	Methodology	Limitations	Metric
[2, 36] 2014	Replaces fully connected layers with convolutional layers for dense pixel-wise prediction using deconvolutions.	Detailed spatial information is lost from shallow layers due to pooling operations and is not recovered during deconvolutions.	65.30% mIoU
[36] 2015	An encoder–decoder architecture that utilizes max-pooling indices from the encoder for precise upsampling in the decoder.	The Backbone encoder increases complexity and struggles with understanding long-range dependencies resulting in poor inference time, and poor accuracy for larger objects.	57.00% mIoU
[39] 2015	Maintains high-resolution feature representations throughout the network by using parallel branches at different resolutions which are fused across these branches to preserve fine details and capture multi-scale context.	Multiple parallel branches and frequent fusion of high-level information causing high parameter count (70.3M params) and computational load (1206.3 GFLOPS), making it impractical for real-time deployment.	81.6% on val.
[41] 2017	Atrous Spatial Pyramid Pooling module to aggregate features at multiple receptive fields and avoids an explicit decoder and instead directly produces high-quality segmentation maps from deep feature layers by interpolation.	Bulky because of ResNet101 backbone and uses additional training data.	78.50% on val.
[34, 39] 2018	An enhanced U-Net architecture that leverages series of nested, dense skip pathways in addition to the traditional U-Net's encoder-decoder arrangement.	ResNet101 backbone and nested skip connections significantly increase the number of parameters (59.5M) and memory usage (748.5 GFLOPS).	75.50%
[42] 2018	Builds on DeepLabv3 by adding a decoder module for better spatial detail recovery.	Uses a Dilated-Xception-71 backbone making it resource intensive due to high parameter count with only marginal improvement in performance, also trained on extra data.	79.55% on val.

Transformer-based Pioneers

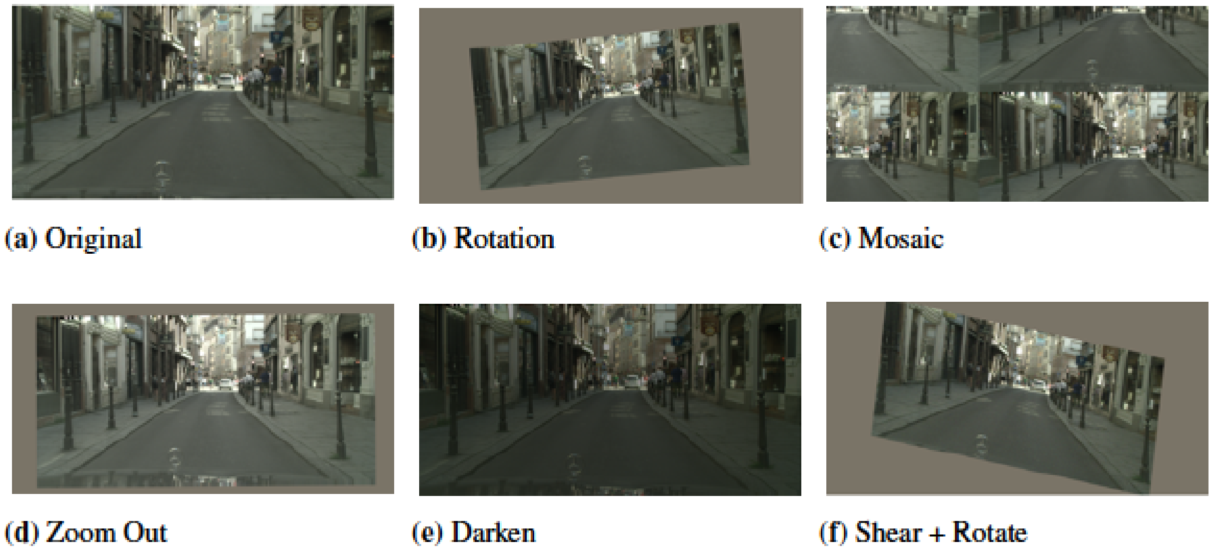
The success of self-attention mechanisms in NLP [19, 48] led to their adaptation in vision transformers (ViTs) [21]. Models like SegFormer [20], VLTSeg [49], and SERNet-Former [50] demonstrate superior performance in capturing long-range dependencies and achieve exemplary results in urban scene segmentation. The KMaX-DeepLab [30] approach combined transformer strengths with CNN efficiency, achieving state-of-the-art results on Cityscapes without the need for additional data. Despite achieving impressive accuracy, these models remain constrained by their network size, computational demands, and reliance on large quantities of labeled data to attain optimal performance, often necessitating the incorporation of additional training data. Conversely, the success of transformers has popularized the use of attention mechanisms [51–53] in computer vision tasks, and numerous models have demonstrated strong performance [17, 18, 54] without incurring the substantial computational overhead typically associated with transformer-based architectures. Table 2.2 summarizes the review of transformer-based models in the recent literature, along with their performance in Cityscapes, the urban scenes dataset.

Table 2.2: A summary of key Transformer-based models in urban scene segmentation

Ref.	Methodology	Limitations	Metric
[20] 2021	Introduces a positional-encoding-free, hierarchical Transformer encoder and a lightweight MLP decoder for a reduced complexity approach to segmentation.	Uses additional coarse data to achieve optimal performance and large GFLOPS (1447.6) make it impractical for real-time performance (2.5 FPS).	83.1% on test.
[30] 2022	Integrates k-means clustering and transformer-based attention mechanism to produce refined segmentation.	Despite not using additional training data, model size is extremely bulky with large backbone (232M params) and computationally complex (1673 GFLOPS), which significantly affect inference speed (3.1 FPS).	83.5% on val.
[49] 2023	Leverages the Vision Transformer by dividing an image into fixed-size patches and processing them with transformer layers to capture long-range dependencies and global context using self-attention mechanisms.	Large model size (304M params) and data hungry, uses 2 additional synthetic training datasets.	86.5% on test
[50] 2024	Hybrid model that combines the strengths of encoder-decoder architectures and vision transformers.	Uses a large backbone (44.2M params) requiring additional training data to achieve optimal performance.	84.8 % on test

Beyond Architectural Innovations

In addition to architectural innovations, data augmentation is widely adopted in deep learning to enhance model generalization and robustness [55, 56]. By pragmatically applying photometric (e.g., color jitter, noise injection) and geometric (e.g., rotation, elastic deformations) perturbations to training samples, augmentation artificially expands the effective dataset diversity (cf. Figure 2.3). This approach addresses two fundamental challenges: mitigating data scarcity—particularly critical in domains with expensive annotations, and improving out-of-distribution generalization by forcing models to learn invariant features beyond superficial appearances. Advanced variants like RandAugment [57] automate perturbation selection, while adversarial augmentation exposes models to worst-case distortions. The technique's universality is evidenced by its adoption across tasks—from classification (via simple crops/flips) to self-supervised learning (where augmentations define pretext tasks). However, task-agnostic augmentation risks disrupting critical features (e.g., occluding key anatomical structures in X-rays), necessitating domain-aware policies. As deep learning increasingly prioritizes data efficiency, augmentation's role evolves beyond mere preprocessing to become integral to loss design, and many models adopt augmentation strategies



(a) Original

(b) Rotation

(c) Mosaic

(d) Zoom Out

(e) Darken

(f) Shear + Rotate

Figure 2.3: Examples of data augmentations applied to an image.

to leverage its benefits. Many of the models discussed in the literature review apply task-specific augmentations for this reason.

The Limitations of Reducing Label Reliance

Reducing reliance on labeled data for training has become a highly pursued area of research due to the substantial cost and time associated with large data annotation. As a result, semi-supervised and unsupervised approaches have become integral to the discussion of achieving high-fidelity segmentation; however, the performance degradation associated with reducing labeled train data prevents such models from achieving the performance benchmarks met by fully supervised models [58, 59]. Table 2.3 summarizes well-known semi-supervised and unsupervised methods on Cityscapes.

Table 2.3: Literature summary of semi-supervised and unsupervised learning methods on Cityscapes

Ref.	Methodology	Limitations	Metric
[59] 2019	Semi-supervised approach leveraging a knowledge graph derived from a large-scale text corpus to capture semantic consistencies across categories to generate synthetic images from unlabeled data to reveal underlying structural information, while incorporating a pyramid architecture in the discriminator to capture multi-scale contextual information for improved parsing.	Reduced labeled data prevents the generator from adapting the static knowledge base to the true data distribution, causing biased pseudo-labels that misalign with the image features, degrading performance.	70.60% on val (with 1/4 train).
[58] 2022	Semi-supervised learning method that reduces feature discrepancies between labeled and unlabeled data using cross-set region-level augmentation and pixel-wise contrastive learning. Stabilizes training with dynamic confidence region selection to focus on high-confidence areas for loss calculation.	Reducing labeled data leads to noisier initial pseudo-labels which can propagate and amplify errors causing performance to lag behind supervised contemporaries.	65.50% on val (with 1/4 train).
[11] 2024	Unsupervised method that leverages a spectral technique that provides both semantic and structural cues by constructing an eigenbasis from a semantic similarity matrix of deep image features, combined with color affinity information, to learn object-level representations.	Reliance on color affinity and spectral clustering fail on objects with complex appearances or lighting variations. The dependence on imperfect pseudo-labels lead to error accumulation, reducing accuracy.	22.1%.
[60] 2025	Unsupervised method that combines visual representations, depth, and motion cues for pseudo-label training and a panoptic self-training strategy to eliminate the need for object-centric training data.	Using scene flow and depth cues for pseudo-labels assumes motion coherence perfectly aligns with object boundaries, which fails for non-rigid objects, occlusions, or dynamic backgrounds limiting semantic groupings and generalization.	26.80% on val.

2.2 Chapter Summary

The discussion in this chapter has illuminated key trends and challenges in the field of semantic segmentation, particularly in the context of autonomous systems and urban scene understanding. While deep learning has revolutionized segmentation tasks, enabling impressive advancements in performance, it has also brought forward critical issues related to computational efficiency and data dependence. The comparative analysis between CNN-based and transformer-based models has shown that although transformers offer state-of-the-art results, they often come with substantial computational costs.

Furthermore, the reliance on large labeled datasets remains a significant hurdle, despite efforts like semi-supervised learning and data augmentation to mitigate the issue. This points to a critical gap between achieving high accuracy and maintaining efficiency, especially in real-time applications. As the demand for AI-driven systems grows, addressing this gap becomes even more urgent, particularly in resource-constrained environments.

This critique sets the stage for the subsequent chapters, which examine model design through two key lenses: accuracy and efficiency. The next chapter investigates a model that prioritizes accuracy, exploring methods to improve segmentation performance by addressing a key challenge in the field. While not constrained by efficiency considerations, this exploration aims to deepen understanding of architectural contributions. The subsequent chapter then translates these insights into a model that reflects the thesis's core objective, and proposes a lightweight, efficient architecture designed to reconcile accuracy with computational efficiency. By focusing on optimizing architectures for real-time deployment and addressing the environmental concerns of large-scale training, this work contributes to the ongoing effort to make AI more efficient, scalable, and sustainable for practical applications.

Chapter 3

Developing a Dual-Network Multi-Class Segmentation Model

3.1 Overview

Deep learning-based methods have driven significant advances in semantic segmentation in recent years. However, complex scenarios and diverse environmental conditions continue to pose challenges, particularly when objects exhibit highly variable shapes and appearances across different scenes. This chapter introduces a novel approach to improving segmentation accuracy by systematically integrating a complementary pair of convolutional neural networks designed to classify pixels for semantic segmentation.

The proposed methodology harnesses a primary multi-class segmentation network and a secondary binary-segmentation boundary network in a supervised learning setting to mitigate the limitations and biases of the former, presenting a balanced solution for improved segmentation outcomes. The effectiveness of this approach is demonstrated through ablation studies conducted on the CamVid benchmark dataset [1], setting a foundation for advancements in this area.

The rationale for integrating these complementary models stems from their distinct yet synergistic capabilities in addressing fundamental limitations of segmentation architectures. While conventional segmentation models often exhibit reduced performance in precisely delineating fine object boundaries and capturing small or thin structures, specialized binary segmentation networks trained explicitly for edge reconstruction can provide critical high-frequency spatial information. This approach leverages the fine-grained semantic understanding of the primary segmentation model while augmenting it with the boundary-specific refinement capacity of the binary segmentation network. The combined framework is designed to enhance overall segmentation accuracy while mitigating classification errors, a crucial advancement for applications demanding high-precision segmentation, such as autonomous driving or medical imaging, where topological accuracy is paramount.

This chapter investigates the feasibility of a two-model segmentation approach, systematically evaluating whether augmenting a primary segmentation model with a supplementary network yields significant improvements in feature representation and boundary delineation compared to standalone architectures. A positive empirical outcome would not only validate the proposed methodology but also establish a foundation for future research into more sophisticated synergetic frameworks that achieve robust multi-class semantic segmentation.

3.2 The Challenge in Urban Scene Segmentation

Urban driving scenes are characterized by high visual complexity, dynamic elements, and frequent occlusions. Existing datasets often exhibit significant class imbalance, where dominant categories such as roads and sky appear far more frequently than rare but critical classes like pedestrians and traffic signs. Additionally, improving accuracy with supervised learning can lead to increasingly complex models, which can hinder real-time performance essential for autonomous systems.

At the same time, the supervised methods rely on end-to-end training with fully annotated datasets, achieving strong results in homogeneous regions but often struggling with boundary ambiguity, class imbalance, and sensitivity to object scale. This can result in blurred or smoothed boundaries, misclassification of underrepresented objects, and inconsistent segmentation of small or thin structures. These limitations arise from a range of architectural challenges, including the loss of spatial and structural cues in favor of high-level semantic features, weak long-range feature retention, information loss during pooling operations, inadequate global context integration, and architectural redundancies.

Therefore, effective segmentation in this context requires a careful trade-off between model capacity—-to accurately capture fine details and rare classes—and computational efficiency. The proposed methodology addresses these challenges by integrating targeted strategies, discussed in the following sections, to mitigate the limitations of existing supervised approaches.

3.2.1 Edge Context in Semantic Segmentation

Recent advancements in semantic segmentation have increasingly emphasized the integration of multi-source information to enhance feature discrimination and localization accuracy. Among these, the fusion of edge-aware features with semantic representations has emerged as a particularly effective strategy, demonstrating significant improvements in segmentation performance, especially in fine-grained boundary delineation [61].

The explicit incorporation of boundary information serves as a structural prior, guiding the segmentation network toward more precise classifications at object boundaries. Studies, such as [62] and [63], have demonstrated that edge-aware feature learning mitigates the common issue of blurred boundaries in segmentation outputs, particularly in complex scenes with occlusions or fine/thin structures. By leveraging an auxiliary edge detection network, these approaches extract high-frequency spatial details that are often lost in deep convolutional networks due to successive pooling and strided operations. The extracted edge features are then fused with multi-level semantic features, either through skip connections, attention mechanisms, or feature concatenation, thereby enriching the hierarchical representation with explicit boundary constraints. This fusion strategy not only refines segmentation masks along object contours but also enhances the model's ability to disambiguate between adjacent regions of similar appearance. For instance, the work [64] employs a module to strengthen the propagation of same segment region features which are isolated by learned boundaries, effectively minimizing misclassifications at edges. Liu et al. [65] introduces a multi-branch architecture where edge predictions are iteratively refined alongside semantic features. Such methods underscore the importance of boundary guidance as a means to enforce geometric consistency in segmentation outputs. By explicitly modeling edge-semantic interactions, these approaches achieve sharper transitions between classes, reducing reliance on post-processing refinement modules. In summary, the integration of edge information into semantic segmentation frameworks provides a principled way to encode structural constraints, leading to improved robustness and accuracy. This proposed model aims to achieve this by exploring a feature-enrichment approach to leverage nuanced feature recovery for improved segmentation outcomes.

3.2.2 Attention Mechanisms in Semantic Segmentation

The advent of attention mechanisms in deep neural networks (DNNs), coupled with their successful application in natural language processing (NLP) [19, 54], has significantly advanced research in deep learning-based semantic segmentation. The attention mechanism can model long-range dependencies and correlations to enhance task-specific contextual information while hiding less pertinent features. Several types of attention mechanisms highlight distinct aspects of feature maps. A spatial attention block aims to boost the spatial regions that contain the most relevant information in a feature map and can be expressed mathematically as (3.1).

$$
\begin{aligned}
&\mathrm{A} = \sigma((W_q \cdot X) \cdot (W_k \cdot Y)^T), \text{ and} \\
&\mathrm{Z} = A \cdot (W_v \cdot Y),
\end{aligned}
\tag{3.1}
$$

where A is the attention score computed for the two input feature maps to the attention module, X and Y, and Z is the attention-refined output feature map. Hence, W_q, W_k, and W_v are learnable weight matrices used to determine query, key, and value representations, respectively, obtained by computing dot products with the input feature maps. The attention score is computed by applying the sigmoid activation function ($\sigma = \frac{1}{1+e^{-x}}$) to the dot product of the query $(W_q \cdot X)$ and the transposed key $(W_k \cdot Y)^T$. The attention scores A are then used to weight the values $(W_v \cdot Y)$ to produce refined output feature maps, Z, emphasizing the important clues. The attention-refined output feature maps are then integrated back into the network and forward propagated, thus achieving the necessary emphasis on relevant regions in the input image. Spatial attention can be useful to resolve ambiguities in cluttered regions, small objects, and enhances boundary localization by emphasizing high-gradient areas. In this way, to ameliorate image segmentation performance, Ozan *et al.* [54] introduced an Attention U-Net (Att U-Net) that focuses on relevant regions of the input image with varying levels of weightage during training. This approach has shown great improvement in biomedical applications [17, 18, 54] and is continued to be adapted for other domains. In theory, a full attention strategy enhances focus on relevant features at each decoding layer. However, this approach risks propagating noise through the network, potentially complicating the model without significant gains. Thus, this paper investigates a modified attention-gating strategy, offering a simpler yet more effective alternative to the multi-level attention commonly employed in existing models. The goal is to achieve comparable, if not superior, results with reduced complexity by reducing architectural redundancies.

3.2.3 Atrous Spatial Pyramid Pooling (ASPP)

While the U-Net architecture inherently enables multi-scale feature fusion through its skip connections, the standard convolution operations used in U-Net do not fully exploit multi-scale feature extraction, as they lack the ability to capture context at varying receptive fields [66]. The ASPP introduced in DeepLabv2 [40] addressed this through parallel dilated convolutions at various rates. The large receptive fields improved global context understanding, and atrous convolutions minimized information loss typically caused by max-pooling operations, resulting in enhanced performance.

3.3 Methodology

A detailed overview of the proposed model consisting of two networks is provided in Figure 3.1.

The model consists of three main components:

(i) **Edge Network (ENet)**: This network is a U-Net-like encoder-decoder architecture that is is pre-trained to generate Sobel edge features from the input RGB images. It is trained on the

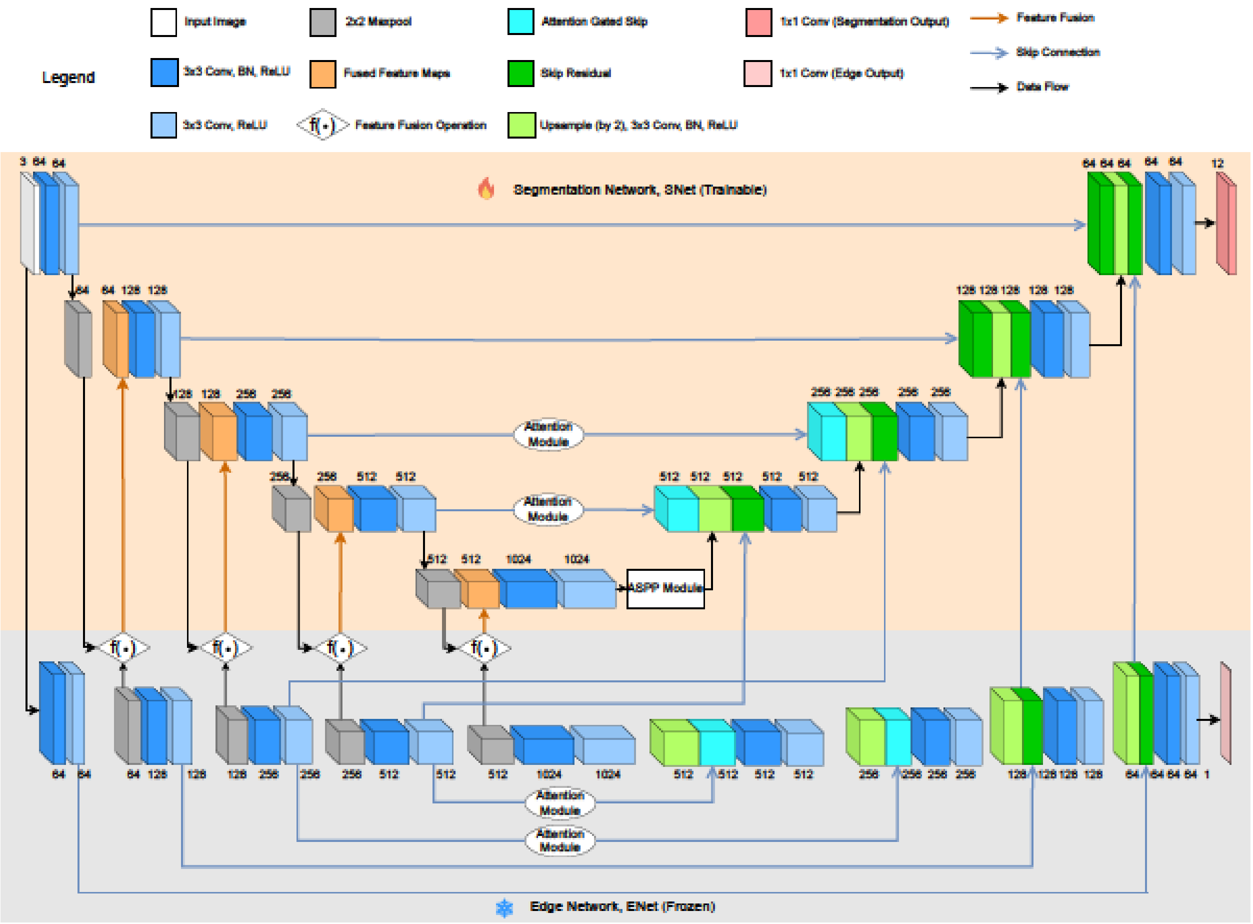


Figure 3.1: Proposed architecture and network details describing the interactions between the networks, as well as the overall pipeline from input to segmentation maps. Top: SNet, Segmentation Network. Bottom: ENet, Edge Network.

CamVid train set and shares boundary related features with the encoder and decoder layers of the second network designed for semantic segmentation.

(ii) **Segmentation Network (SNet)**: This network is also a U-Net-like encoder-decoder architecture with subtle differences from the ENet–the incorporation of an ASPP module.

(iii) **Strategic Feature Fusion**: Feature sharing is enabled through strategically placed feature fusion sites to leverage the collaboration of both networks for unified decision-making.

3.3.1 Model Architectures

Edge Network (ENet)

Table 3.1 provides the layer-wise architectural detail of the ENets's encoding sub-net. It receives a mini-batch of 8 RGB images of size 384×512, and it is trained to detect Sobel edges from the CamVid benchmark dataset by minimizing a Binary Cross-Entropy loss (Equation (1.2)) using the hyperparameter settings summarized in Table 3.2. The encoder pathway consists of successive 3×3 convolutions for feature extraction, each followed by maxpool layers that progressively halve the spatial resolution until reaching 1/16th of the original size, culminating in 1024 feature maps at the deepest layer. Residual features from before each maxpooling operation are stored as skip connections (L2, L5, L8, L11) for the decoding process.

Table 3.1: ENet: Encoding Sub-Network Architecture

Layer ID	Layer Type $A(k, s)$	Output Shape $[b, D, H, W]$	Input
Input	Input Layer	$[b, 3, 384, 512]$	mini-batch
L1	Conv(3,1)→BN→ReLU	$[b, 64, 384, 512]$	Input
L2	Conv(3,1)→ReLU	$[b, 64, 384, 512]$	L1
L3	Maxpool(2,2)	$[b, 64, 192, 256]$	L2
L4	Conv(3,1)→BN→ReLU	$[b, 128, 192, 256]$	L3
L5	Conv(3,1)→ReLU	$[b, 128, 192, 256]$	L4
L6	Maxpool(2,2)	$[b, 128, 96, 128]$	L5
L7	Conv(3,1)→BN→ReLU	$[b, 256, 96, 128]$	L6
L8	Conv(3,1)→ReLU	$[b, 256, 96, 128]$	L5
L9	Maxpool(2,2)	$[b, 256, 48, 64]$	L8
L10	Conv(3,1)→BN→ReLU	$[b, 512, 48, 64]$	L9
L11	Conv(3,1)→ReLU	$[b, 512, 48, 64]$	L7
L12	Maxpool(2,2)	$[b, 512, 24, 32]$	L11
L13	Conv(3,1)→BN→ReLU	$[b, 1024, 24, 32]$	L12
L14	Conv(3,1)→ReLU	$[b, 1024, 24, 32]$	L13

Table 3.2: Training Hyperparameter of the ENet

Hyperparameter	Value
Learning rate	0.001
Optimizer	Adam
# of Epochs	100
Batch size	8
Loss Function	Binary Cross-Entropy Loss

The decoding process occurs in two distinct phases. Table 3.3 on page 36 summarizes the layer-wise detail of the decoding pathway, which produces the class-wise probability distribution. First, the attention-based sub-net upsamples the features by factor 2, combining them with attention-weighted skip connections (L20, L29) and halving the number of feature maps via 3×3 convolutions after each upsampling step. Next, the full decoding sub-net repeats this process but without attention weighting the skip connections from the encoding layers (L2, L5). By the end of the decoder pathway, the resolution is restored to 384×512, and a final convolutional layer acts as the classifier to produce the edge mask (L41). Figure 3.2 illustrates the ENet's training and validation progress over 100 epochs, the 90th epoch is where it achieved optimal recall (cf. Equation (1.6)) and minimized loss.

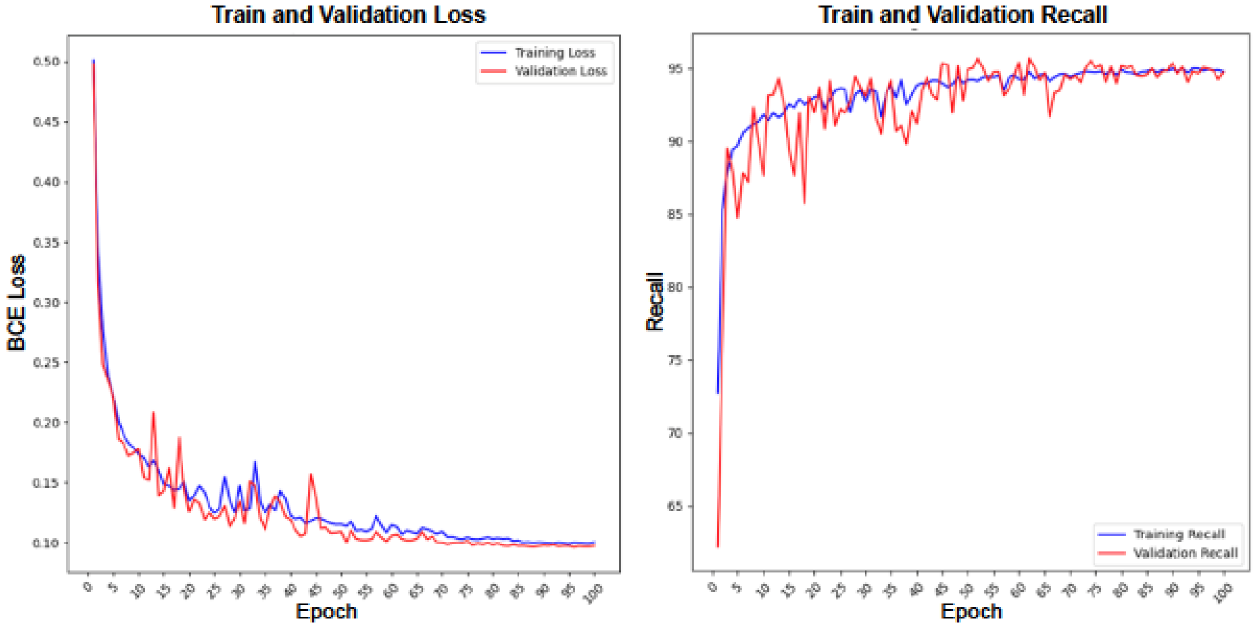


Figure 3.2: Training progress of the ENet for 100 epochs.

Table 3.3: Layer-wise Connectivity Pattern of the Decoding Pathway of the ENet

Layer ID	**Layer Type** $A(k, s)$	**Output Shape** $[b, D, H, W]$	**Input**
Decoding Sub-Net with Attention			
L15	Upsampling Block	$[b, 512, 48, 64]$	L14
L16	Conv(1,1)→BN	$[b, 256, 48, 64]$	L15
L17	Conv(1,1)→BN	$[b, 256, 48, 64]$	L11
L18	Add→ReLU	$[b, 256, 48, 64]$	L11, L15
L19	Conv(1,1)→BN→Sigmoid	$[b, 1, 48, 64]$	L18
L20	Multiply	$[b, 512, 48, 64]$	L18, L11
L21	Concat	$[b, 1024, 48, 64]$	L20, L15
L22	Conv(3,1)→BN→ReLU	$[b, 512, 48, 64]$	L21
L23	Conv(3,1)→ReLU	$[b, 512, 48, 64]$	L22
L24	Upsampling Block	$[b, 256, 96, 128]$	L23
L25	Conv(1,1)→BN	$[b, 256, 96, 128]$	L23
L26	Conv(1,1)→BN	$[b, 256, 96, 128]$	L8
L27	Add→ReLU	$[b, 256, 96, 128]$	L8, L23
L28	Conv(1,1)→BN→Sigmoid	$[b, 1, 96, 128]$	L27
L29	Multiply	$[b, 256, 96, 128]$	L27, L8
L30	Concat	$[b, 512, 96, 128]$	L8, L23
L31	Conv(3,1)→BN→ReLU	$[b, 256, 96, 128]$	L30
L32	Conv(3,1)→ReLU	$[b, 256, 96, 128]$	L31
Decoding Sub-Net w/o Attention			
L33	Upsampling Block	$[b, 128, 192, 256]$	L32
L34	Concat	$[b, 256, 192, 256]$	L33, L5
L35	Conv(3,1)→BN→ReLU	$[b, 128, 192, 256]$	L34
L36	Conv(3,1)→ReLU	$[b, 128, 192, 256]$	L35
L37	Upsampling Block	$[b, 64, 384, 512]$	L36
L38	Concat	$[b, 128, 384, 512]$	L36, L2
L39	Conv(3,1)→BN→ReLU	$[b, 64, 384, 512]$	L38
L40	Conv(3,1)→ReLU	$[b, 64, 384, 512]$	L39
L41	Conv(1,1)→$f(\cdot)$	$[b, 1, 384, 512]$	L40

Total number of trainable parameters: 34,846,853

Note: $A(\cdot)$ - Operation type, k - Kernel size, s - stride rate, b - batch size, D, H, W - depth, height, and width of the feature map, BN - Batch Normalization, Upsampling Block - Upsample(2,2)→Conv(3,1)→BN→ReLU, $f(\cdot)$ - Classifier (Sigmoid)

Segmentation Network (SNet)

The segmentation network receives a mini-batch of 4 RGB images as input data simultaneously as the ENet and operates as the primary multi-class label predictor. It is trained on CamVid train samples to minimize Categorical Cross-Entropy loss between predicted and target pixel classes given by Equation (1.1) and evaluated using mIoU and accuracy metrics (Equations (1.5), (1.4), respectively). The hyperparameter settings are summarized in Table 3.6. Also designed using the U-Net as inspiration, it is almost identical to the edge network except for one key structural difference. An ASPP module is incorporated at the bottleneck to perform multi-scale feature processing to leverage high-dimensional feature maps enriched with edge information to improve delineated segmentation. The layer-wise details of the ASPP module are provided in Table 3.4. It involves four parallel 3×3 Convolution (Conv) kernels with atrous rates of 1, 6, 2, and 18, respectively, for feature extraction.

Table 3.4: SNet: ASPP Module Architecture

Layer ID	Layer Type $A(k, s, d)$	Output Shape $[b, C, H, W]$	Input
SNet: ASPP Module Architecture			
Input	Previous Layer	$[b, 1024, 24, 32]$	P22
A1	Conv(1,1)→BN→ReLU	$[b, 512, 24, 32]$	Input
A2	Conv(3,1,6)→BN→ReLU	$[b, 512, 24, 32]$	Input
A3	Conv(3,1,12)→BN→ReLU	$[b, 512, 24, 32]$	Input
A4	Conv(3,1,18)→BN→ReLU	$[b, 512, 24, 32$	Input
A5	AdaptiveAvgPool2d(1)	$[b, 1024, 1, 1]$	Input
A6	Conv(1,1)→BN→ReLU	$[b, 512, 1, 1]$	A5
A7	Upsample(’bilinear’)	$[b, 512, 24, 32]$	A6
A8	Concat	$[b, 2560, H, W]$	A1, A2, A3, A4, A7
ASPP Output	Conv(1,1)→BN→ReLU	$[b, 1024, 24, 32]$	A8

Note: $A(\cdot)$ - Operation type, k - Kernel size, s - stride rate, b - batch size,
D, H, W - depth, height, and width of the feature map, BN - Batch Normalization.

Batch normalization (BN) and ReLU activation are applied at the end of parallel dilated convolutions, and the outputs are finally spliced with an adaptive average pooling output of the input feature map. A 1×1 Conv is used to resize the channel dimensions of the resulting ASPP feature maps to that of the ASPP block Input and is passed to the decoding pathway.

Table 3.5: SNet: Encoding Sub-Network Architecture

	Layer Type	**Output Shape**	
Layer ID	$A(k, s)$	$[b, D, H, W]$	**Input**
	SNet: Encoding Sub-Net Architecture		
Input	Input Layer	$[b, 3, 384, 512]$	mini-batch
P1	Conv(3,1)→BN→ReLU	$[b, 64, 384, 512]$	Input
P2	Conv(3,1)→ReLU	$[b, 64, 384, 512]$	P1
P3	Maxpool(2,2)	$[b, 64, 192, 256]$	P2
P4	Concat	$[b, 128, 192, 256]$	P3, L3
P5	Conv(1,1)	$[b, 64, 192, 256]$	P4
P6	Conv(3,1)→BN→ReLU	$[b, 128, 192, 256]$	P5
P7	Conv(3,1)→ReLU	$[b, 128, 192, 256]$	P6
P8	Maxpool(2,2)	$[b, 128, 96, 128]$	P7
P9	Concat	$[b, 256, 96, 128]$	P8, L6
P10	Conv(1,1)	$[b, 128, 96, 128]$	P9
P11	Conv(3,1)→BN→ReLU	$[b, 256, 96, 128]$	P10
P12	Conv(3,1)→ReLU	$[b, 256, 96, 128]$	P11
P13	Maxpool(2,2)	$[b, 256, 48, 64]$	P12
P14	Concat	$[b, 512, 48, 64]$	P13, L9
P15	Conv(1,1)	$[b, 256, 48, 64]$	P14
P16	Conv(3,1)→BN→ReLU	$[b, 512, 48, 64]$	P15
P17	Conv(3,1)→ReLU	$[b, 512, 48, 64]$	P16
P18	Maxpool(2,2)	$[b, 512, 24, 32]$	P17
P19	Concat	$[b, 1024, 24, 32]$	P18, L12
P20	Conv(1,1)	$[b, 512, 24, 32]$	P19
P21	Conv(3,1)→BN→ReLU	$[b, 1024, 24, 32]$	P20
P22	Conv(3,1)→ReLU	$[b, 1024, 24, 32]$	P21

Note: $A(\cdot)$ - Operation type, k - Kernel size, s - stride rate, b - batch size, D, H, W - depth, height, and width of the feature map, BN - Batch Normalization.

Table 3.5 summarizes the layer-wise detail of the Segmentation network's encoding sub-net developed in this work. While the SNet works similarly to the ENet, it receives specific intermediate outputs from the ENet as edge signals, which are propagated through the network at different stages. For example, the SNet's encoding sub-net propagates the skip connections from the ENet's encoding sub-net (L3, L6, L9, L12) along with its own subsampled features (P3, P7, P12, P17) through concatenation and convolution. Similarly, the decoding process is divided into two phases like the ENet, but it involves upsampling combined features from both the networks' skip connections. By the end of the decoder pathway, the resolution is restored to 384×512, and a final convolutional layer acts as the classifier to produce the semantic class probability map (Out). Table 3.7 on page 40 summarizes the layers used to build the decoding pathway. The detailed description of the feature fusion strategies used to enable feature sharing between the ENet and SNet is provided in the next Section 3.3.1. Figure 3.3 illustrates the SNet's training and validation progress over 150 epochs, the 110th epoch is where it achieved optimal recall and minimized loss.

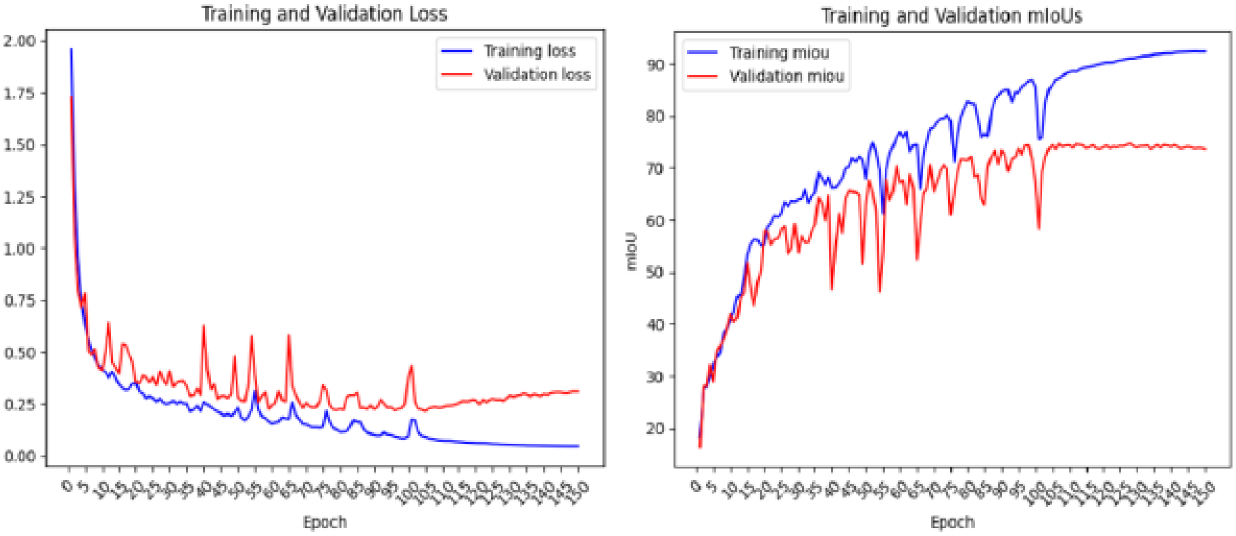


Figure 3.3: Training progress of the SNet for 150 epochs.

Table 3.6: Training Hyperparameter of the SNet

Hyperparameter	**Value**
Learning rate	0.001
Optimizer	Adam
# of Epochs	150
Batch size	4
Loss Function	Categorical Cross-Entropy Loss

Table 3.7: Layer-wise Connectivity Pattern of Decoding Pathway of the SNet

Layer ID	Layer Type	Output Shape	Input
Decoding Sub-Net with Attention			
P23	Previous Layer	$[b, 1024, 24, 32]$	ASPP Output
P24	Upsampling Block	$[b, 512, 48, 64]$	P23
P25	Conv(1,1)→BN	$[b, 256, 48, 64]$	P24
P26	Conv(1,1)→BN	$[b, 256, 48, 64]$	P17
P27	Add→ReLU	$[b, 256, 48, 64]$	P26, P25
P28	Conv(1,1)→BN→Sigmoid	$[b, 1, 48, 64]$	P27
P29	Multiply	$[b, 512, 48, 64]$	L28,P17
P30	Concat	$[b, 1536, 48, 64]$	P29, P24, L11
P31	Conv(3,1)→BN→ReLU	$[b, 512, 48, 64]$	P30
P32	Conv(3,1)→ReLU	$[b, 512, 48, 64]$	P32
P33	Upsampling Block	$[b, 256, 96, 128]$	LP32
P34	Conv(1,1)→BN	$[b, 256, 96, 128]$	P33
P35	Conv(1,1)→BN	$[b, 256, 96, 128]$	P12
P36	Add→ReLU	$[b, 256, 96, 128]$	P35, P34
P37	Conv(1,1)→BN→Sigmoid	$[b, 1, 96, 128]$	P36
P38	Multiply	$[b, 256, 96, 128]$	P37, P12
P39	Concat	$[b, 768, 96, 128]$	P38, P33, L8
P40	Conv(3,1)→BN→ReLU	$[b, 256, 96, 128]$	P39
P41	Conv(3,1)→ReLU	$[b, 256, 96, 128]$	P40
Decoding Sub-Net w/o Attention			
P42	Upsampling Block	$[b, 128, 192, 256]$	P41
P43	Concat	$[b, 384, 192, 256]$	P42, P7, L5
P44	Conv(3,1)→BN→ReLU	$[b, 128, 192, 256]$	P43
P45	Conv(3,1)→ReLU	$[b, 128, 192, 256]$	P44
P46	Upsampling Block	$[b, 64, 384, 512]$	P45
P47	Concat	$[b, 192, 384, 512]$	P46, P2, L2
P48	Conv(3,1)→BN→ReLU	$[b, 64, 384, 512]$	P47
P49	Conv(3,1)→ReLU	$[b, 64, 384, 512]$	P48
Out	Conv(1,1)→$f(\cdot)$	$[b, 12, 384, 512]$	L40
Total number of trainable parameters: 56,514,832			

Note: $A(\cdot)$ - Operation type, k - Kernel size, s - stride rate, b - batch size,
D, H, W - depth, height, and width of the feature map, BN - Batch Normalization,
Upsampling Block - Upsample(2,2)→Conv(3,1)→BN→ReLU, $f(\cdot)$ - Softmax Classifier

Feature Fusion Strategy

The edge features from the encoding pathway of the ENet, are fused with the encoder block outputs of the SNet using simple depth-wise concatenation and 1×1 Conv to resize the feature maps to the required size. The decoder pathway of the SNet receives features from the SNet's encoder pathway as well as the ENet's encoder pathway. Figure 3.4 demonstrates the feature fusion and skip connection strategy that facilitates feature sharing from the ENet to the SNet.

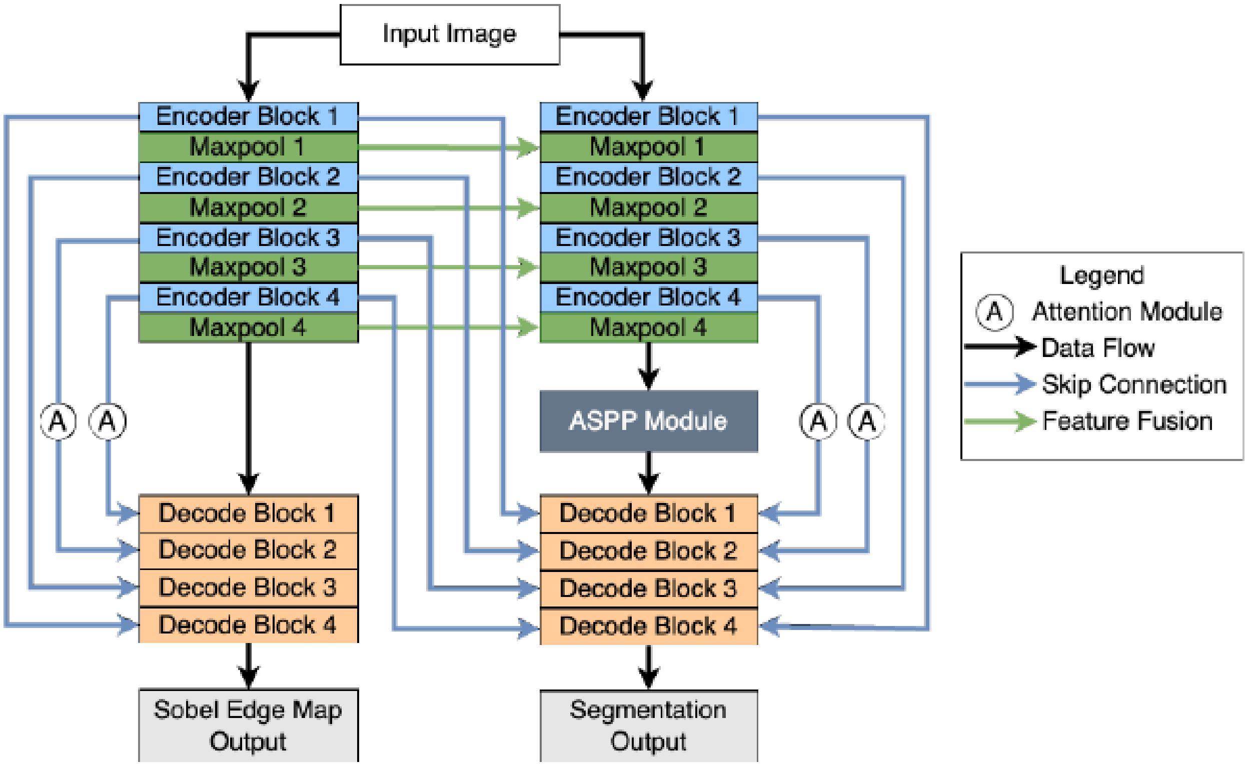


Figure 3.4: The feature fusion and skip connection locations employed between the two subnets.

3.3.2 Network Justification

In order to determine the ideal architectural arrangement of the dual network that enables optimal feature sharing, a methodical sanity analysis is conducted to develop the final architecture. To start, the baseline, i.e. the basic U-Net [3], and an Att U-Net [54] were trained from scratch. It was found that the Att U-Net performed better than the basic U-Net with a 2.00% mIoU improvement. This prompted further analysis by incrementally adding one attention block at a time, starting from the deepest encoder layer (with the smallest spatial dimension) and progressing outward to the bottom layer (with the largest spatial dimension), to determine the optimal number of attention blocks. This revealed that the addition of the 3rd and 4th attention blocks had negligible improvement since the deeper feature maps are more information-rich. Reducing the attention blocks to two at the deepest encoding layers also reduced the number of trainable parameters while achieving almost the same mIoU. The full-attention strategy achieved a mIoU of 64.55%, while the two-attention-block approach (abbr. 2-Att) delivered a comparable yet marginally improved mIoU of 64.63%.

Subsequently, to reach an optimal fusion model using the ENet and the SNet, systematic experiments were performed as tabulated in Table 3.8. First, the pooled edge features from the ENet's encoder pathway were fused with the features of the SNet's encoder pathway (Exp1: M1). Then, the bottleneck feature maps were fused in addition to the encoder-pooled outputs (Exp2: M2). Then, the ENet's decoder block outputs were also fused with the SNet's decoder block outputs (Exp3: M3). Later, the encoder pathway was unfused, leaving only the bottleneck feature maps and decoder block outputs fused between the subnets (Exp4: M4), and finally, the bottleneck was unfused, resulting in only the decoder block outputs between the networks remaining fused (Exp5: M5).

Table 3.8: Sanity analysis of the feature fusion strategies used to build the proposed model. % change is w.r.t the baseline U-Net

Model	Feature Fusion Strategy	mIoU (%)	# of Tr. Params	% Change
Baseline	No strategy	63.29	39,391,244	-
Exp1: M1	Encoder layers 1, 2, 3, 4	65.43	35,544,848	**+3.38**
Exp2: M2	Encoder layers 1, 2, 3, 4; Bottleneck	64.89	37,643,024	+2.52
Exp3: M3	Encoder layers 1, 2, 3, 4; Bottleneck, Decoder layers 1, 2, 3, 4	64.99	38,340,304	+2.69
Exp4: M4	Decoder layers 1, 2, 3, 4; Bottleneck	63.73	37,643,024	+0.70
Exp5: M5	Decoder layers 1, 2, 3, 4	63.67	35,544,848	+0.60

Note: Models M1 - M5 represent different strategies described in the above paragraph.

Table 3.8 shows the layers that were fused between the SNet and ENet, their performance in terms of mIoU, and complexity in terms of the number of trainable parameters. The results concluded that feature fusion along the encoder pathway omitting the fusion of bottleneck feature maps (M1), produced the best results at 65.43% mIoU without adding much computational complexity. At this stage, the ENet was unfrozen to explore whether fine-tuning could improve performance.

While this nearly doubled the trainable parameters to 70,391,701, the added complexity led to only a marginal improvement of 0.69%, from 65.43% to 65.88%. As a result, the ENet was kept frozen in subsequent experiments.

In the next step, the ASPP module's viability was tested at the bottleneck (abbr. ASPP-B) of the SNet. This module was omitted in the ENet because the existing configuration, which achieved 93% recall (computed using Equation (1.6)), did not require the additional complexity for acceptable edge detection. Finally, the skip connections from the ENet were concatenated with the skip connections of the SNet along the decoder pathway (proposed model, ECASeg) with the rationale that edge context might also guide the upsampling process. Table 3.9 summarizes the test outcomes of the systematic addition of each architectural enhancement after sanity analysis.

Table 3.9: Ablation study of modifications made to achieve the final proposed model, ECASeg

Model	**Enhancement**	**mIoU** (%)	**# of Tr. Params**
Baseline	No enhancement	63.29	39,391,244
2-Att	Two-Attention-Strategy in Layer 3, 4	64.63	34,847,568
M1	Feature Fusion in Encoder blocks	65.43	35,544,848
ASPP-B	ASPP module in Bottleneck	66.23	53,381,392
ECASeg	ENet and SNet Skip Concatenation	**66.53**	56,514,832

3.3.3 Environment

The proposed model is developed using Python 3.10 and its open-source native libraries, along with the PyTorch framework. Model development, training, and evaluation are conducted on a node within the Compute Canada Beluga Cluster. The system is equipped with an Intel Gold 6148 Skylake CPU running at 2.4 GHz, with a memory allocation of 64 GB RAM. Training is performed on a single NVIDIA Tesla V100 GPU, connected via NVLink, with 16 GB memory.

3.3.4 Dataset

The publicly available CamVid Database [1] is employed to train and evaluate the proposed model. It contains collections of high-quality 30Hz driving scene video sequences with dense per-frame annotations. The dataset includes over ten minutes of footage, with semantic labels provided at 1Hz (and partially at 15Hz). There are 701 frames of size 720×960 annotated for 11 object classes. The dataset is divided into train, validation, and test sets, subsuming 369, 100, and 231 samples. The samples are resized, retaining their aspect ratio to have a spatial dimension of 384×512, and their pixel values are normalized to $[0, 1]$. No augmentations are applied during preprocessing.

Table 3.10: Summary of CamVid Database [1]

Class Label	Sky	Building	Column Pole	Road	Sidewalk	Tree	Sign Symbol	Fence	Car	Pedestrian	Bicyclist
% Pixel Occurrence	18.04	20.79	1.04	25.98	6.69	10.76	0.17	0.87	4.15	0.56	0.30

Table 3.10 shows the distribution of the 11 label classes for all labeled data present in the dataset. The class distribution exhibits significant imbalance, reflecting real-world driving scenarios where roads dominate pixel coverage compared to sparse classes like bicycles. Despite its moderate size, the dataset's diverse scenes and comprehensive annotations make it particularly suitable for evaluating urban scene understanding tasks.

3.3.5 Quantitative Analysis

For a fair comparison, all the models were trained from scratch with the same conditions as the proposed model, and their performance was tested on the CamVid holdout test set. The baseline U-Net [3] achieved test accuracy of 90.82%, and mIoU of 63.29%. The Att U-Net [54] performed better than the baseline U-Net, reaching overall test accuracy of 91.21%, and mIoU of 64.55%. However, the proposed model outperforms both prior models with a test accuracy of 91.86%, and mIoU of 66.53%, enhancing the segmentation results by 5.12 percentage improvement compared to the baseline U-Net. Table 3.11 provides a comparison of the proposed model's performance against existing models' performances with respect to test accuracy and mIoU. The respective % change in ECASeg's mIoU from its counterparts is also included. The quantitative analysis demonstrates positive findings, yet its effectiveness is to be validated across a broader range of real-world scenarios.

Table 3.11: Quantitative analysis summary. The % change is calculated w.r.t test mIoU between the proposed model and the respective model in comparison

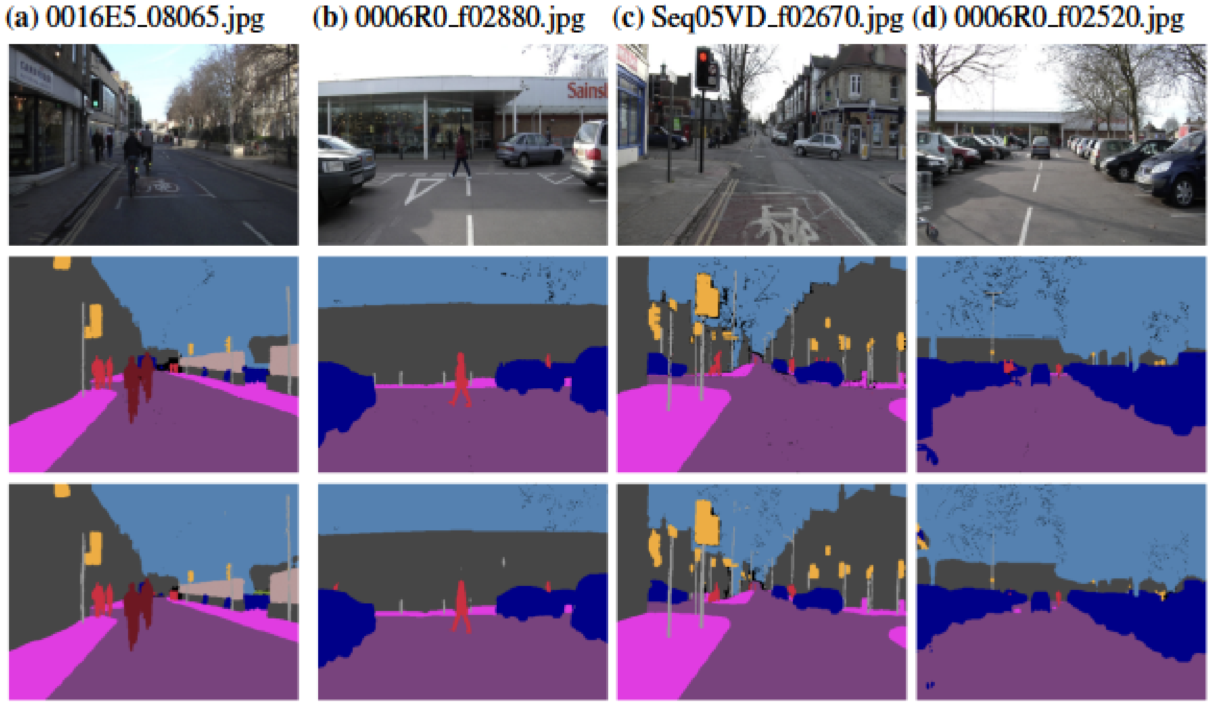
Model	U-Net		SegNet	FCN	LargeFOV	Cyclic	GSAUNet	ECASeg
	Baseline	Att	[36]	[36][2]	[36][40]	Net [67]	[31]	(this work)
Acc (%)	90.82	91.21	84.00	83.90	85.95	91.38	91.44	91.86
mIoU (%)	63.29	64.55	46.30	45.00	50.18	62.98	65.47	66.53
% Change	+5.12	+3.07	+43.69	+47.84	+32.58	+5.64	+1.62	-

3.3.6 Qualitative Analysis

Figure 3.5 on page 46 provides a few samples of the proposed ECASeg's predictions on the test set, demonstrating that even difficult object classes like pedestrians, cyclists, and poles are captured satisfactorily despite varying lighting conditions and degrees of occlusion. Even with the reduction of attention blocks, the proposed model sufficiently captures relevant fine-grain details in the final segmentation masks. Upon further observation, it still exhibits poor object boundary resolution for underrepresented classes and objects with intricate shapes. Besides, the ENet subnet, which was trained to detect Sobel edges, may have introduced noise into the feature maps as Sobel edges are not always effective in isolating precise object boundaries. This can be addressed by leveraging sophisticated learning-based edge detection methods or advanced boundary-aware algorithms that focus on isolating object semantic boundaries rather than merely detecting gradient variations. The under-utilization of the generated edge maps also points to an area for further exploration, where late-fusion strategies and edge-based confidence scores can be employed to guide final pixel-wise predictions.

3.4 Chapter Summary and Findings

This chapter demonstrates the effectiveness of a dual-network framework combining a specialized Sobel edge network and a specialized segmentation network for multi-class segmentation. The experimental results validate the hypothesis that augmenting feature representations with boundary constraints and attention strategies can improve performance over individual models. The proposed



(a) 0016E5_08065.jpg (b) 0006R0_f02880.jpg (c) Seq05VD_f02670.jpg (d) 0006R0_f02520.jpg

Figure 3.5: Prediction samples of the model on CamVid test. Row # 1 - 3: Input images, ground truths, and predicted segmentation maps. The image IDs are provided for reproducibility.

approach achieves a 5.12% improvement over the baseline U-Net and a 3.07% improvement over the baseline Attention U-Net.

Qualitative analysis highlights the strengths of the framework by delineating thin structures like poles as well as distant pedestrians. However, object boundaries are still smoothened slightly, which points to a major limitation. Employing transformation techniques or a late fusion approach could better integrate edge information with semantic features, potentially leading to more refined segmentation results. Given the critical importance of accurate segmentation in applications such as autonomous driving, improving the model's ability to precisely distinguish object boundaries is essential for ensuring safety and reliability. The architectural enhancements introduced in this methodology are promising, but there is ample opportunity for further refinement. While the model's success could be attributed to explicitly preserving and propagating multi-scale edge features, a unified architecture leveraging advanced multi-scale mechanisms may similarly achieve edge-aware feature retention without separate edge detection networks. There is also significant potential in exploring transfer learning in domains like remote sensing or indoor positioning, where the model's adaptability to different visual contexts can be tested. In parallel, addressing the persistent challenges of minority class segmentation, improving boundary accuracy, and optimizing the model to reduce computational overhead and inference time remain key research objectives. This phase of the research addresses the initial hypothesis that integrating supplementary features improves overall feature representation, and a complimentary network can augment the performance of a primary network under the same hyperparameter conditions. The insights gained from this exploration guide the next approach towards achieving the goals outlined in this thesis.

Chapter 4

An Efficient Network with Smart Scaling

4.1 Overview

Transformer-based supervised semantic segmentation approaches have achieved many state-of-the-art (SOTA) performances across several domains. However, such models tend to be large and computationally complex and need large amounts of labeled data for fine-tuning to achieve optimal performance. As we advance in a rapidly evolving AI-driven era, there is a growing responsibility to develop computationally efficient approaches that aspire toward SOTA results while operating within the constraints of finite resources. Interestingly, increasing model size alone does not guarantee better performance [68] as intricate patterns in the data may still be poorly captured, and overfitting becomes more imminent. In this regard, this research focuses on improving the performance of low-complexity models via architectural innovations and additions as opposed to implementing larger CNNs. Enriched architectures exhibit the potential to perform competitively with fewer training samples than Transformer-based counterparts by overcoming limitations such as loss of spatial information, poor global context, and weak feature representation that typically plague CNN approaches. Therefore, this chapter aims to find an efficient and lightweight architecture whose latent abilities can be unlocked by reevaluating the efficiency of each of its components and the incorporation of additional methods like channel and spatial attention, depth-wise separable convolution (DS Conv), and multi-scale feature modules.

4.2 Feature Pyramid Networks

Multi-scale feature modules like Feature Pyramid Networks (FPNs) [69] have significantly advanced semantic segmentation by addressing the inherent challenges of scale variation in pixel-level classification. Traditional convolutional networks suffer from a fundamental limitation: As features propagate deeper into the network, high-resolution spatial details are progressively lost due to pooling and strided convolutions, making small object segmentation particularly difficult. FPNs mitigate this by constructing a hierarchical feature pyramid that combines high-level semantic information from deep layers with fine-grained spatial details from shallow layers. The top-down pathway with lateral connections upsamples high-level semantically rich feature maps and merges them with spatially rich low-level features via lateral skip connections. This retains the fine details necessary for segmentation intricate object boundaries and delicate structures.

Moreover, similar to how backbones extract hierarchical features, FPNs generate multi-resolution feature maps that capture object information at multiple scales, which is beneficial when there are large-scale variations of the same objects in a dataset. By maintaining high-resolution feature maps at multiple levels, FPNs help preserve gradients for small objects, preventing their features from being "washed out" in deep layers. This is crucial for long-range feature propagation, where traditional CNNs fail to retain small object details. Since its introduction, several enhanced FPNs have been constructed to exploit these characteristics. For example, PSPNet [70] appends a pyramid pooling layer to an FPN to capture multi-region context. The EfficientDet framework [71] employs a Bidirectional Feature Pyramid Network (BiFPN) which introduces a combined top-down and bottom-up pathway that preserves low-level details and reinforces them in deeper layers. A learnable feature weighting mechanism dynamically balances contributions across scales to prevent larger objects from dominating the feature space, and efficient cross-scale connections improve gradient flow. The bidirectional design not only strengthens feature reuse but also creates more direct gradient propagation paths, leading to improved optimization during training and improved sensitivity to detail.

4.3 Advanced Attention Mechanisms

Attention mechanisms operate on the principle of modeling long-range dependencies and correlations to enhance task-specific contextual cues while hiding less pertinent features. For example, in the previous chapter, spatial attention was investigated to boost edge-semantic relationships. On the other hand, SENet [72] introduced channel attention by using global average pooling and fully connected layers with nonlinear activations to enhance important channel-wise feature representations. Similarly, the spatial attention module in [51] uses global average and max pooling to highlight important spatial regions and suppress irrelevant features. This work applies both channel and spatial attention to improve segmentation. Spatial attention is computed by applying depth-wise average and max pooling over the channel dimension, as follows.

$$F_{\text{avg}}(h, w) = \frac{1}{C} \sum_{c=1}^{C} F_c(h, w), \tag{4.1}$$

$$F_{\text{max}}(h, w) = \max_{c \in [1, C]} F_c(h, w), \tag{4.2}$$

where $F_c(h, w)$ is the feature at spatial location (h, w) and channel c; and C is the number of channels. The average and max pooled features are concatenated along the channel dimension to form a combined feature map:

$$F_{\text{cat}} = \text{concat}(F_{\text{avg}}, F_{\text{max}}) \in \mathbb{R}^{2 \times H \times W}, \tag{4.3}$$

where H and W are the height and width of the input feature map. Then, a convolution (Conv) with a kernel (K) of size 7×7 is applied to the concatenated feature map:

$$M_s = \sigma(K * F_{\text{cat}}), \quad \text{and} \quad F' = M_s \odot F, \tag{4.4}$$

where $*$ is the Conv operation and $\sigma(\cdot)$ is a sigmoid activation. The resulting spatial attention weights M_s are applied to the original feature map by element-wise multiplication to produce F', the refined feature map.

The channel attention mechanism is applied in parallel and begins with a squeeze operation by aggregating global spatial information using both adaptive average and adaptive max pooling across the spatial dimensions for a given feature map $F \in \mathbb{R}^{C\times H\times W}$:

$$\mathrm{f}_{\mathrm{avg}} = \mathrm{AdptAvgPool}_S(F), \quad \text{and} \quad \mathrm{f}_{\mathrm{max}} = \mathrm{AdptMaxPool}_S(F), \tag{4.5}$$

where $\mathrm{f}_{\mathrm{avg}}, \mathrm{f}_{\mathrm{max}} \in \mathbb{R}^{C\times 1\times 1}$ are pooled feature vectors. Each pooled vector is passed through a shared multi-layer perceptron with 2 fully connected layers and ReLU activation:

$$\mathrm{f}'_{\mathrm{avg}} = W_2\mathrm{ReLU}(W_1\mathrm{f}_{\mathrm{avg}}), \quad \text{and} \quad \mathrm{f}'_{\mathrm{max}} = W_2\mathrm{ReLU}(W_1\mathrm{f}_{\mathrm{max}}), \tag{4.6}$$

where W_1 and W_2 are learnable weight matrices. The channel attention map is computed by summing the two excitation outputs and applying the sigmoid function:

$$M_c = \sigma(\mathrm{f}'_{\mathrm{avg}} + \mathrm{f}'_{\mathrm{max}}), \quad \text{and} \quad F' = M_c \odot F, \tag{4.7}$$

where $M_c \in \mathbb{R}^{C\times 1\times 1}$ is the channel attention mask. The original feature map is reweighted using element-wise multiplication with the channel attention map to produce F', the refined feature map. The outputs of the spatial and channel attention modules are combined using a learnable weight-based fusion mechanism called fast normalized fusion, introduced in [71]:

$$F_{\mathrm{final}} = \frac{w_1 \cdot F_s + w_2 \cdot F_c}{\sum w_i + \epsilon}, \tag{4.8}$$

where w_1, w_2 are learnable fusion weights and $w \leftarrow \mathrm{ReLU}(w)$, ensuring a dynamic balance between spatial and channel attention outputs. ϵ is a small positive constant to prevent division by zero, and F_{final} is the refined feature map that combines spatial (F_{s}) and channel (F_{c}) attention refined outputs. This fusion strategy adaptively emphasizes the most relevant features, enhancing the representation power of the network. The resulting module is named the Dual Attention Refinement (DAR) Module.

4.4 Methodology

The contributions of the proposed methodology are summarized as follows: (i) Reducing model complexity with a lightweight backbone and enhancing traditional Deeplabv3 capabilities with the strategic addition of attention mechanisms to highlight important features, (ii) introducing a depth-wise and point-wise feature pyramid module for capturing spatial and semantic context, and (iii) employing two complementary modules to extract distinct types of multiscale information to fortify the decoding pathway using a strategic feature fusion. Figure 4.1 illustrates an overview of the proposed semantic segmentation model.

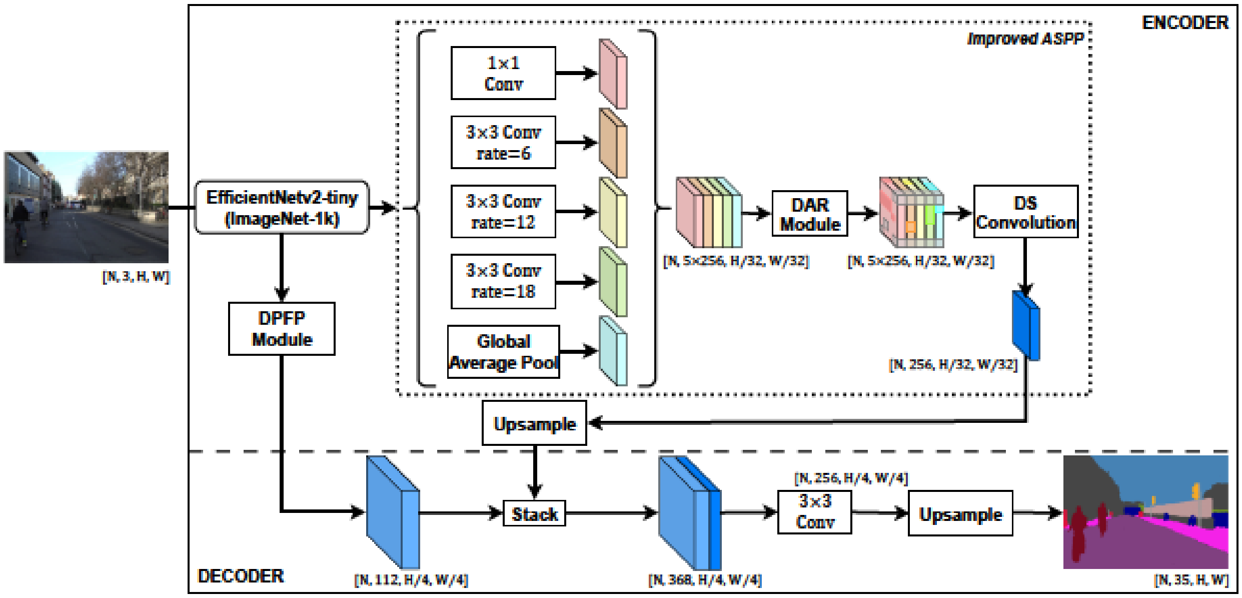


Figure 4.1: An architectural overview of the proposed model.

4.4.1 The Backbone Network

Traditional implementations of the DeepLabv3 setup involve highly complex and computationally heavy backbones which can make fine-tuning the architecture challenging. In this work, a lightweight CNN–EfficientNet-v2(Tiny)–pre-trained on ImageNet-1k is considered as the backbone. It has fewer parameters and a quicker training time than the popular ResNet and Xception implementations.

4.4.2 The Improved ASPP Network

The ASPP network includes a DAR module (Section 4.3) and a depth-wise separable convolution (DS Conv) to condense DAR-refined features. The ASPP module uses global average pooling and 3 × 3 Conv with dilation rates of 6, 12, and 18, each followed by batch normalization and ReLU, to extract semantic context, which is further refined by the DAR module. The output is reshaped using DS Conv and upsampled 4×. In the decoder, a depth-wise point-wise feature pyramid (DPFP) captures multi-scale semantic and spatial information from different stages in the backbone. These features are fused with ASPP outputs via skip connection and channel-wise concatenation.

4.4.3 The Depth-wise Point-wise Feature Pyramid (DPFP) Module

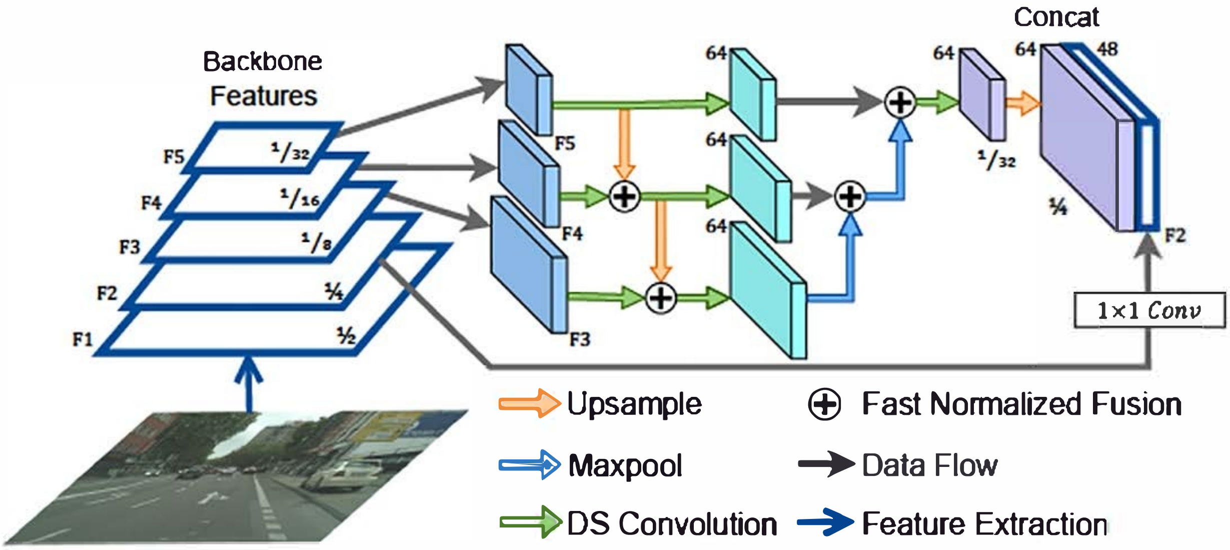


Figure 4.2: Design specifications of the Depth-wise Point-wise Feature Pyramid (DPFP) module.

The backbone extracts hierarchical information in 5 stages, where each stage progressively halves the spatial resolution while increasing the number of channels. Features extracted by the last layer of the backbone (F_5) are passed to the improved ASPP, while feature outputs from the four later layers of the backbone, F_2 to F_5, are fed to the DPFP module. The DPFP module aligns features from a feature pyramid constructed using the last three backbone features, F_3 to F_5, through a series of depth-wise separable (DS) convolutions, upsampling/ downsampling operations, and fast normalized fusion of features at the same scale. The aligned features are concatenated with 1×1 convolved F_2 along the channel dimension as depicted in Figure 4.2.

4.4.4 A Smart Scaling for Multi-level Feature Learning

The resulting features from the multi-scale modules, DPFP and ASPP, are stacked channel-wise again and are followed by a 3×3 convolution, BN, and ReLU activation function before being upsampled by a factor of 4 using bilinear interpolation to restore the segmentation predictions to the input image size. A Softmax classifier refines the output to produce a semantic class probability map representing the objects found in the input image. In this approach, the ASPP and DPFP act as two complementary modules to extract distinct types of multiscale information.

The first module utilizes multiple atrous rates applied to the deepest backbone layer, which are inherently rich in semantic information. The varying atrous rates enable the capture of long-range dependencies within these semantically rich features at multiple spatial scales, enhancing global context understanding. The second module captures multi-scale information by integrating features from multiple backbone layers, which contain both spatial and semantic information at different levels of abstraction. This integration is performed through fast normalized fusion, facilitating the effective distillation of spatial and semantic features across different stages of feature extraction. As a result, spatial information is preserved and propagated over longer ranges when passed as skip connections during the decoding process.

The fusion of these two feature representations introduces a trade-off between different multi-scale information sources. The extracted features vary not only in terms of spatial scales—defined by receptive field variations—but also in terms of localization and semantic content. The latter includes both detailed object characteristics and the contextual relationships between objects and their spatial positioning within the image. Through this process, the network learns to adaptively balance and propagate these features, resulting in a “smart scaling” mechanism that dynamically adjusts the interaction between early and deep feature representations. This approach enhances both spatial consistency and semantic coherence throughout the network, leading to improved segmentation performance.

4.4.5 Environment

The model is implemented in Python 3.10 using open-source libraries and the PyTorch framework. Development, training, and evaluation are performed on Compute Canada's Beluga, Narval, and Mist clusters. Beluga features an Intel Gold 6148 CPU (2.4GHz) and $4\times$ NVIDIA Tesla V100 GPUs (16GB). Narval uses an AMD CPU (2.65GHz) and $4\times$ NVIDIA A100SXM4 GPUs (40GB). Mist is equipped with an IBM Power9 SMT4 CPU and $4\times$ NVIDIA V100 Volta GPUs (32GB).

4.4.6 Datasets

Cityscapes [47]

It contains diverse daytime urban driving scenes throughout several seasons with varied weather conditions. There are 5,000 finely annotated images of size 1024×2048 and divided into training, validation, and test sets, which contain 2975, 500, and 1525 samples, respectively. Although there are 30 semantic classes, only 19 are considered in the evaluation. Images are resized to 768×1536 for training. Since the test ground truths are not available publicly, the validation set is used as a holdout set in this work. A small subset of the train data (275 samples) without overlap is used as a validation set to track train progress. Inference is performed on non-augmented images that are only resized and normalized. Also, for preliminary experimentation and ablation study, a miniset comprising 1000 training samples and 275 non-overlapping validation samples from the training set was created, with the original validation set used in its entirety as a holdout. All other protocols for the miniset remain consistent with those applied to the full set.

CamVid [1]

For cross-validation, the CamVid database is also utilized. Dataset details are described in Section 3.3.4. The images are resized to 704×960 and all other preprocessing steps and inference protocols followed for the Cityscapes dataset are applied to the CamVid dataset as well. A batch size of 4 is used for training baseline and proposed models.

LoveDA [6]

For cross-domain validation, the Land-cOVEr Domain Adaptive semantic segmentation dataset is utilized. It contains high spatial resolution remote sensing images of size 1024 × 1024, captured in Nanjing, Changzhou, and Wuhan. It contains 2713 urban scenes and 3274 rural scenes that were collected from 18 spatially independent areas using advanced ArcGIS geo-spatial software. The dataset of 5987 images is split into train, validation, and test sets with 2522, 1669, and 1796 images, respectively. Evaluation of 7 semantic land-cover classes–background, building, road, water, barren, forest, agriculture–can be conducted using the publicly available validation set or submitted to the evaluation server for test results since the test ground truths are private. The images are not resized, but all other preprocessing steps and inference protocols followed for the Cityscapes dataset are applied to the LoveDA dataset as well.

Figure 4.3 shows a sample input image and its corresponding ground truth mask from each dataset used in this work.

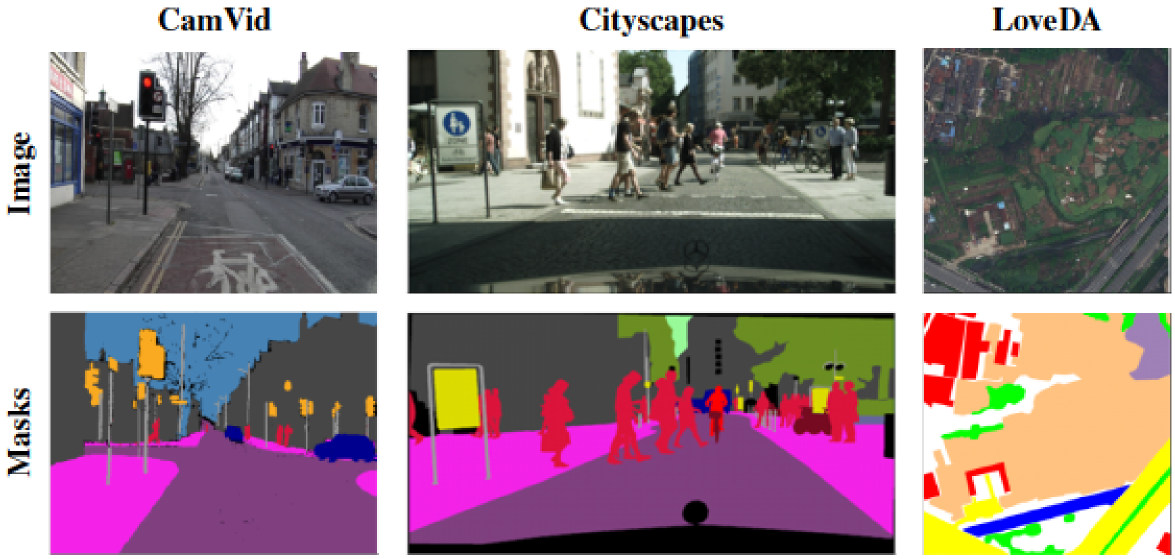


Figure 4.3: A sample input image and its corresponding ground truth mask across datasets used in this work.

4.4.7 Data Preprocessing: An Online Data Augmentation Strategy

To ensure the model learns robust feature representations that are invariant to perspective shifts and changes in photometric appearance [55, 57], a range of color and geometric augmentations are applied. These include color jitter, random noise, blur, horizontal flip, shear, random rotation, and crop. These augmentations are applied only to the training data and are generated on the fly during data loading with a random probability. Also note that inputs are normalized based on ImageNet-1k normalization for better convergence and generalization [44].

4.4.8 Model Training

The miniset is utilized to run preliminary experiments, with the best-performing configurations then applied during the full training of the proposed model. The train hyperparameters (cf. Table 4.1) remain consistent across all experiments and are maintained in the final training phase described in the following paragraph. However, due to the reduced size of the miniset, the 'mini' experiments are conducted for only 50 epochs. The impact of key enhancements incorporated into the final training phase is assessed through ablation with the miniset. Figure 4.4 presents the validation progress over 50 epochs of the ablation experiments discussed in Section 4.4.9.

Table 4.1: Train Hyperparameters of the Proposed Method on Cityscapes

Hyperparameter	Value
Learning rate	0.0001
Optimizer	Adam
# of Epochs	100
Batch size	4
Loss Function	Categorical Cross-entropy Loss

For the final training phase, the proposed model is trained from scratch using Adam optimizer [73] with a learning rate of 0.0001, minimizing the categorical cross-entropy objective function and other hyperparameters outlined in Table 4.1. The objective function is defined in Equation (1.1) and evaluation is done using mIoU and efficiency metrics (Equations (1.5), (1.7), respectively). Figure 4.5 illustrates the training and validation progress over 100 epochs; the model

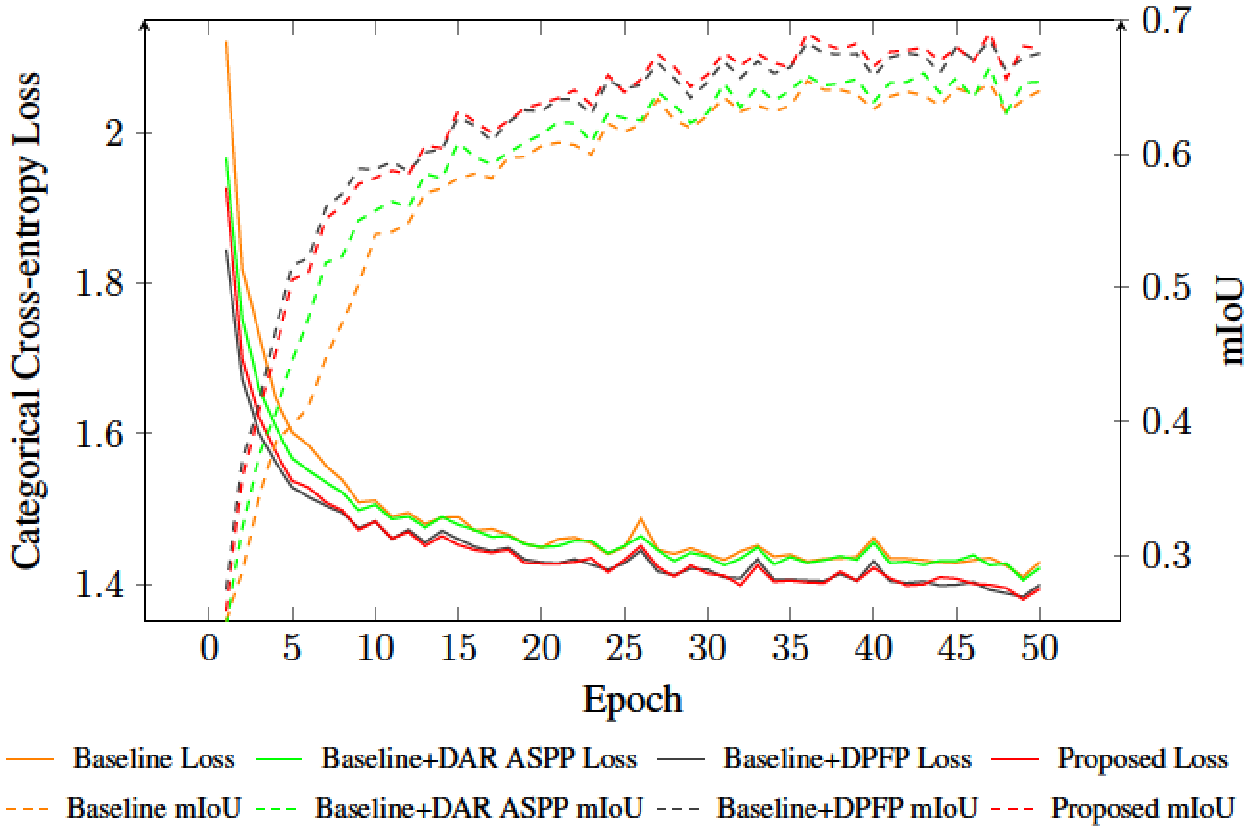


Figure 4.4: Training progress plots of the ablation experiments on Cityscapes miniset.

is first trained for 95 epochs with the train (2700 samples) and validation subsets (275 samples), followed by 5 epochs using the complete train set (2975 samples). The model achieved optimal mIoU and minimal loss at the 98th epoch.

4.4.9 Comparative Study

Initially, several lightweight backbones were tested in a DeepLabv3 setup. The EfficientNetv2 pre-trained on ImageNet1k in tiny and small configurations produced the best results. Still, the tiny configuration was chosen because the latter only performed marginally better for a significant parameter increase. This model forms the baseline for a meaningful analysis of the effectiveness of the enhancements introduced in the proposed method. Table 4.2 summarizes the ablation study of the key components in the proposed framework. The ablation process begins with training the baseline model on the train miniset while monitoring validation loss to mitigate overfitting, fol-

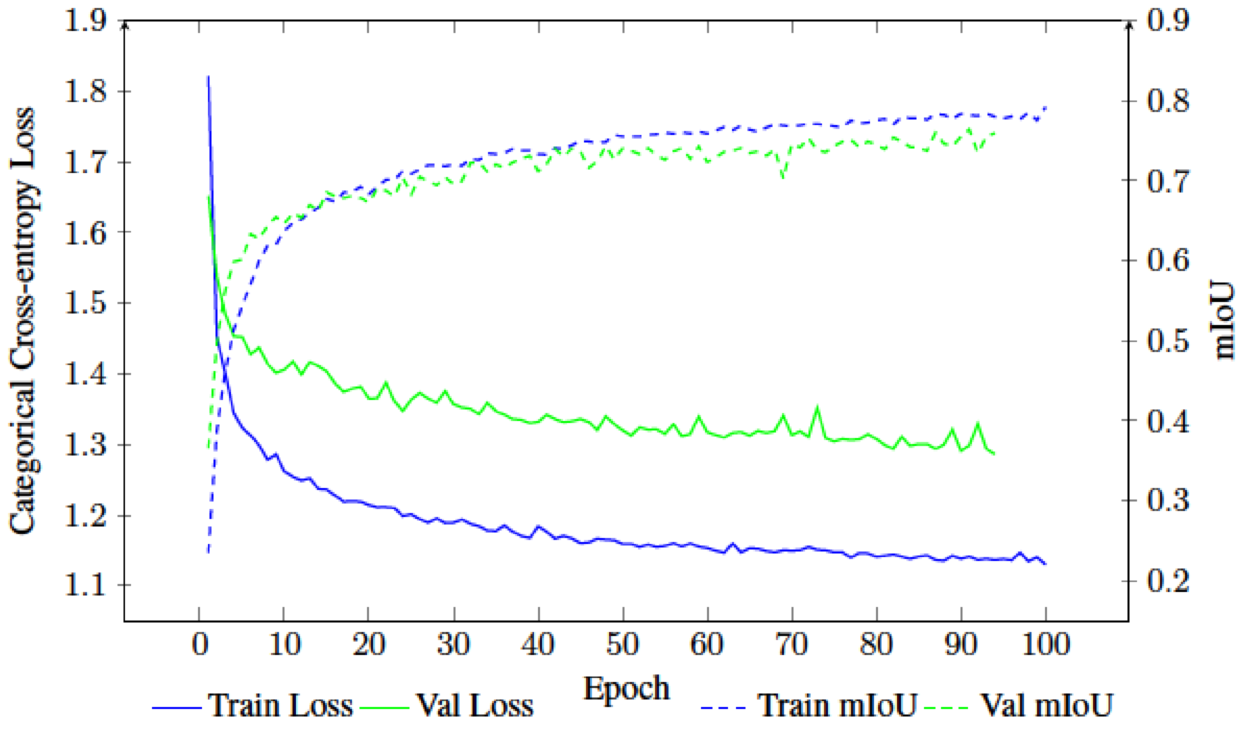


Figure 4.5: Training progress plot of the proposed podel for 100 epochs on Cityscapes.

lowed by sequentially incorporating the improved ASPP, the DPFP module with standard ASPP, and finally, both the DPFP module and the improved ASPP. For a fair comparison across experiments, the best mIoU achieved by each configuration and the corresponding epoch at which it was obtained are reported. The results demonstrate that the proposed configuration achieves 67.44% mIoU on the Cityscapes validation set, surpassing all intermediate versions.

Table 4.2: Ablation Study on Cityscapes Miniset

Experiment	**mIoU** (%)	**Epoch**
Baseline	62.75	46
Baseline + Improved ASPP	63.44	46
Baseline + DPFP	66.87	46
Baseline + DPFP + Improved ASPP	67.44	41

Thus, for the final training phase, the baseline and proposed models were trained from scratch on the full train set. The proposed model achieved 73.78% mIoU and the baseline achieved 67.55% on the Cityscapes validation dataset. Table 4.3 shows an improvement across almost all classes by the proposed method. The proposed model has a very slight increase in Params and GFLOPS after the architectural additions, achieving an increase of 6.23 percentage points in mIoU%.

Table 4.3: Class-wise performance comparison in Cityscapes Validation Set

Method	Performance (mIoU %) w.r.t Semantic Segmentation Classes																		
	Bicycle	*Building*	*Bus*	*Car*	*Fence*	*Motorcycle*	*Person*	*Pole*	*Rider*	*Road*	*Sidewalk*	*Sky*	*Terrain*	*Traffic Light*	*Traffic Sign*	*Train*	*Truck*	*Vegetation*	*Wall*
Baseline	0.64	0.90	0.71	0.93	0.57	0.42	0.70	0.38	0.49	0.97	0.82	0.90	0.57	0.53	0.66	0.61	0.66	0.90	0.47
Proposed	0.72	0.91	0.83	0.94	0.58	0.59	0.77	0.53	0.54	0.96	0.77	0.93	0.61	0.62	0.72	0.76	0.80	0.91	0.50

Table 4.4: Quantitative analysis on Cityscapes. Change is calculated w.r.t mIoU of the baseline

Model	**mIoU** (%) ↑	**Params** (M) ↓	**GFLOPS** ↓	**Change** ↑	**Efficiency** (%) ↑
Baseline	67.55	14.9	84.3	–	–
SwiftNetRN-18 [74]	65.30	11.8	52	-2.25	–
GSAUNet [31]	71.92	NA	NA	+4.37	–
KMaX DeepLab with ResNet50 [30]	79.70	56	434	+12.15	1.60
Panoptic-DeepLab with Xception-71 [29]	80.50	47	548	+12.95	1.41
SegFormer with MiT-B5 [20]	82.40	85	1460	+14.85	0.53
KMaX DeepLab with ConvNeXt-L [30]	83.50	232	1673	+15.95	0.40
SS-DeepSeg (this work)	73.78	15.6	110.2	+6.23	4.74

↓ - *Lower is better,* ↑ - *Higher is better*

'NA' denotes that the information is not available in the literature.

Table 4.4 summarizes the comparative analysis on Cityscapes, in which only methods that do not use extra training data and test-time augmentation are considered for a fair comparison. It is evident that while the proposed model does not achieve the highest segmentation mIoU, it demonstrates superior efficiency at 4.74%, i.e. achieves greater performance gains relative to the additional computational cost incurred. Cross-validating the approach on the second dataset, CamVid, the proposed model achieves a test mIoU of 76.96% after 119 epochs when training from scratch with only the CamVid train set. The pretraining on the Cityscapes train set first, demonstrated an increase in test mIoU at 77.22%. The test results on CamVid compared against competitive models are summarized in Table 4.5.

Table 4.5: Performance of various models on CamVid Test

Model	mIoU(%)
GSAUNet [31]	65.47
SwiftNetRN-18 [74]	65.70
CyclicNet [67]	62.98
VideoGCRF [75]	67.00
†VideoGCRF [75]	75.20
SS-DeepSeg (this work)	76.96
†SS-DeepSeg (this work)	77.22

†: *Pretrained on additional data, i.e. Cityscapes train set.*

Validating the model's performance on LoveDA, the proposed model achieves a competitive test mIoU of 51.71% after training from scratch for 33 epochs. The state-of-the-art results are 57.36% mIoU achieved using an ensemble model that combines three UNet architectures whose individual performances are also included for comparison alongside contemporary models in Table 4.6. While the proposed approach does not achieve SOTA, ensemble techniques are often bulky and computationally expensive. The ensemble technique does not report its parameters or GFLOPS and demonstrates only marginal improvement in performance over its individual components, failing to justify the increased size and complexity of a multi-transformer approach.

Table 4.6: Performance of various models on LoveDA Test

Method	Per-class IoU (%)							mIoU (%)
	Background	Building	Road	Water	Barren	Forest	Agriculture	
DeepLabV3+ (ResNet50) [6]	42.97	50.88	52.02	74.36	10.40	44.21	58.53	47.62
UNet++ (ResNet50) [6]	42.85	52.58	52.82	74.51	11.42	44.42	58.80	48.20
HRNet (W32) [6]	44.61	55.34	57.42	73.96	11.07	45.25	60.88	49.79
UNetFormer (ResNet18) [76]	45.70	58.80	54.90	79.60	20.10	46.00	62.50	52.40
‡UNet (ConvFormer-M36) [7]	45.17	60.81	58.00	81.48	28.27	46.90	62.96	54.80
‡UNet(EfficientNet-B7) [7]	45.88	57.57	58.92	80.69	28.24	47.88	66.31	55.07
‡UNet (MaxViT-S) [7]	48.59	60.47	63.40	81.17	27.02	48.10	64.40	56.16
⋆UNet Ensemble [7]	49.09	61.12	63.71	82.36	30.15	49.29	65.82	57.36
SS-DeepSeg (this work)	44.41	51.53	53.49	76.62	23.96	44.65	64.14	51.71

⋆: *Ensemble with models denoted by ‡ symbol.*

4.4.10 Qualitative Analysis

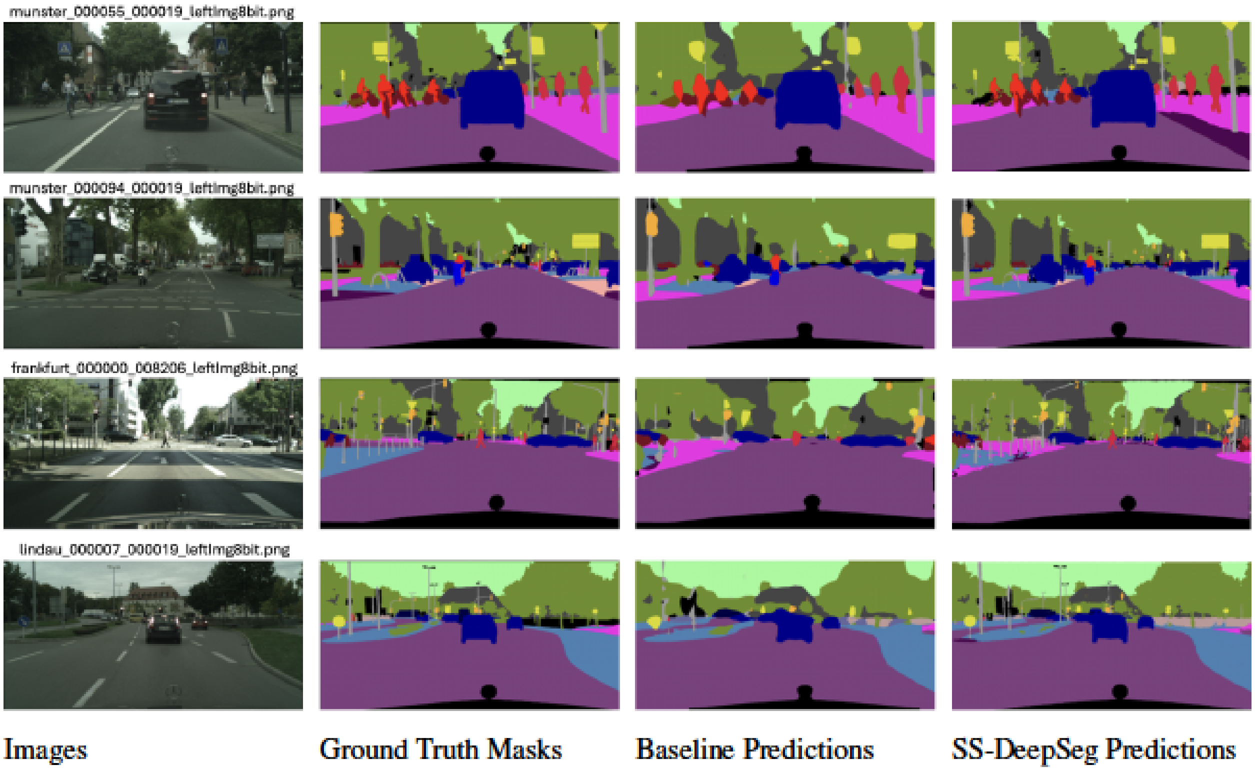

munster_000055_000019_leftImg8bit.png
munster_000094_000019_leftImg8bit.png
frankfurt_000000_008206_leftImg8bit.png
lindau_000007_000019_leftImg8bit.png
Images
Ground Truth Masks
Baseline Predictions
SS-DeepSeg Predictions

Figure 4.6: Sample of predictions on Cityscapes validation set. Col. # 1 - 4: Input images, ground truths, baseline predictions, and proposed model predictions, respectively.

The strengths of the proposed approach are evident in the qualitative results. The predictions on the Cityscapes validation set are shown in Figure 4.6. Here, challenging scenes from the Cityscapes validation set are used to compare performance. The selected images (IDs are provided for reproducibility) include variations in illumination, low-light conditions, high object density, occlusions, and distant small objects. It is immediately apparent that the baseline model struggles with severe boundary smoothing, a limitation commonly seen in DeepLab-based models, leading to imprecise segmentation. In contrast, the proposed model demonstrates robustness to illumination changes, improved performance on small objects, and superior boundary delineation. Notably, it effectively segments thin structures such as poles, highlighting its ability to capture fine-grained details. The proposed model's strengths can be attributed to its hierarchical feature balancing and attention-guided refinement that effectively leverage multi-scale spatio-semantic features. Unlike the baseline, which primarily relies on a global context aggregation mechanism, the proposed approach incorporates a specialized weighting mechanism designed to preserve and propagate vital spatial details (DPFP module) throughout the pipeline while attentively emphasizing contextual relationships (improved ASPP with DAR module). This targeted design enhances fine-grained segmentation, particularly in regions where the baseline struggles to capture delicate details.

Predictions of the proposed approach on the CamVid test set are illustrated in Figure 4.7. The model demonstrates robust performance across most scenes, even under varying lighting conditions. It effectively captures thin structures like poles and accurately identifies intricate object classes like pedestrians, even when they are at considerable distances. However, some limitations persist, object boundaries appear slightly eroded and there are misclassifications between visually similar categories, such as wall/building, road/sidewalk, pedestrian/bicyclist.

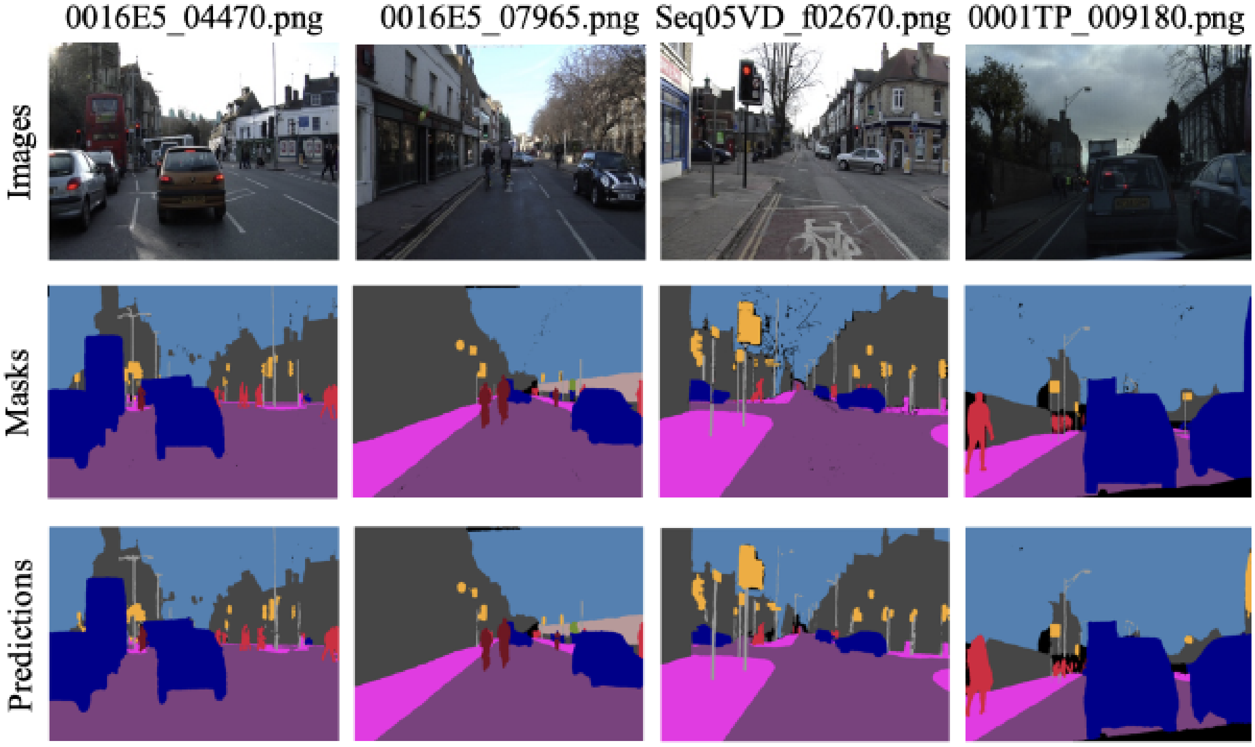


Figure 4.7: Predictions on the CamVid Test set, IDs provided for reproducibility.

Predictions on the LoveDA validation set are shown in Figure 4.8. LoveDA contains challenging urban and rural scenes and the relatively low mIoU value substantiates the level of difficulty. From observation, the model segments water bodies (blue) well which is supported by the high class mIoU. However, boundary erosion of varying degrees is in evident across all classes. Forests and agricultural areas (green and beige, respectively) are difficult to differentiate, possibly due to their spectral similarities. Small and intricate objects like building clusters (red) and sparse trees (green) are particularly challenging leading to missed classifications, and background class (white) exhibits high intra-class variance resulting in false positives.

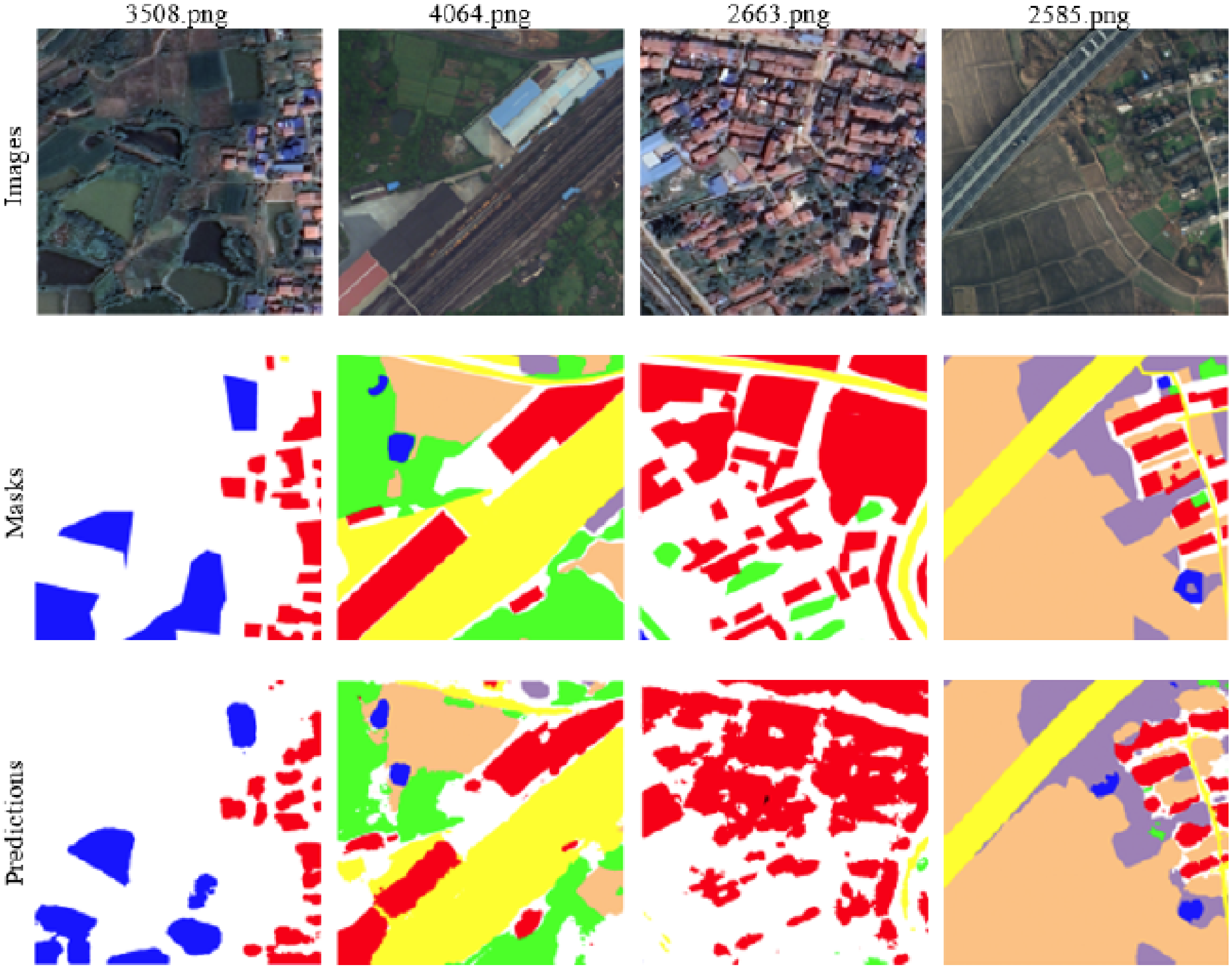


Figure 4.8: Proposed method's predictions on the LoveDA validation set. Image IDs provided for reproducibility.

4.5 Chapter Summary

This work presents a lightweight segmentation model that achieves improved performance compared to the baseline while maintaining efficiency in terms of computational cost and parameters. Despite not reaching SOTA performance, the model demonstrates a promising trade-off between accuracy and efficiency, particularly when contrasted with transformer-based models that require significantly higher FLOPs and parameter counts. From the analysis, small architectural modifications, i.e., 0.7M additional parameters, can yield notable performance improvements while overcoming certain network limitations. Additionally, qualitative evaluation highlights that the model effectively delineates most objects; however, challenges persist in poorly represented classes and complex structures, an inherent issue in driving datasets. To address these limitations, future research avenues include exploring data-centric strategies such as advanced augmentation strategies to increase the occurrence of underrepresented samples. These include techniques like generative adversarial networks or diffusion-based synthesis of realistic rare-class samples. Another method is dynamic adaptive upsampling of rare classes during dataloading to unsure a more balanced exposure during training. Class-aware augmentation methods like CutMix can be adapted to strategically insert rare-class regions into images, thereby increasing the rare-class instances during training. Additionally, architectural refinements like neural architecture search-optimized decoding and training optimization techniques such as pre-text task training and self-supervised learning are possible strategies to enhance generalization. By incorporating such strategies to diversify training data, and further refining architectural choices, the proposed approach can be enhanced to bridge the gap between efficiency and accuracy in real-world segmentation applications.

Chapter 5

Conclusion

This thesis methodologically investigated two approaches: a multi-model and a lightweight label-efficient solution to addressing key challenges in complex scene semantic segmentation. Through effective model design and iterative improvements, the proposed approaches achieve improved accuracy-complexity trade-offs.

The contributions of this thesis include the development of novel architectural components, namely the DPFP module and DAR module, and the systematic application of augmentation strategies to improve label efficiency. The experimental analysis on benchmark datasets shows that the proposed strategies can lead to real-time performance on edge devices for autonomous systems while maintaining accurate segmentation results. However, the performance in rare classes remains suboptimal due to dataset bias, which points to the need for more advanced sampling and augmentation strategies to boost representation for rare classes. Building upon this research, several critical avenues emerge as future directions to advance efficient and sustainable visual perception systems.

First, developing adaptive class imbalance strategies, such as learned loss weighting and constrained diffusion-based synthesis, will maximize the utility of labeled data while addressing scene heterogeneity.

Second, the fundamental challenge of scaling lightweight architectures requires innovations in dynamic capacity allocation, potentially through gated modular networks combined with task-aware distillation protocols that preserve model efficiency during domain transfer. In tandem, Neural Architecture Search can play a pivotal role in identifying optimal gating structures and modular pathways, thus, acting as a complimentary tool that enables optimal configuration of dynamic capacity allocation ensuring that lightweight models maintain robustness across domains.

Most crucially, the field's transition to 3D perception (LiDAR) demands a shift toward sustainable point cloud processing, where techniques like hardware-aware sparse convolutions, geometric-fidelity preserving compression, and carbon-accounted training regimens must co-evolve with emerging sensor technologies. This necessitates new benchmarking frameworks that jointly quantify accuracy, latency, and energy expenditure across the model life cycle, from training algorithms to deployment-aware quantization schemes. Realizing this vision will require tight integration of architectural innovations, spatiotemporal representation learning, and environmentally conscious design principles, ultimately enabling high-performance yet sustainable perception systems for next-generation vision systems.

Bibliography

[1] G. J. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-definition ground truth database,” *Patt. Recog. Lett.*, 2008.

[2] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Trans. on Patt. Analy. and Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Med. Image Comput. and Comp.-Assist. Interv.–MICCAI 2015: 18th Intl. Conf., Munich, Germany, 2015, Proc., Part III 18.* Springer, 2015, pp. 234–241.

[4] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” in *Intl. conf. on learning representations*, 2015.

[5] N. Jahan, T. Akilan, and T. M. Nguyen, “Improved semi-supervised attention gan for semantic segmentation,” in *IEEE Pacific Rim Conf. on Comm., Comp. and Signal Process.*, 2024, pp. 1–6.

[6] J. Wang, Z. Zheng, A. Ma, X. Lu, and Y. Zhong, “Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation,” in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung, Eds., vol. 1. Curran Associates, Inc., 2021. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/4e732ced3463d06de0ca9a15b6153677-Paper-round2.pdf

[7] I. Dimitrovski, V. Spasev, S. Loshkovska, and I. Kitanovski, "U-net ensemble for enhanced semantic segmentation in remote sensing imagery," *Remote sensing (Basel, Switzerland)*, vol. 16, no. 12, pp. 2077–, 2024.

[8] H. Wang, C. Liu, Y. Cai, L. Chen, and Y. Li, "Yolov8-qsd: An improved small object detection algorithm for autonomous vehicles based on yolov8," *IEEE Transactions on Instrumentation and Measurement*, vol. PP, pp. 1–1, 01 2024.

[9] C. Wang, B. Liu, C. He, L. Cong, and S. Wan, "Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–12, 11 2023.

[10] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman, "Unsupervised semantic segmentation by distilling feature correspondences," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=SaKO6z6Hl0c

[11] C. Kim, W. Han, D. Ju, and S. J. Hwang, "Eagle: Eigen aggregation learning for object-centric unsupervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 3523–3533.

[12] S. Karimijafarbigloo, R. Azad, A. Kazerouni, Y. Velichko, U. Bagci, and D. Merhof, "Self-supervised semantic segmentation: Consistency over transformation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2654–2663.

[13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9726–9735.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231591445

[15] L. Chen, W. Wu, C. Fu, X. Han, and Y. Zhang, “Weakly supervised semantic segmentation with boundary exploration.” Berlin, Heidelberg: Springer-Verlag, 2020, p. 347–362. [Online]. Available: https://doi.org/10.1007/978-3-030-58574-7_21

[16] Z. Wang, J. Zhang, Z. Liu, S. Chen, and D. Lu, “An improved u-net network for medical image segmentation,” in *2023 IEEE 10th International Conference on Cyber Security and Cloud Computing (CSCloud)/2023 IEEE 9th International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2023, pp. 292–297.

[17] J. Li, J. Shen, P. Xie, Z. Wei, X. Wang, L. Zhang, and Y. Zhang, “Attention-guided network with densely connected convolution for skin lesion segmentation,” *IEEE Trans. on Medical Imaging*, vol. 39, no. 5, pp. 1446–1457, 2020.

[18] X. Zhang, G. Gao, and Z. Wu, “Attention dense-u-net for automatic breast mass segmentation,” *Journal of Healthcare Engineering*, vol. 2020, p. Article ID 6157657, 2020.

[19] P. Chowdhary, *Natural Language Processing*. Springer Nature, 04 2020, pp. 603–649.

[20] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” in *Neural Info. Process. Sys. (NeurIPS)*, 2021.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[22] A. Adam and C. Ioannidis, “Automatic road sign detecion and classification based on support vector machines and hog descriptos,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-5, pp. 1–7, 2014. [Online]. Available: https://isprs-annals.copernicus.org/articles/II-5/1/2014/

[23] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation,” in *Computer Vision*

– *ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–15.

[24] M. Islam, D. Zhang, and G. Lu, "Automatic categorization of image regions using dominant color based vector quantization," in *Proceedings of the Digital Image Computing: Techniques and Applications*, A. Robles-Kellyand and T. Caelli, Eds. United States of America: IEEE, Institute of Electrical and Electronics Engineers, 2008, pp. 191 – 198, digital Image Computing Techniques and Applications 2008, DICTA 2008 ; Conference date: 01-12-2008 Through 03-12-2008.

[25] M. Y. Yang and W. Förstner, "A hierarchical conditional random field model for labeling and classifying images of man-made scenes," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 196–203.

[26] D. Hartmann, D. Müller, I. S. Rey, and F. Kramer, "Assessing the role of random forests in medical image segmentation," *ArXiv*, vol. abs/2103.16492, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232417880

[27] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using k -means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764–771, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915014143

[28] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *IEEE/CVF Conf. on Comp. Vis. and Patt. Recog. (CVPR)*. IEEE, 2020, pp. 12 472–12 482.

[30] Q. Yu, H. Wang, S. Qiao, M. Collins, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "k-means mask transformer," in *European Conf. on Computer Vision*. Springer, 2022, pp. 288–307.

[31] S. P. S. Sangita B. Nemade, "Semantic segmentation using gsaunet," *ICT Express*, vol. 9, no. 1, pp. 1–7, 2023.

[32] N. Jahan, T. Akilan, and T. Suresh, "Improving pavement crack segmentation using attention mechanism and self-gated activation," in *IEEE Canadian Conf. on Electrical and Computer Engg. (CCECE)*. IEEE, 2024, pp. 853–858.

[33] X. Li and D. Chen, "A survey on deep learning-based panoptic segmentation," *Digital Signal Processing*, vol. 120, p. 103283, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200421003225

[34] Z. Zhou, M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support - 4th International Workshop, DLMIA 2018 and 8th International Workshop, ML-CDS 2018 Held in Conjunction with MICCAI 2018*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, 2018, pp. 3–11.

[35] E. Sahragard, H. Farsi, and S. Mohamadzadeh, "Advancing semantic segmentation: Enhanced unet algorithm with attention mechanism and deformable convolution," *PloS one*, vol. 20, no. 1, pp. e0 305 561–, 2025.

[36] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. on Patt. Analy. and Mach. Intell.*, vol. 39, no. 12, 2017.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[38] S. Fang, B. Zhang, and J. Hu, "Improved mask r-cnn multi-target detection and segmentation for autonomous driving in complex scenes," *Sensors*, vol. 23, no. 8, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/8/3853

[39] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, oct 2021. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.2983686

[40] 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. on Patt. Analys. and Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[41] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," in *Proc. of the IEEE Conf. on CVPR*, 2017, pp. 5067–5075.

[42] ——, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. of the Euro. conf. on comp. vis. (ECCV)*, 2018, pp. 801–818.

[43] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:17127188

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[45] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[46] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[47] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conf. on CVPR*, 2016, pp. 3213–3223.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural info. process. sys.*, 2017, pp. 5998–6008.

[49] H. Ding, C. Liu, S. Wang, and X. Jiang, "Vlt: Vision-language transformer and query generation for referring segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 6, pp. 7900–7916, 2023.

[50] S. Erişen, "Sernet-former: Segmentation by efficient-resnet with attention-boosting gates and attention-fusion networks," in *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, 2024, pp. 1–6.

[51] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. of the European Conf. on Comp. Vis. (ECCV)*, September 2018.

[52] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *2020 IEEE/CVF Conf. on Comp. Vis. and Patt. Recog. (CVPR)*, 2020, pp. 11 531–11 539.

[53] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[54] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 220–228.

[55] S. K. Ravindran and C. Tomasi, “Randmsaugment: A mixed-sample augmentation for limited-data scenarios,” *arXiv preprint arXiv:2311.16508*, 2023.

[56] E. Hoffer, T. Ben-Nun, I. Hubara, N. Giladi, T. Hoefler, and D. Soudry, “Augment your batch: Improving generalization through instance repetition,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[57] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in *Advances in Neural Info. Process. Sys.*, vol. 33, 2020, pp. 18 613–18 624.

[58] Y. Zhou, H. Xu, W. Zhang, B. Gao, and P.-A. Heng, “C3-semiseg: Contrastive semi-supervised segmentation via cross-set learning and dynamic class-balancing,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 7016–7025.

[59] M. Qi, Y. Wang, J. Qin, and A. Li, “Ke-gan: Knowledge embedded generative adversarial networks for semi-supervised scene parsing,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5232–5241.

[60] O. Hahn, C. Reich, N. Araslanov, D. Cremers, C. Rupprecht, and S. Roth, “Scene-centric unsupervised panoptic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[61] J. Jin, W. Zhou, R. Yang, L. Ye, and L. Yu, “Edge detection guide network for semantic segmentation of remote-sensing images,” vol. 20, pp. 1–5, 2023, conference Name: IEEE Geoscience and Remote Sensing Letters.

[62] H. Ma, H. Yang, and D. Huang, “Boundary guided context aggregation for semantic segmentation.”

[63] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection.”

[64] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, “Boundary-Aware Feature Propagation for Scene Segmentation,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 6818–6828.

[65] J.-J. Liu, Q. Hou, and M.-M. Cheng, “Dynamic feature integration for simultaneous detection of salient object, edge and skeleton,” *IEEE transactions on image processing*, vol. 29, pp. 1–1, 2020.

[66] J. Shi, Z. Gao, and A. Wang, “Multi-scale image semantic segmentation based on aspp and improved hrnet [j],” *Chinese Journal of Liquid Crystals and Displays*, vol. 36, no. 11, pp. 1497–1505, 2021.

[67] G. Wu and Y. Li, “Cyclicnet: an alternately updated network for semantic segmentation,” *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 3213–3227, 2021.

[68] Z. Wu, C. Shen, and A. van den Hengel, “Wider or deeper: Revisiting the resnet model for visual recognition,” *Patt. Recog.*, pp. 119–133, 2019.

[69] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[70] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. of the IEEE Conf. on CVPR*, 2017, pp. 2881–2890.

[71] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in *IEEE/CVF Conf. on Comp. Vis. and Patt. Recog. (CVPR)*, June 2020, pp. 10 778–10 787.

[72] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. of the IEEE conf. on comp. vis. and patt. recog.*, 2018, pp. 7132–7141.

[73] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.

[74] M. Oršić and S. Šegvić, “Efficient semantic segmentation with pyramidal fusion,” *Patt. Recog.*, vol. 110, p. 107611, 2021.

[75] S. Chandra, C. Couprie, and I. Kokkinos, “Deep spatio-temporal random fields for efficient video segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[76] L. Wang, R. Li, C. Zhang, S. Fang, C. Duan, X. Meng, and P. M. Atkinson, “Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 190, pp. 196–214, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271622001654

Appendix

A Permission to Reprint

A.1 Elsevier Permission to Reprint

In reference to Elsev ier copyrighted material which is used with permission in this thesis, Elsevier does not endorse any of Lakehead University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing Elsevier copyrighted material for advertising or promotional purposes, or for creating new collective works for resale or redistribution, please go to https://www.elsevier.com/about/policies/copyright/permissions to learn how to obtain a License from RightsLink.

A.2 IEEE Permission to Reprint

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Lakehead University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html and https://www.ieee.org/publications/rights/author-rights-responsibilities.html to learn how to obtain a License from RightsLink.

B Source Code

The source codes of this thesis are available on GitHub.