成都理工大學中英合作办学
CDUT Sino-British Collaborative Education

OXFORD BROOKES UNIVERSITY

UNDERGRADUATE PROJECT REPORT

Project Title:	**Semantic Segmentation on Deep Learning Application**
Surname:	**Feng**
First Name:	**Xiang**
Student Number:	**202018010119**
Supervisor Name:	**Dr Grace Ugochi Nneji**
Module Code:	**CHC 6096**
Module Name:	**Project**
Date Submitted:	**May 5, 2023**

Chengdu University of Technology Oxford Brookes College

Chengdu University of Technology

BSc (Single Honours) Degree Project

Programme Name: Projects

Module No.: CHC 6096

Surname: Feng

First Name: Xiang

Project Title: Semantic Segmentation on Deep Learning Application

Student No.: 202018010119

Supervisor: Dr. Grace Ugochi Nneji

Date submitted: May 5 2024

A report submitted as part of the requirements for the degree of BSc (Hons) in Computer Science

At

Chengdu University of Technology Oxford Brookes College

Declaration

Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed on-line at https://www.brookes.ac.uk/students/sirt/student-conduct/ . If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.

Signature: Feng Xiang Data: 22/04/2024

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Library.

| agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing use of the Library.

Signature: Feng Xiang Data: 22/04/2024

Acknowledgment

I'd like to extend my sincerest appreciation to my Supervisor, Dr. Grace Ugochi Nneji, for her unwavering guidance and support during the culmination of my undergraduate project. Dr. Nneji's expertise, patience, and encouragement have been instrumental in shaping the trajectory of this endeavor. Additionally, I wish to express gratitude to Joojo Walker, the module leader, and all the other educators who have imparted their knowledge and offered invaluable advice throughout my undergraduate journey.

Furthermore, I am grateful for the resources and facilities made available through the collaborative efforts of Oxford Brookes University and Chengdu University of Technology, which have provided an exceptional environment for academic growth.

Lastly, to my cherished family and friends, your enduring love and support have been a constant source of strength. I am profoundly thankful for your presence in my life.

Table of Contents

List of Figures

List of Table

Abstract:

Semantic segmentation is an important research direction in the field of computer vision , which plays a key role in many daily life applications, such as self-driving, medical image analysis and video monitoring. Nevertheless, today's semantic segmentation techniques still face challenges such as complex scene structure, multi-scale object recognition and changing environmental conditions, which bring very huge disturbances to semantic segmentation. To deal with these challenges, this study designs a high-performance semantic segmentation framework. The model does this by fusing multiple advanced deep learning techniques, such as using the ASPP module with deeply separable convolution to capture multi-scale contextual information, adopting the Transformer module to enhance model's ability to capture global dependency, and applying multi-scale pooling to optimize the model's ability to process features at different scales. This module fusion method significantly improves the accuracy and robustness of semantic segmentation, and achieves excellent segmentation results on diverse datasets while increasing the segmentation speed. By evaluating the model on test datasets, the model demonstrates its excellent performance in handling complex image segmentation tasks. The core contribution of this research is to propose an efficient and accurate semantic segmentation framework integrating deep separable convolution, ASPP, Transformer and multi-scale pooling, which brings new research directions and application potentials to the field of semantic segmentation.

Keywords: Semantic Segmentation, Computer Vision, Deep Learning, Technological integration, ASPP, Transformer Module, Multi-scale Pooling

Abbreviations

CNN: Convolutional Neural Network

ASPP: Atrous Spatial Pyramid Pooling

ResNet: Residual Network

MIOU: Mean Intersection Over Union

GAP: Global Average Pooling

DWC: Depth-wise Convolution

SGD: Stochastic Gradient Descent

PA: Pixel Accuracy

CPA: Class Pixel Accuracy

FPS: Frames Per Second

ReLU: Rectified Linear Unit

BN: Batch Normalization

TN: True Negative

TP: True Positive

FP: False Positive

FN: False Negative

GUI: Graphical User Interface

GPU: Graphics Processing Unit

Glossary

Semantic Segmentation : A technique that segment an image into multiple part and each part is labeled with different categories, which could understanding scene and object in the image.

Deep Learning : Learning from large amounts of data by using network with multiple layers of processing units, and is widely used for tasks such as image recognition, speech recognition, and semantic segmentation.

Image Augmentation : This is a technique where a series of transformations (e.g., rotation, scaling, cropping, etc.) are applied to the training image to increase the variety of the dataset artificially. This helps the model learn more generalized features, which improves performance on unseen data.

Feature Extraction : In deep learning, feature extraction usually means using high-level features learned from previous layers of the model that are helpful for the task at hand.

Muti-scale Feature Fusion : This is a technique that can combine features from different resolutions of an image to capture information ranging from coarse to detailed.

Attention Mechanism : Attention Mechanism allows deep learning models to dynamically focus on important parts of the input data. It plays a key role in improving the explainable and performance of the model.

Stochastic Gradient Descent (SGD) : It is an optimization algorithm used to minimize the loss function of the model during the training process.The SGD optimizer can effectively reduce the consumption of computational resources and speed up the training process.

Atrous Spatial Pyramid Pooling (ASPP) : ASPP is a technique to capture multi-scale information by using null convolution with different sampling rates.ASPP can effectively improve the feature resolution and model performance in semantic segmentation tasks.

Cross-entropy Loss : It is a loss function commonly used in classification tasks to quantify the difference between the probability distribution predicted by the model and the true label. It is essential for training high-performance classification models.

Dice Loss : It is a loss function used in image segmentation tasks. It is particularly suitable for dealing with the problem of category imbalance, where the accuracy of segmentation is improved by optimizing the model to the overlap of the segmented regions of the Jung family.

Chapter 1 Introduction

1.1 Background

With urban traffic congestion and frequent accidents becoming an increasing problem, self-driving vehicles are widely recognized as a potential solution to reduce accident rates and improve traffic flow **[1]**. However, the application of self-driving in urban environments faces a number of challenges, which include dealing with complex backgrounds, variable weather conditions, and diverse traffic scenarios **[2]**. These factors make it difficult for traditional computer vision techniques to be adapted, increasing the difficulty of realizing self-driving technology **[2]**.

In this case, semantic segmentation is especially crucial as a machine vision technique that allows a fine understanding of the surrounding environment **[3]**. It can relate each pixel in an image to a semantic category of roads, buildings, vehicles, and pedestrians **[4]**. This can help self-driving systems to recognize and understand their surroundings more accurately, providing a powerful environment perception tool for vehicles **[5]**.

Although semantic segmentation technology shows great potential in self-driving, it still faces many challenges in practical applications, including how to effectively deal with complex scene structures, adapt multi-scale object recognition, and deal with changing environmental conditions **[6]**. These challenges require semantic segmentation models not only have better accuracy and robustness, but also need to realize real-time and fast image processing to adapt the real-time decision-making requirements for self-driving vehicles **[7]**.

Therefore, it has become an urgent problem for how to make the semantic segmentation technology adapt the application requirements of self-driving vehicles in complex urban environments. This study is devoted to in-depth research on semantic segmentation technology, aiming to develop a high-accuracy real-time semantic segmentation model to provide safer and more efficient self-driving technology for urban transportation systems to promote sustainable urban development.

1.1.1 Convolutional Layers

The convolutional layer is the core building block of a convolutional neural network and is responsible for executing most of the computations. It requires several components, including input data, filters, and feature maps. There is also a feature detector, also known as a kernel or filter, which moves through the various sense fields of the image, checking for the presence of features. This process is called convolution **[8]**.

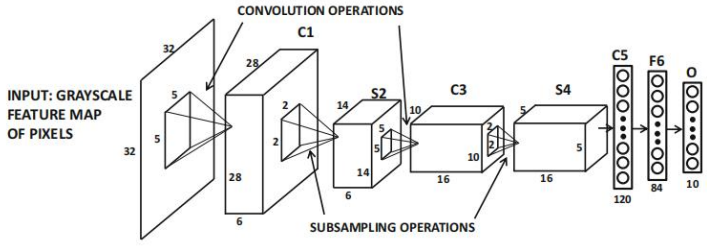


Figure 1. convolutional architecture**[29]**

1.1.2 Pooling Layers

The pooling layer, also known as the downsampling layer, is a data dimensionality reduction operation that aims to reduce the number of parameters within the input data. Similar to the convolutional layer, the pooling operation lets the filter scan the entire input, but the difference is that this filter has no weights. It has two types of pooling: maximum pooling picks out the largest value and average pooling calculates the average value. This process loses some information about the data, but it allows the neural network to run more efficiently to avoid the risk of overfitting **[8]**.

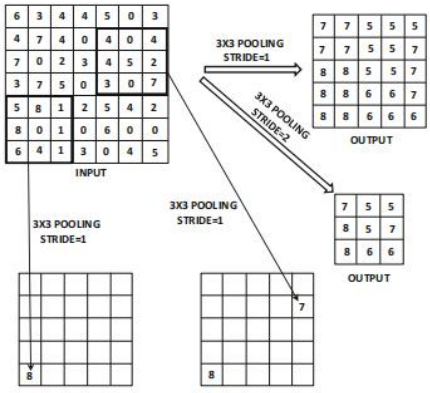


Figure 2. Pooling schematic**[29]**

1.1.3 Up-Sampling

Up-sampling increases the size of the data by inserting new pixels or feature points, using methods such as transposed convolution to recover detail and support accurate prediction.

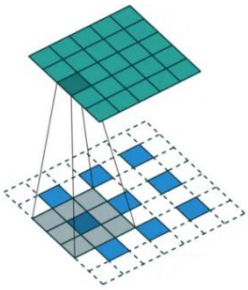


Figure 3. Up-Sample process

1.1.4 ASPP

ASPP module improves the accuracy of semantic segmentation by processing images in parallel using null convolution with different sampling rates, effectively capturing information at different scales. This method expands the sensory field of the model, allowing it to simultaneously understand both details and broader contextual information in the image, and is particularly suitable for dealing with size variations in images, thus improving segmentation performance. In short, ASPP allows the model to better adapt to targets of different sizes and improve semantic segmentation **[9]**.

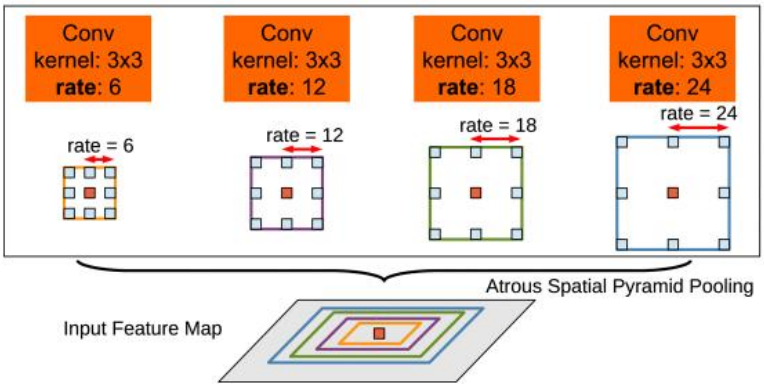


Figure 4. ASPP Module**[19]**

1.1.5 Loss Functions

Cross-entropy Loss **[21]**: Cross-entropy loss measures the discrepancy between model predictions and actual labels, and is a key tool for optimizing model accuracy in tasks dealing with classification and semantic segmentation.

Dice Loss **[22]**: Dice Loss is a loss function based on Dice coefficients for evaluating the similarity of two samples, which is particularly suitable for the case of category imbalance in image segmentation, and significantly improves the accuracy and performance of semantic segmentation by optimizing the overlap rate between predicted and real labels.

1.1.6 Optimization Algorithms

SGD is a optimizer for training deep learning models. It updates the model by using a small portion of randomly selected data with the aim of reducing errors. This approach is faster than updating the model with all of the data and is particularly suitable for large data sets. **[20]**.

1.1.7 LinkNet

As shown in Figure 5, LinkNet is a light weight,efficient neural network structure for semantic segmentation that uses an encoder-decoder architecture. Special jump connections solve the gradient loss problem. The encoder catches image features and decoder recover image details. The encoder gradually reduces the image resolution and extracts the features and the decoder restores the resolution while fusing the features and assigning semantic categories to each pixel **[24]**.

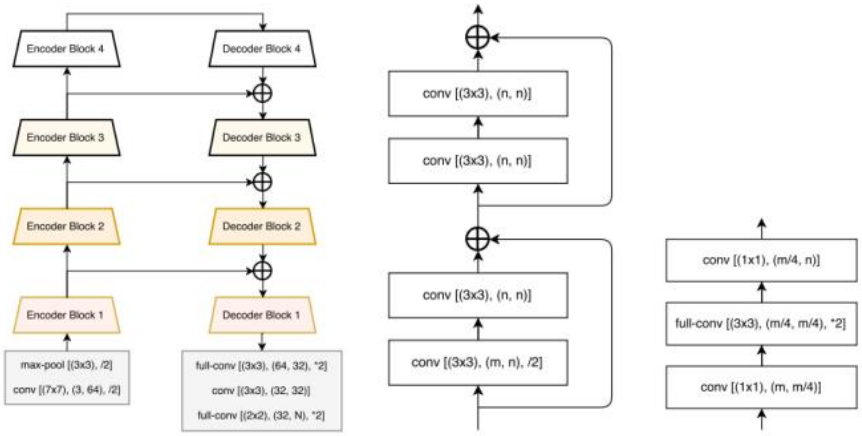


Figure 5. LinkNet structure diagram and encoder-decoder**[24]**

1.1.8 U-Net

U-Net depicted in Figure 6 is a deep learning model for image segmentation. It uses an encoder-decoder architecture and jump connections to reduce images, extract features and recover to original size, which solves the gradient loss problem, improves accuracy and preserves features at all layers [25].U-Net is able to catch image features accurately, meanwhile its simple architecture is easy to modify.

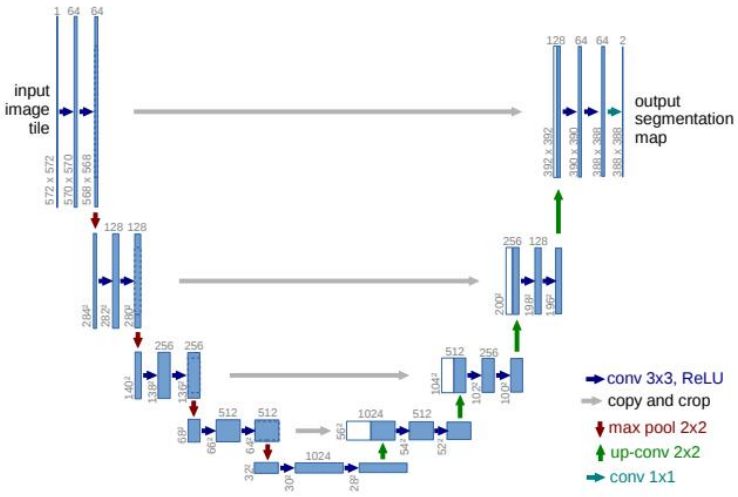


Figure 6. U-Net model **[25]**

1.1.9 Dilated Convolution

In the field of deep learning and computer vision, dilation convolution can be used to improve the performance of convolutional neural networks. It expands the network's receptive field by adding spaced "holes" between the elements of the convolutional kernel, which adds no additional computational burden and allows the network to observe a wider region of the input data **[27].**

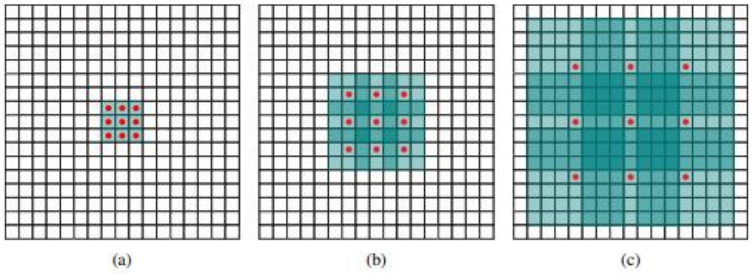


Figure 7. Dilated Convolution **[27]**

1.1.10 Transformer Encoder

The Transformer module, originally designed for processing sequential task in natural language, has been successfully applied in computer vision as an effective tool to complement traditional convolutional neural networks. Integrating Transformer into a CNN improve the model's understanding of the overall context of the image. This combination exploits both the efficiency of CNN in extracting local visual features and the power of Transformer in dealing with long-range dependencies. In visual tasks such as image classification, object detection, and semantic segmentation, this approach

improve model performance by better capturing both the detail and the overall structure of an image. **[28]**

1.2 Aim

The primary aim of this project is to create a high-performance semantic segmentation model by using deep learning technique. The model integrates varies of advanced deep learning technique, including the ASPP module based on DWC, the Transformer module, and multi-scale pooling. It aim to significantly improve the accuracy and robustness of the semantic segmentation model in complex scene structure, multi-scale object recognition and diverse environmental condition. In addition, to enhance the applicability and interactivity of the project, this project plan to use Flask build a website to implement GUI, which will display the segmentation result of the semantic segmentation model, aiming able to intuitively see the result of the model processing various images. In conclusion, this project aim to provide a novel semantic segmentation framework and promote the development and popularization of semantic segmentation technology in practical application by developing a user-friendly GUI.

1.3 Objectives

In order to achieve the main goal of this project, the following specific objectives have been set:

- In-depth research and experimentation of advanced deep learning techniques: research and implement multiple deep learning modules, such as DWC-based ASPP module, Transformer module, edge detection module, and multi-scale pooling module, and try to integrate them. Aims to improve the performance of semantic segmentation models through these advanced techniques.
- Improve the accuracy and robustness of the model in complex environments: through the above techniques, the accuracy and stability of the model in complex scene structures, multi-scale object recognition, and diverse environmental conditions are improved to meet the common challenges in semantic segmentation.
- Evaluating model performance: evaluate the accuracy and robustness of the model using evaluation metrics (e.g., accuracy, recall, MIOU, PA, Dice coefficients, Kappa scores, FPS) by running the model on a test dataset. This will provide an objective benchmark for comparing different model architectures and configurations.
- Developing a Graphical User Interface (GUI): Build a website using the Flask framework to implement a GUI for presenting semantic segmentation results so that

users can easily upload images, start the segmentation process and view the model outputs intuitively, enhancing the applicability and interactivity of the project.

- Contribute to the development and popularization of semantic segmentation technology: Through the development of an easy-to-use GUI and a high-performance semantic segmentation model, This project provide new ideas and tools for the research, development, and application of semantic segmentation technology, and promote the development and popularization of this technology in practical applications.

1.4 Project Overview

1.4.1 Scope

The core goal of this project is to develop a high-performance semantic segmentation model using deep learning techniques, aiming to significantly improve accuracy and robustness in complex scenarios, multi-scale object recognition, and variable environmental conditions, which will contribute to the realization of self-driving technologies. The scope of the project includes the design and implementation of a semantic segmentation model that integrates multiple deep learning techniques, in particular the fusion of the DWC-based ASPP module, the Transformer module, and multi-scale pooling techniques. This integration strategy works on extracting richer image features as well as reducing the training parameters of the model to improve its efficiency and accuracy. In addition to this, the project creates a website to implement a GUI through the Flask framework to visualize the results of semantic segmentation to enhance the user's application experience and interactivity. Through the GUI, users can upload images for segmentation and instantly view the processing results of the model. Finally, the project will evaluate the performance of the model on a test dataset and compare it with other existing models to show the advantages of proposed model. Through the above efforts, this project is not only devoted to promote the innovation of semantic segmentation technology, but also to promote the popularization and application of this technology in real-world applications through the development of an easy-to-use GUI.

1.4.2 Audience

For government and city planners, this project enhances traffic management efficiency. Realtime road condition sensing in autonomous driving can optimize traffic flow and reduce congestion **[2]**. Autonomous driving also mitigates accidents caused by human factors, enhancing road safety. For drivers, it ensures safe, fatigue-free driving, and

selects optimal routes based on real-time conditions, saving time, reducing stress, and improving travel efficiency **[8]**. For the urban environment, improved traffic management and reduced congestion cut emissions, enhancing city air quality.

Chapter 2 Background Review

2.1 Summary of Related Literature

The research background of semantic segmentation techniques centers around a central task in the field of computer vision - understanding the category to which each pixel in an image belongs **[4]**. This technique enables computers not only to recognize the objects present in an image, but also to accurately classify the boundary of each object, which is an important foundation in the field of image processing and analysis **[3]**. Nowadays, the introduce of deep learning, especially the application of CNN, greatly improves the accuracy and efficiency of semantic segmentation, allowing machines to process more complex image data, recognize and segment multiple objects in an image **[23]**. The progress of this technology not only promotes the development of the computer vision field, but also brings innovative possibilities for a number of application fields, such as self-driving, medical image analysis, and environmental monitoring **[23]**.

By combing through the related literature, the MIOU of some existing semantic segmentation models will be compared, and below is Table 1 which summarizes the performance comparison of different semantic segmentation models by different researchers.

Author	Model	MIou	Dataset
Badrinarayanan et.al. **[9]**	SegNet	56.1%	Cityscapes
Abdigapporov et.al.**[10]**	BiFPN	56.4%	Cityscapes
Paszke et.al. **[11]**	ENet	58.3%	Cityscapes
Poudel et.al. **[12]**	Fast-SCNN	68%	Cityscapes
Yu et al **[13]**	BiSeNet	69%	Cityscapes
Fourure et al **[14]**	GridNet	69.5%	Cityscapes
Chen et al **[15]**	Deep-Lab CRF	70.4%	Cityscapes
Lin et al **[16]**	RefineNet	73.6%	Cityscapes
Li et.al.**[17]**	BiAttnNet	74.7%	Cityscapes
My Model	DeepSegASPP+Transformer	78%(11class)	Cityscapes

My Model	DeepSegASPP+Transformer	71%(19class)	Cityscapes

Table 1. Comparison of results

First, the SegNet model proposed by Badrinarayanan et al **[9]**. is a convolutional network using a nonlinear upsampling technique that achieves a MIOU of 56.1%, which is more suitable for simple scene understanding. Later, the BiFPN network developed by Abdigapporov et al.**[10]** achieved a MIOU of 56.4% by enhancing the multi-scale fusion of features. ENet designed by Paszke et al. **[11]** is a lightweight network designed to satisfy the requirements of real-time applications with 58.3% MIOU, while Fast-SCNN proposed by Poudel et al. **[12]** achieves 68% MIOU by optimizing the computational efficiency, and both networks are suitable for semantic segmentation applications on mobile.

BiSeNet developed by Yu et al. **[13]** uses dual network structure to achieve a model that balances speed and performance with 69% MIOU. Fourure et al.**[14]**'s GridNet model enhances feature fusion through its grid structure and achieves 69.5% MIOU. Chen et al. **[15]'**s DeepLab-CRF, which is constructed using ASPP with CRF technology, achieves 69.5% MIOU on the edge processing was refined and achieved 70.4% MIOU. RefineNet proposed by Lin et al. **[16]** enhanced the information fusion by multipath refinement technique and achieved 73.6% MIOU. BiAttnNet model by Li et al. **[17]** with the introduction of bi-directional attention mechanism drastically improved the segmentation accuracy and achieved 74.7% MIOU.

Proposed model achieves higher mIoU on the CityScapes dataset, showing superior performance in recognition of complex urban scenes. This not only exceeds the accuracy of other models, but also brings important advances to computer vision systems for urban scenes. Compared to other research, proposed model not only reset the performance benchmark for semantic segmentation, but also introduces an innovative approach for accurate and efficient parsing of images by utilising cutting-edge deep learning techniques.

Chapter 3 Methodology

3.1 Approach

These following aspects will be followed in this project:

- The Cityscapes dataset is used, containing 2975 training images, 500 validation images and 500 test images.
- Data preprocessing includes data balancing and enhancement to improve model generalization. Specifically, evaluating pixel classes and ignoring classes with fewer pixels. Different weather is simulated to adjust the saturation, hue, contrast and random cropping of the images.
- The model will use ASPP module, Transformer encoder, Edge Detection and Contextual Enhancement modules and select Resnet50 as the backbone network.

3.2 Dataset

The Cityscapes dataset is a large dataset focusing on urban street scenes, widely used in computer vision and autonomous driving research **[33]**. It contains high-resolution images from 50 different cities and provides accurate pixel-level annotations for about 5,000 images covering 30 different categories, such as roads, pedestrians, etc **[33]**. The dataset also includes a number of other images that have been annotated to provide a better understanding of the city's streetscape. In addition, the dataset includes about 20,000 roughly annotated images, enriching the training data. In this project 2975 images will be used as training dataset, 500 as validation and 500 as testing .

3.3 Pre-processing

3.3.1 Data Balancing

When processing the Cityscapes semantic segmentation dataset, the key step was to assess the balance of the categories by counting the number of pixels in each category. This involves selecting a representative validation dataset and traversing its image annotations to count the number of pixels. The relationship between the different categories can be visualized through Figure 8. The analysis shows that the road and building categories have more samples, while specific vehicles and pedestrians have fewer. To improve the accuracy of segmentation, categories with few pixels will be ignored in this project to avoid their influence on the main categories.

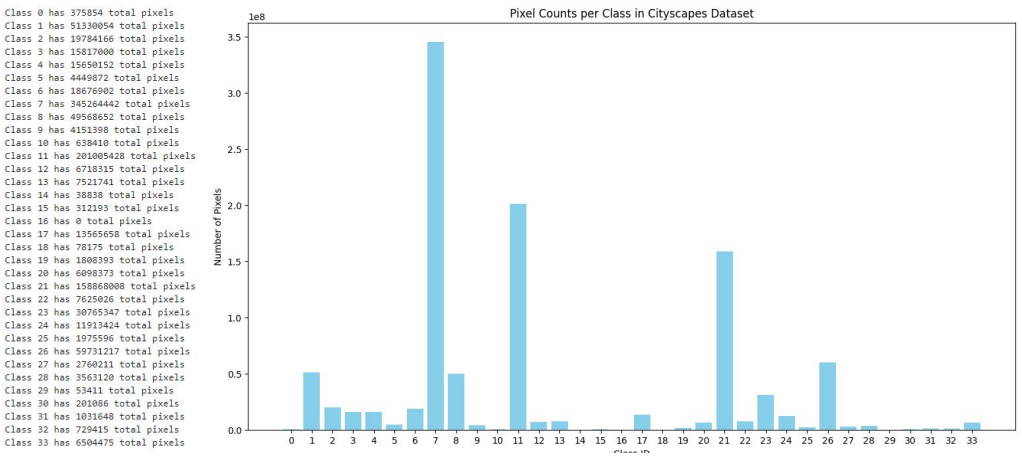


Figure 8. Pixel class profile of the original cityscapes dataset

3.3.2 Data Enhancement

Data enhancement techniques play a key role in developing image segmentation models for urban scenes. Due to the variety of urban weather conditions, this project first added a weather changing data enhancement operation to the original cityscapes dataset to simulate the visual effects under different weather conditions such as foggy, rainy and snowy days. This is shown in the figure below:

Figure 9.Weather Change Effect

In addition, to further improve the model's adaptability and robustness to features such as urban image lighting, this project use several stochastic transformation techniques. A series of diverse training samples are generated by adjusting the saturation, hue, and contrast of the images and implementing random cropping. Specifically, the image saturation is randomly varied between 0.5 and 1.5, the hue can change by up to 0.2, and the contrast is adjusted between 0.5 and 1.5 to ensure the model can handle images under different lighting condition. Additionally, by randomly cropping images with a resolution of 1024*2048 to 384*384, this approach reduces the model's training parameters and avoid the interference of extreme values.

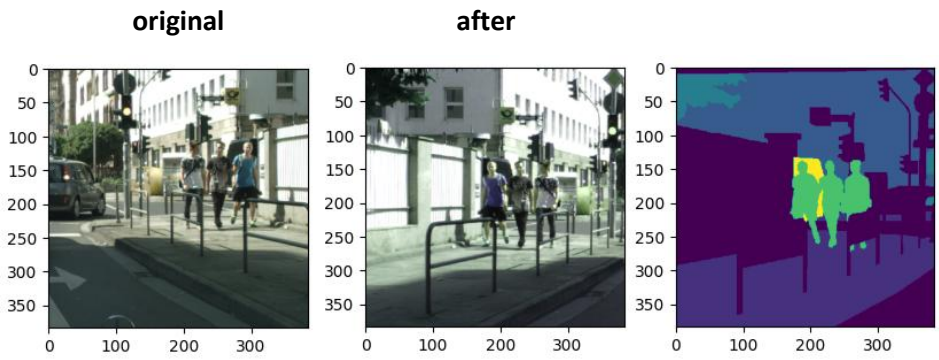


Figure 10. Image Enhancement Processing

In addition to this, data augmentation part also tried to add an edge enhancement effect with original image, this step aim to improve the model's ability to capture the image edge information and further enhance the model's accuracy in recognizing the boundaries of object in complex urban scenes. However, as far as the results of the training are concerned, the results of this are very unsatisfactory and this data enhancement approach is abandoned. The enhancement results are shown below:

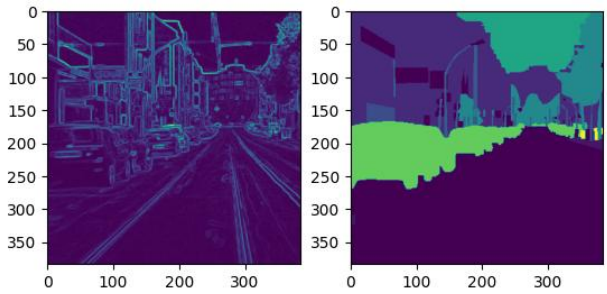


Figure 11. Edge Enhancement Processing

3.4 Component Modules and Model Architecture

With the description of the aforementioned concepts, it becomes more straightforward to grasp the network model utilized in this project. The following section will provide a detailed introduction to the modules and architecture of the model employed within this project.

3.4.1 Convolution-based ASPP Module:

The objective of the ASPP (Atrous Spatial Pyramid Pooling) module is to enhance the capture of multi-scale information from images, thereby increasing the precision of semantic segmentation. As depicted in Figure 12, this module represents a modified version of the ASPP module, where the original four dilated convolutions of varying dilation rates have been reduced to three layers, with dilation rates of 6, 12, and 18, respectively. This significant reduction in dilated convolution layers considerably decreases redundant computations and training parameters. Such a streamlined

architecture not only diminishes the model's training duration but also aids in preventing over-fitting, markedly improving the model's generalization capability during segmentation. Moreover, the conventional convolution layers have been substituted with depthwise separable convolutions, which substantially reduce the model's parameter count and computational load. Compared to standard dilated convolutions, depthwise separable convolutions achieve better performance while significantly reducing computational complexity and memory requirements.

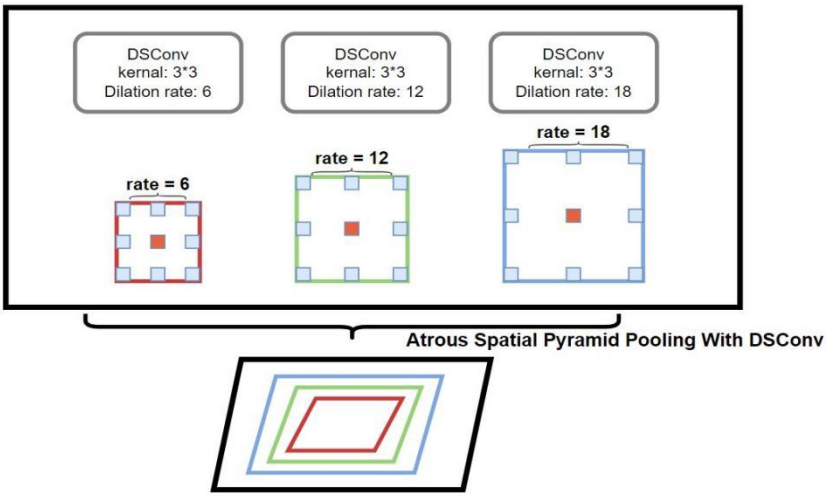


Figure 12. Modified ASPP module

Figure 13 illustrates the flowchart of the improved ASPP (Atrous Spatial Pyramid Pooling) module. Initially, the input X is fed into both an upsampled average pooling layer and four depthwise separable convolutions with varying dilation rates (DR=1, DR=6, DR=12, DR=18). Subsequently, the upsampled global context feature map (y_pool) is fused with the outcomes of the four depthwise separable convolutions with different dilation rates (y1, y2, y3, y4). Finally, a rich set of multi-scale features is obtained as the output of this module. This module enhances the model's capability to perceive features of varying sizes, thereby improving its performance in processing objects across different scales.

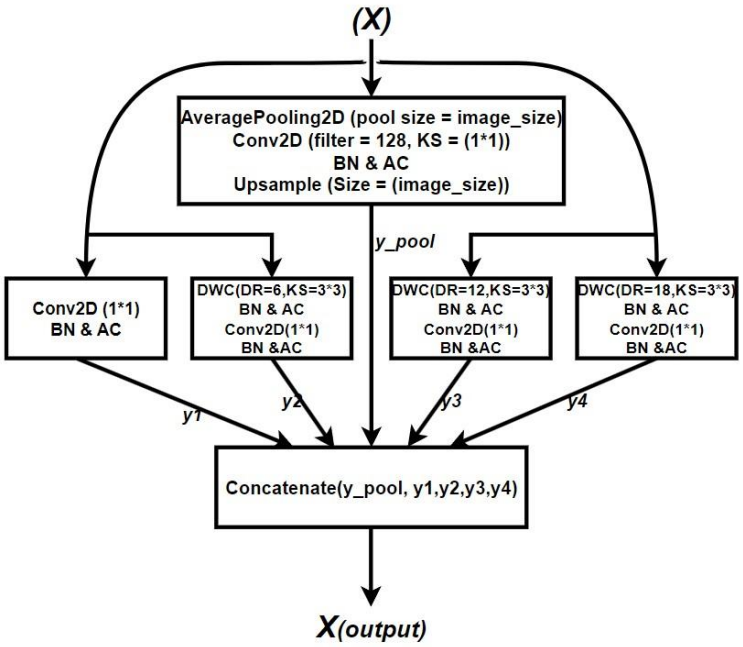


Figure 13. Modified ASPP Module Flow Diagram

3.4.2 Transformer Encoder Module:

Figure 14 presents the flowchart of the Transformer Encoder module, whose input originates from the output of the ASPP module that has been reshaped into a sequence. Initially, the input features undergo layer normalization and a multi-head attention mechanism. This process aids in stabilizing the training process and allows the model to focus on different parts of the input features simultaneously. This capability enhances the model's ability to capture the global dependencies between pixels, significantly improving semantic segmentation performance. The epsilon value is set to 1^{e-6} to ensure computational stability, the key dimension is set to 128 to enhance the model's capacity to capture diverse features, and the number of heads is set to 4 for parallel computation of attention.

This module is appended to the end of the ASPP module because, while the ASPP module aids the model in **observing** the image at different scales, it enables the model to understand the connections between different parts of the image. It is akin to equipping the model with both a **telescope** and a **microscope**, allowing not only the observation of the overall shape of objects within the image but also the details of these objects. Thus, the model can perceive both the local information of each pixel and a broader range of global information.

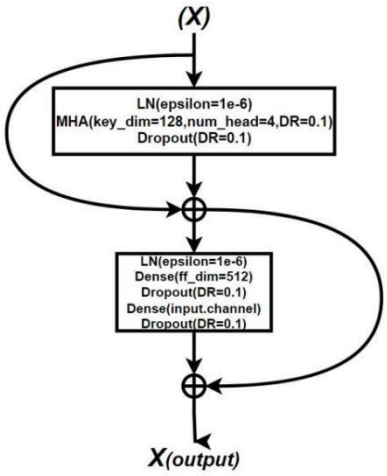


Figure 14. Transformer Module Flow Diagram

3.4.3 Edge Detection Module:

Figure 15 depicts the edge detection module designed for this project. In this module, depthwise separable convolutions are integrated with the SE (Squeeze-and-Excitation) attention mechanism to achieve efficient edge detection. At first, the module extract features through the DWC. Next, these features will be sent to the SE module for **excitation** operation, which making the model more focused on the feature channels that are crucial for edge detection task. Finally, the module uses a 1*1 convolutional kernel to generate the final result. It ensure that the module is able to effectively highlight the features that are important for edge detection, thus enhancing the overall performance of the task.

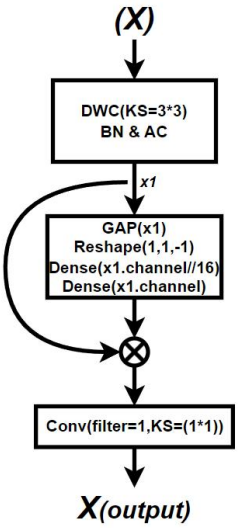


Figure 15. Edge Detection Module Flow Diagram

3.4.4 Context Enhancement Module:

Fig. 18 shows the flowchart of the designed Context Enhancement module, which is aimed at making the model more focused on local details and global environmental information in the scene. Firstly, the input data for this module is processed through three convolutional layers with different dilation rates in order to capture multi-scale features. These features are then combined through concatenation and weighting operations (Add and Multiply) to generate an integrated feature set. In particular, the weighting process determines the fusion weights of each feature by means of a convolutional layer activated by a Sigmoid function, enabling the module to make full use of various contextual information during feature fusion, thus improving the accuracy of segmentation. Next, the module use a global average pooling operation to extract the global information of the image, which is further processed by two connected layers with activation functions ReLU and Sigmoid, respectively, to enhance the global features. Through this processing, the module help the model to recognize the overall scene layout and category distribution more accurately, which in turn enhances the semantic understanding of complex scenes.

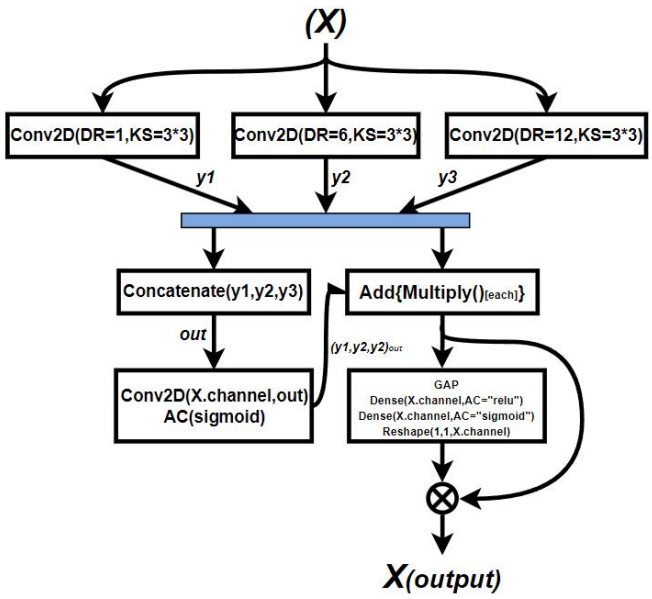


Figure 16. Context Enhancement Module Flow Diagram

3.4.5 ResNet50:

ResNet50 is a residual network with 50 convolutional layers, designed to mitigate the training difficulties that occur with increasing network depth. The network include residual modules that utilize concatenation of input and output to prevent the issue of vanishing gradients. The primary reason for selecting ResNet50 as the backbone network for the semantic segmentation model is to balance classification effectiveness with computational efficiency. This choice ensure that the model maintain high accuracy while also being relatively efficient to train and run, making it suitable for a wide range of semantic segmentation tasks **[26]**.

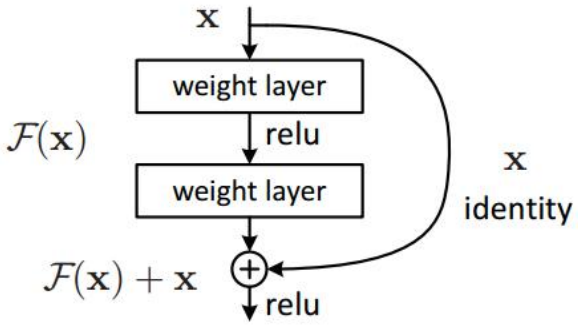


Figure 17. Residual module **[26]**

3.4.6 Overall Architecture of the Model:

Figure 18 show the architecture of the model, with an input size of 384x384x3. Initially, the input data is directed into two separate paths: the edge detection module and ResNet50 equipped with pretrained weights. The former enhances the model's sensitivity to image edges, while the latter accelerates model training and improves generalization capabilities. Subsequently, weights from three layers of the pretrained ResNet50 model—namely, activation_9, activation_23, and activation_39—are utilized.

The weights from the activation_9 layer, representing low-level features, undergo convolutional layer processing and multi-scale pooling, followed by concatenation with the output of the convolutional layer. This step aids in capturing fundamental image information, beneficial for maintaining image details.

Activation_23 represents a higher-level feature layer within ResNet50. Its weights are fed into the context enhancement module to produce output that assists the model in understanding more complex image content and contextual relationships, aiding in the recognition of complex objects.

Following this, the activation_39 layer, which provides even higher-level abstract features compared to the activation_23 layer, has its output directed into an ASPP module based on depthwise separable convolutions, and then the output is fed into a Transformer encoder. The ASPP module enhances the model's ability to process different parts of an image, while the Transformer encoder leverages attention mechanisms to improve segmentation accuracy.

Subsequently, the results from the edge detection module and the final outputs from activation_9, activation_23, and activation_39 are combined. Finally, the combined result is passed through two layers of depthwise separable dilated convolutions based on a residual structure to obtain the final segmentation output. This significantly reduces the consumption of computational resources, ensuring that the model maintains high performance while keeping computational costs low.

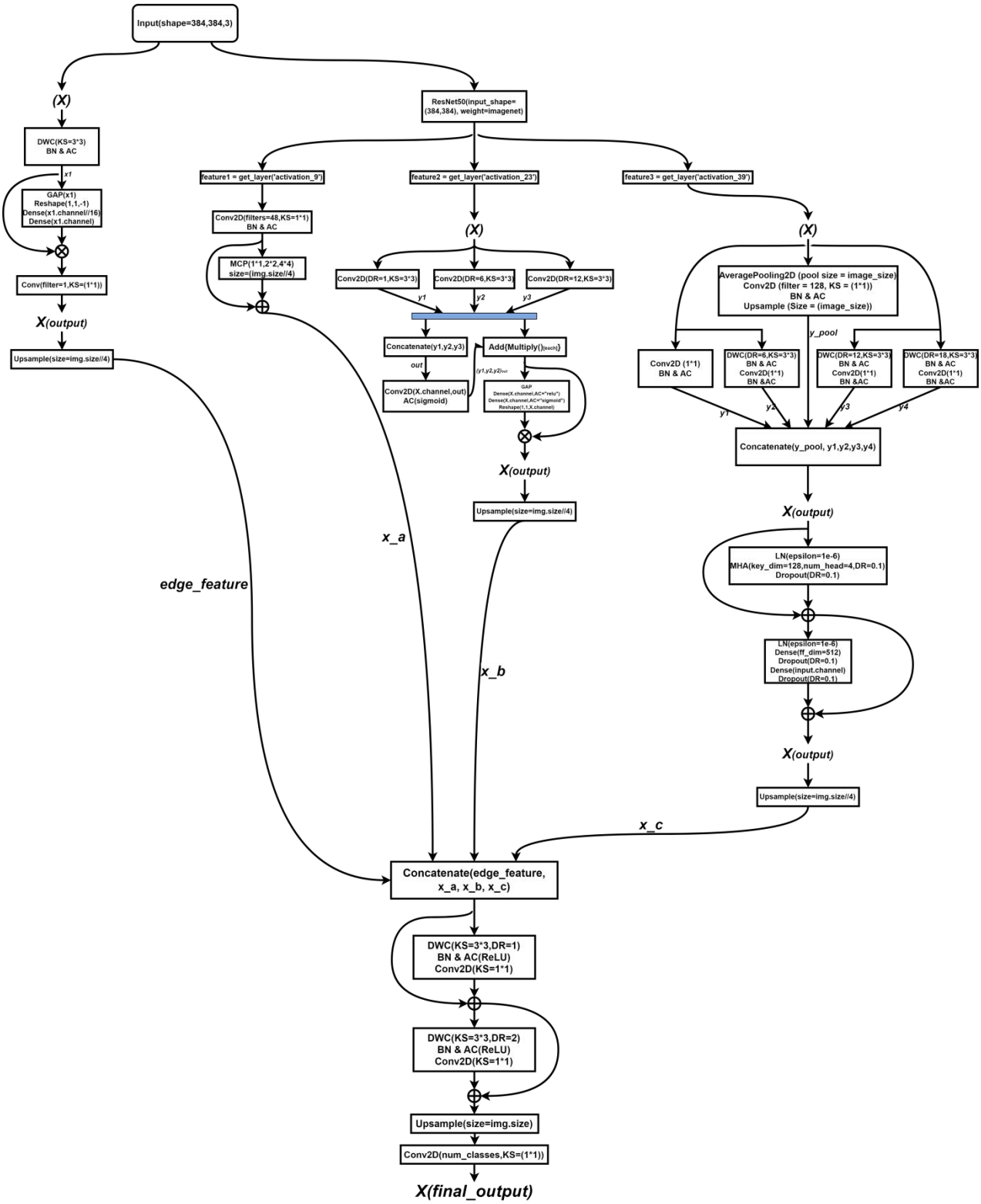


Figure 18. Model Architecture

3.5 Technology

The part information is shown in Table 2.

Software	Framework	Tensorflow 2.10.0 Cudatoolkit 11.8.0 Cudnn 8.9.2.26
	Language	Python 2.9
	Libraries	Numpy Matplotlib Pandas Keras 2.10.0 Glob3
	Version management plan	GitHub
	Operation System	Windows 10
Hardware	Central processing unit(CPU)	Intel Xeon Gold 6142 Processor 22M Cache 2.60 GHz (Cloud Server)
	Graphic Processing Unit(GPU)	NVIDIA GeForce RTX 3090 (Cloud Server)

Table 2 The technologies of the project

3.6 Testing and Evaluation plan

3.6.1 Data testing

- 500 test images from the Cityscapes dataset were selected to ensure coverage of all annotation categories.
- Verify the annotation completeness of each image in the test set to ensure that there is no missing data.
- Check that the test set contains images from different cities, different weather and different time periods to ensure comprehensiveness.
- Apply the same preprocessing steps to the test set as to the training set, including image resizing and normalization, to eliminate the effect of processing differences on the test results.

(1) Mean Intersection over Union(MIoU):

MIoU [15] is mainly used to measure the degree of overlap between the segmentation results predicted by the model and the true results. In formula, 'k' represents the category number, 'Pii' represents number of overlapping pixels, and 'Pji' represents the number of misassigned pixels.' 1/(k+1)' is the average weight to ensure that each category contributes equally to the MIOU. Thus, MIOU is affected by the category number, the positive sample number, and the pixel overlap between different categories.

$$\boldsymbol{MIoU} = \frac{\mathbf{1}}{\boldsymbol{k}+\mathbf{1}} \sum_{\boldsymbol{i}=\mathbf{0}}^{\boldsymbol{k}} \frac{\boldsymbol{pii}}{\sum_{\boldsymbol{j}=\mathbf{0}}^{\boldsymbol{k}} \boldsymbol{p\,ij} + \sum_{\boldsymbol{j}=\mathbf{0}}^{\boldsymbol{k}} \boldsymbol{p\,ji} - \boldsymbol{pii}} \quad \text{(equation 1)}$$

(2) Accuracy:

It indicates the ratio of the number of samples correctly predicted by the model to the total number of samples. In semantic segmentation, Accuracy is equal to the total number of correctly classified pixels divided by the total number of pixels in the image and it gives a quick overview of how well the model performs on the entire dataset.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP}+\boldsymbol{TN}}{\boldsymbol{TP}+\boldsymbol{TN}+\boldsymbol{FP}+\boldsymbol{FN}} \quad \text{(equation 2)}$$

- TP denotes the number of true cases (the number of samples that the model correctly predicts to be in the positive category)
- FN is the number of false negative cases (the number of samples that the model incorrectly predicts as negative)
- FP denotes the number of false positive cases (the number of samples that the model incorrectly predicts as positive)
- TN denotes the number of true negative cases (the number of samples correctly predicted by the model to be in the negative category).

(3) Loss Function:

In this project, combined loss function have been used. This loss function is a combination of the Cross Entropy Loss Function, which takes into account the accuracy of the predicted probability distribution, and the Dice Loss, which focuses on the overlap of the shapes between the predicted and real labels. This combined loss function not only allows the model to learn accurate pixel categorization and overall region similarity but also alleviates the problem of category imbalance in semantic segmentation tasks.

Categorical Cross-Entropy Loss is used to measure the difference between the probability distribution predicted by the model and the probability distribution of the true label. Here, M is the number of categories, $y_{o,c}$ is 1 if the true label of category c is observed and 0 otherwise, and $p_{o,c.}$ is the predicted probability that category c is observed.

$$\boldsymbol{Categorical\ Cross\ Entropy\ Loss} = -\sum_{c=1}^{M} \boldsymbol{y_{o,c}\ log\ (p_{o,c})} \quad \text{(equation 3)}$$

This following Dice Loss is applicable to multi classification task. Where Y is the binarized matrix of true labels, P is the predicted probability matrix, yi and pi are the true and predicted values respectively, and N is the total number of pixel points.

$$\boldsymbol{Dice\ Loss} = 1 - \frac{2\sum_{i}^{N} y_i p_i}{\sum_{i}^{N} y_i^2 + \sum_{i}^{N} p_i^2} \quad \text{(equation 4)}$$

Portfolio losses are realized through a weighted sum, where α and β are weighting parameters used to adjust the relative importance of the two loss terms.

$$Combined\ Loss = \alpha \times Categorical\ Cross\ Entropy\ Loss + \beta \times Dice\ Loss \quad \text{(equation 5)}$$

(4) Precision:

Precision measures how many of all the samples classified as positive instances by the model are true instances. Precision is calculated using the following formula:

$$\boldsymbol{Precision} = \frac{TP}{TP+FP} \quad \text{(equation 6)}$$

Precision also ranges from 0 to 1, with closer to 1 indicating that the model is more accurate in the samples classified as true cases.

(5) Recall:

Recall measures the ability of the model to correctly identify positive examples, also recall is known as True Positive Rate or Sensitivity.Recall is calculated as follows:

$$\boldsymbol{Recall} = \frac{TP}{TP+FN} \quad \text{(equation 7)}$$

The Recall Rate ranges from 0 to 1, the closer to 1 means the better performance of the model in identifying positive examples.

(6) Kappa

Kappa is used to measure the consistency of two evaluators in a classification task beyond pure chance consistency. It is commonly used to evaluate the performance of machine learning models, especially in consistency tests for data labeling.

$$\boldsymbol{K = \frac{P_o - P_e}{1 - P_e}} \quad \text{(equation 8)}$$

Where $\mathrm{P_o}$ is the observed consistency and $\mathrm{P_e}$ is the chance consistency, it takes values between -1 and 1, with higher values indicating better consistency.

(7) PA

Pixel accuracy is one of the most intuitive metrics for evaluating the performance of an image segmentation model, which calculates the percentage of all correctly categorized pixels out of the total pixels.

$$\boldsymbol{PA = \frac{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} TN_i}{\sum_{i=1}^{n} TP_i + FP_i + FN_i + TN_i}} \quad \text{(equation 9)}$$

In particular, TPi, TNi, FPi, and FNi represent the number of true cases, true negative cases, false positive cases, and false negative cases of the i th category, respectively, and n is the total number of categories.

(8) CPA

The category accuracy calculates the proportion of pixels correctly categorized in each category out of the total pixels in that category, which is then averaged over all categories.

$$\boldsymbol{CPA_i = \frac{TP_i}{TP_i + FN_i}} \quad \text{(equation 10)}$$

For each category i, its CPA is given by the following equation, where TPi and FNi are defined as above.

(9) Dice Coefficient

Dice coefficient is a statistical tool that measures the similarity of two samples and is commonly used in medical image segmentation. It calculates the ratio of the size of the intersection between twice the predicted and true labels to the sum of the respective sizes of the predicted and true labels.

$$\boldsymbol{Dice\ Coefficient = \frac{2\sum_{i=1}^{N} y_i p_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} p_i}} \quad \text{(equation 11)}$$

Here, Y is the set of real labels, P is the set of predicted labels, and yi and pi represent the value of each pixel point in the real and predicted labels, respectively.

(10) FPS

FPS is a measure of the speed of image processing, especially important in video processing or real-time systems, and indicates how many frames per second the model can process.

$$FPS = \frac{1}{Average.Processing.Time} \quad \text{(equation 12)}$$

The average processing time is the average time it takes for the model to process a single frame of an image, and the unit is usually seconds.The higher the value of FPS, the faster the model can process.

Chapter 4 Results

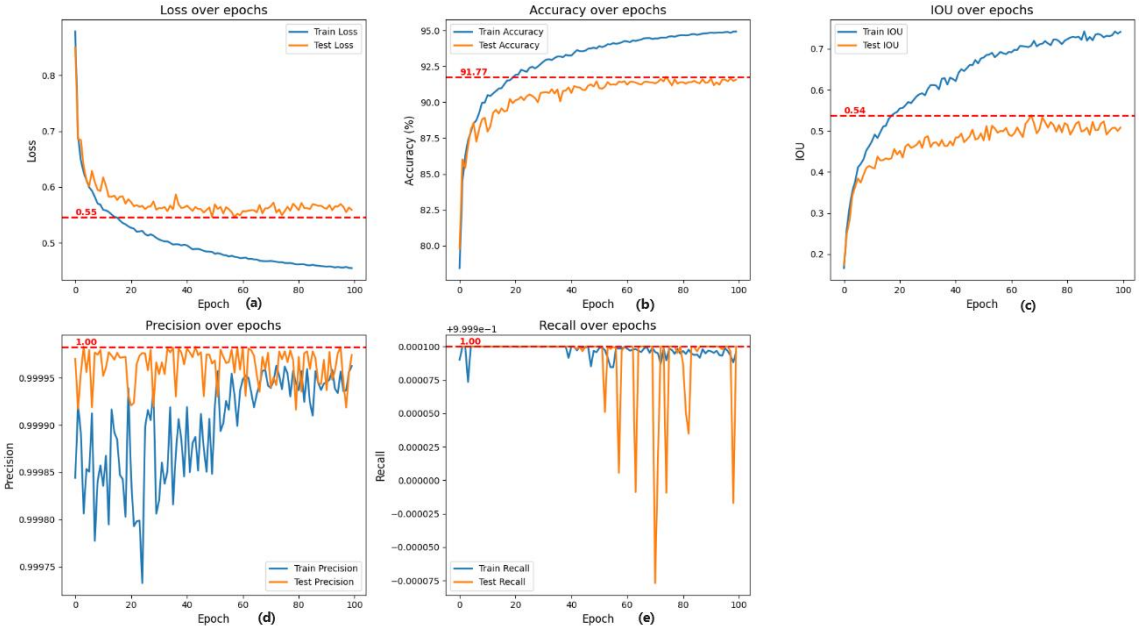
4.1 Results of Model Training

4.1.1 Final Result

The following result is the final result of proposed model, which was run under the use of a combined loss function (Categorical Cross Entropy Loss+Dice Loss). Also, this project have used SGD optimizer based on weight decayed as well as polynomial decayed learning rate scheduling to ensure the segmentation results. In the SGD optimizer, this project used an initial learning rate of 0.01 and 1.2 as a power of polynomial decay, which ensures that the learning rate decreases slowly early in the training and accelerates later on, which is well suited for tasks that require long periods of time to explore the parameter space at a high learning rate. After that, also set the weight decay value to 0.0001, which can be used for regularization to avoid overfitting. Finally, setting the momentum of the SGD optimizer to 0.9, which ensures that the model strikes a certain balance between speeding up training and avoiding excessive oscillations.

In this project, model training was attempted using 34, 19, 15 and 11 categories of the Cityscapes dataset. Initial experiments showed that the mIoU decreased when the number of categories increased, this is because more categories means more complex and diverse target, which increase the training difficult. Therefore, this project consider training with fewer categories to find a balance between performance and effect. After a detailed comparison, the 11-category dataset is found to be the most suitable, which not only cover the key and common objects in the urban scene, but also reduce the training burden, making it a solution that balances performance and effect.

- **34 categories**

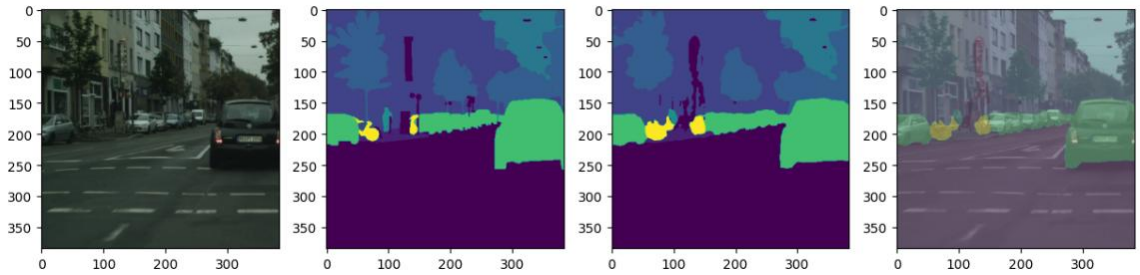


(a)Loss=0.55;(b)Acc=91.77;(c)MIOU=0.54;(d)Precision=1;(e)Recall=1

Figure 19. ,34 Categories Metric

CLASS	unlabeled	Ego_vehicle	Rectification_border	Out_of_roi	Static	Dynamic	Ground	Road	Sidewalk	Parking	Rail_track	Building	Wall	Fence	Guard_rail	Bridge	Tunnel	OVERALL
MIOU	0	0.93	0.73	0.22	0.25	0.13	0.38	0.97	0.74	0.44	0.61	0.86	0.47	0.44	0.65	0.52	0	
CLASS	Pole	Rolegroup	Traffic_Light	Traffic_sign	Vegetation	Terrain	Sky	Person	Rider	Car	Track	Bus	Caravan	Trailer	Train	Motorcycle	Bicycle	0.53
MIOU	0.39	0.17	0.47	0.59	0.88	0.49	0.92	0.73	0.53	0.89	0.29	0.67	0.56	0.09	0.87	0.47	0.68	

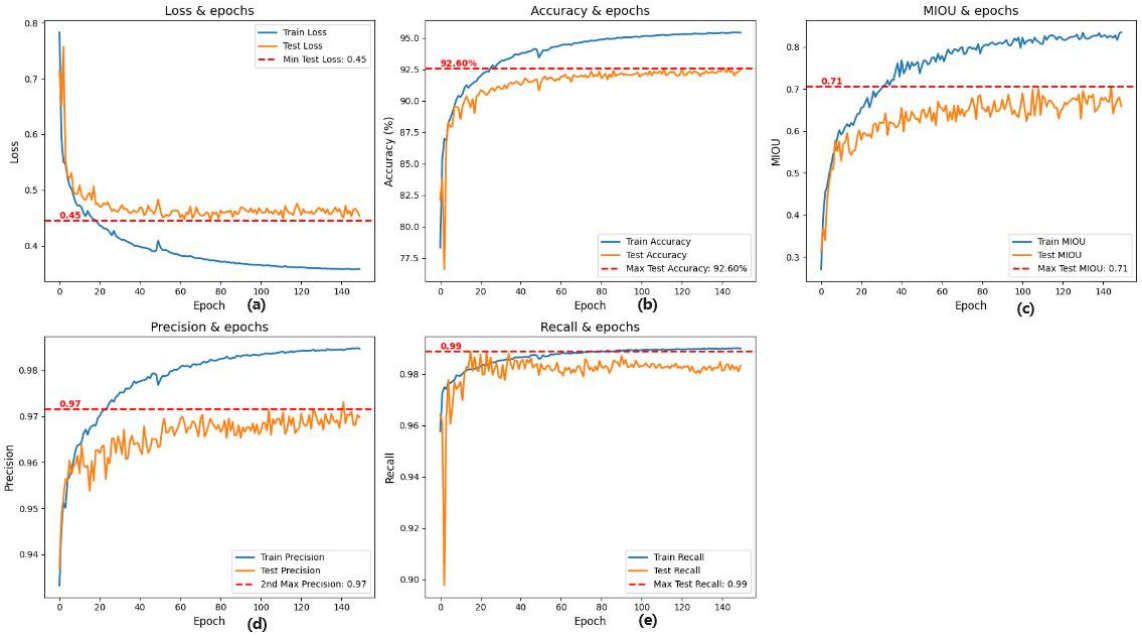
Table 3.Each Categories of MIOU for 34 categories



Original Image Real Labels Predicted Label Fusion results

Figure 20.Segmentation Result for 34 categories

- **19 categories**



(a)Loss=0.45;(b)Acc=92.60;(c)MIOU=0.71;(d)Precision=0.97;(e)Recall=0.99

Figure 21.19 Categories of Metric

CLASS	Ego vehicle	Rectification_border	Road	Sidewalk	Rail_track	Building	Guard_rail	Bridge	Traffic_sign	Vegetation	Terrain	Sky	Person	Rider	Car	Bus	Caravan	Train	Bicycle	OVERALL
MIOU	0.60	0.70	0.96	0.76	0.61	0.84	0.56	0.74	0.56	0.88	0.69	0.91	0.77	0.59	0.9	0.63	0.56	0.73	0.64	0.717

Table 4.Each Categories of MIOU in for categories

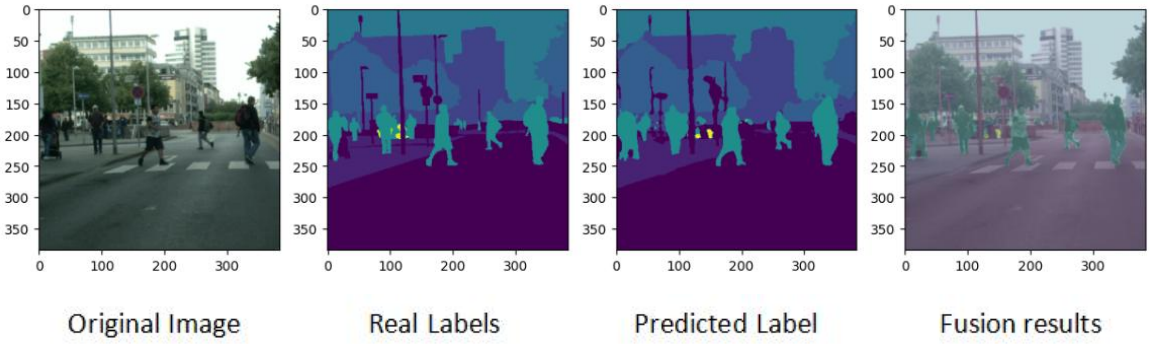
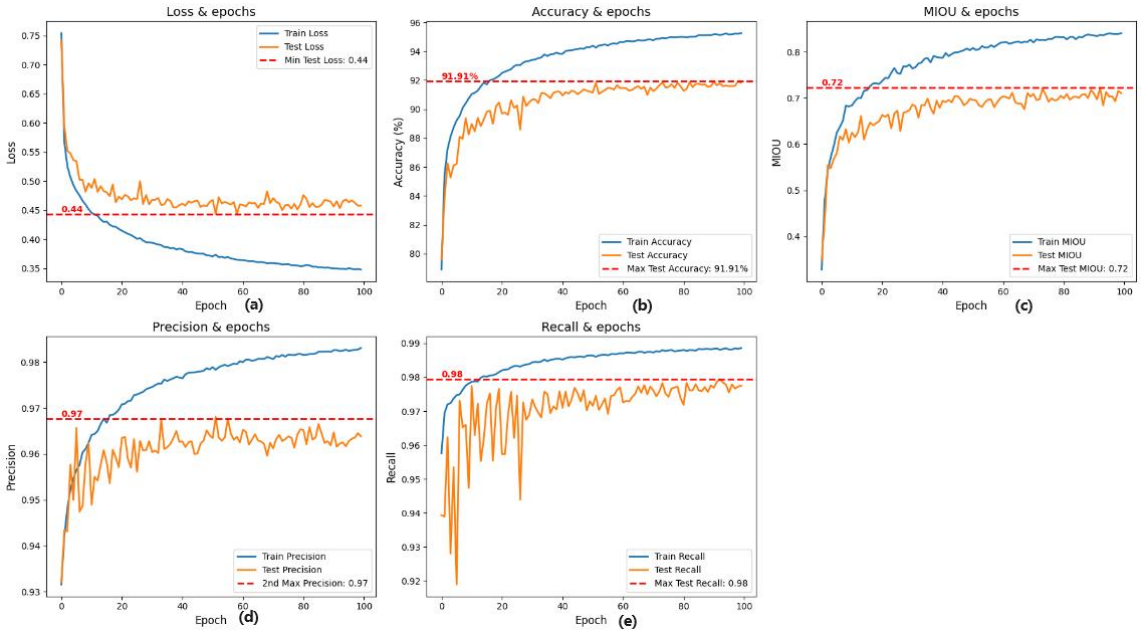


Figure 22.Segmentation Result for 19 categories

- **15 categories**



(a)Loss=0.44;(b)Acc=91.91;(c)MIOU=0.72;(d)Precision=0.97;(e)Recall=0.98

Figure 23. 15 Categories of Metric

CLASS	Ground	Road	Sidewalk	Building	Wall	Bridge	Vegetation	Sky	Person	Rider	Car	Truck	Bus	Train	Bicycle	OVERALL
MIOU	0.54	0.6	0.94	0.71	0.52	0.85	0.32	0.55	0.88	0.91	0.74	0.89	0.63	0.74	0.64	0.70

Table 5.Each Categories of MIOU in for 15 categories

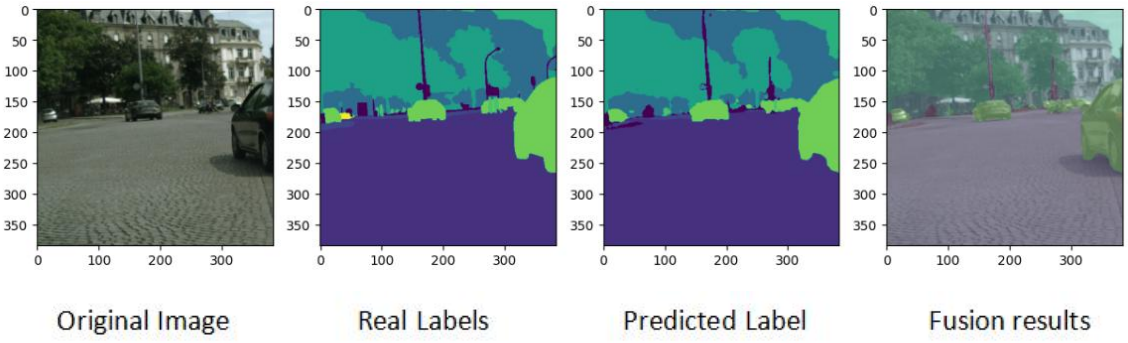
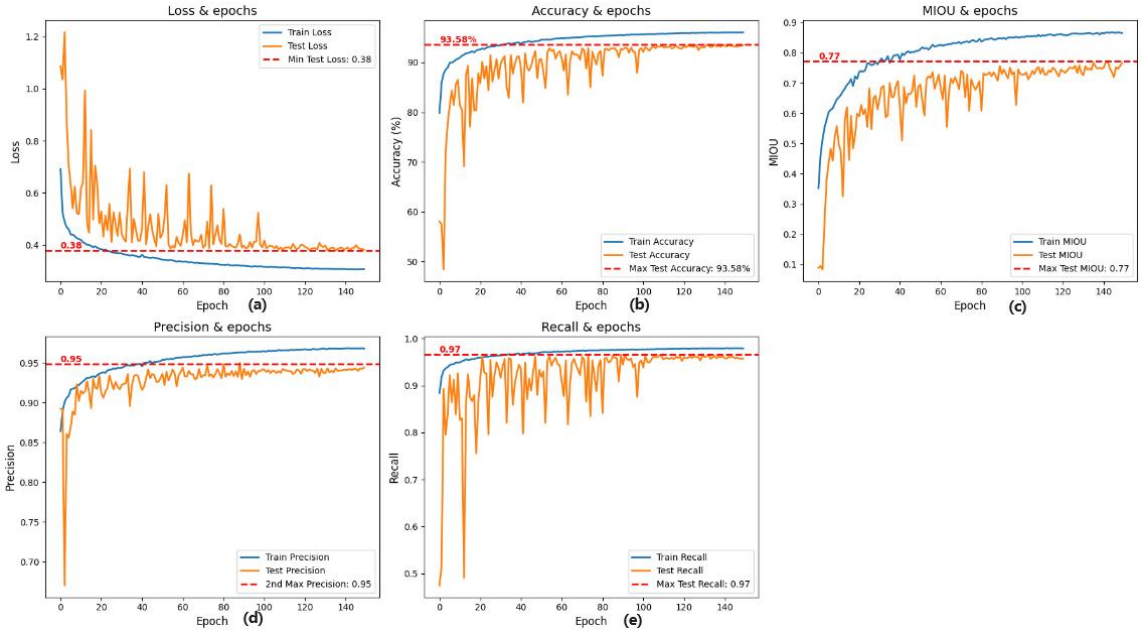


Figure 24.Segmentation Result for 15 categories

- **11 categories**



(a)Loss=0.38;(b)Acc=93.58;(c)MIOU=0.77;(d)Precision=0.95;(e)Recall=0.97

Figure 25.11 Categories of Metric

CLASS	Road	Sidewalk	Building	Vegetation	Sky	Person	Rider	Car	Truck	Bus	Bicycle	OVERALL
MIOU	0.91	0.69	0.84	0.88	0.94	0.79	0.57	0.9	0.57	0.7	0.58	0.761

Table 6.Each Categories of MIOU in for 11 categories

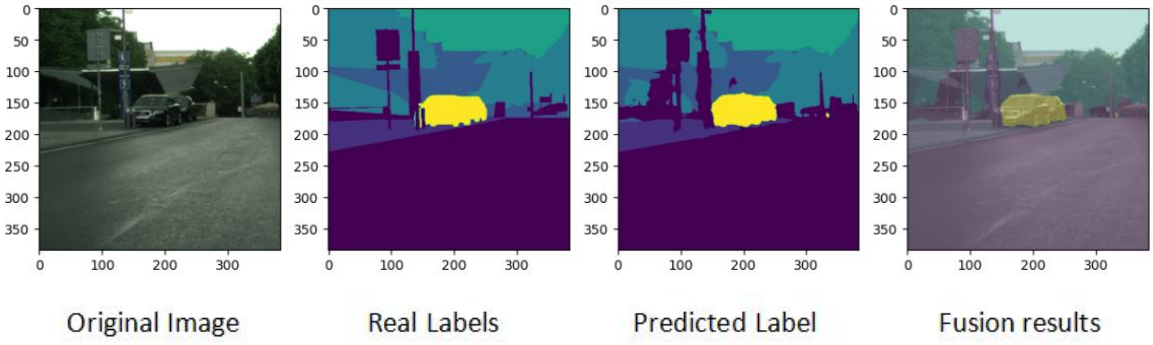


Figure 26.Segmentation Result for 11 categories

No. of categories	Accuracy	Loss	Precision	Recall	MIOU	PA	CPA	Dice Coefficient	Kappa	FPS
34 categories	91.77%	0.55	1	1	0.54	0.9516	0.6315	0.6510	0.8902	150(A6000)
19 categories	92.60%	0.45	0.97	0.99	0.71	0.9307	0.7188	0.7413	0.9037	149(A6000)
15 categories	91.91%	0.44	0.97	0.98	0.70	0.921	0.8114	0.8193	0.8931	148(A6000)
11 categories	93.58%	0.38	0.95	0.97	0.77	0.9316	0.8510	0.8529	0.8972	63.28(3090Ti)

Table 7. Comparison of the performance of different categories

In Table 7, the datasets using 34, 19, and 15 categories were trained on an A6000 graphics card, which caused significant resource consumption. In contrast, the dataset with 11 categories was trained on an RTX3090Ti graphics card. Considering the balance between training efficiency and computational resources, this project will use the 11-category dataset for subsequent training efforts.

(1) Accuracy

According to Figure 27, the model finally obtained an accuracy of 93.81. It can be clearly seen that the validation accuracy oscillates greatly in the early stage of training, and is relatively stable in the later stage. This is due to the fact that the learning rate is set too high in the early stage, and the weights of the model are too aggressive leading to too much change in each step, thus causing sharp fluctuations in the accuracy rate. In the later stage of training, the learning rate gradually stabilizes by decaying to a relatively small value. For the semantic segmentation of complex scenes and multi-category fine-grained segmentation, the accuracy rate reaches 93.81% indicating that the model can recognize and segment various objects in the image very well.

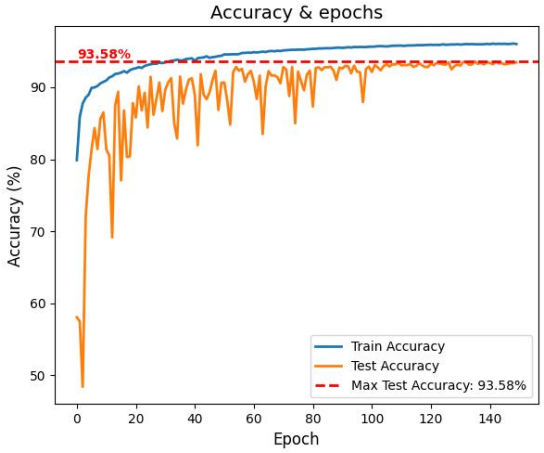


Figure 27. Accuracy

(2) Loss

The change of loss is shown in Figure 28, which also has a strong vibration in the early stage and stabilizes in the later stage. This is also due to the high learning rate in the early stage of training, the update step of the model weights may be too large, resulting in the model "jumping" on the surface of the loss function, which causes sharp fluctuations in the loss. In addition, because it was in the 512 * 1024 size of the image randomly cropped out of the 384 * 384 size of the image, which makes it difficult for the model to learn all the features in the pre-training period, with the introduction of more data and the model's adaptation to the distribution of data, the size of the loss gradually stabilized. However, in the semantic segmentation task, the loss function is mainly used as a metric to optimize the model during training rather than an indicator to evaluate the model's performance, which can only help to determine whether the model is learning from the data and whether it is moving in the direction of reducing the prediction error.

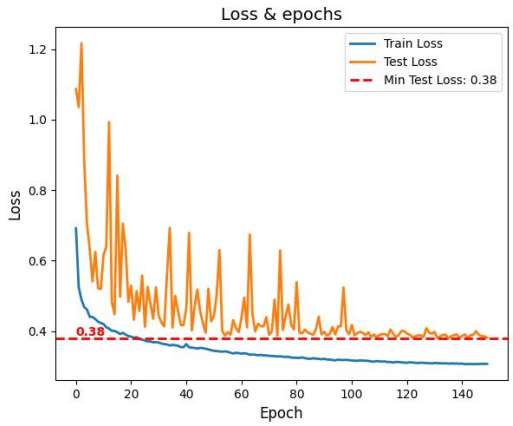


Figure 28. Loss

(3) MIOU

The figure below shows the rising curve of MIOU during the training process. It can be clearly seen that the value of MIOU reaches 0.77, which is an okay result on the Cityscapes dataset, indicating that the model has a high generalization ability when dealing with complex urban street scenes. Based on the images, it can be seen that the training set consistently outperforms the test set, which is an expected situation since the model learns directly on the training set. However, the MIOU of the test set stabilizes at a high level, which means that the model has better convergence. In the pre-training period, the size of the MIOU is much smaller than that of the validation set, which is due to the fact that randomly crop the original image, crop a smaller region of the original

image for training to get a larger field of view, which is good for the model to learn more details.

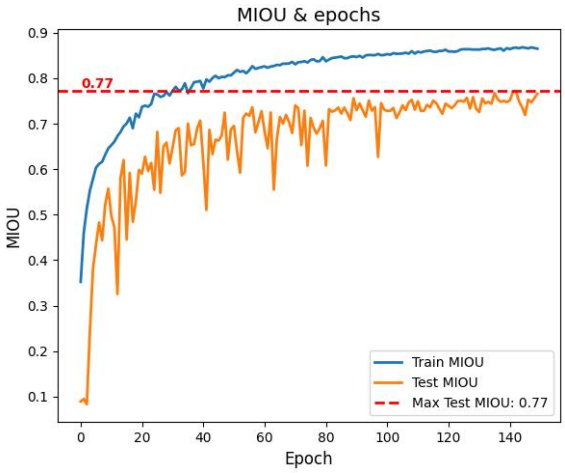


Figure 29. MIOU

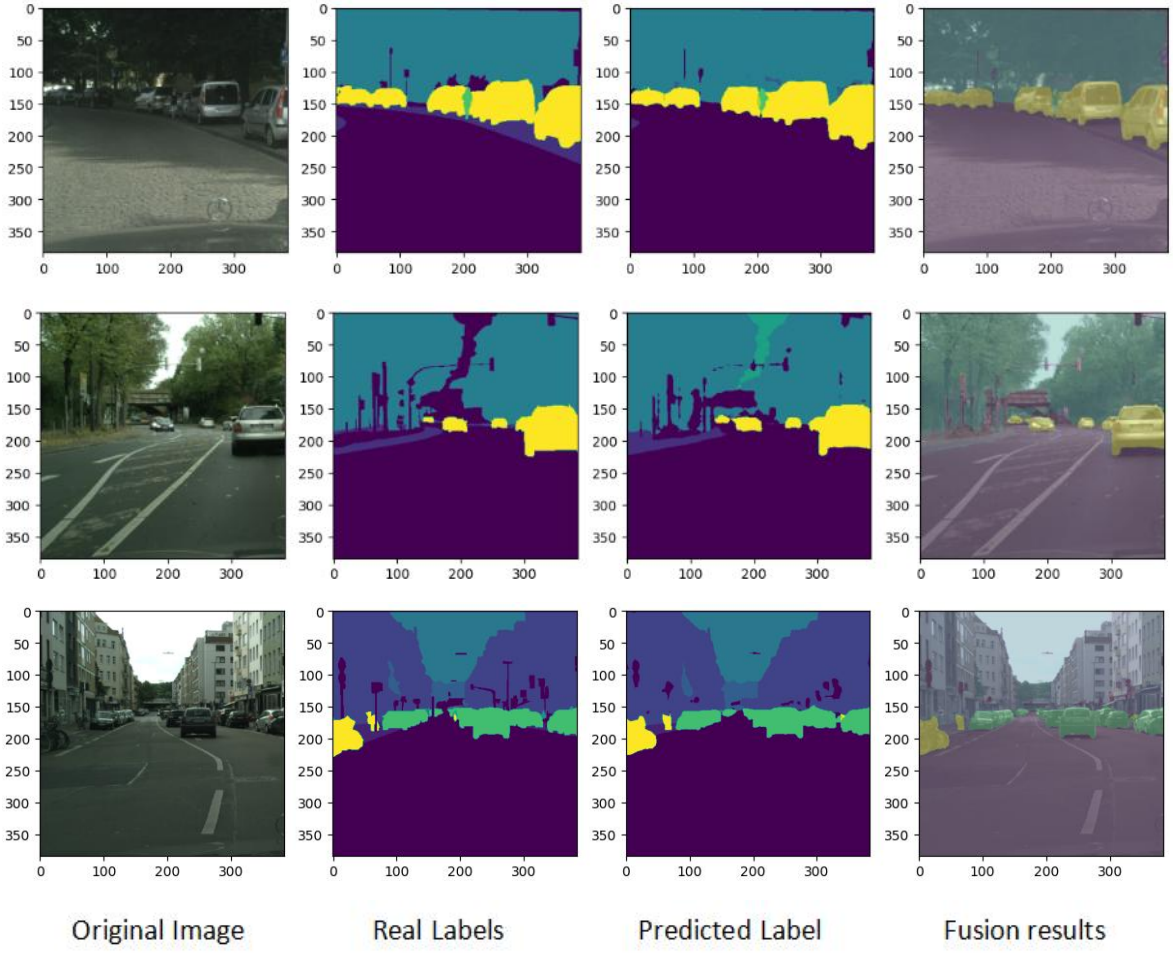


Figure 30. Segmentation Result

(4) Precision

As can be seen from the figure below, the model's accuracy is 0.95, which is a high result, showing that 95% of the samples that the model predicts as positive categories do belong to positive categories, which shows that the model is very precise and stable in determining pixel categories, showing that the model has a high generalisation ability. In addition, the training and testing accuracies are very close to each other, which also shows that the model has a high generalisation ability on both the training and validation sets. At about 80 epochs, the accuracy reaches a steady state and does not change much at subsequent times.

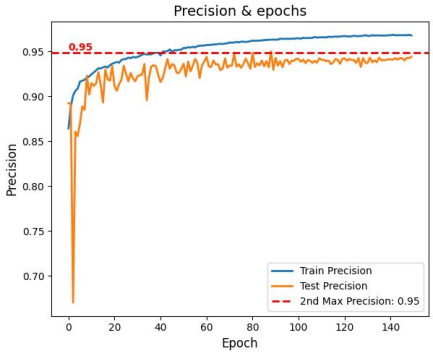


Figure 31. Precision

(5) Recall

The recall of the model reached 0.97, which is a relatively high value, which means that the model was able to recognize the majority of positively classified samples in the dataset. In addition, the results of the training and validation sets are very similar and remain high, indicating that the model has good recognition ability on both datasets and there is no overfitting. Although the recall fluctuates more drastically in the early stages, it stabilizes at about 100 epochs, which is due to the fact that the model did not learn enough features in the early stages.

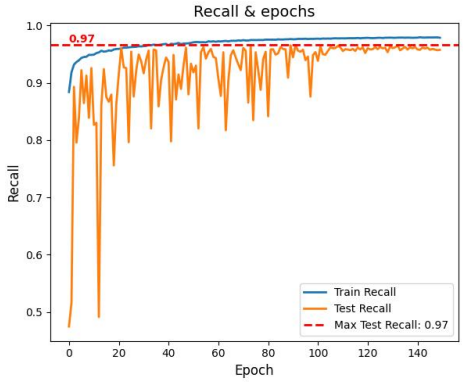


Figure 32. Recall

(6) Other Metrics

PA	CPA	Dice Coefficient	Kappa	FPS
0.9316	**0.8510**	**0.8529**	**0.8972**	63.28

Table 8. Other Metrics

According to the above table it can be seen that the pixel accuracy (PA) of the model reaches 0.9316, which indicates that the model correctly classifies 93.16% of the pixel points, which means that the model has a better performance on Cityscapes, which is a more complex dataset. Also, the average pixel accuracy reached 0.8510, which indicates that the model has an average accuracy of 85.1% on each category and it shows the performance of the model on all categories.

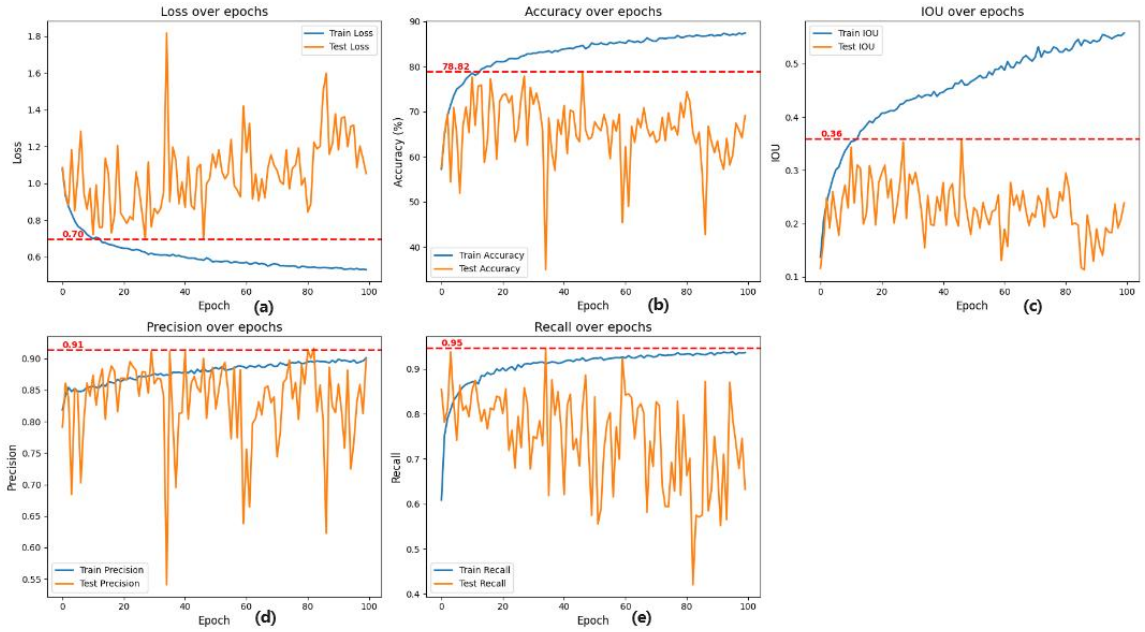
The Dice coefficient reached 0.8529, and although this metric is commonly used for medical image segmentation, it gives a good indication of the model's performance for similar cases like data imbalance. This metric is similar to IOU, which also measures the overlap between predicted segmentation and true segmentation, and 0.8529 means that the model has good segmentation performance.

The kappa score reaches 0.8972 and this metric measures the classification accuracy while considering data imbalance. And the kappa score of 89.72% indicates that the model has a very reliable classification performance.

Finally, the model achieves an FPS of 63, indicating that the model can process 63 frames per second while processing the video stream. This is very important for real-time semantic segmentation, as it ensures that the model can respond very quickly to complex scene inputs while ensuring accuracy.

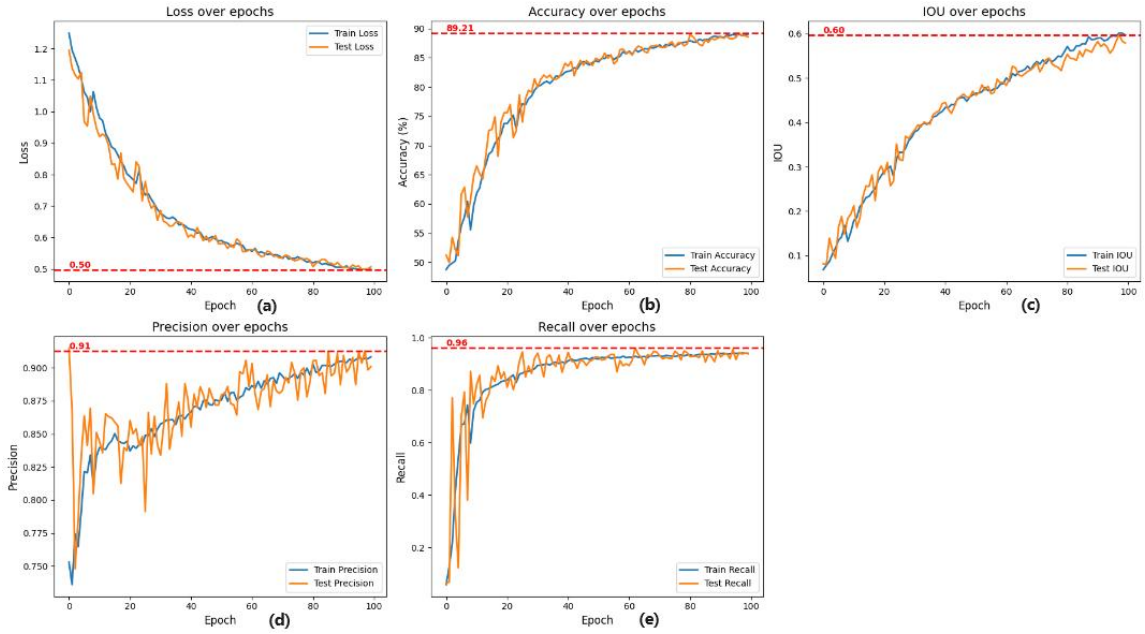
4.2 Comparison with Other Models & fine-tuning

4.2.1 Comparison of Common Models



(a)Loss=0.70;(b)Acc=78.82;(c)MIOU=0.36;(d)Precision=0.91;(e)Recall=0.95

Figure 33. Metric of LinkNet



(a)Loss=0.50;(b)Acc=89.21;(c)MIOU=0.60;(d)Precision=0.91;(e)Recall=0.96

Figure 34. Metric of UNet

Model	Accuracy	Loss	Precision	Recall	MIOU	PA	CPA	Dice Coefficient	Kappa	FPS
LinkNet	78.82	0.70	0.91	0.95	0.36	0.68	0.30	0.3084	0.49	61.8
UNet	89.21	0.50	0.91	0.96	0.60	0.89	0.67	0.6732	0.83	33.5
proposed model	93.58	0.38	0.95	0.97	0.76	0.9316	0.8510	0.8529	0.8972	63

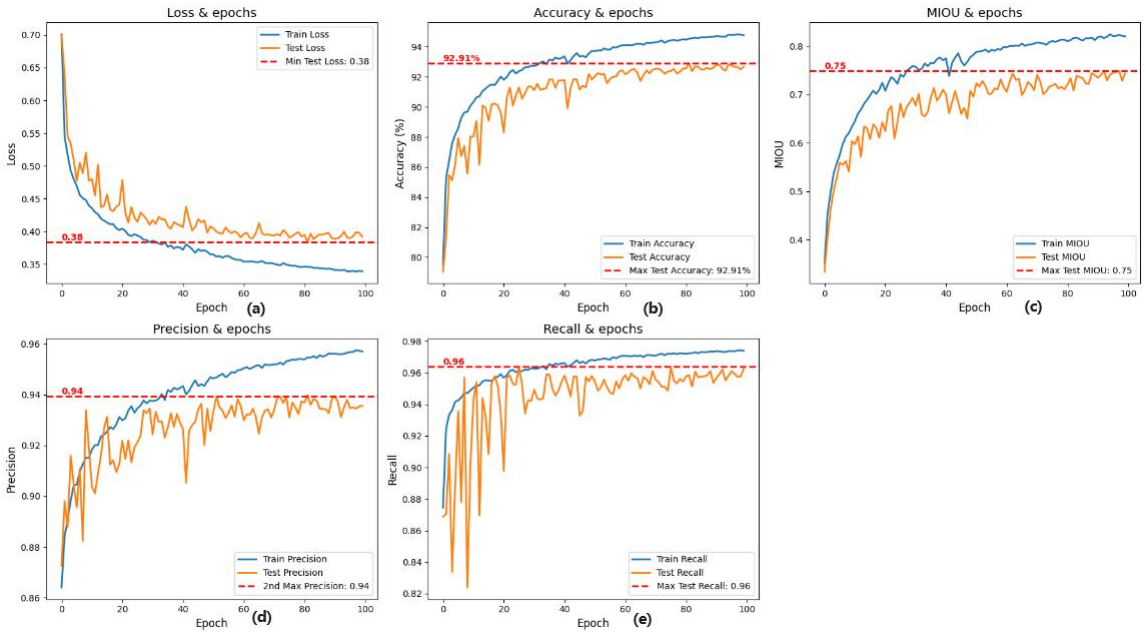
Table 9. Comparison of Common Models

In this research, proposed model is compared with common semantic segmentation models LinkNet and UNet through comparative analysis. The results show that the proposed model significantly outperforms the compared models in several key performance metrics. Specifically, the accuracy of the proposed model reaches 93.58%, which is much better than the 78.82% of LinkNet and 89.21% of UNet; in terms of the value of loss function, loss value of the proposed model is only 0.38, while loss values of LinkNet and UNet are 0.70 and 0.50, respectively, which shows a higher error rate. In addition, proposed model performs well in terms of precision, recall, MIOU, PA, CPA, Dice coefficient and Kappa coefficient, especially in terms of MIOU, proposed model reaches 0.76, which is much higher than the 0.6 of LinkNet and UNet, which fully reflects its efficient segmentation ability. In terms of processing speed, the FPS of proposed model is much larger than these two models. Overall, the proposed model outperforms LinkNet and UNet in all metric comparisons.

4.2.2 Comparison of Other backbones

In this study, different backbone networks was integrated into this semantic segmentation model and evaluated the training results of the model to get the best semantic segmentation model. (The following results are all generated based on RTX3090Ti training)

- Result of InceptionV3



(a)Loss=0.38;(b)Acc=92.91;(c)MIOU=0.75;(d)Precision=0.94;(e)Recall=0.96

Figure 35. InceptionV3 Backbone Metric

CLASS	Road	Sidewalk	Building	Vegetation	Sky	Person	Rider	Car	Truck	Bus	Bicycle	OVERALL
MIOU	0.89	0.68	0.85	0.86	0.91	0.71	0.46	0.9	0.57	0.88	0.62	0.756

Table 10.Each class IOU with InceptionV3

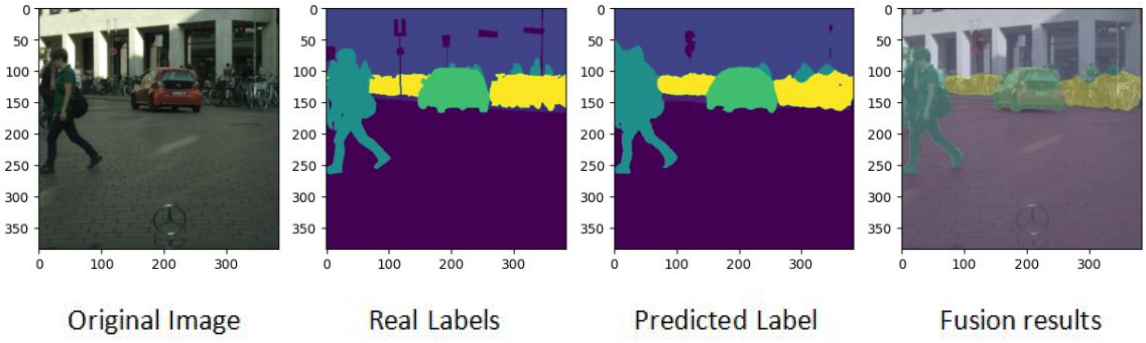
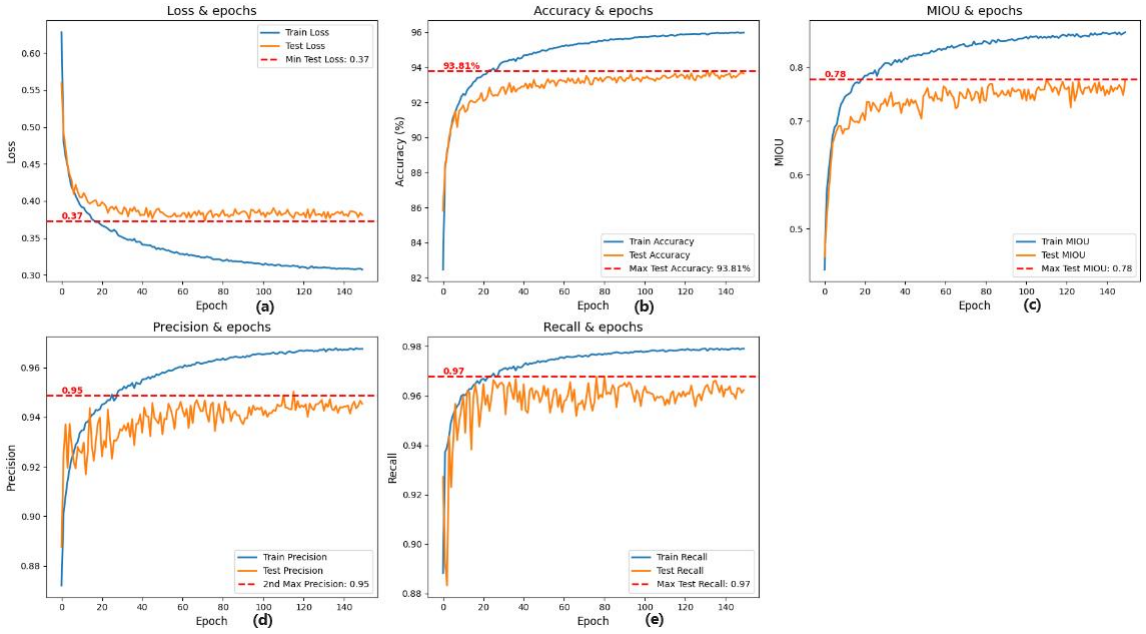


Figure 36. Segmentation Result for InceptionV3

- Result of Xception



(a)Loss=0.37;(b)Acc=93.81;(c)MIOU=0.78;(d)Precision=0.95;(e)Recall=0.97

Figure 37. Xception Backbone Metric

CLASS	Road	Sidewalk	Building	Vegetation	Sky	Person	Rider	Car	Truck	Bus	Bicycle	OVERALL
MIOU	0.91	0.72	0.85	0.92	0.70	0.41	0.91	0.9	0.31	0.78	0.58	0.73

Table 11.Each class IOU with Xception

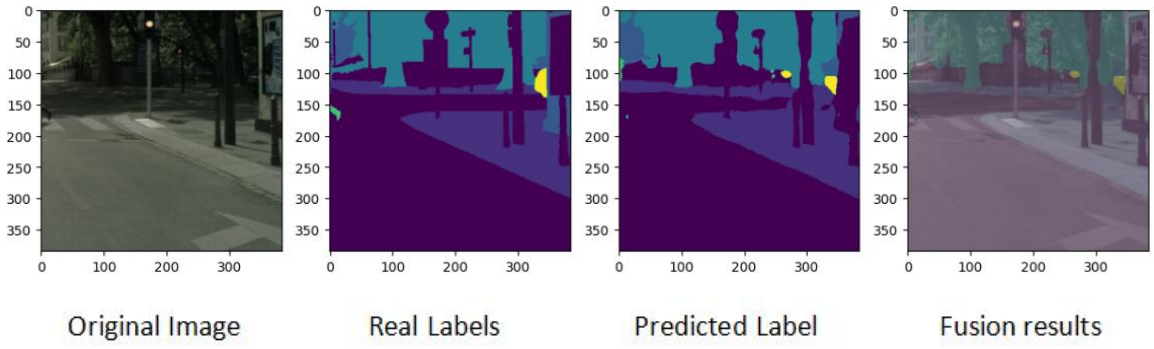
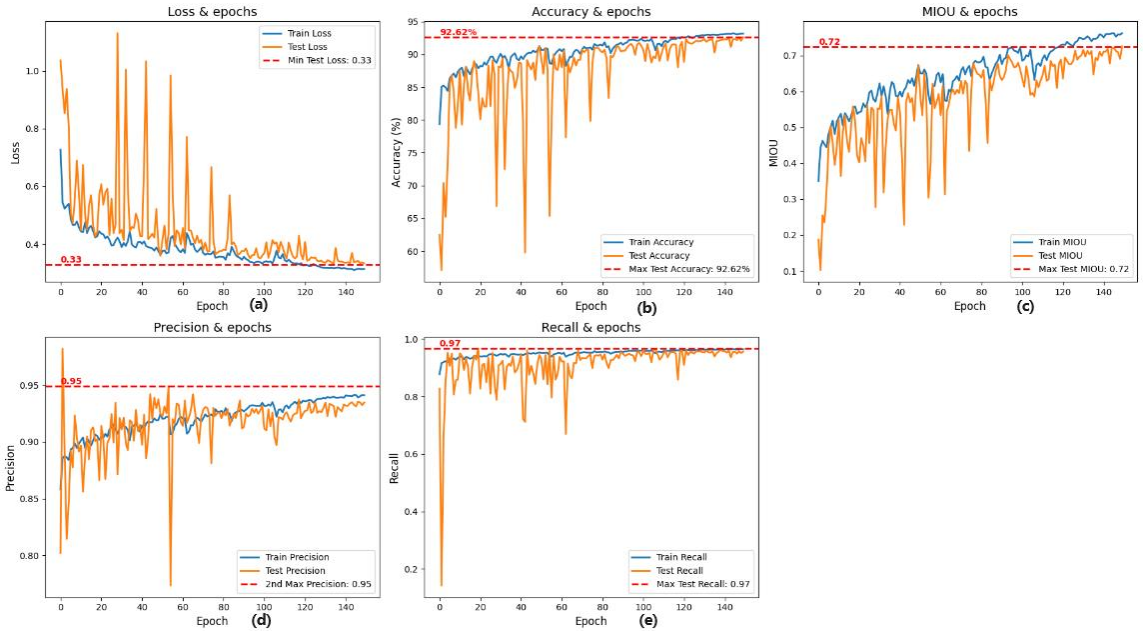


Figure 38.Segmentation Result for Xception

- Result of DenseNet121



(a)Loss=0.33;(b)Acc=92.62;(c)MIOU=0.72;(d)Precision=0.95;(e)Recall=0.97

Figure 39. DenseNet121 Backbone Metric

CLASS	Road	Sidewalk	Building	Vegetation	Sky	Person	Rider	Car	Truck	Bus	Bicycle	OVERALL
MIOU	0.89	0.65	0.83	0.86	0.91	0.67	0.46	0.89	0.70	0.57	0.59	0.732

Table 12. Each class IOU with DenseNet121

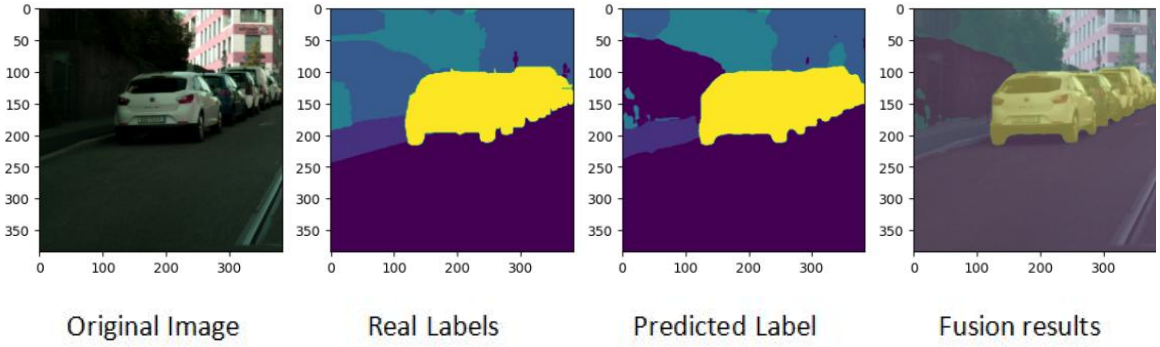
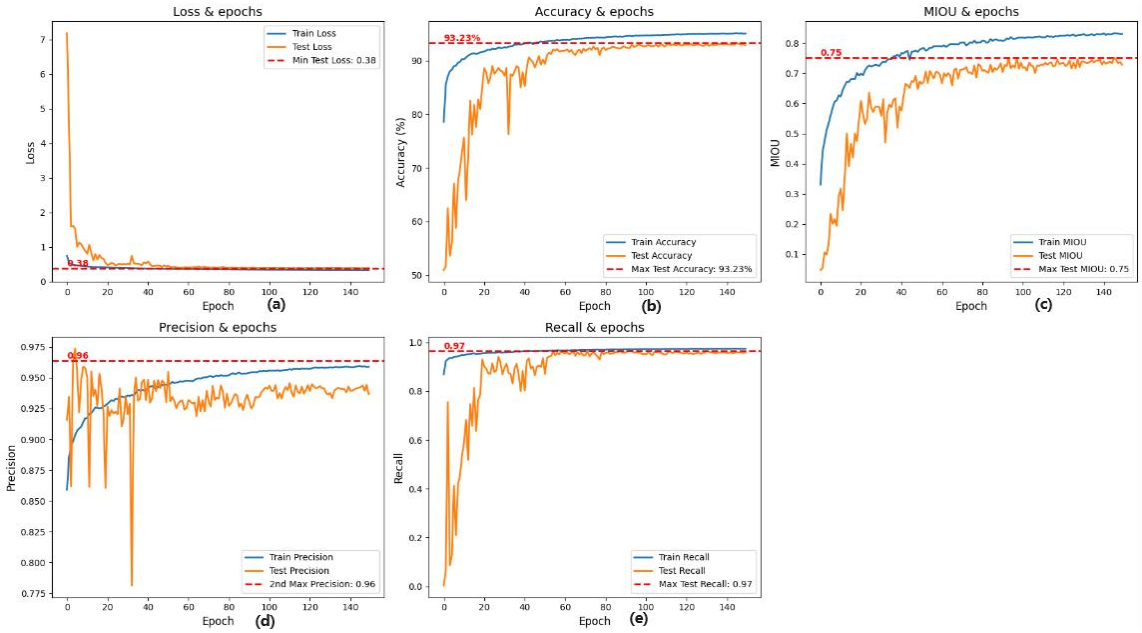


Figure 40. Segmentation Result for DenseNet121

- Result of MobileNetV2



(a)Loss=0.38;(b)Acc=93.23;(c)MIOU=0.75;(d)Precision=0.96;(e)Recall=0.97

Figure 41. MobileNetV2 Backbone Metric

CLASS	Road	Sidewalk	Building	Vegetation	Sky	Person	Rider	Car	Truck	Bus	Bicycle	OVERALL
MIOU	0.90	0.69	0.83	0.88	0.91	0.73	0.50	0.89	0.46	0.62	0.54	0.73

Table 13. Each class IOU with MobileNetV2

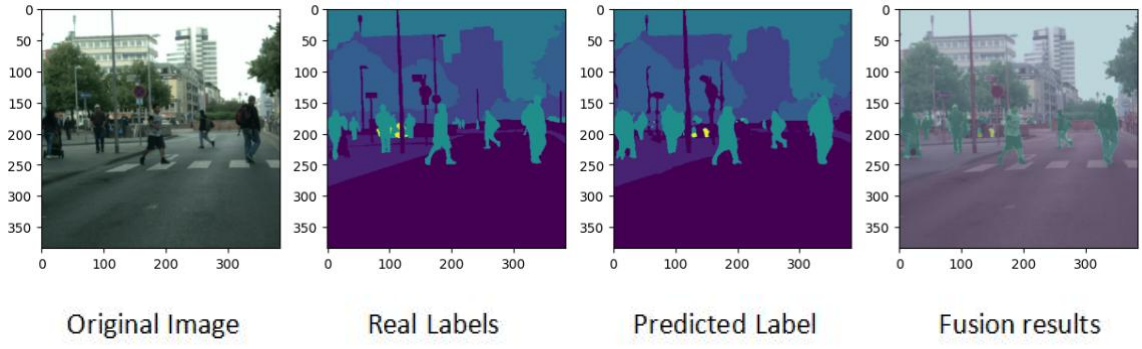
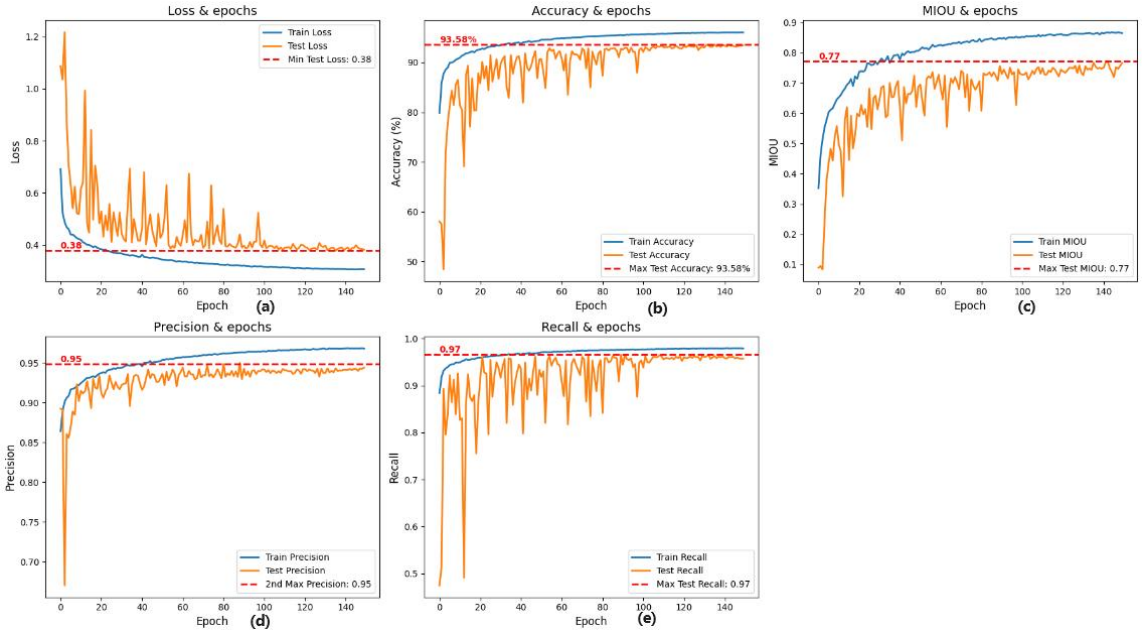


Figure 42. Segmentation Result for MobileNetV2

- Result of ResNet50



(a)Loss=0.38;(b)Acc=93.58;(c)MIOU=0.77;(d)Precision=0.95;(e)Recall=0.97

Figure 43. ResNet50 Backbone Metric

CLASS	Road	Sidewalk	Building	Vegetation	Sky	Person	Rider	Car	Truck	Bus	Bicycle	OVERALL
MIOU	0.91	0.69	0.84	0.88	0.94	0.79	0.57	0.9	0.57	0.7	0.58	0.761

Table 14. Each class IOU with ResNet50

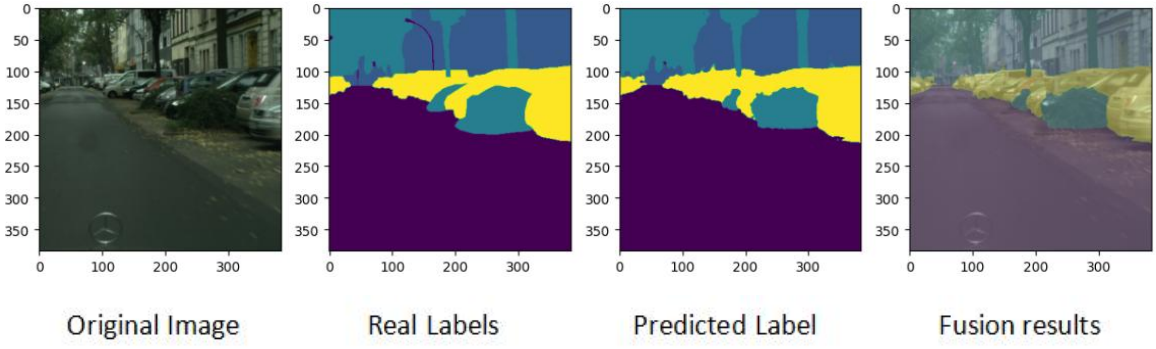


Figure 44. Segmentation Result for MobileNetV2

With the above results, the following Table 15 and Figure 45 is obtained:

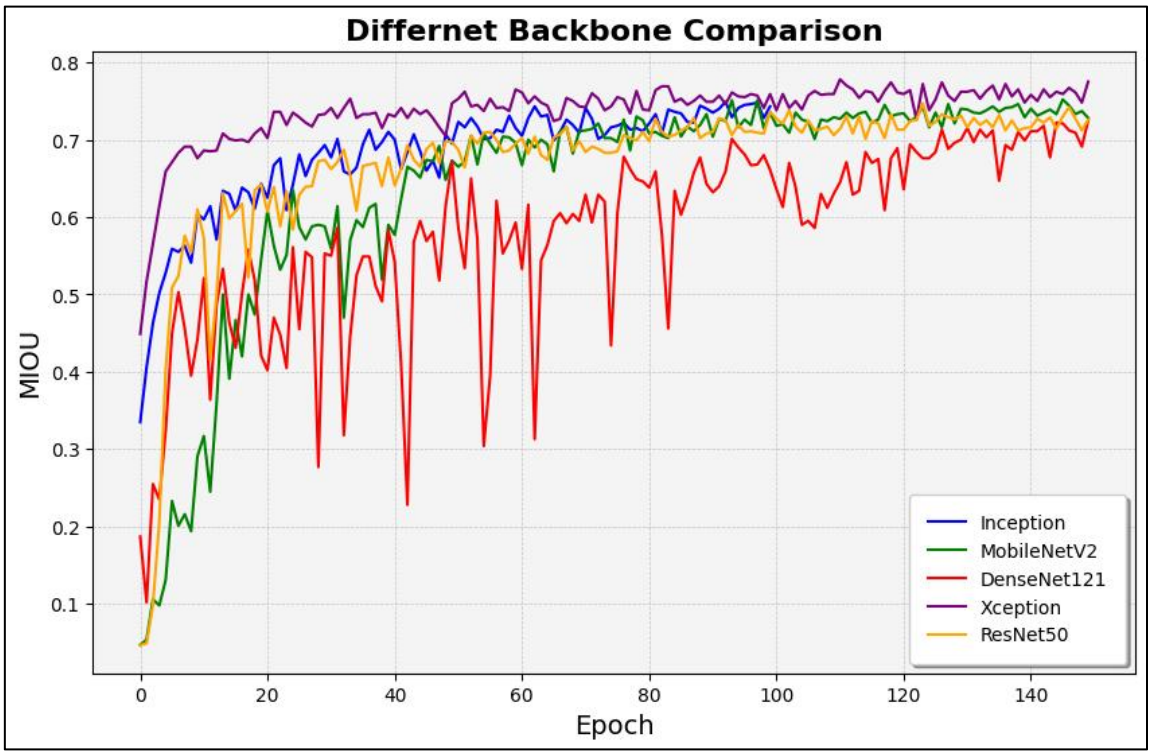


Figure 45 Different backbone comparison

Backbone	Accuracy	Loss	Precision	Recall	MIOU	PA	CPA	Dice Coefficient	Kappa	FPS	Weight
InceptionV3	92.91%	0.38	0.94	0.96	0.73	0.9276	0.8386	0.8422	0.8934	62.67	65.4MB
Xception	93.81%	0.37	0.95	0.97	0.73	0.9352	0.8592	0.8567	0.9078	63.28	122.4MB
DenseNet121	92.62%	0.33	0.95	0.97	0.72	0.9233	0.8377	0.8342	0.8902	62.23	44.1MB
MobileNetV2	93.23%	0.38	0.96	0.97	0.72	0.9292	0.8278	0.8294	0.9875	63.96	30.8MB
proposed model(ResNet50)	93.58%	0.38	0.95	0.97	0.77	0.9316	0.8510	0.8529	0.8972	62.14	60.1MB

Table 15. Comparison of Other backbones models

The table 15 shows that the performance of using Xception as backbone is best. It is better than the other backbone networks in most of the performance metrics, especially in accuracy (93.81%), PA (0.9352), and CPA (0.8592), which shows the best performance. This shows that the Xception backbone network is not only able to capture the features of the image more efficiently when dealing with the Cityscape dataset, but also improves the accuracy of the segmentation.

However, from the point of view of model size and efficiency, although using Xception as the backbone network gives the best segmentation results, it's model size is nearly twice the size of the other backbone networks, which means it is not applicable to some

mobile devices. Based on this concern, using MobileNetV2 as the backbone network is the most appropriate, it not only has the highest results in terms of FPS, but also has the advantage of being suitable for mobile devices due to the size of the model (30.8MB). In addition, its MIOU is not as high as ResNet50 and Xception, but the difference is small, which is not a big difference in segmentation performance. Therefore, MobileNetV2 is very suitable for use in resource-constrained conditions.

From the perspective of resource consumption, ResNet50 maintains the high performance metrics while the model size is relatively small (60.1MB), which strikes a good balance between model efficiency and performance.

In summary, under resource-constrained conditions, using MobileNetV2 as the backbone network is most appropriate. Under resource-sufficient conditions, higher accuracy and MIOU can be obtained by using a large model such as Xception. If there is a balance between performance and efficiency, ResNet50 would be a better fit.

4.2.3 Models in literature

Author	Model	MIou	Accuracy	Recall	FPS	Para(M)
Badrinarayanan et.al. **[20]**	SegNet	56.1%	*	*	*	29.46
Abdigapporov et.al.**[21]**	BiFPN	56.4%	89.6	79.8	65.7	*
Paszke et.al. **[22]**	ENet	58.3%	*	*	46.8	0.4
Poudel et.al. **[23]**	Fast-SCNN	68%	83.5	*	123.5	1.11
Yu et al **[24]**	BiSeNet	69%	65.5	*	65.5	14.1
Fourure et al **[25]**	GridNet	69.5%	*	*	*	*
Chen et al **[26]**	Deep-Lab CRF	70.4%	*	*	*	15.2
Lin et al **[27]**	RefineNet	73.6%	80.6	*	*	*
Li et.al.**[28]**	BiAttnNet	74.7%	*	*	89.2	2.2
My Model	proposed model	76%	93.58	0.97	63	10

Table 16. Comparison of Other models in literature

Through in-depth analysis and comparison, proposed model is compared with other well-known semantic segmentation models, demonstrating its outstanding performance and innovations in various key performance metrics.

First, in terms of MIoU, proposed model leads all compared models with a score of 76%, including the recent top performers BiAttnNet (74.7%) and RefineNet (73.6%). This result demonstrates the superior performance of proposed model in accurately segmenting the categories to which each pixel of an image belongs.

In terms of accuracy, proposed model achieves a high score of 93.58%, which is much better than BiFPN (89.6%) and RefineNet (80.6%), demonstrating its superior ability in correctly identifying image categories. Specifically, proposed model also achieves 0.97 in recall, showing that it is able to re-identify positive class samples almost perfectly, which is excellent among all the models listed.

For FPS (frames per second), proposed model ensures good real-time processing with a performance of 65 FPS. Although slightly lower than Fast-SCNN's 123.5 FPS, this processing speed is a reasonable balance between accuracy and real-time performance, given proposed model's significant advantages in other performance metrics.

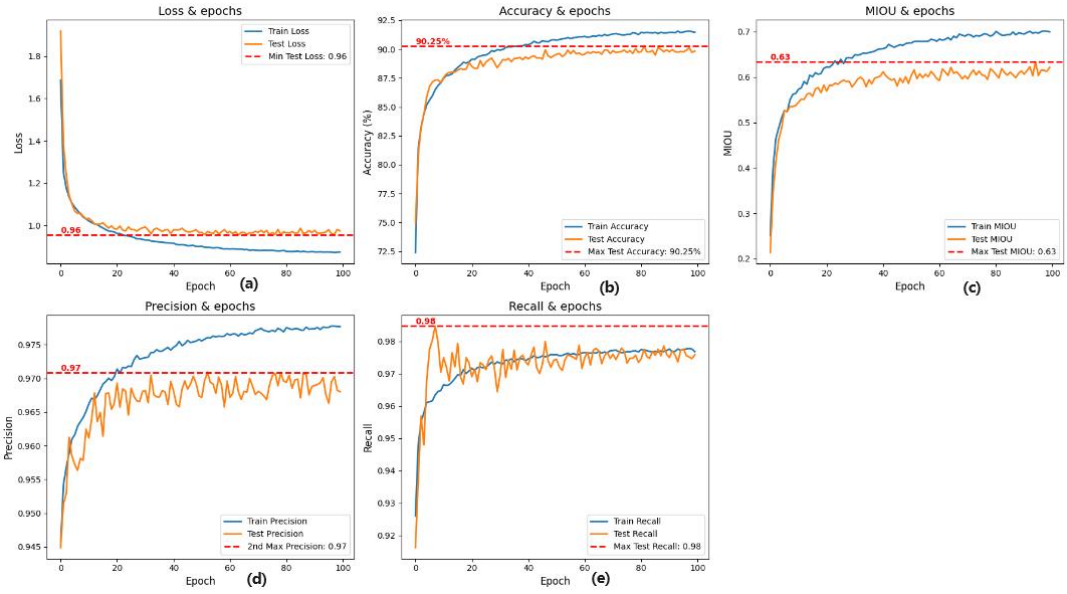
In terms of the number of model parameters (Para(M)), proposed model has a parameter count of 10M, which still achieves the best segmentation performance while maintaining a lower complexity compared to other models. This is in comparison to ENet (0.4M) and Fast-SCNN (1.11M), which have a much lower number of parameters, further demonstrating the results of proposed model in optimizing the model structure and improving efficiency.

In summary, proposed model performs well in the semantic segmentation task, not only leading in key metrics such as MIoU, accuracy and recall, but also achieving an excellent balance between processing speed and model efficiency. These results fully demonstrate the efficiency and sophistication of proposed model, providing new perspectives and solutions for future image processing and analysis tasks.

4.2.4 Model Fine-Tuning

Below is a part of the model fine-tuning to show the impact of different loss functions, optimizer and disabling different modules on the results.

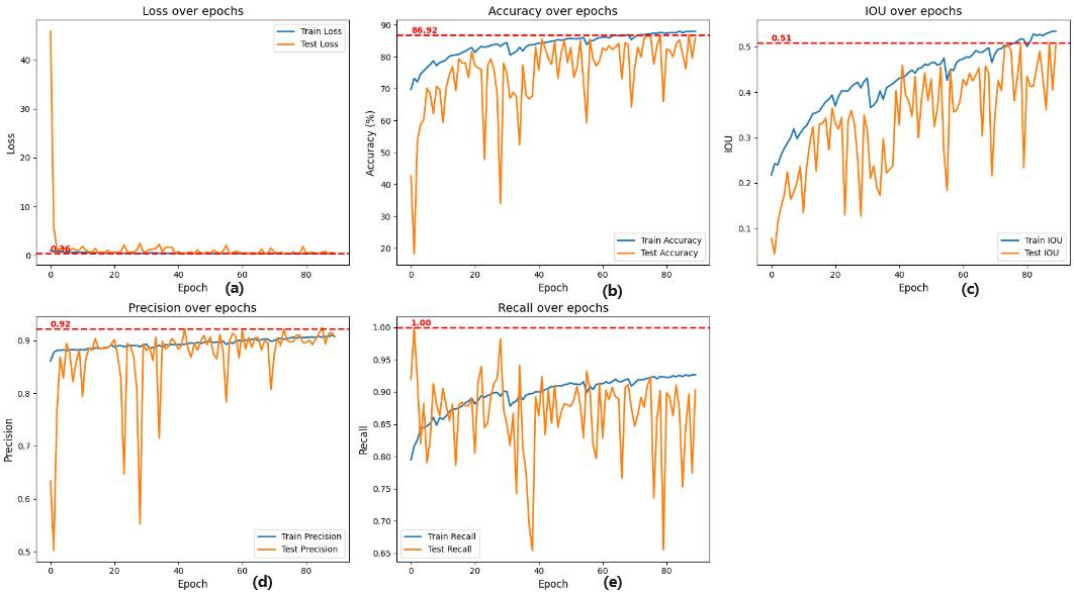
(1) Train with focal loss



(a)Loss=0.96;(b)Acc=90.25;(c)MIOU=0.63;(d)Precision=0.97;(e)Recall=0.98

Figure 46. Metric with focal loss

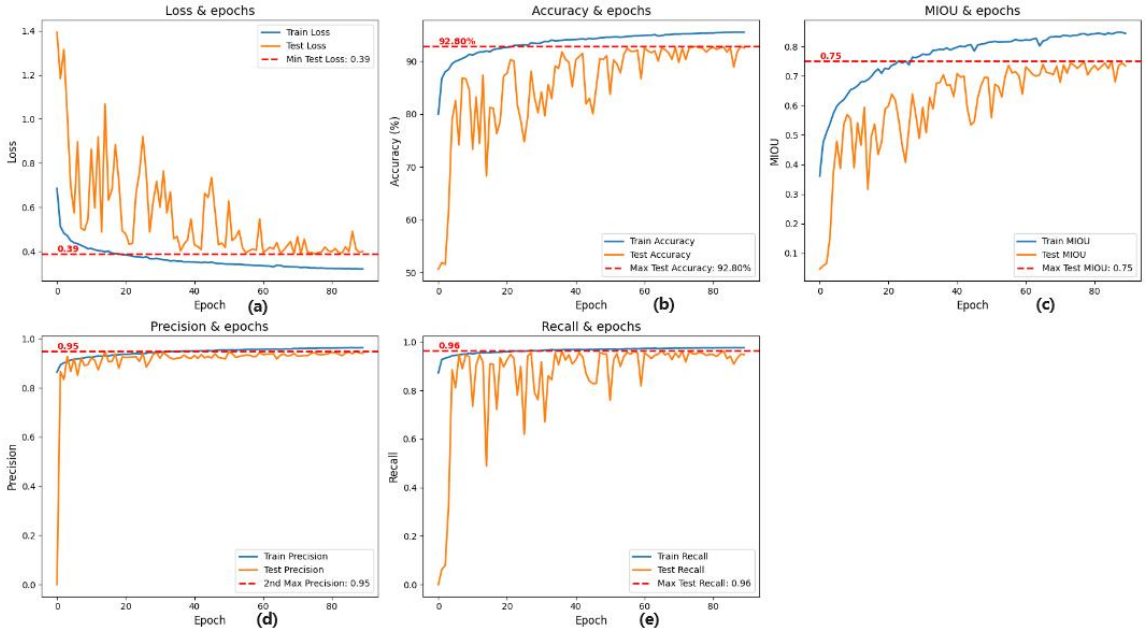
(2) Train with Adam&sparsecategoricalcrossentropy



(a)Loss=0.36;(b)Acc=86.92;(c)MIOU=0.51;(d)Precision=0.92;(e)Recall=1

Figure 47. Metric with adam& sparsecategoricalcrossentropy

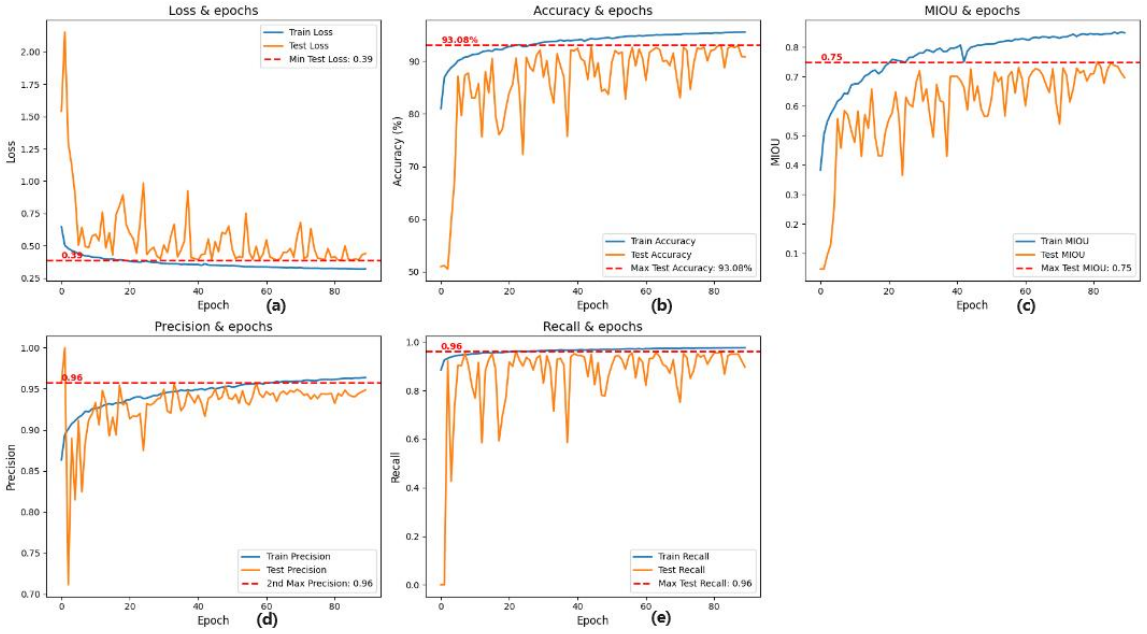
(3) Train without context enhancement



(a)Loss=0.39;(b)Acc=92.80;(c)MIOU=0.75;(d)Precision=0.95;(e)Recall=0.96

Figure 48. Metric without context enhancement

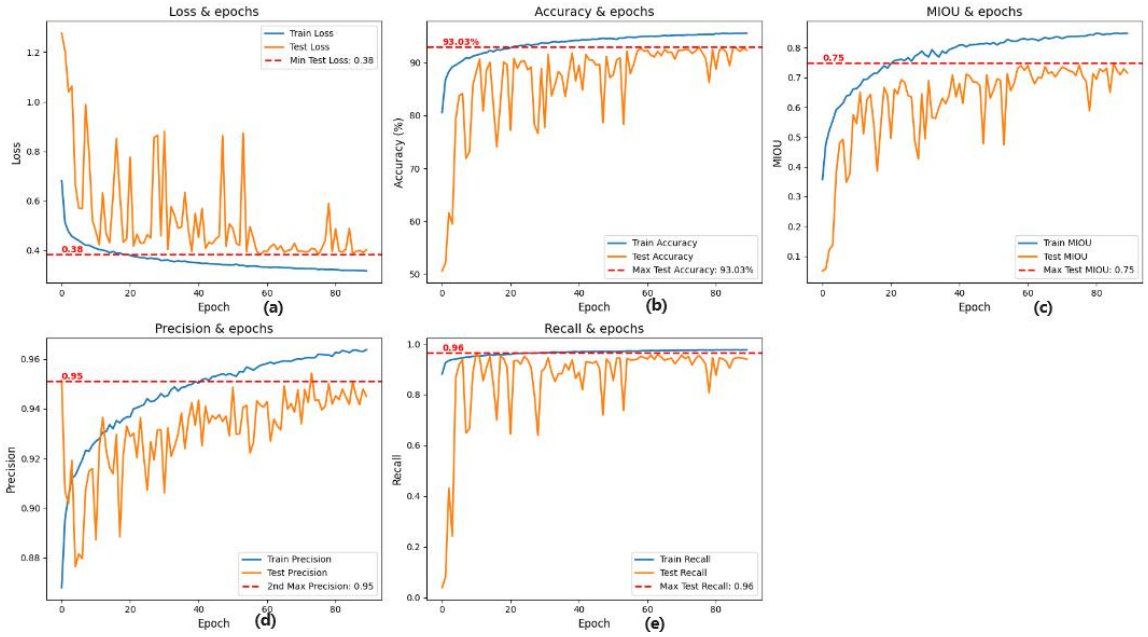
(4) Train without edge detection module



(a)Loss=0.39;(b)Acc=93.08;(c)MIOU=0.75;(d)Precision=0.96;(e)Recall=0.96

Figure 49. Metric without edge detection

(5) Train without transformers module



(a)Loss=0.38;(b)Acc=93.03;(c)MIOU=0.75;(d)Precision=0.95;(e)Recall=0.96

Figure 50. Metric without transformers

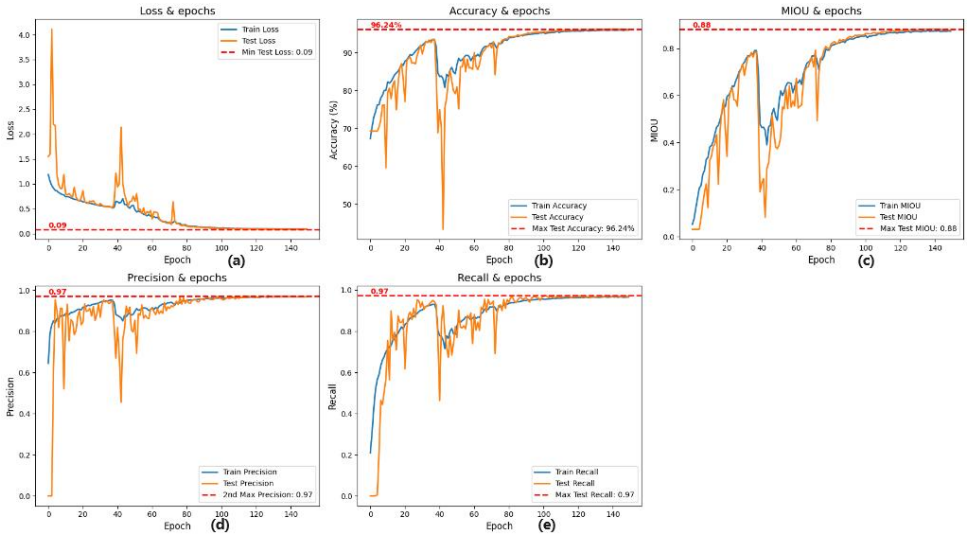
Model	Accuracy	Loss	Precision	Recall	MIOU	PA	CPA	Dice Coefficient	Kappa	FPS
(1)	90.25	0.96	0.97	0.98	0.63	0.90	0.73	0.74	0.87	148.16
(2)	86.92	0.36	0.92	1	0.51	0.86	0.61	0.62	0.80	65.25
(3)	92.80	0.39	0.95	0.96	0.75	0.93	0.82	0.84	0.89	66.90
(4)	93.08	0.39	0.96	0.96	0.75	0.90	0.78	0.80	0.86	66.23
(5)	93.03	0.38	0.95	0.96	0.75	0.92	0.82	0.83	0.88	67.00
proposed model	93.58	0.38	0.95	0.97	0.76	0.9316	0.8510	0.8529	0.8972	63.80

Table 17. Comparison of fine-tuning result

From the results, the focal loss function and the Adam optimizer are not very suitable for this project. However, the context enhancement module as well as the edge detection module do improve the performance of the model. Transforms module is effective in improving the training speed of the model although it makes the FPS slower.

4.2.5 Other Dataset Train

To ensure the generalisability of the model, this project also attempted to train the model using the VOC 2012 **[34]** dataset and the following results are presented.



(a)Loss=0.09;(b)Acc=96.24;(c)MIOU=0.88;(d)Precision=0.97;(e)Recall=0.97

Figure 51. VOC 2012 Result

CLASS	Person	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dinning_Table	Dog	Horse	Motorbike	Potted_plant	Sheep	Sofa	Train	TV/Monitor	Void/Edges	Unlabeled/BG	Overall
MIOU	0.97	0.89	0.53	0.88	0.88	0.89	0.96	0.94	0.94	0.81	0.94	0.95	0.94	0.91	0.87	0.88	0.41	0.94	0.95	0.95	0.96	0.48	0.86

Table 18. Each class IOU with VOC 2012 Dataset

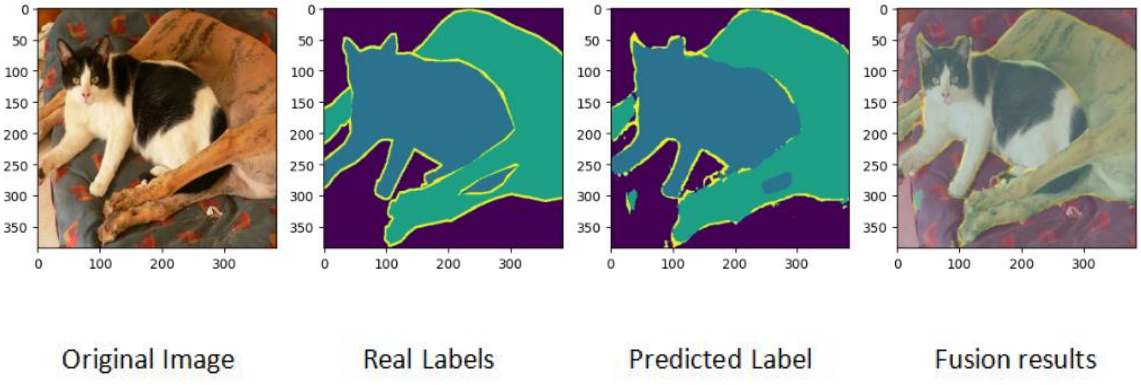


Figure 52. Segmentation Result for VOC 2012 Result

As can be seen from the above graph, a MIOU of 0.86 was obtained, which is at a more advanced level.

4.3 GUI Demonstration

In this project, to research semantic segmentation of urban datasets, an interactive website has been designed and implemented with the aim of improving the user experience and demonstrating the practical application of semantic segmentation techniques. The website provides four main functions: image segmentation, video segmentation, real-time segmentation and other dataset segmentation.

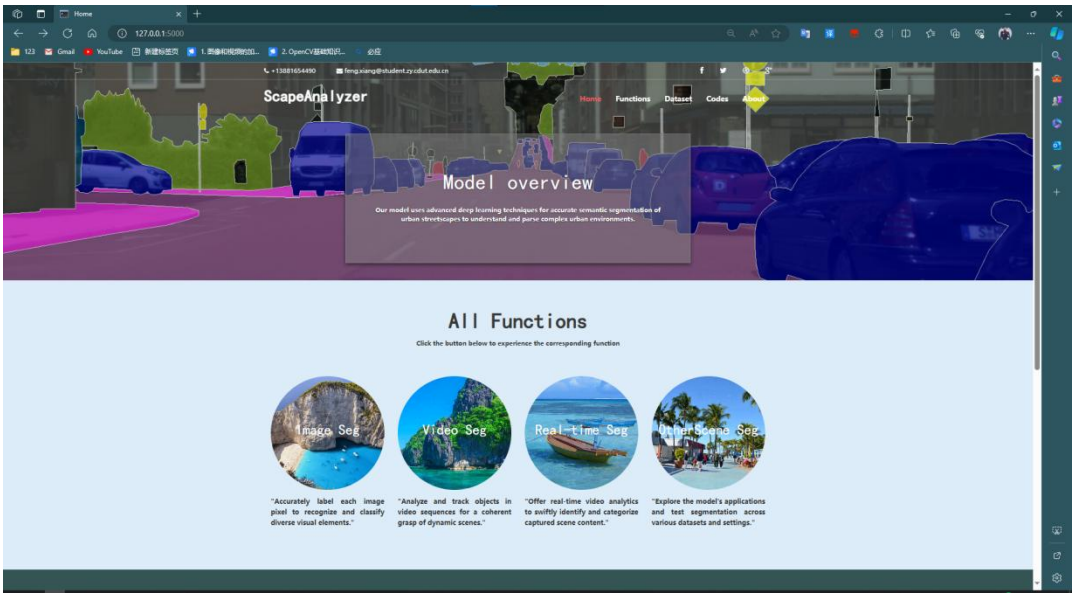


Figure 53. Web GUI

Figure 54 shows the image segmentation function, where users can upload a cityscape static image and the system will automatically segment and label different city elements.

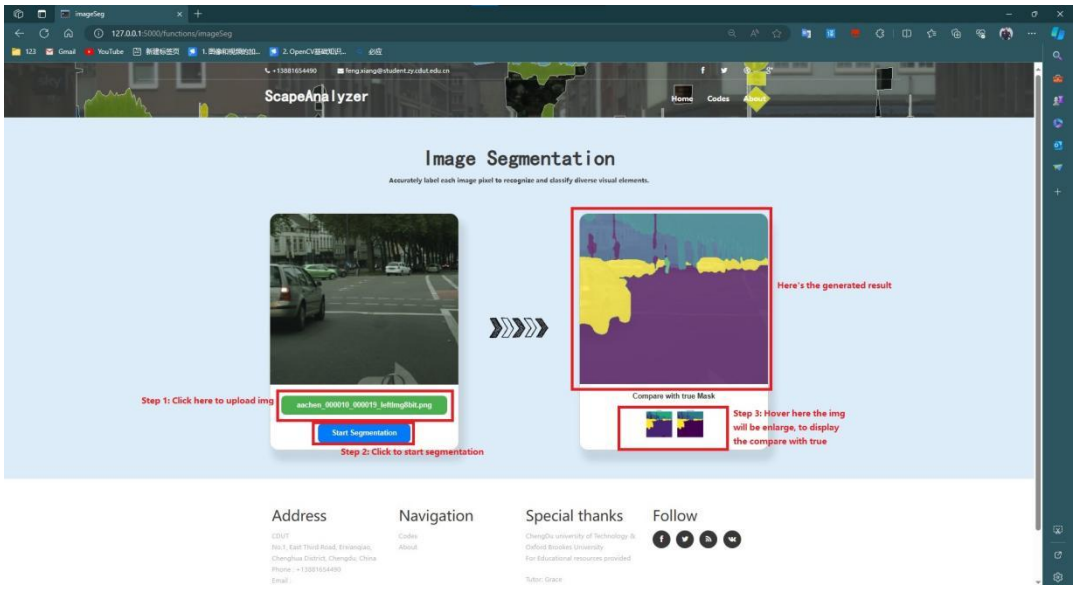


Figure 54. Image Segmentation

As shown in Figure 55, in the video segmentation section, users can upload video data and the system will analyse and segment the dynamic city scenes in the video frame by frame.

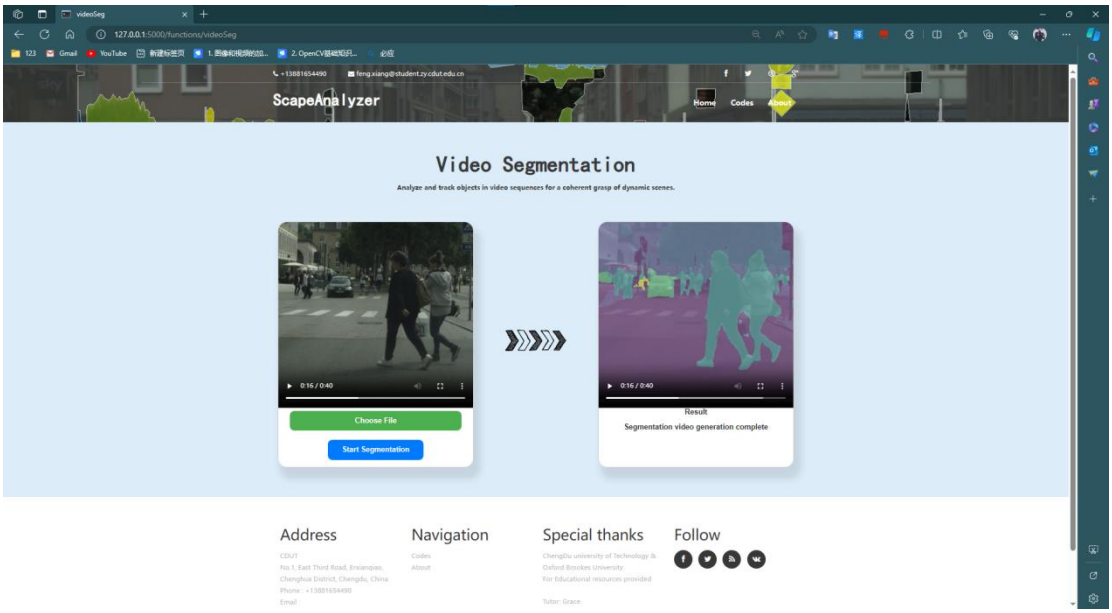


Figure 55. Video Segmentation

Figure 56 is the real-time segmentation function, which enables real-time semantic segmentation of video streams, allowing users to perform real-time semantic segmentation through the webcam.

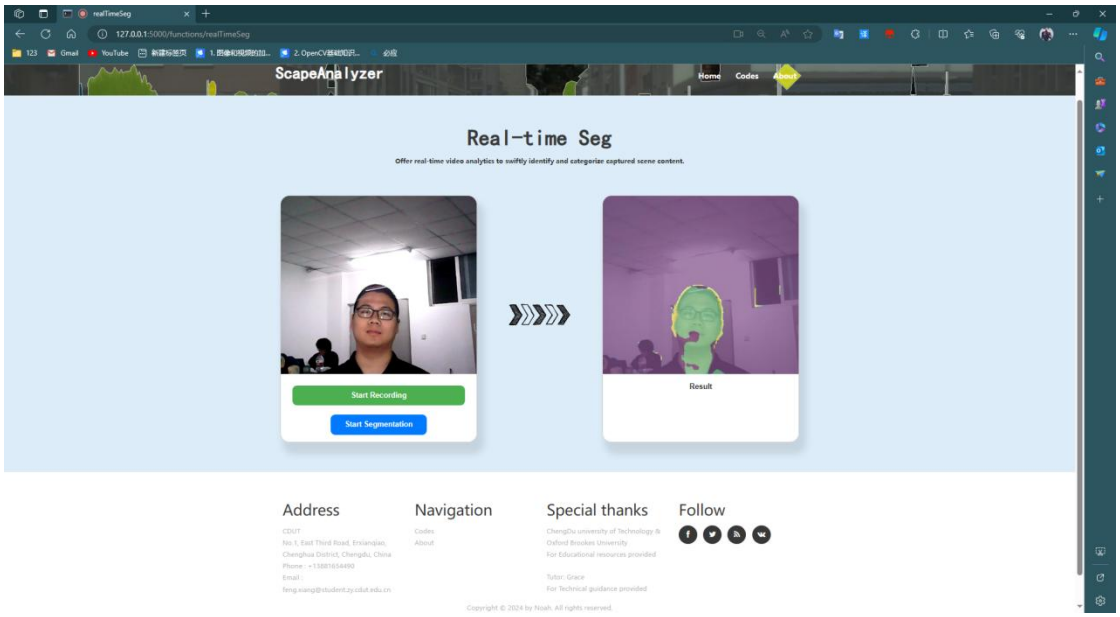


Figure 56. Real-Time Segmentation

Finally, as shown by Figure 57, the website also provides other scenario segmentation functions, which demonstrate the applicability and usefulness of the system.

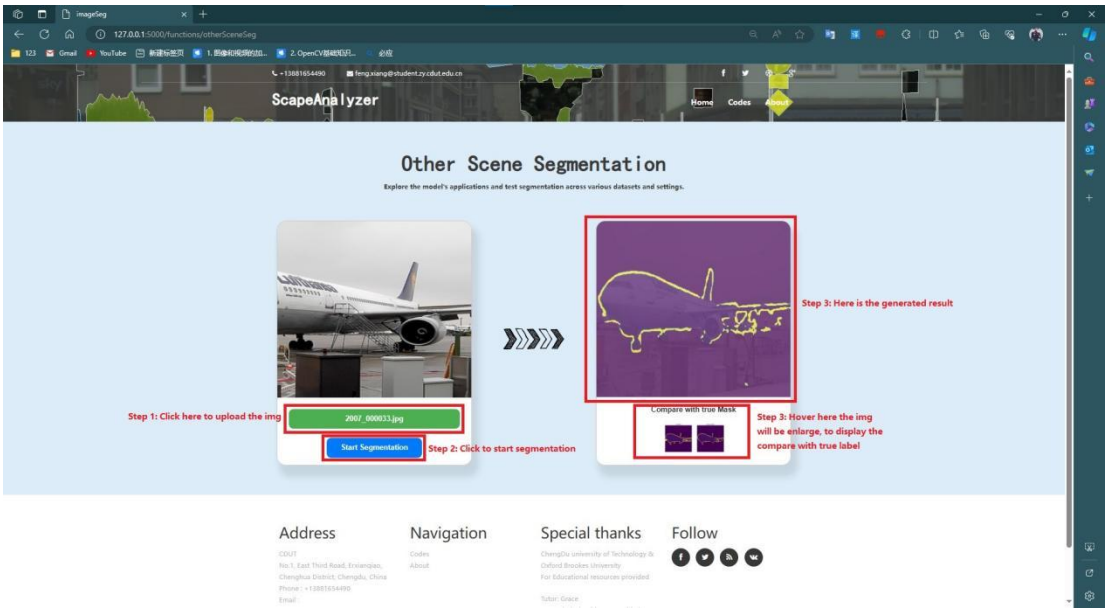


Figure 57. Other Scene Segmentation

Chapter 5 Professional Issues

5.1 Project Management

5.1.1 Activities

The activities and completion of the project are shown in Table 19.

Objectives	Activities	State
Ob1: Background Review	A1:Read current papers on semantic segmentation of urban datasets to understand the basics related to semantic segmentation. A2:Create a table to compare the performance of models appearing in different papers. A3:Analyse the applications of urban datasets and the challenges associated with them.	Completed
Ob2: Choose Dataset and split it.	A1:Download the Cityscapes dataset, this consists of two files, one is the original city image dataset and the other is the corresponding segmentation map dataset. A2:Cityscapes dataset has been divided into train, validation and test datasets. Complete the dataset fusion and re-divide it 7:1.5:1.5 A3:Generate a bar chart of the division of the dataset.	Completed
Ob3: Pre-processing the dataset	A1:The dataset categories were set using the official Cityscapes dataset script to generate datasets with 34, 19, 15, and 11 categories, respectively. A2:The dataset is normalized by randomizing the image saturation, hue, contrast, and randomly cropping the images to ensure that the corresponding segmentation maps are also randomly cropped. A2: Randomize the weather for some of the images in the dataset, including randomizing foggy, rainy and snowy days.	Completed
Ob4: Construct the model by using customer module	A1:Read the paper to understand the commonly used frameworks and modules of semantic segmentation models, and analyse the advantages and disadvantages of the relevant modules, and summarize the optimization scheme. A2:Keep trying to build a basic semantic segmentation model. Choose a simple backbone model to complete the basic feature extraction function, and input the results into the constructed semantic segmentation model.	Completed

	A3:Complete the fine-tuning of the SGD optimizer parameters and replace different loss functions, this includes a combination of loss functions using cross-entropy, Focal loss, Dice Loss and other loss functions. A4:Define metrics for model evaluation	
Ob5: Train and compare the result	A1: Input datasets with different number of categories into the model for training and get the optimal category dataset. A2: Replace the backbone network of the model and compare the outputs to evaluate the efficiency and performance of the model. A3:Compare common semantic segmentation models, including LinkNet as well as U-Net. A4:Compare the performance of the current model with existing semantic segmentation models.	Completed
Ob6: Design a GUI	A1:Design and draw the structure of the GUI. A2:Understand and identify the technology to be used to complete the GUI, Flask or PyQt. A3:Implement the code for the GUI.	Completed

Table 19. Activities and State of Completion

5.1.2 Schedule

Figure 58 shows the time planning Gantt chart, where the orange bars show the work completed. Table 20 shows the time planning schedule for the project. All objectives were completed on time or ahead of schedule.

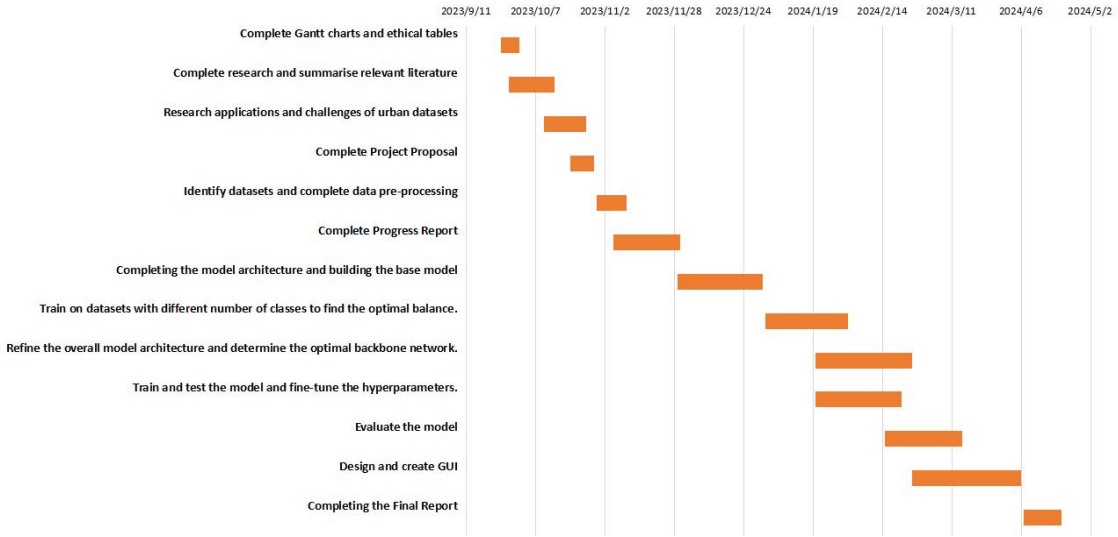


Figure 58. Time planning Gantt chart

Task	Start Date	End Date	Duration (days)
Complete Gantt charts and ethical tables	2023/9/24	2023/10/1	7
Complete research and summarize relevant literature	2023/9/27	2023/10/14	17
Research applications and challenges of urban datasets	2023/10/10	2023/10/26	16
Complete Project Proposal	2023/10/20	2023/10/29	9
Identify datasets and complete data pre-processing	2023/10/30	2023/11/10	11
Complete Progress Report	2023/11/5	2023/11/30	25
Completing the model architecture and building the base model	2023/11/29	2023/12/31	32
Train on datasets with different number of classes to find the optimal balance.	2024/1/1	2024/2/1	31
Refine the overall model architecture and determine the optimal backbone network.	2024/1/20	2024/2/25	36
Train and test the model and fine-tune the hyperparameters.	2024/1/20	2024/2/21	32
Evaluate the model	2024/2/15	2024/3/15	29
Design and create GUI	2024/2/25	2024/4/6	41
Completing the Final Report	2024/4/7	2024/4/21	14

Table 20. project timetable

5.1.3 Project Data Management

This project uses Github for project data management. The CODES folder stores the project code and the LITERATURE folder stores the literature used in the project. Also, the PROJECT_PIC folder stores all the result images of the project and the REPORT folder stores all the project reports. Finally, VIDEO holds the demo video of the project.

https://github.com/NOAHORFX/Project_Data

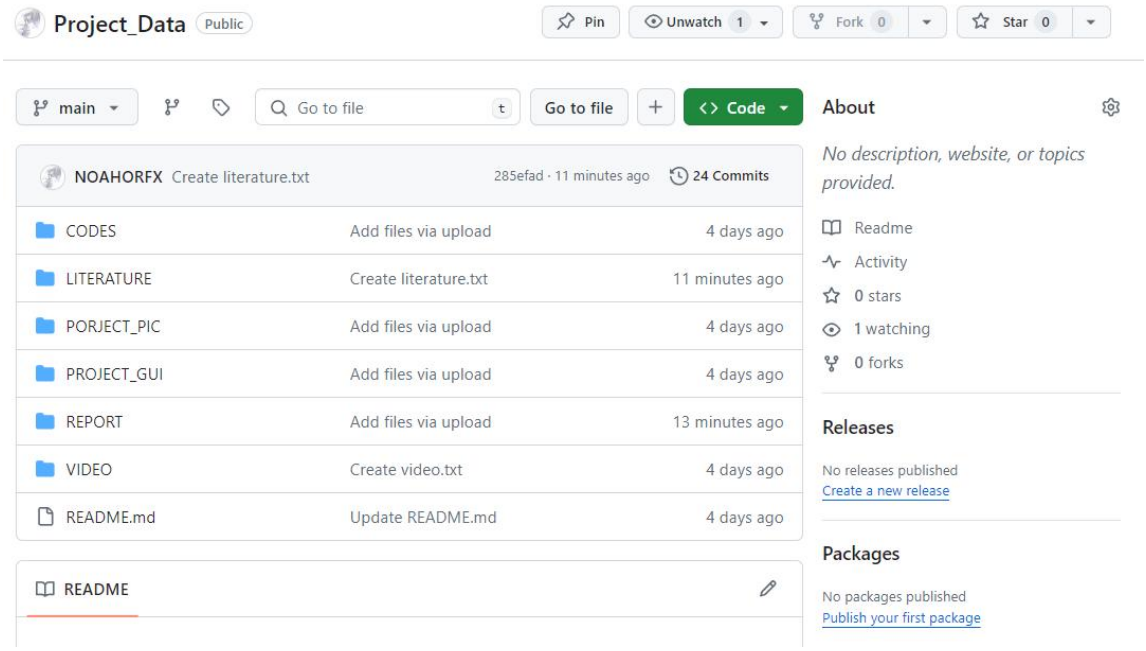


Figure 59. Structure of the data management folder

5.1.4 Project Deliverable

a. Project Proposal

b. Ethics Form

c. Weekly Report

d. Progress Report

e. Final Report

f. Project Codes

g. Project PPT

5.2 Risk Analysis

Table 21 shows the risks that have been dealt with and the solutions to the risks.Table 22 summarize the potential risks that may occur in the future and how they can be prevented.

State	Potential Risk	Potential Causes	Severity	Likelihood	Risk	Mitigation
Resolved	Insufficient processing capability for high-resolution images	Limited computing resources	4	2	8	Rent remote GPU resources
	Category imbalance issue	Low occurrence rate of certain categories in the dataset	3	3	9	Adjust category weights in the loss function; Data augmentation
	Inadequate edge detection	Limited model capability for detailed processing	3	2	6	Incorporate advanced edge detection algorithms; Optimize with post-processing techniques

Table 21. Resolved Risk

State	Potential Risk	Potential Causes	Severity	Likelihood	Risk	Mitigation
Future	Computational demands exceed mobile device capabilitie	High memory requirements when loading the model	3	3	9	Design a detailed test plan.

	s					
	Lack of model generalization	**Landscape style differences among cities in different countries**	**3**	**4**	**12**	**Implement version control strategy at start.**
	Segmentation capability in dynamic scenes	**Insufficient model understanding of dynamic scenes**	**4**	**3**	**12**	**Integrate mechanisms like LSTM to improve processing of dynamic scenes**

Table 22. Potential future risks

5.3 Professional Issues

5.3.1 Legal Issues

The legal issues need to focus on the legal use of the data and ensure that user privacy is protected **[29]**. For example, when using publicly available cityscape datasets, even if it is legally obtained, this project need to check carefully whether relevant information that identifies individuals, such as face features, licence plate numbers, etc., is inadvertently captured in these images, which may violate privacy-protecting regulations such as the GDPR **[30]**. Therefore, it might be necessary to introduce technical methods for data desensitization, such as fuzzing of personally identifiable information.

5.3.2 Social Issues

In social issues, need to ensure that the model has a wide range of recognition abilities for urban scenes from different regions and cultural backgrounds. For example, the urban landscape of a particular country is very different from another country, and if the training dataset is too much biased towards a particular region, the model may not be able to accurately recognize urban landscapes from other regions. Therefore, it is important to introduce diverse datasets in the data collection and model training stage to enhance the model's generalization and fairness.

5.3.3 Ethical Issues

Considering ethical issues, the goal of the project is to promote the maximum benefit to society. For example, by improving urban traffic management and increasing the safety of self-driving vehicles, traffic accidents will be reduced and road use will become more efficient. In the process, it is important to ensure that the application of the technology does not increase social inequality, for example, avoiding the deployment of advanced self-driving technology only in affluent areas while ignoring the needs of lower-income areas **[31]**.

5.3.4 Environmental Issues

Environmental issues are also worth considering. While the project itself, in digital form, seems to have little environmental impact, but the large-scale data processing and model training behind it actually consumes a lot of computational resources, which in turn increases energy demand **[32]**. For example, training a state-of-the-art deep learning model may require hundreds of hours of running GPU, resulting in huge carbon emissions **[32]**. To deal with this problem, it is possible to explore more efficient model architectures that reduce the need for computational resources or use green energy to power data centre operations.

Chapter 6 Conclusion

In this research, a high-performance semantic segmentation framework is designed to solve the challenges that today's semantic segmentation techniques encounter when reprocessing complex scene structures, multi-scale object recognition, and diverse environmental conditions. The model construction is achieved by combining a variety of advanced deep learning techniques, including sampling deep separable convolution-based ASPP module, Transformers module, edge feature extraction module, and multi-scale pooling. By training and evaluating on the Cityscapes dataset, the model achieves 93% accuracy as well as 0.76 MIOU, showing good accuracy, robustness and speed. In addition to this, this project completed several comparison experiments, including training the model on datasets with different number of categories, comparing it with common semantic segmentation models, comparing it using different backbone networks and comparing it with the model proposed in the paper. This demonstrates the significant advantages of the semantic segmentation framework proposed in this project in terms of accuracy, robustness and efficiency as well as its wide potential for practical applications.

However, the current project has some limitations. Firstly, limited by training resources, the project had to adopt strategies such as image cropping to adapt to model training, a practice that may lead to the loss of important global feature information and affect the model's understanding of complex scenes. In addition, the inherent category imbalance in the dataset also poses difficulties for model training, especially for the less frequently occurring categories, such as bicycles and street signs. To address these problems, this study attempts to mitigate them by adjusting the weights of the categories in the loss function, and although this strategy fails to completely solve the problem, it presents a valuable reference for subsequent research. In terms of image edge detection, although the model integrates the convolution operation for edge detection, it is still insufficient in dealing with fine features, and try to compensate for this by post-processing techniques such as morphological operations, but this instead exposes the model's limitations in dealing with fine features, as well as adding additional computational costs.

Despite these limitations, proposed model performed well on the validation dataset, increasing computational speed by effectively reducing the number of parameters while ensuring high accuracy. This demonstrates that even under resource-constrained conditions, efficient and accurate segmentation of complex urban scenes can be achieved through proper application and optimization of deep learning techniques.

For the future work of this project, this project will focus on further improving the model's computational efficiency, generalization ability and adaptability to dynamic environments. And plan to explore more lightweight model frameworks and apply techniques such as model pruning and knowledge distillation to improve the inference speed of the models. For the segmentation problem of dynamic scenes, the project will try to integrate temporal processing mechanisms such as LSTM into the network so that the model can better capture and understand the temporal relationships in dynamic scenes. Through these efforts, this research will rise to new heights and contribute to the development of the field of self-driving. Overall, this project provides some technical insights into the challenges of semantic segmentation in self-driving and points out the direction for future exploration, demonstrating the potential of deep learning applications in the development of traffic-only mentality.

References

[1] Duarte, F. & Ratti, C., 2018. The Impact of Autonomous Vehicles on Cities: A Review. Journal of Urban Technology, 25(4), pp.3-18. Available at: https://doi.org/10.1080/10630732.2018.1493883

[2] Janai, J. et al. (2020) 'Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art', Foundations and Trends® in Computer Graphics and Vision, 12(1–3), pp. 1–308. Available at: https://doi.org/10.1561/0600000079.

[3] H. Hu, H. Cai, Z. Ma, and W. Wang, "Semantic segmentation based on semantic edge optimization," in Proc. 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), Changchun, China, 2021, pp. 612-615, doi: 10.1109/EIECS53707.2021.9587939.

[4] Hao, S., Zhou, Y. and Guo, Y. (2020) 'A Brief Survey on Semantic Segmentation with Deep Learning', Neurocomputing, 406, pp. 302–321. Available at: https://doi.org/10.1016/j.neucom.2019.11.118.

[5] Li, J. et al. (2021) ' Lane-DeepLab: Lane semantic segmentation in automatic driving scenarios for high-definition maps ', Neurocomputing, 465, pp. 15 – 25. Available at: https://doi.org/10.1016/j.neucom.2021.08.105.

[6] Y. Guo and B. Yang, "A Survey of Semantic Segmentation Methods in Traffic Scenarios" in Proc. 2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM), Xiamen, China, 2022, pp. 452-457, doi: 10.1109/MLCCIM55934.2022.00083.

[7] K. Singh, V. Varshney, and M. Rajl, "Image Segmentation Role in Self Driving Car," in Proc. 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 861-864, doi: 10.1109/ICAC3N56670.2022.10074404.

[8] Martínez-Díaz, M. and Soriguera, F. (2018) ' Autonomous vehicles: theoretical and practical challenges', XIII Conference on Transport Engineering, CIT2018, 33, pp. 275 – 282. Available at: https://doi.org/10.1016/j.trpro.2018.10.103.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, 2017, doi: 10.1109/TPAMI.2016.2644615.

[10] S. Abdigapporov, S. Miraliev, V. Kakani, and H. Kim, “Joint Multiclass Object Detection and Semantic Segmentation for Autonomous Driving,” IEEE Access, vol. 11, pp. 37637–37649, 2023, doi: 10.1109/ACCESS.2023.3266284.

[11] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation,” CoRR, vol. abs/1606.02147, 2016. Available: http://arxiv.org/abs/1606.02147

[12] R. P. K. Poudel, S. Liwicki, and R. Cipolla, “Fast-SCNN: Fast Semantic Segmentation Network,” CoRR, vol. abs/1902.04502, 2019. Available: http://arxiv.org/abs/1902.04502

[13] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation,” CoRR, vol. abs/1808.00897, 2018. Available: http://arxiv.org/abs/1808.00897

[14] D. Fourure, R. Emonet, É. Fromont, D. Muselet, A. Trémeau, and C. Wolf, “Residual Conv-Deconv Grid Network for Semantic Segmentation,” CoRR, vol. abs/1707.07958, 2017, [Online]. Available: http://arxiv.org/abs/1707.07958

[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834–848, 2018, doi: 10.1109/TPAMI.2017.2699184.

[16] G. Li, L. Li, and J. Zhang, “BiAttnNet: Bilateral Attention for Improving Real-Time Semantic Segmentation,” IEEE Signal Processing Letters, vol. 29, pp. 46–50, 2022, doi: 10.1109/LSP.2021.3124186.

[17] G. Lin, A. Milan, C. Shen, and I. D. Reid, “RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation,” CoRR, vol. abs/1611.06612, 2016, [Online]. Available: http://arxiv.org/abs/1611.06612

[18] C. C. Aggarwal, Neural Networks and Deep Learning: A Textbook. Cham, Switzerland: Springer International Publishing AG, part of Springer Nature, 2018. doi: 10.1007/978-3-319-94463-0.

[19] 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 Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834-848, Apr. 1, 2018, doi: 10.1109/TPAMI.2017.2699184.

[20] X. Wang, L. Yan, and Q. Zhang, "Research on the Application of Gradient Descent Algorithm in Machine Learning," in Proc. 2021 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 2021, pp. 11-15, doi: 10.1109/ICCNEA53019.2021.00014.

[21] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in Proc. ICML 2023. [Online]. Available: https://arxiv.org/pdf/2304.07288.pdf

[22] S. Kato and K. Hotta, “Adaptive t-vMF dice loss: An effective expansion of dice loss for medical image segmentation,” Computers in Biology and Medicine, vol. 168, p. 107695, 2024, doi: https://doi.org/10.1016/j.compbiomed.2023.107695.

[23] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," Neurocomputing, vol. 338, pp. 321–348, 2019. Available: https://doi.org/10.1016/j.neucom.2019.02.003.

[24] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation," CoRR, vol. abs/1707.03718, 2017. [Online]. Available: http://arxiv.org/abs/1707.03718

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," CoRR, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," CoRR, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[27] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," in 2016 International Conference on Learning Representations (ICLR), 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1511.07122

[28] C.-F. (Richard) Chen, Q. Fan, and R. Panda, "CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 357-366.

[29] "The Act | The Artificial Intelligence Act." https://artificialintelligenceact.eu/the-act/ (accessed Mar. 7, 2024).

[30] "General Data Protection Regulation (GDPR) – Official Legal Text," gdpr-info.eu. [Online]. Available: https://gdpr-info.eu/. [Accessed: Mar. 18, 2024].

[31] A. Das, Y. Xian, Y. He, Z. Akata, and B. Schiele, "Urban Scene Semantic Segmentation with Low-Cost Coarse Annotation," 2022. [Online]. Available: arXiv:2212.07911 [cs.CV]

[32] Physics World. (2021, November 10). The huge carbon footprint of large-scale computing. Retrieved from https://physicsworld.com/a/the-huge-carbon-footprint-of-large-scale-computing/

[33] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, 2010.