UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Research and Development

BACHELOR THESIS

By

Nguyen Anh Quan – BI12-365

Data Science

Title:

ROAD LANE MARKINGS SEGMENTATION

Supervisor: Dr. DOAN Nhat Quang – ICT Lab

Hanoi, 2024

TABLE OF CONTENTS

ACKNOWLEDGEMENT

This project is my contribution to a research topic of the USTH ICT Lab.

“First and foremost, I would like to extend my sincere gratitude to the faculty and staff of the University of Science and Technology of Hanoi, particularly those from the ICT Department and the Data Science Major. Your dedication to equipping invaluable specialized knowledge, alongside lessons in ethics and life skills, has been vital in shaping us into skilled, capable, and well-rounded individuals. Learning and working with you in the past years has played a very important role in my personal development, career, and the progress of this thesis.

Secondly, I would like to send my deepest appreciation to my supervisor, Dr. Doan Nhat Quang, for his invaluable guidance and support throughout the completion of this thesis. I feel grateful for the time I had working with him during the internship, and for the knowledge, experience, and encouragement he gave me during the difficult times. It is an honor for me to be his intern in the USTH ICT Lab and I am extremely proud of that.

To my beloved family, thank you for the endless love and support you have given me, your encouragement and trust are the most valuable strength and motivation to help me keep trying on this important journey.

Many thanks to all those whose contributions to my achievements may not have been mentioned. Your support has been irreplaceable, and I am immensely thankful beyond any expression.

And finally, to those who are willing to spend your time reading my work, thank you and I hope you will have a wonderful time as I had when writing it.”

LIST OF ABBREVIATIONS

ADAS	Advanced Driver Assistance Systems
BiSeNet	Bilateral Segmentation Network
CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
CP	Context Path
SP	Spatial Path
ARM	Attention Refinement Module
FFM	Feature Fusion Module
JPG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
XML	Extensible Markup Language
PNG	Portable Network Graphic
mIoU	Mean Intersection over Union
ICT	Information and Communication Technology
USTH	University of Science and Technology of Hanoi

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Road lane marking segmentation is a critical component in the development of Advanced Driver Assistance Systems (ADAS), autonomous driving technologies, and road surface repair and maintenance. This study focuses on the segmentation of road lane markings from video and image data to enhance the accuracy and reliability of the lane detection system. By using and comparing deep learning techniques, particularly Convolutional Neural Network (CNN) models, this research utilizes semantic segmentation approaches to efficiently identify and segment several important lane markings under various conditions of the environment namely the lighting, weather, and road textures from a large dataset collected in different road situation from vehicles dash cam. This project focuses on implementing a complete workflow of a deep learning problem, especially computer vision, from collecting more data and enriching the dataset to preprocessing the images, and training the models, to achieve a modern method of segmentation of road markings used in road safety and autonomous vehicles.

Keywords: Road Markings, Autonomous Driving, Road Safety, Visibility Enhancement, Image Processing, Deep Learning, Computer Vision, Semantic Segmentation, BiSeNet, U-Net.

I. INTRODUCTION

1. Context and Motivation

As population growth and construction activities increase, the demand for and density of transportation also rise. This development assists the growth of a country's transport infrastructure, however, it poses certain problems among which are accidents and carnages that lead to damage of vehicles, not to mention the many fatalities that occur in such mishaps. As a response to these problems, scientists and engineers working in their respective fields have created Advanced Driver Assistance Systems (ADAS), designed to minimize the number of accidents and to help drivers improve their road use in a safer manner [1]. Furthermore, what has become specific in the past several years, large companies that are focused on technology, have shown great progress in transport technology, the example of which can be called autonomous vehicles – the success of modern engineering.

These breakthroughs have been largely driven by the application of Deep Learning, particularly Computer Vision, in embedded devices that enhance road safety by automatically identifying dangerousness while traveling and improving visibility under varying weather, lighting, and road conditions. One of the important elements of road infrastructure is lane markings, which include travel directions, special lanes, and pedestrian crossings. Misunderstanding or lack of visibility of these markings can lead to dangerous situations, including traffic violations or more severe consequences affecting other road users or pedestrians. Moreover, the blur or disappearance of these markings affects the driving experience heavily, and manual inspection is often time-consuming and labor-intensive. These disadvantages also happen when drawing street maps including road lane markings manually. Therefore, collecting image data through dash cams and automatical animating while driving offers a more efficient alternative for monitoring road markings, and city map drawing.

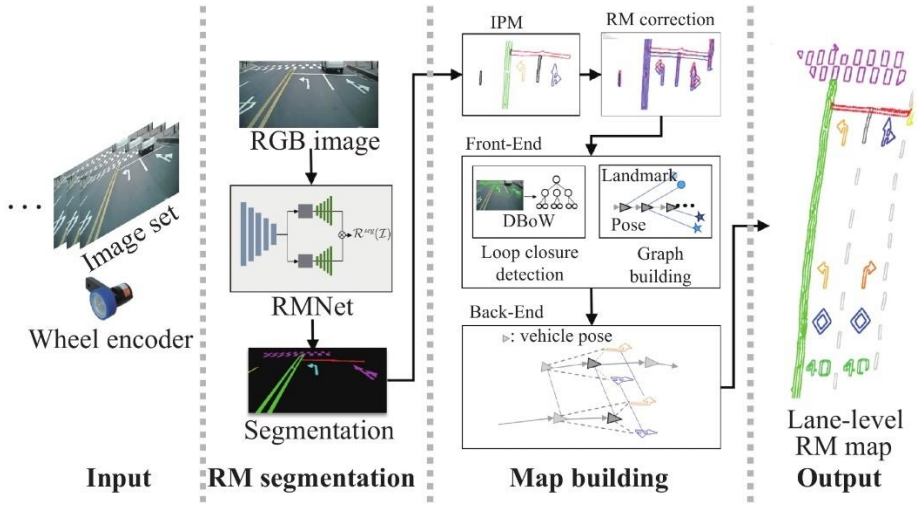

Figure 1: Example of Road Markings Map Rendering Using Image Semantic Segmentation[5]

As the recognition of these markings as well as their clear display is necessary for the correct understanding of the road by drivers and the steering of self-driving vehicles, this study examined the potential of Convolutional Neural Network (CNN) models to detect and highlight the road marking using a variety of Image Segmentation techniques, so that drivers and autonomous systems can enhance their abilities to understand and navigate roadways.

The study will simulate a complete process of a project using Deep Learning: we will search for the dataset and collect, and label additional external data. Next, enrich the dataset with augmentation methods and then preprocess it with image processing methods. Then the research utilizes a method of Image Segmentation: Semantic Segmentation, to accurately detect and differentiate road lane markings under various environmental conditions. By employing models of Semantic Segmentation that classify each pixel in the image, providing an overall view of lane markings, the research evaluated the strengths and limitations of the solutions in terms of scores and performance, with the goal of enhancing the reliability of lane detection systems in and autonomous vehicles. After that, a complete conclusion was drawn to close the project.

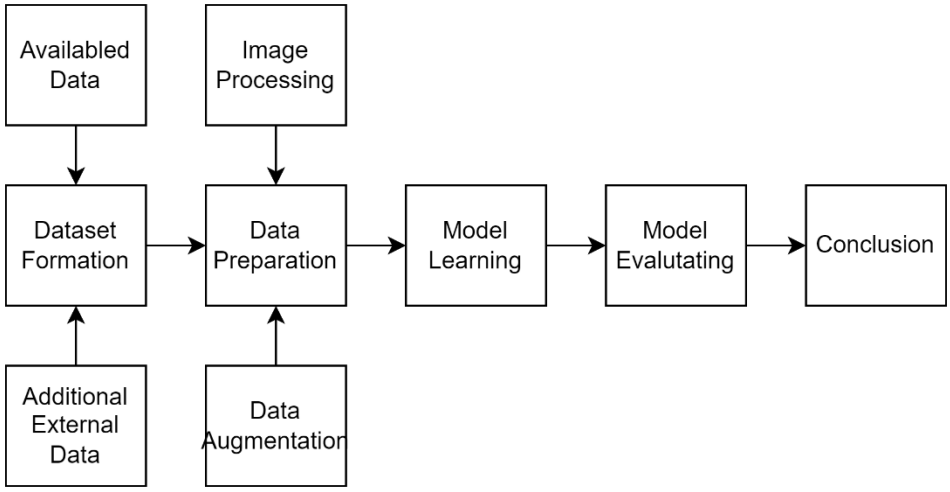

Figure 2 Workflow of the Process utilized in the Project

This thesis is motivated by the increasing utilization of dashcams in daily driving and the need for advanced technologies to enhance road safety. Despite the widespread usage of those camcorders, the use of segmentation technology research in vehicles especially for road lane marking is still relatively limited in Vietnam. Since there are so many instances of vehicular accidents that result in loss and damage to families' assets, it is high time to find solutions. We conducted this research to attain that technology and promote the development of this technology in our country.

2. Objectives

The project aims to achieve the following objectives: First, we will collect, label, enrich, and provide a comprehensive dataset to enhance the training efficiency of segmentation models. The next step of the study is to train and compare segmentation models to fulfill the particular requirements of the given problem suitably. After that, we will describe a contemporary approach that employs the Deep Learning technique, specifically CNN models that aim at enhancing traffic safety and developing new technologies to help people build safer roads for consumers and automobiles.

3. Contribution

Our contribution to this project is to collect a dataset used for Image Segmentation combined with collecting and labeling data collected on the street outside and using methods to create a complete dataset used for training. Next, we trained and applied transfer learning to models of Semantic Segmentation such as U-Net, and BiSeNet with pre-trained backbones and then compared them to propose a modern solution to this problem.

4. Related Works

This problem has been extensively studied by many authors with various approaches and datasets. A summary of those related works is presented in Table 1.

Table 1 List of the Previous Approaches of Road Lane Markings Segmentation

Year	Authors	Method	Name	Dataset
2022	Oshada Jayasinghe, Sahan Hemachandra, et al.	SSD-MobileNet-v1, SSD-Inception-v2, Mask-RCNN-Inception-v2, Mask-RCNN-ResNet50	CeyMo: See More on Roads - A Novel Benchmark Dataset for Road Marking Detection	CeyMo
2022	Junjie Wu, Wen Liu, Yoshihisa Maruyama	MSA-DCNN	Automated Road-Marking Segmentation via a Multiscale Attention-Based Dilated Convolutional Neural Network Using the Road Marking Dataset	RMD
2017	Tom Bruls, Will Maddern, Akshay A. Morye, and Paul Newman	U-Net	Mark Yourself: Road Marking Segmentation via Weakly-Supervised Annotations from Multimodal Data	Oxford RobotCar dataset, CamVid dataset
2022	W. Jang, J. Hyun, J. An, M. Cho, and E. Kim	RMNet	A Lane-Level Road Marking Map Using a Monocular Camera	Semantic Road Mark Mapping (SeRM) Dataset

First, it is essential to highlight the work of **Oshada Jayasinghe, Sahan Hemachandra, et al.** [3], who introduced the **CeyMo** dataset for road lane markings segmentation. High-resolution input images (1920×1080) containing various types of traffic, lighting, and weather conditions are provided in the dataset. The authors tried two object detection models (SSD-MobileNet-v1, SSD-Inception-v2) and two instance segmentation models (Mask-RCNN-Inception-v2, Mask-RCNN-ResNet50). These results were obtained among those models, and in particular, Mask-RCNN-ResNet50 was the model with the maximum Macro F1 Score of 88.33 while Mask-RCNN-Inception-v2, SSD-Inception-v2, and SSD-MobileNet-v1 have Macro F1 Scores of 85.75, 82.88, and 80.93 respectively. . The CeyMo dataset also provides both input images and pixel-wise segmentation masks, making it an essential resource for semantic segmentation tasks and inspiring the goals of this project.

Next, in 2022, **Junjie Wu, Wen Liu, and Yoshihisa Maruyama** [2] introduced the MSA-DCNN, a novel multiscale attention-based dilated convolutional neural network for automated road-marking segmentation. The research utilized a dataset called (RMD) with an image size of 1920 x 1080 and their method combines multiscale attention and dilated convolution to capture spatial-context information effectively. The proposed model achieved a mIoU of 0.7488 which is an impressive result in the field of semantic segmentation.

In 2017, **Tom Bruls, Will Maddern, Akshay A. Morye, and Paul Newman** [4] introduced a weakly-supervised system for real-time road marking segmentation using images from a monocular camera. By leveraging multimodal sensor data, they generated large quantities of annotated images with minimal manual labeling, which were used to train a U-Net model. Their approach achieved the IoU score of 0.612 using the Oxford RobotCar dataset and the CamVid dataset.

Lastly, **W. Jang, J. Hyun, J. An, M. Cho, and E. Kim** [5] developed a complete system for automatic road marking map generation using lane marking segmentation. They used the SeRM (Semantic Road Mark Mapping) dataset, consisting of pixel-accurate annotated street scene images with a resolution of 672 × 1280. Their custom-built model, RMNet,

achieved a mIoU score of 0.6526, demonstrating the potential of their method for lane-level road marking segmentation and map generation.

5. Thesis Outline

This section provides an overview of the structure and content of the thesis. Following the introduction and problem statement discussed in **Section I - Introduction**, the thesis is organized as follows:

Section II - Data Acquisition and Understanding: This section introduces the dataset used in the research, detailing the collection methods and important features of the dataset. It also covers information from external datasets which when added to the study.

Section III - Methodologies: In this section, we present the complete methodology employed in the research. This includes data augmentation and filling techniques, preprocessing methods, the models used for training, and the evaluation criteria applied to assess model performance.

Section IV - Experiments: This section provides a comprehensive analysis of the experimental setup, results and evaluates the performance of the models both quantitatively and qualitatively. Results are presented to compare the effectiveness of the methods used.

Section V - Conclusion: The work is concluded with a summary of the main results and an evaluation of the approaches. Additionally, difficulties, suggestions, and potential directions for future improvements in addressing the problem are discussed.

II. DATA ACQUISITION AND UNDERSTANDING

This Section will depict an overall view of the whole dataset used in this research, the additional data, and the way of collecting and labeling.

1. CeyMo Dataset

To achieve the solution of the project, a dataset called CeyMo: See More on Roads was selected (which was published in 2022) [3]. This dataset was created to provide better options for the road lane markings segmentation problem and address the limitations found in existing publicly previous datasets. CeyMo seeks to overcome limitations such as the lack of challenging road scenes, insufficient prominence given to lane markings, the absence of evaluation scripts, annotation formats, and lower resolutions. Therefore, the images of this dataset can help the model learn more diversely.

Figure 3 Example of the dataset with polygon class annotation

The dataset contains 2887 total images of road scenarios in Sri Lanka of 1920 x 1080 resolution with 4706 road markings belonging to 11 classes. The author split the dataset into 2099 images for training and 788 for validating. The conditions of the scenes have been divided into six categories: normal, crowded, dazzling light, night, rain, and shadow.

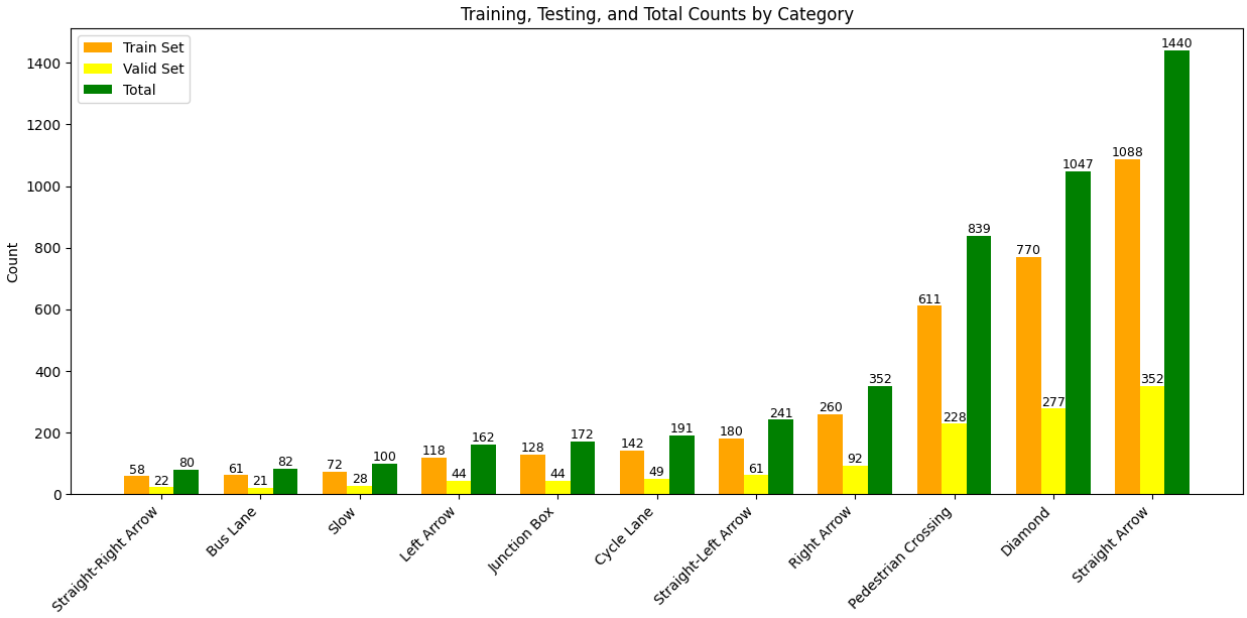

Figure 4 Distribution of Training, Validation, and Total Counts by Road Marking Category

The bar chart of Figure 4 visualizes the distribution of different road lane markings across the training set, test set, and the total dataset. A total of 4706 labels are captured inside the whole dataset. The figure also shows the dominance of the classes like 'Straight Arrow', 'Pedestrian Crossing', and 'Diamond', and the less frequently represented classes are "Bus Lane" and "Slow". In this distribution, we can clearly see how well each class is represented within the dataset of each class of road lane markings.

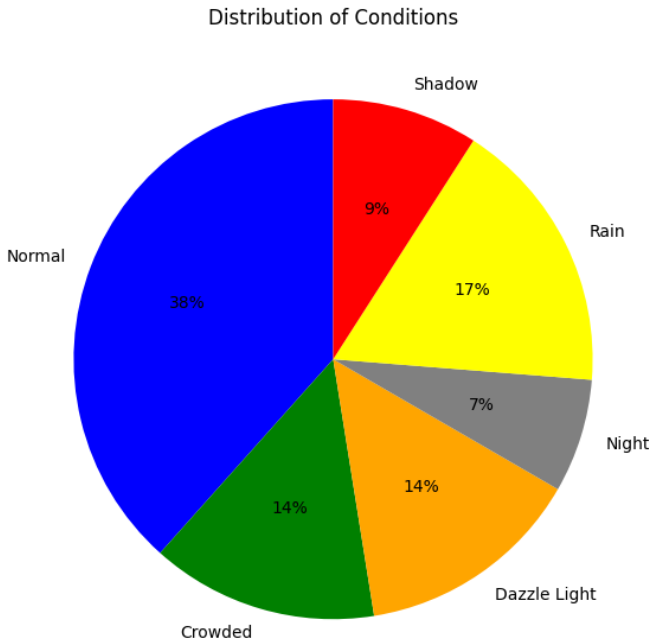

Figure 5 Distribution of Scene Conditions in the dataset

The pie chart of Figure 5 illustrates the distribution of the dataset across different environmental conditions. The largest portion of the dataset was collected under "Normal" conditions (38%), followed by "Rain" (17%), and "Night" (14%). Other conditions, such as "Crowded," "Dazzle Light," and "Shadow," take a smaller amount of the dataset. This distribution shows the variety of scenarios captured in the dataset, ensuring that the trained models can generalize well across different real-life conditions.

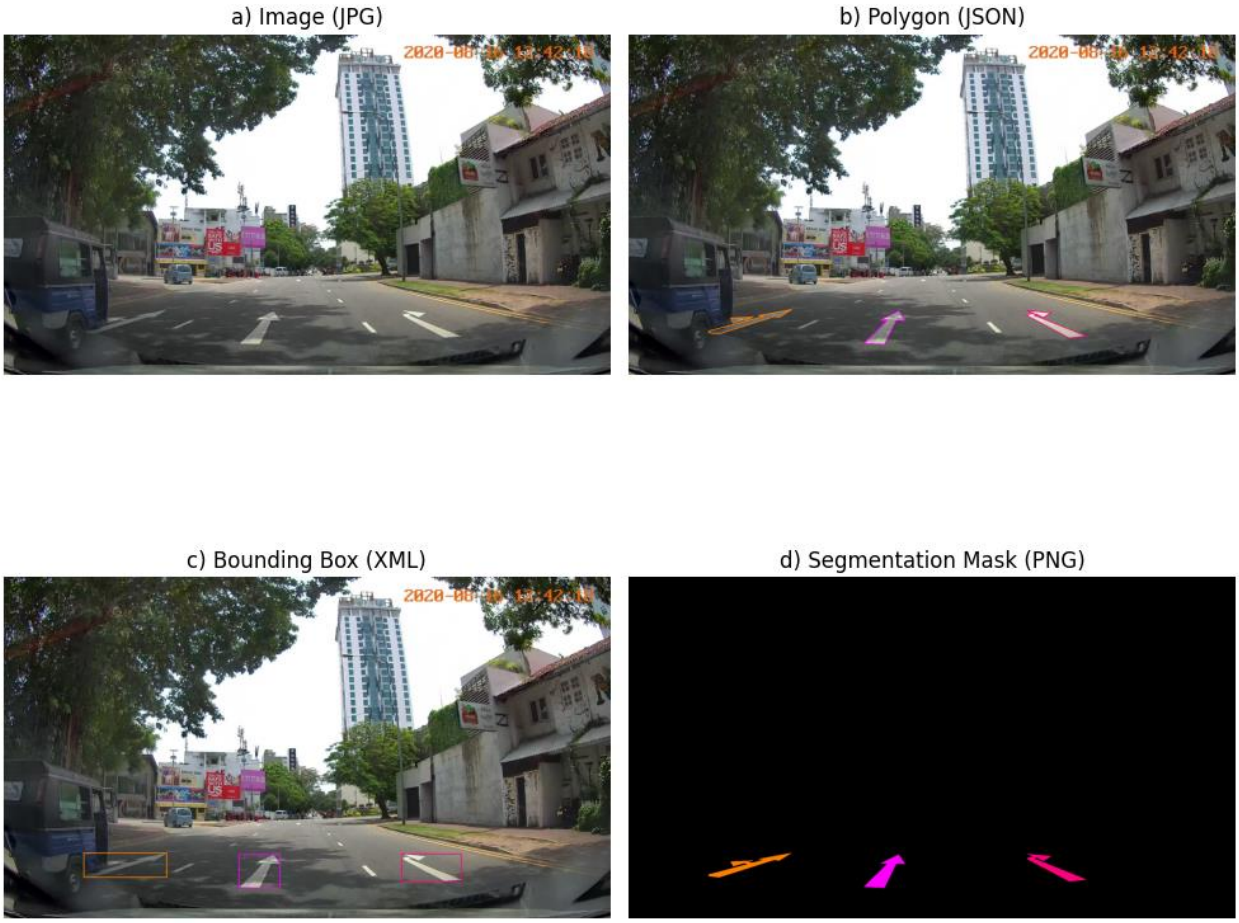

Figure 6 Visualization of dataset image, annotations with file format:

a) Original Image as JPG files, b) Polygon annotations in JSON files, c) Bounding Box annotations in XML files, d) Segmentation Masks in PNG files

The dataset also provides annotations in three formats: polygons, bounding boxes, and pixel-level segmentation masks. The polygon annotations in JSON format are considered as the ground truth and bounding box annotations in XML format and segmentation masks in PNG format are provided as additional annotations. The images are captured using a dash cam of the author's vehicles. In this project, we primarily choose the PNG segmentation masks as those annotations are the required output of semantic segmentation models and use some techniques to encode the mask images into a digital format that fits the model.

Table 2 Table of Class Definitions and RGB Color Label

Road Marking Class	Color Code	Class Description
Bus Lane (BL)	(0,255,255)	Indicates lanes specifically designated for bus traffic, usually marked to prioritize public transport.
Cycle Lane (CL)	(0,128,255)	Designates lanes reserved for cyclists, providing a safe space for bicycle traffic on the road.
Diamond (DM)	(178,102,255)	A reminder that drivers are about to encounter a Pedestrian Cross
Junction Box (JB)	(255,255,51)	Marked areas at intersections to prevent vehicles from blocking the junction and maintain traffic flow.
Left Arrow (LA)	(255,102,178)	Indicates lanes where vehicles are required or permitted to turn left.
Pedestrian Crossing (PC)	(255,255,0)	Designates areas for pedestrians to safely cross the road, often marked with zebra stripes or similar.
Right Arrow (RA)	(255,0,127)	Indicates lanes where vehicles are required or permitted to turn right.
Straight Arrow (SA)	(255,0,255)	Indicates lanes where vehicles are required or permitted to continue straight.
Slow (SL)	(0,255,0)	Warns drivers to reduce speed, typically found near pedestrian crossings or sharp turns.
Straight-Left Arrow (SLA)	(255,128,0)	Indicates lanes where vehicles are permitted to either continue straight or turn left.
Straight-Right Arrow (SRA)	(255,0,0)	Indicates lanes where vehicles are permitted to either continue straight or turn right.

As mentioned above, the dataset consists of 11 different classes which are the specific lane markings that are important in the road. Each class in the mask and the polygon annotations is assigned a certain color in RGB format for better visualization and to facilitate the training process.

2. Additional External Dataset

Given the relatively small size of the CeyMo dataset and the fact that some road markings in CeyMo closely resemble those commonly seen on Vietnamese streets (such as pedestrian crossing marks, arrows, and diamond boxes), we recognized the need to supplement our data. To address this, we collected and manually labeled an additional 482 images. These images were extracted from six videos, each approximately five seconds in duration, recorded during daily commutes on the streets of Hanoi.

These images were taken at a resolution of 1920x1080 using an iPhone 8 Plus camera while on motorcycle travel to offer a complete picture of road surfaces to reality. This additional dataset is intended to highlight the specific road markings and environmental conditions encountered in Vietnam that may not be present in the first CeyMo dataset.

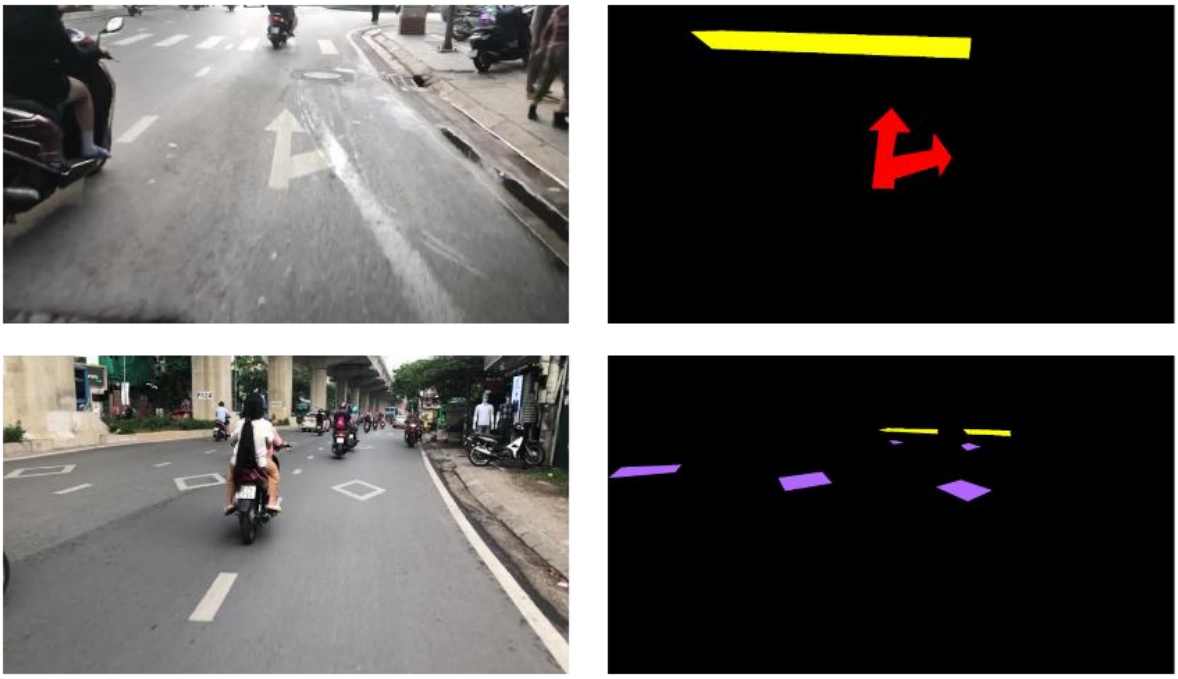

Figure 7 Examples of Additional External Data with Self-Annotated Masks

To ensure the quality of the annotations, we employed professional annotation tools such as Labelme and CVAT. With these tools, we were able to precisely label each image multiple times, ensuring that the set accurately represents the road markings that are relevant to our research. This external dataset is included to increase the diversity of the training data and thus improve segmentation model performance and generalizability.

III. METHODOLOGIES

This section covers all the scientific methods, including the theoretical background model architectures and evaluation.

1. Theoretical Background

1.1. Machine Learning and Deep Learning

Machine learning (ML) is a subset of artificial intelligence that allows machines to learn from data and make predictions or decisions based on patterns identified during training. In ML, models are trained to recognize patterns by processing data and using these patterns to make informed predictions or decisions [11]. The traditional approach to machine learning, as shown in the first illustration of Figure 8, involves a step-by-step pipeline: from input data, pre-processing, and feature extraction, to applying a classification or prediction model that generates an output.

However, traditional machine learning methods rely heavily on manual feature engineering, where users must select the features that the model uses. This limitation can be handled by using deep learning.

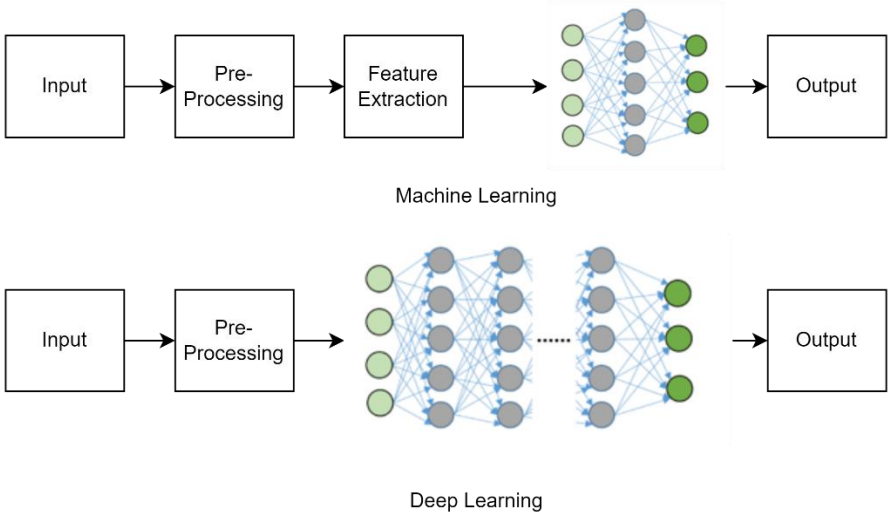

Figure 8 Comparison of Machine Learning and Deep Learning workflow

Deep learning (DL), a more advanced subset of machine learning, eliminates the need for manual feature extraction by automatically learning hierarchical features from the raw input data [6]. The second illustration visualizes a deep learning model, where a neural network with multiple layers or more than three layers (often called a deep neural network) learns these features through multiple transformations. Each layer learns to represent more complex patterns, ultimately generating a highly accurate prediction. This automatic feature extraction process, depicted in the Deep Learning workflow image of Figure 8, allows DL models to perform tasks in the fields of computer vision, natural language processing, etc.

1.2. Convolutional Neural Network (CNN)

In this study, the data here are images, and the networks used are Convolutional Neural Network (CNN) models, which are a specialized kind of deep learning architecture to process grid-like data. A common representation of the image using convolutional operations is to scan across the input image using some convolutional operations for capturing local patterns such as edges or textures to build up a hierarchical representation of the image. CNN models consist of several key layers - convolutional layers, pooling layers, and fully connected layers - that progressively learn more complex features from the data [7]. This ability to automatically detect meaningful features is efficient in tasks such as image segmentation, which will be shown in the models that are utilized in the next sections.

1.3. Computer Vision

Computer vision is the field of artificial intelligence that enables computers to interpret and understand visual information from the world. Through computer vision techniques, machines can analyze and process images and videos to extract meaningful information [6]. This capability is essential in real-world applications such as facial recognition, autonomous driving, and medical imaging, where machines need to "see" and interpret their surroundings.

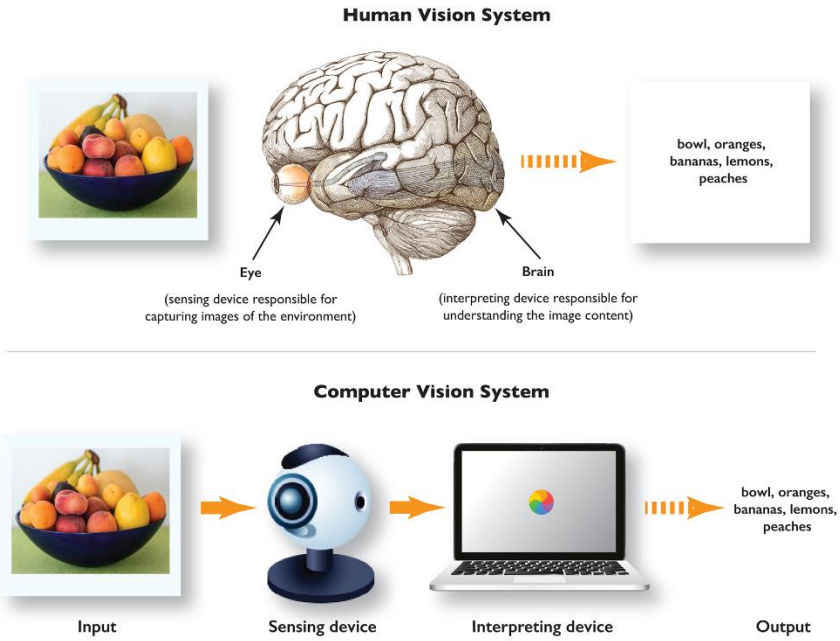

Figure 9 Comparison of computer vision and human in processing visual input. Source: https://kili-technology.com/data-labeling/computer-vision/top-computer-vision-applications

In fact, there are many ways to solve computer vision problems. But in this research what we use mostly is Deep Learning with CNN models and certain processing techniques for the image. This combination makes this approach to solve it technical and using this method it allows us to change the models for the task of road lane marking segmentation.

1.4. Image Segmentation

Image Segmentation is a crucial process in computer vision that involves segmenting an image into meaningful regions or objects, providing a clearer understanding and detailed analysis of the visual data.

1.5. Semantic Segmentation

Semantic Segmentation is a pixel-level classification method in the field of Image Segmentation that assigns a class label to each pixel of an image [7]. The goal is to provide a total view of the different classes present in the image. As mentioned in previous sections, the CNN models were used in this study to perform Semantic Segmentation. Given a raw

image, the model outputs segmented masks by assigning each pixel a color indicating the class to which it belonged. The process is illustrated in Figure 10 below.

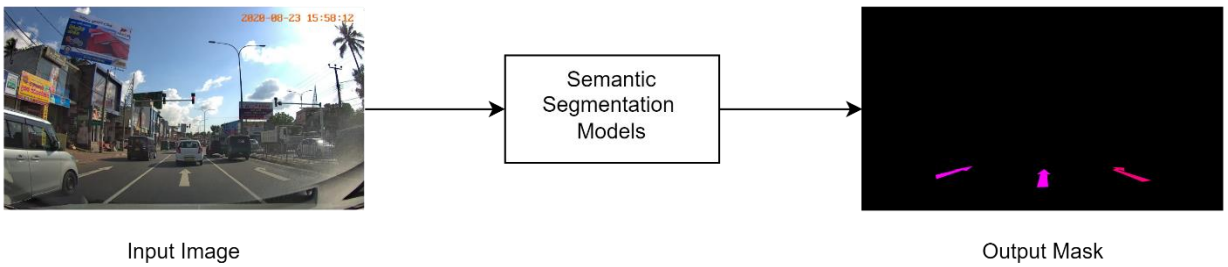

Figure 10 Example of Input and Output of the Semantic Segmentation Models

In the context of road lane markings, semantic segmentation creates a global grasp of the road scene by positioning every pixel according to the lane marking type. This approach is extremely useful in assessing the quality of road markings (intact or destroyed, dissolved) and in helping to clarify them for users to see while moving on the road.

2. Data Preparation

Before applying any training, it is notable to ensure that the dataset is well-prepared to maximize the performance of the models. In this part, proper data augmentation and preprocessing techniques will be taken. These adjustments operated for a rich and diverse dataset.

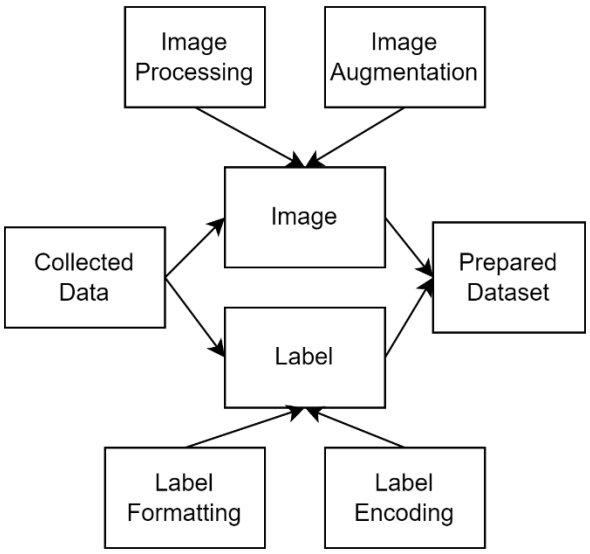

Figure 11 Workflow of the Data Preparation Process

2.1. Data augmentation

Data augmentation was used to provide a variety of road scenes on the street in this research, allowing the model to learn effectively across different locations, terrains, and lighting.

2.1.1. Image Flipping

This includes reversing the image along the vertical axis (vertical flipping) and horizontal axis (horizontal flipping). Technically, pixels are repositioned to reflect the opposite side of the image. The horizontal flip makes the scene like on the opposite lanes on the one-way road, while vertical flipping brings a perspective of a driver on the inverse lane on the two-way road.

Figure 12 Original Image and the Flipping Results

These techniques were used to replicate the symmetric view of the drivers while looking at roads and road markings, enhancing the model's ability to generalize with two-way markings or vision in different road lanes.

2.1.2. Image Rotation

We performed rotating the image by 90 degrees and a small range such as 7 degrees using a rotation matrix to rearrange the pixels while preserving the original scale and aspect ratio.

Figure 13 Original Image and Rotation Results

Rotations are useful in showing markings in the road that intersects with the camera view, and the small angle rotation helps create various cases where the camera is unstable over bumpy, uneven terrain.

2.1.3. Brightness adjustment

This method modifies the intensity of all pixels by multiplying their values by a scaling factor, with a factor greater than 1 for brightness increment and smaller than 1 for decreasing it. In this approach, we chose 1.17 and 0.83.

Figure 14 Original Image and Brightness Adjustment Results

The alteration generates more data on driving in the afternoon or evening from morning images and vice versa.

The augmentation process was implied for all the images and annotations and created 21,465 images for training.

2.2. Data preprocessing

After the data was enriched, the preparation continued with preprocessing the data to fit the model training requirements, this includes resizing the image for model input size and switching label formats.

2.2.1. Image processing

Image resizing

Before inputting the data into training, the size of the image must match the input size of the models. In this experiment, the image size was adjusted below to achieve the model's prerequisite:

Table 3 Input Size and Shape Specifications for Models

Model	Size and Shape (Width x Height x RGB Channels)
U-Net	512 x 512 x 3
BiSeNet	720 x 720 x 3

Image Normalization

As the semantic segmentation models contain a pre-trained backbone with the weight extracted from the ImageNet dataset, a normalization method should be taken to follow the training requirements and enhance the model's ability. Specifically, the normalization process adjusts the pixel values of the images to have a mean of (0.485, 0.456, 0.406) and a standard deviation of (0.229, 0.224, 0.225) for the RGB channels, respectively.

2.2.2. Label Processing

Label Formatting

As demonstrated in the Data Understanding part, only PNG image ground truth masks are chosen to train specifically semantic segmentation models. However, encoding techniques must be taken to suit the computational process of the models

One – hot encoding

This is the crucial step that is needed with the ground truth masks before feeding the data into semantic segmentation models. One-hot encoding converts each pixel in the image into a vector of binary values, the vector is the same length as the number of classes with all elements set to 0 except for the one element corresponding to the class of pixel, which is set to 1. For example, with 11 road marking classes and 1 background class, each pixel in the image is represented by a 12-dimensional vector:

Background: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Class 1 (Bus Lane): [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Class 2 (Cycle Lane): [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

…

Class 11 (Straight-Right Arrow): [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

This method transforms categorical data (pixel value) into a numerical format that the model can understand and perform calculations.

3. Model Architecture

A total of 2 CNN models were used in this study, including U-Net and BiSeNet. Both are well-established models that are famous for their effectiveness in semantic segmentation tasks.

3.1. U-Net

In this research, we applied transfer learning techniques using a U-Net model with a ResNet34 encoding backbone that is pre-trained from the famous ImageNet dataset. U-Net model is a fully convolutional encoder-decoder network for pixel-level segmentation tasks, aided by skip connections that help preserve spatial context [7]. About the information, the input shape of the model is 512 x 512 x 3, the output shape is 512 x 512 x 12 with 158 layers and 23706124 trainable parameters.

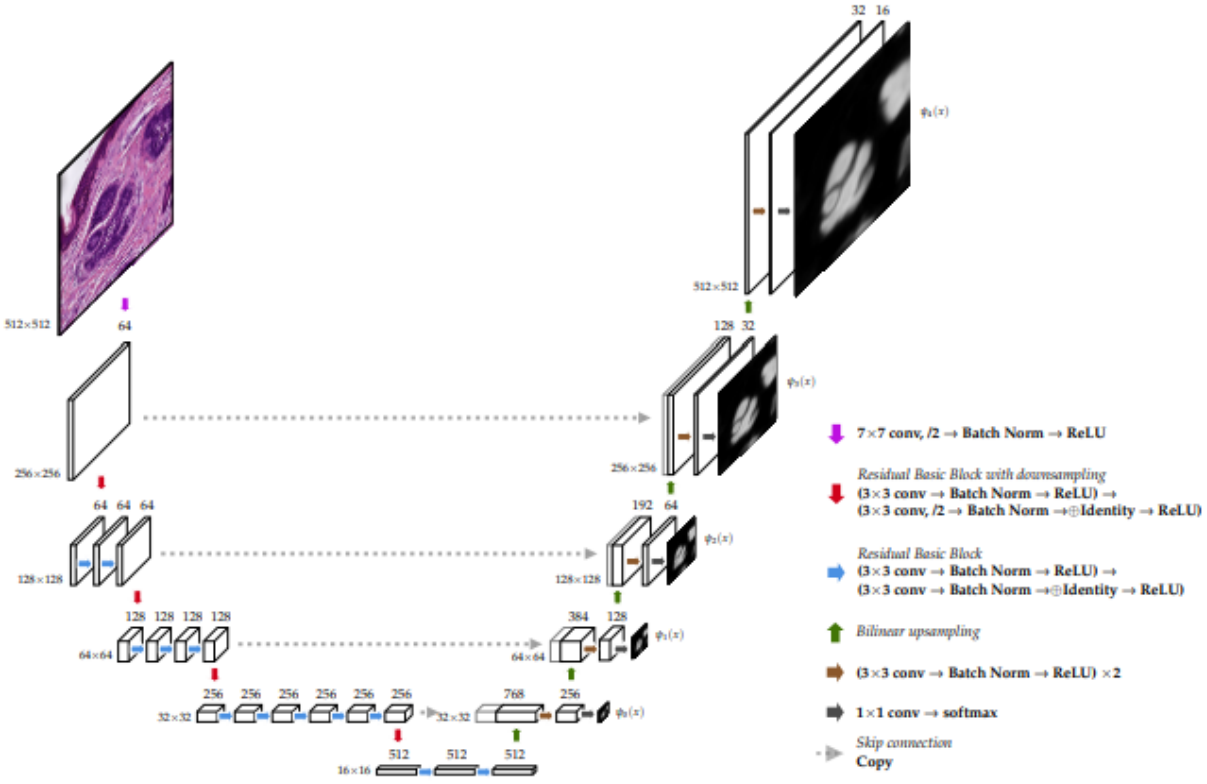

Figure 15 U-Net model architecture[9]

The encoder in our U-Net model is designed similarly to the original, with modifications to add the ResNet34 backbone. The encoder consists of five blocks that downsample the spatial resolution, each block in the encoder contains two 3x3 convolutions, with the first convolution using a stride of 2 to halve the spatial dimensions while doubling the number of feature channels. This downsampling technique allows the network to learn complex features while reducing the computational load. Additionally, both encoders include an initial 7x7 convolution with 64 filters and a stride of 2, which reduces the input resolution and enriches the information gain, allowing the model to maintain larger feature sizes without significantly increasing memory usage.

The decoder in the U-Net architecture aims to regenerate the construction of the segmentation mask by upsampling the encoded features, with the same number of blocks as the encoder, restoring the spatial resolution to the original input size. The blocks consist of bilinear upsampling followed by concatenation with the corresponding encoder block's output through skip connections.

Besides these additions, the decoder carries additional modifications, such as extra 1x1 convolutions to produce intermediate point maps used in deep supervision and fusion, contributing to better segmentation accuracy. Batch normalization and ReLU activation are applied in the decoder to stabilize the training process and improve model performance. Softmax activation function is used in the final layer of U-Net, converting the raw output values into probabilities that sum to 1 across the classes, and return the segmentation map of shape 512 x 512 x 12 that represents the predicted probabilities for each class at every pixel in the image [9].

3.2. BiSeNet

In addition to the U-Net model, we also used the BiSeNet (Bilateral Segmentation Network) for this research. BiSeNet model is composed of two primary pathways: the Spatial Path and the Context Path. The input shape of the model is 720 x 720 x 3 and the output shape is 720 x 720 x 12, with 171 layers and 23175056 trainable parameters.

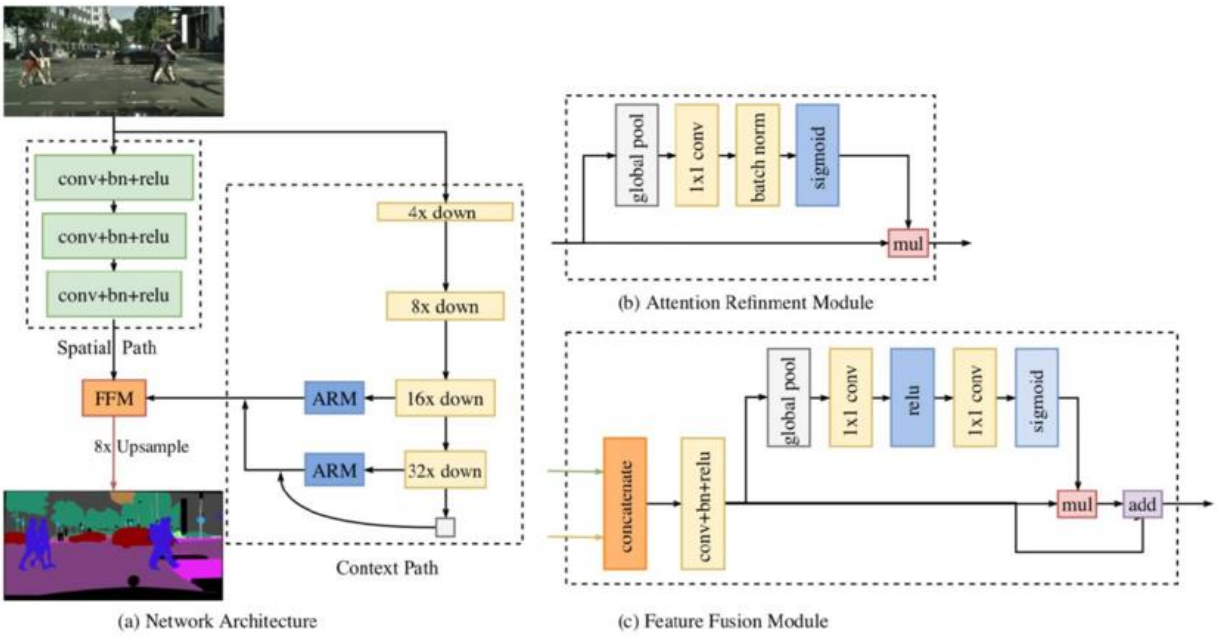

Figure 16 BiSeNet model architecture[10]

First, the Spatial Path (SP) is designed with a small stride to help preserve the spatial information and also generate features at high resolution. Spatial Path contains a total of 3 blocks, each block has 1 convolutional layer with stride 2. Next is a batch normalization

and then a ReLU activation layer. This configuration results in the feature map outputs that were downsampled into 1/8 the size of the original input, allowing the SP to encode detailed spatial information efficiently while maintaining a manageable computational complexity. Secondly, while the Spatial Path focuses on retaining spatial information, the Context Path (CP), emphasizes expanding the receptive field to introduce a broader contextual understanding of the scene. This receptive field enables the network to understand the relationships between different objects and features within the image. To achieve a large receptive field efficiently, we applied transfer learning while using ResNet18 as the backbone, to quickly downsample the feature map to 32 times and then combine it with global average pooling to make a more diverse receptive field, which contains the most global context information inside the CP.

Within the Context Path, the Attention Refinement Module (ARM) is utilized to refine feature maps at each stage. ARM employs global average pooling to capture the global context and computes an attention vector that guides feature learning by emphasizing relevant features. This attention mechanism refines the output of each stage in the CP without requiring upsampling, thereby reducing computational costs. ARM integrates contextual information more effectively, enhancing the network's ability to focus on important features within the image.

The last part of the model architecture is the Feature Fusion Module (FFM), this is the merge of outputs of SP and CP. Given that the SP outputs low-level features rich in detail (e.g., edges and boundaries), and the CP outputs high-level semantic features, the FFM is essential for fusing these types of information into a cohesive feature representation. The fusion process begins by concatenating the outputs of the two paths, followed by batch normalization to balance and scale the features appropriately. The combined features then pass through pooling and convolution layers, which compute the final weights and refine the fused feature map [10].

The activation used in the final layer of BiSeNet is also a Softmax function, converting the learned features into a probabilistic segmentation map of shape 720 x 720 x12.

4. Prediction Process

Since the output of the semantic segmentation model consists of probability maps for class predictions, the format of these outputs needs to be adjusted to achieve better visual results.

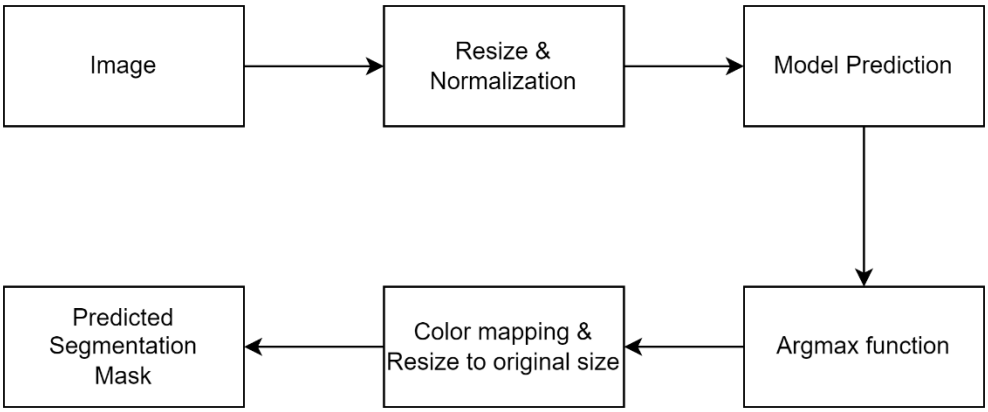

Figure 17 Prediction workflow

The figure illustrates the process of generating the segmentation mask using our models. Initially, the image starts to be resized and normalized before feeding into the model. Next, the model returns the probability maps as mentioned above, and then the class with the highest probability is selected by applying the argmax function along the class dimension. This function converts the maps into a single-channel image where each pixel value represents a class. The last step is mapping the label into distinct colors following the given color of the dataset and resizing the prediction image to the original size for visual enhancement.

5. Evaluation Metric

5.1. Cross Entropy Loss:

Cross-entropy loss was used as the evaluation metric for semantic segmentation, calculating the difference between predicted class probabilities and the true class labels for each pixel. This loss function encourages the model to accurately classify each pixel by penalizing incorrect predictions, improving segmentation performance across classes.

$$\text{Loss} = -\sum_{i=1}^{C} y_i \log(p_i)$$

C: The index of the current class in the summation, ranging from 1 to C.

y_i: The true label for class i at each pixel, where y_i = 1 if the pixel belongs to class i and 0 otherwise.

p_i: The predicted probability that the pixel belongs to class i

5.2. Mean Intersection over Union (mIoU):

The metric for evaluating the semantic segmentation models in this research is mean Intersection over Union, which measures the average overlap between the predicted segmentation masks and the ground truth masks across all classes, calculated as the ratio of the intersection area to the union area for each class and then averaged. This describes how well the predicted segment aligns with the true boundaries of the object in the mask.

$$\text{mIoU} = \frac{1}{C}\sum_{i=1}^{C}\frac{|P_i \cap G_i|}{|P_i \cup G_i|}$$

C: the total number of classes.

P_i: The predicted set of pixels for class i.

G_i: The ground truth set of pixels for class i.

$|P_i \cap G_i|$: The number of pixels in the intersection of the predicted and ground truth sets for class i.

$|P_i \cap G_i|$: The number of pixels in the union of the predicted and ground truth sets for class i.

IV. EXPERIMENTS

After understanding all the materials and methods, the experiments are conducted to measure data and model performance. This section will delve into the setup, result, and analysis

1. Experiment Setup

The experiment was operated using the online virtual environment and GPU of Kaggle and Google Colab as our personal devices and computational units were insufficient. The GPUs are NVIDIA TESLA P100 (Kaggle) and NVIDIA TESLA T4 (Google Colab). Python was employed with additional useful libraries and frameworks for Image Processing and Deep Learning such as cv2, Pytorch, etc.

For the training process, as noted in each model's characteristics, we applied transfer learning techniques to enhance model performance. The semantic segmentation models utilized backbones that were pre-trained on the ImageNet dataset, allowing the models to leverage the rich feature representations learned from a large and diverse set of images.

For the model hyperparameters, the learning rate was set at 0.0001. The models were trained for 100 epochs with the size of 8 images per batch due to the large size of the training dataset and the lack of computing memory. Moreover, we shuffled the dataset after each epoch to increase the randomness of each training initialization and validate the model after training for 1 epoch. Adam optimizer was used and the model was automatically saved if the validation metric evolved. The best and last model weights were updated after each training epoch. Subsequently, the loss and metric values were recorded inside a text file for training continuation and to enable visualization of the results

To assess the new dataset quality, we performed training the models on the original data with no modifications, and the new dataset that was created to analyze the differences between them.

2. Experiment Results

Following the setup and training of the models, we acquired, evaluated, and compared the results using key performance metrics such as dataset quality, scores, and prediction speed.

2.1. Dataset Quality

As outlined earlier, the models were trained with two different versions of the dataset: the transformed dataset (which included additional data and augmentation methods), and the original dataset with only preprocessing techniques. As shown in Table 4 below, we see a clear improvement in the measurement of the mIoU metric by using the new dataset.

Table 4 Dataset quality comparison using mIoU

Model + Dataset version	Metric (mIoU)
U-Net (Original data)	0.55
BiSeNet (Original data)	0.57
U-Net (Additional + Augmented data)	**0.61**
BiSeNet (Additional + Augmented data)	**0.64**

Both U-Net and BiSeNet showed notable gains in mIoU with the new version of the image set. Specifically, BiSeNet mIoU score rose from 0.57 to 0.64, and U-Net improved from 0.55 to 0.61. These results indicate that both models tend to generalize more effectively when trained on the enriched dataset, highlighting the impact of data augmentation and enhancement techniques on segmentation efficiency.

With this result, the models trained on the transformed dataset were selected as the main models for further measurement and evaluation.

2.2. Metric Scores

The training process for both U-Net and BiSeNet models was visualized through the two key metrics: the loss function and the mIoU score over 100 epochs on the validation set

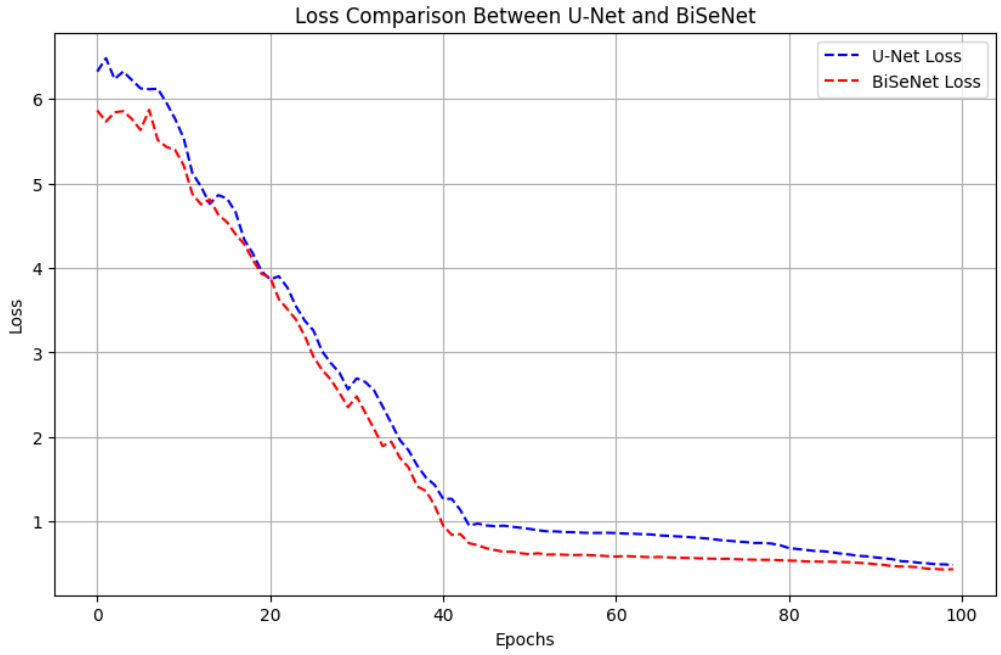

Figure 18 Loss Comparison Graph of the two models on the validation set

Figure 18 visualizes the loss reduction in validating both of the models. As expected, a decrease is demonstrated in the loss, indicating the ability to learn and optimize the output of the two models. Moreover, it can be observed that BiSeNet started with a lower loss and converged more quickly than U-Net. This described that BiSeNet achieved an outstanding performance, likely due to its dual-path architecture that captures a great number of features in spatial and contextual information. Lastly, it is obvious that the models minimized the fluctuations at about 42 epochs signaling that the models have reached near-optimal performance.

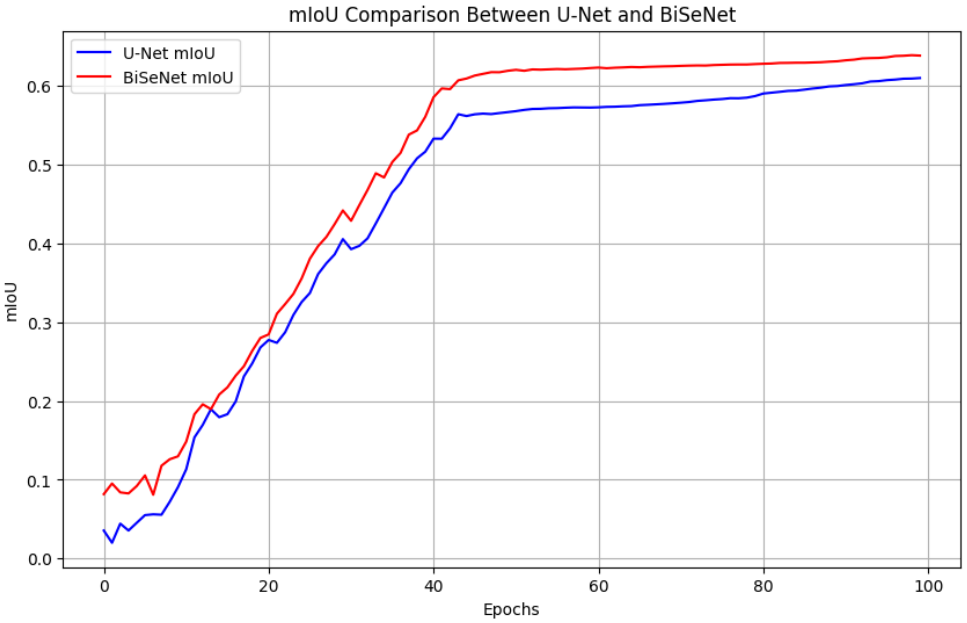

Figure 19 mIoU Comparison Graph of the two models on the validation set

Additionally, Figure 19 tracks the improvement of the mean Intersection over Union (mIoU) score for U-Net and BiSeNet models across epochs. Similar to the loss graph, BiSeNet outperforms U-Net, reaching a higher mIoU value in fewer epochs. While both models show a steady increase in mIoU during the initial stages, BiSeNet consistently maintains a performance advantage over U-Net, achieving a final mIoU of approximately 0.64 compared to U-Net's 0.61.

The results clearly indicate that in the context of road lane marking segmentation, the BiSeNet model is likely to achieve better than U-Net with higher mIoU scores and a more rapid decrease of loss during training. This suggests that BiSeNet not only delivers better performance metrics but also has the potential to offer more accurate and visually effective results in practical applications.

2.3. Metric by Classes

After getting an overview of the training quality, we delved into detail about the model performance based on each class. The table below compares the Intersection over Union (IoU) scores for different road marking classes between the U-Net and BiSeNet models.

Table 5 Comparison of IoU score of the models by classes

	Model	
Class	**U-Net**	**BiSeNet**
Background	**0.82**	0.79
Bus Lane (BL)	0.50	**0.57**
Cycle Lane (CL)	0.52	**0.55**
Diamond (DM)	**0.71**	0.70
Junction Box (JB)	0.61	**0.65**
Left Arrow (LA)	**0.57**	0.54
Pedestrian Crossing (PC)	0.65	**0.68**
Right Arrow (RA)	0.59	**0.62**
Straight Arrow (SA)	0.76	**0.79**
Slow (SL)	0.53	**0.60**
Straight-Left Arrow (SLA)	0.60	**0.63**
Straight-Right Arrow (SRA)	0.48	**0.55**
mIoU	0.61	**0.64**

Overall, we found that BiSeNet outperforms U-Net in most classes with an overall mean IoU (mIoU) of 0.64 while U-Net performs at 0.61. BiSeNet achieves better performance in classes such as “Bus Lane”, “Cycle Lane”, and “Slow” as well as “Straight-Right Arrow”, which suggests its capability to attribute more complex and detailed road lane markings. U-Net is slightly better in a few classes, like “Background” and “Diamond”. BiSeNet performs well across nearly all of the other classes and reaches comparable results in “Pedestrian Crossing” and “Straight Arrow” compared to the U-Net model.

3. Prediction and Error Analysis

In this section, we visualize the prediction of the models with the validation set and discuss the quality and error of the outcomes.

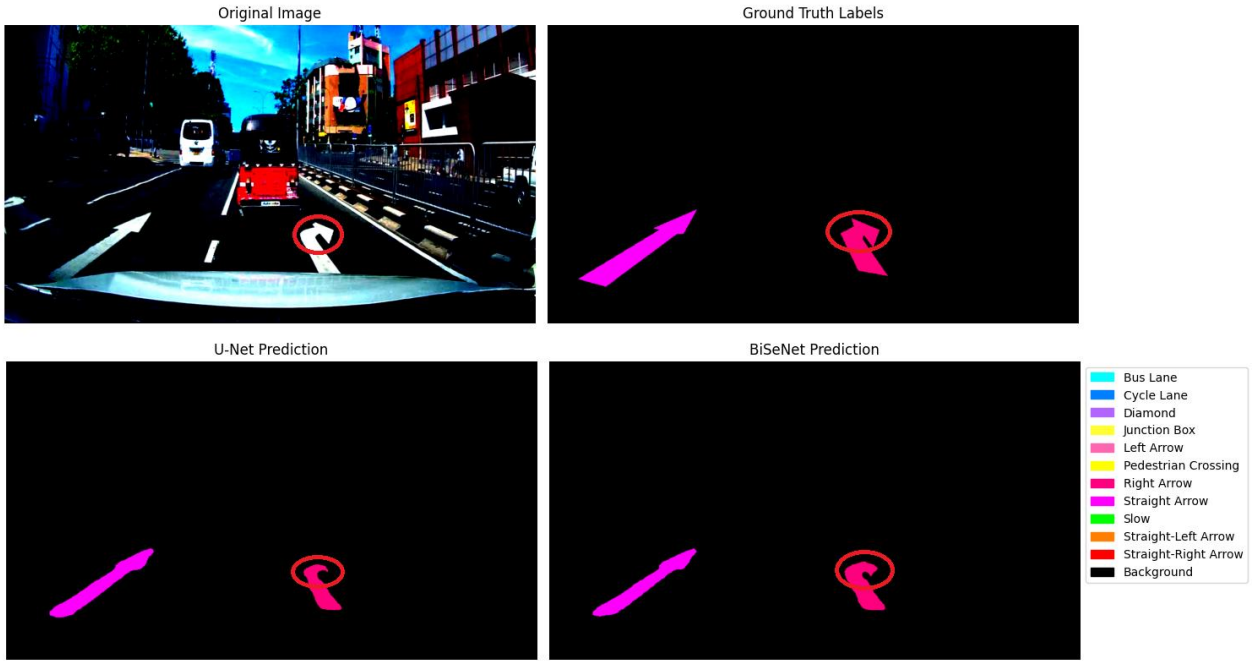

Figure 20 Prediction of U-Net and BiSeNet models compares with Original Image and Ground Truth Labels

Based on the results discussed in previous sections and the model predictions visualized in the figure above, it can be concluded that while the predicted shapes of the road markings are not always precise, the models have been relatively successful in identifying the correct locations and classifying most of the pixels belong to a class. As seen in Figure 20, both U-Net and BiSeNet managed to segment most of the road markings, though the shape of the Right Arrow in the model's prediction is not fully accurate compared to the ground truth labels. This inconsistency suggests that the models still struggle with correctly capturing more complex or intricate shapes, which may require further refinement in model architecture or training strategies.

Figure 21 Effect of Weather, Camera Angle and Markings Quality on model performance

Another important factor influencing the prediction results is environmental conditions, such as weather, and the quality of the markings, which may be blurred or damaged. In the example above ò Figure 21, the straight arrow marking on the road has lost part of its left side, which has impacted the BiSeNet model, as it only predicted the right side of the arrow; while U-Net, unfortunately, only predicted the tip of the arrow. Additionally, both models failed to predict the diamond marking on the left side of the frame. This is likely due to the perspective distortion caused by the camera angle, which makes it difficult for the models to accurately recognize the shape of the diamond. These observations highlight the practical applicability of the model, particularly BiSeNet, in evaluating the quality of road markings for street improvement projects. However, a larger, more diverse dataset is also required to enhance the model's ability to handle varied conditions.

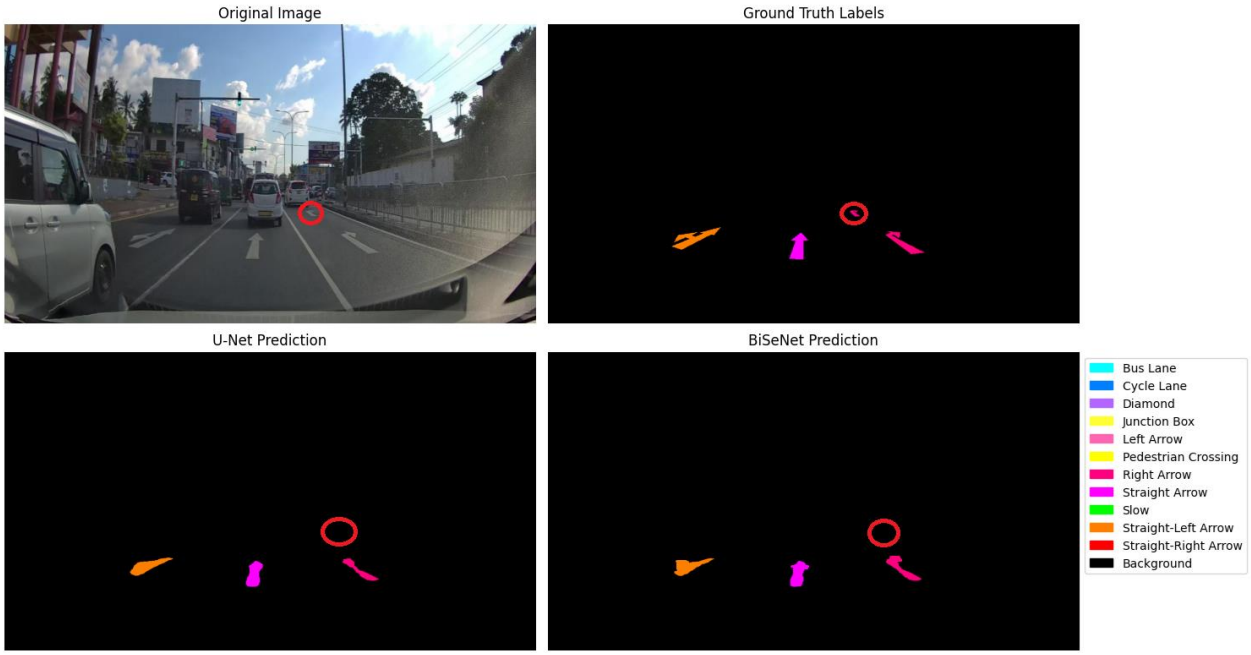

Figure 22 Effect of distance on model performance

In addition to the previously discussed factors, distance plays a critical role in the models' abilities to make accurate predictions. As shown in Figure 22 above, both U-Net and BiSeNet failed to predict the second right arrow markings on the right of the image. The long distance from the camera makes the shape of the markings harder to recognize, leading to incorrect predictions. This issue could be addressed by using more advanced image acquisition devices that can highlight distant objects more clearly. Furthermore, enhancing the model with additional training data that includes distant markings in the view could significantly improve its performance in this scenario.

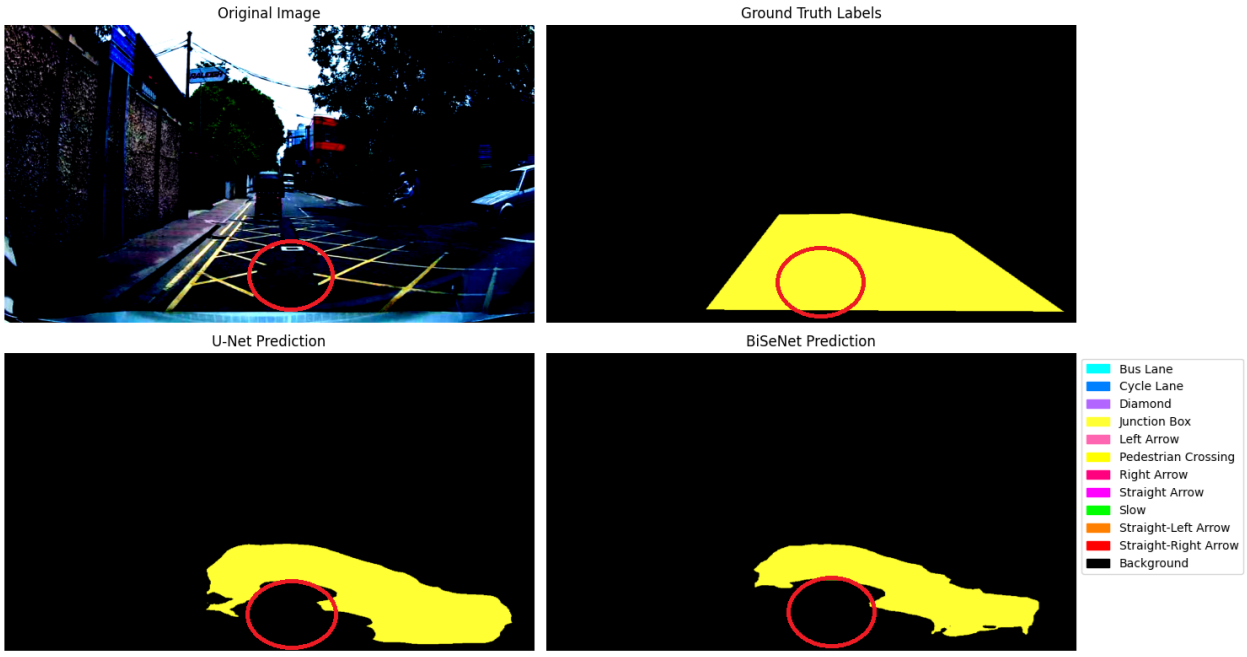

Figure 23 Effect of Annotation Quality on model performance

Finally, another factor that impacts the model's accuracy is the subjectivity of the author in labeling the dataset. For example, in Figure 23, the junction box is split in the original image, yet the author labeled the entire area as a single entity. As a result, the model misclassified regions where the junction box line is not present and this affects the model score calculation. To address this issue, it is crucial to have professional and accurate labeling of the dataset classes, ensuring that the model can accurately learn the features and make more reliable predictions.

V. CONCLUSION

1. Conclusion

This study has successfully demonstrated the application of deep learning techniques, particularly Convolutional Neural Network (CNN) models, in road lane marking segmentation, a critical task for Advanced Driver Assistance Systems (ADAS) and autonomous driving technologies. During the research period, we employed various semantic segmentation models including U-Net and BiSeNet to generate colorful masks for segmenting each of the road lane marking classes under many environmental conditions (such as changes in lighting, weather, and road texture) provided by the CeyMo dataset. Our results illustrate that BiSeNet outperformed U-Net in the metric and also visual results, as it reached the mIoU of 0.64, slightly greater than 0.61 of U-Net, showing an advantage in handling more complex road markings and providing a better mask for visualization.

One of the outstanding factors from this research is the significant impact of data augmentation and preprocessing on model performance. Both models improved segmentation accuracy when trained on an enriched dataset that included additional images and augmentation techniques such as image flipping, rotation, and brightness adjustments. The mean Intersection over Union (mIoU) scores for both U-Net and BiSeNet increased notably after applying data enhancement. This underscores the importance of diverse and well-prepared data in enhancing model generalization and performance and confirms that we have reached the objectives of creating a better dataset for the training of the models.

However, several factors still affected the prediction of both models, including environmental conditions, distance, and marking quality. For example, rain and poor-quality roads made pavement markings inconsistent while predicting. Additionally, the distance from the camera affected the models' abilities to correctly identify markings, with distant markings being not seen or misclassified. Lastly, the annotation quality must also be considered to increase the results and good metric calculation.

In conclusion, we have proposed a viable solution that meets the objectives of this research. Nevertheless, there are still several areas that require improvement to transform this approach into a fully functional product that can be used in professional transportation assisting systems.

2. Limitations and Future Works

Throughout the process of completing this thesis, we faced several limitations that impacted both the quality of the dataset and the model training. Firstly, despite our efforts to enrich the dataset by collecting external data and applying augmentation techniques, the number of images was still insufficient to fully represent all the classes in the dataset. Moreover, manually collecting, filtering, and labeling these images required a considerable amount of time, effort, and understanding, which further complicated the process. Additionally, computing resources were also a significant challenge. The device utilized in this research was a personal computer of a student, which was not equipped to handle the task of training large deep learning models. Consequently, we relied heavily on virtual machines and cloud environments provided by Kaggle and Google Colab and the limited access to these platforms, due to time and funding constraints, led to the training interruption, making it difficult to store and combine results efficiently. Our limited resource availability also constrained the number of models we could implement, and thus we did not provide a more extensive analysis of different semantic segmentation models for the task. Another major challenge was the quality of the dataset annotations. Some classes were not labeled accurately at the pixel level, both in the original dataset and the additional data we collected. This negatively affected the models' performance during training and prediction. Lastly, the camera view used to collect data is still not optimal in many aspects. , particularly for distant road markings, making it harder for the models to learn effectively.

To address these drawbacks in our research or the study of researchers or engineers who are likely to inherit or improve the project, we propose several solutions. First, expanding the dataset to be larger and including more diverse road conditions is essential. Contributing to that, a detailed evaluation and more accurate labeling techniques should be employed to

create a professional dataset for training. Upgrading data collection equipment to capture clearer and more detailed images, especially of distant road markings, would also improve the quality of the data. In addition, increasing computing resources like running on GPUs with continuous training time will prevent the training process from being interrupted and enable us to test a variety of models. Lastly, significant improvement in a more professional workflow through collaboration with scientists and engineers would be needed to take the research from progress to practical use.

VI. REFERENCES

[1] A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk, "Review of Advanced Driver Assistance Systems (ADAS)," *AIP Conference Proceedings*, vol. 1906, no. 1, Nov. 2017

[2] J. Wu, W. Liu, and Y. Maruyama, "Automated Road-Marking Segmentation via a Multiscale Attention-Based Dilated Convolutional Neural Network Using the Road Marking Dataset," *Remote Sens.*, vol. 14, no. 18, pp. 4508, Sept. 2022.

[3] O. Jayasinghe, S. Hemachandra, D. Anhettigama, S. Kariyawasam, R. Rodrigo, and P. Jayasekara, "CeyMo: See More on Roads - A Novel Benchmark Dataset for Road Marking Detection," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3104-3113, Jan. 2022.

[4] T. Bruls, W. Maddern, A. A. Morye, and P. Newman, "Mark Yourself: Road Marking Segmentation via Weakly-Supervised Annotations from Multimodal Data," *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 2018.

[5] W. Jang, J. Hyun, J. An, M. Cho, and E. Kim, "A lane-level road marking map using a monocular camera," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 187–204, Jan. 2022.

[6] LeCun, Y., Bengio, Y., & Hinton, G., "Deep learning." *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[7] Ronneberger, O., Fischer, P., & Brox, T., "U-Net: Convolutional networks for biomedical image segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, pp. 234-241, May. 2015.

[8] Gabriela Csurka, Riccardo Volpi and Boris Chidlovskii, "Semantic Image Segmentation: Two Decades of Research", Foundations and Trends® in Computer Graphics and Vision: Vol. 14: No. 1-2, pp 1-162, Oct. 2022

[9] J. L. Arrastia, N. Heilenkötter, D. O. Baguer, et al., "Deeply Supervised UNet for Semantic Segmentation to Assist Dermatopathological Assessment of Basal Cell Carcinoma," *Journal of Imaging*, vol. 7, no. 4, pp. 71, Apr. 2021.

[10] C. Yu, J. Wang, C. Peng, et al., "BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation," *Computer Vision - ECCV 2018*, vol. 11218, pp. 334-349, Sept. 2018.

[11] Mahesh, B., "Machine Learning Algorithms - A Review," International Journal of Science and Research, vol. 9, pp. 381-386, 2020.

VII. APPENDICES

APPENDIX 1

Argmax function:

This function is used after the semantic segmentation models return the probability map where each pixel has probabilities for belonging to different classes. The function is then applied to each pixel to select the class with the highest probability. This produces a segmentation mask with each pixel representing the most likely class.

$$\boldsymbol{Class}(\boldsymbol{x},\boldsymbol{y}) = \underset{\boldsymbol{c}}{\boldsymbol{arg\,max}}\, \boldsymbol{P}(\boldsymbol{c} \mid \boldsymbol{x},\boldsymbol{y})$$

Class(x,y): The predicted class for pixel (x, y)

$\underset{\boldsymbol{c}}{\boldsymbol{arg\,max}}$ **:** The argmax function, which returns the class c that maximizes the predicted probability.

$\boldsymbol{P}(\boldsymbol{c} \mid \boldsymbol{x},\boldsymbol{y})$**:** The probability that pixel (x,y) belongs to class c, is typically derived from a softmax output.