A Study on Road Surface Marking Quality Evaluation Using Machine Learning and Computer Vision

by **Boudissa Mehieddine**

July 2024

Thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy
under the supervision of Prof. Hiroharu KAWANAKA

Division of Systems Engineering,
Graduate School of Engineering, Mie University

Certificate of Authorship / Originality

I, Boudissa Mehieddine, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in Systems Engineering Graduate School, Mie University.

This thesis is wholly my own work unless otherwise referenced or acknowledged. Also, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Local Government of Mie Prefecture.

Signature: BOUDISSA

Date: May 10, 2024

Abstract

In the context of rapidly accelerating urban expansion and advancements in deep learning, the need for sustainable and efficient traffic management systems has emerged as a pivotal research area. This dissertation addresses the challenges associated with the deterioration of road surface markings, particularly in the Mie prefecture of Japan.

Central to this research is a comprehensive dataset of 13,000 high-resolution images (3000x1600) sourced from the local government facilities of Mie prefecture. These images span diverse roads and intersections in urban, rural, and even off-road contexts. Captured under varied lighting conditions, they present a slew of challenging samples, including images with glare effects, shadowy regions, deteriorated roads, and traffic signs. Through the development of a semi-assisted annotation tool, an initial subset of 400 images was labeled, serving to train a U-Net model that achieved a Dice score of 78.90%. Recognizing the potential of the extensive dataset, a subsequent phase of assisted labeling was undertaken. This effort used the trained model's inference on all images, facilitating a streamlined labeling process. Ultimately, this method yielded 12,000 labeled images, with about 1,000 images deemed unfit for accurate annotations.

The research's early stages successfully detected and segmented these landmarks, integrating both classical computer vision techniques and deep learning approaches. The introduction of the "Efficient VGG-16" model, a tailored version of the renowned VGG-16, emerged as a significant contribution, adeptly evaluating road surface marking quality and achieving a Mean Squared Error (MSE) of 3.62

Further deepening the research, a comprehensive survey of various segmentation models was conducted, with the U-Net model exhibiting notable superiority in terms of Dice score evaluation.

The current trajectory of this research is marked by several promising endeavors:

The exploration of uncertainty-aware regression to refine road surface marking quality evaluation. The application of Diffusion techniques to enhance road surface marking quality assessment. The integration of PhyCV edge detection for a holistic approach to road surface marking detection and evaluation. Conclusively, this dissertation underscores the transformative potential of advanced computer vision techniques in the realm of traffic management and road safety, heralding a new era of research and practical implementations in the domain.

Acknowledgements

First and foremost, I extend my gratitude to God for granting me strength, perseverance, and guidance throughout this journey.

To my parents, Mokhtar and Fouzia, who instilled in me the values of hard work and perseverance. Your unwavering belief in my abilities has been a driving force behind my achievements.

To my devoted wife, Hanane, whose support and understanding were pivotal during the most challenging times. Your love and companionship have made this journey worthwhile.

To my older brother, Ahmed, for consistently offering guidance and for being a beacon of support, especially as I navigated the complexities of settling in Japan.

To Nakano-san, whose assistance and encouragement often went beyond the call of duty. Your help has been invaluable to my research and adaptation to a new environment.

And to Kawanaka-sensei, my supervisor, for the invaluable mentorship, patience, and unwavering belief in my potential. Your insights and feedback have shaped this work.

This dissertation is a testament to your collective influence and support. I am deeply grateful to each of you.

Boudissa Mehieddine
July 10, 2024
Mie, Japan

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Road surface markings[1] are essential components of transportation infrastructures worldwide. In areas such as Mie prefecture, Japan, the value of these markings becomes even more evident when we consider the challenges faced due to their degradation. The deterioration of road surface markings has real-world consequences[2][3][4]. In 2020 alone, the local government reported a concerning 3% sudden stop rate[5], which can be linked to faded or unclear markings on the roads.

Deterioration occurs due to a variety of reasons[4]. Environmental factors, from extreme weather conditions to pollutants, constantly act against the integrity of these markings. Additionally, the daily grind from vehicular traffic, particularly from heavy-duty vehicles, further accelerates wear and tear. As these markings fade or become unclear, the risks for misinterpretation or lack of visibility for drivers increase, impacting overall road safety.

Addressing this issue goes beyond the mere act of repainting or refurbishing these lines. Regular manual maintenance of road surface markings is a significant endeavor, both in terms of manpower and financial investment. Given the vast expanse of roads and the intricate nature of some of these markings, manual maintenance becomes a resource-intensive task for municipalities.

This research leverages the capabilities of deep learning and computer vision techniques to offer a solution. The goal is to develop models that can autonomously detect, evaluate, and even suggest reconstructions for these road surface markings using high-resolution image data. By automating this process, we aim to make the maintenance of road markings more efficient and effective.

Furthermore, the implications of this research extend to future-forward domains like autonomous vehicle technology. Clear, discernible road markings are crucial for the safe navigation of self-driving cars. Therefore, by ensuring the clarity and quality of these markings, this work indirectly contributes to the advancements in the autonomous vehicle sector.

1.1 Research Objectives and Overview

Road surface markings are essential for guiding drivers and ensuring road safety. Their clarity and maintenance directly correlate with the efficiency of transportation systems. However, regions such as the Mie prefecture in Japan have witnessed deterioration in these markings over time, leading to potential safety risks. This necessitates an in-depth exploration into the challenges and solutions for the detection[5], evaluation[6], and maintenance of these markings. The primary research question this thesis seeks to address is: How can advanced computer vision techniques be employed to detect, evaluate, and reconstruct deteriorating road surface markings, ensuring road safety and promoting urban sustainability? To tackle this overarching question, the research is divided into specific objectives.

1.1.1 Objective 1: Advanced Detection and Segmentation of Road Surface Markings

The foundational step in addressing the degradation of road surface markings is their accurate detection. Incorrect or missed detection can lead to inaccurate evaluations and ineffective interventions. Classical vs. Deep Learning: A synergy of classical computer vision techniques and deep learning models offers a robust method for detection. While classical techniques provide tried-and-tested methodologies, deep learning introduces adaptability and precision, especially when dealing with diverse road conditions and lighting scenarios.

Diverse environmental conditions, varying degrees of deterioration, and different road types present unique challenges. The objective will delve into creating a method that can handle these complexities, ensuring comprehensive and accurate detection across various scenarios.

1.1.2 Objective 2: Comprehensive Quality Evaluation of Detected Road Surface Markings

Significance of Quality Evaluation: Once detected, assessing the condition of road surface markings becomes paramount. An inaccurate evaluation could lead to unnecessary interventions or overlooked deteriorations. Tailored specifically for this task, the "Efficient VGG-16" model emerges as a pivotal tool. It's an optimized version of the renowned VGG-16, designed to evaluate the quality of road surface markings with high precision. The model's efficiency will be benchmarked against real-world datasets, ensuring its applicability and reliability in diverse conditions. Potential challenges in its implementation, as well as solutions to overcome them, will be explored in-depth. Subsequent chapters of this thesis will dive deeper into the methodologies, experiments, datasets, and results related to these objectives. Through this exploration, this research aims to present actionable solutions and insights into the challenges posed by deteriorating road surface markings, and the potential of computer vision techniques in addressing them.

Chapter 2

Literature Review

2.1 Introduction

The realm of road safety and urban planning has witnessed significant advancements with the integration of computer vision techniques. Road surface markings, being pivotal for guiding drivers and ensuring road safety, have emerged as a focal point of research. Their detection and evaluation hold the key to maintaining and enhancing the efficiency of transportation systems. This chapter delves into the extensive body of literature that surrounds the detection and evaluation of road surface markings, as well as the pivotal role of segmentation models in these processes.

2.1.1 Introduction to Road Surface Marking Detection and Evaluation

Road surface markings, while seemingly simple, have a storied history marked by innovation and adaptation. Over the decades, these markings have witnessed substantial transformations in multiple dimensions. From their initial rudimentary designs, we've seen evolutions that have adapted to the changing nature of roads, vehicles, and driving conditions. The materials employed have shifted from basic paints to more enduring and reflective substances, ensuring that they remain visible under varying weather conditions and times of day. Additionally, the techniques for applying these markings have been refined, ensuring longevity and durability even under heavy traffic conditions. The aim has always been clear: to enhance road safety. As urban areas grow and traffic patterns become more complex, the role of clear and durable road surface markings has never been more crucial. They not only guide drivers but also play a pivotal role in traffic management, pedestrian safety, and overall urban planning[7][8][9][10][11].

Detecting road surface markings with high precision is a task fraught with complexities. The very nature of roads – open to the elements, bearing the brunt of vehicular traffic, and subject to wear and tear – means that these markings can deteriorate, fade, or even be obscured. Environmental factors like rain, snow, and shadows, or road conditions like cracks, debris, or water logging, further complicate the detection process. Moreover, the diversity in road types, from busy urban intersections to serene rural paths, introduces variability in how these markings appear. Over the years, researchers have grappled with these challenges, devising methods to enhance detection accuracy. Techniques have ranged from traditional image processing methods to advanced machine learning algorithms. This section will delve deep into these challenges,

Figure 2.1: Road Surface Marking in Countryside

Figure 2.2: Road Surface Marking in Urban Area

highlighting the pioneering solutions that have been proposed, their successes, and the hurdles yet to be overcome. See Figures 2.1 and 2.2 for road surface marking examples. Merely detecting road surface markings isn't enough; understanding their quality is paramount. A faded or broken marking might be as hazardous as no marking at all, misleading drivers or leaving them without essential guidance. Hence, the task extends beyond simple detection to a nuanced evaluation. This evaluation assesses the clarity, visibility, and integrity of the markings. Over time, a myriad of methods and metrics have been proposed to gauge the quality of these markings[12][4][13][14][15][16]. From simple visual inspections to sophisticated computer vision techniques, the spectrum is vast. Each method brings its own set of advantages, offering precision, scalability, or real-time feedback. However, they also come with limitations, be it in terms of accuracy, applicability to diverse conditions, or dependency on specific equipment. This section will explore this multifaceted landscape, shedding light on the various evaluation techniques, their merits, and areas of potential improvement. In Figures 2.3 and 2.4 some examples of markings with challenging conditions are visible.

Figure 2.3: Types of Road Surface Marking

Figure 2.4: Road Surface Marking with Deterioration and Glare Effects

Figure 2.5: Visual Representation of Segmentation Model Output[29]

2.1.2 Deep Dive into Segmentation Models

Segmentation[17][18][19], in the realm of computer vision, stands as a fundamental technique, especially when the task at hand is the precise detection and evaluation of road surface markings. The objective of segmentation is to categorize each pixel in an image into a specific class, making it an indispensable tool for isolating road markings from other elements within an image. By delineating these markings, segmentation sets the stage for subsequent processes, be it evaluation of marking quality or further analysis. The accuracy of the segmentation process directly impacts the efficacy of downstream applications, underscoring its paramount importance in the context of road safety and urban infrastructure management[20][21][22][23].

The domain of segmentation has witnessed a profound transformation over the years. Beginning with classical image processing models[24][25][26][27] that relied heavily on handcrafted features and thresholding techniques, the field has transitioned to harnessing the power of deep learning. Modern architectures, leveraging vast amounts of data and computational prowess, have surpassed their predecessors in both precision and versatility. This section will traverse this evolutionary path, spotlighting seminal models, landmark research, and the paradigm shifts that have shaped the current landscape of segmentation. From early edge detection methods to sophisticated neural networks, the journey of segmentation models encapsulates the broader evolution of computer vision as a discipline. Given the plethora of segmentation models available, a comparative analysis becomes essential[28]. This section will juxtapose the various models, analyzing their efficacy, precision, and adaptability in the context of road surface marking detection.

Given the rich tapestry of segmentation models that have been proposed over the decades, a methodical comparative analysis is imperative. Different models bring to the table varied strengths, be it in terms of computational efficiency, precision in diverse conditions, or robustness against anomalies. In the specific context of road surface marking detection, certain models may outshine others. This chapter will embark on a detailed exploration, juxtaposing the salient models, scrutinizing their methodologies, and evaluating their performance metrics. The aim is to discern the most apt models for road surface marking detection, considering the unique challenges and requirements of this application. A visual representation of the output of a segmentation model is shown in Figure 2.5.

2.2 Road Surface Marking Detection and Evaluation

2.2.1 Detection Method of Road Marking Using LRF Intensity

The paper titled "Detection method of Road marking using LRF intensity of surface" [12] delves deep into a novel technique for road marking detection using the intensity of a Laser Range Finder (LRF). This study aims to overcome the challenges posed by traditional methods that rely on optical cameras. While optical cameras have been the predominant tool in many road marking detection studies, they have inherent limitations. Specifically, they are vulnerable to external factors such as ambient light conditions, which can impede their ability to capture clear images.

In contrast, the intensity of the Laser Range Finder is largely unaffected by such environmental conditions. Drawing upon this advantage, the researchers created road surface texture images using the refractive intensity of the LRF on surfaces. These images, rich in detail and clarity, serve as the foundation for the subsequent detection process. To identify specific types of road markings, the study utilized a straightforward template-matching technique. This methodology proved particularly effective in recognizing certain road markings, such as the Japanese road marking "Tomare".

A significant highlight of the study's findings is the accuracy achieved in real-world scenarios. When the vehicle's speed was maintained at 30 km/h or less, the method demonstrated a commendable ability to detect the aforementioned road marking. Furthermore, the paper reported an impressive detection rate for other essential road signs. Speed limit signs were detected with an accuracy rate of about 92%, and pedestrian crossing warning signs had a detection rate of approximately 87%.

In essence, this work underscores the potential of using Laser Range Finder intensity as a reliable alternative to optical cameras for road marking detection. By offering a solution less susceptible to environmental interferences, it sets a precedent for future research in this domain, emphasizing the importance of accuracy and adaptability in real-world conditions.

In the study "Detection method of Road marking using LRF intensity of surface", several mathematical models are employed to enhance the detection accuracy of road markings using Laser Range Finder (LRF) technology. The intensity of the laser, given by Equation 2.1, is crucial for determining how environmental factors affect the laser's reach and effectiveness. The reflectivity R of the road markings, calculated using Equations 2.2, helps in assessing the visibility and condition of the markings under different lighting conditions. The template matching score S, derived from sliding a predefined template across the image, is used to pinpoint the location and shape of the markings accurately, as shown in Equation 2.3. Finally, the overall accuracy of the detection method is quantified in Equation 2.4, taking into account true and false positives, to evaluate the method's efficiency and reliability in practical scenarios. These equations collectively form a comprehensive framework for advancing road safety technologies by improving the detection and maintenance of road markings.

Figure 2.6 illustrates the challenge of road surface marking deterioration that the Linear Regression Filter (LRF) method aims to address. The top image shows a well-defined road marking with minimal signs of wear. In contrast, the bottom image depicts a significantly deteriorated road marking characterized by visible cracks, fading, and surface irregularities. This degradation can impair the visibility and effectiveness of road markings, posing a potential hazard for traffic safety. The LRF method focuses on reconstructing and enhancing these degraded

markings by accurately predicting and restoring their original patterns. By analyzing and compensating for the noise and wear present in the markings, the LRF method ensures that road markings maintain their clarity and functionality over time, thereby contributing to safer driving conditions.

Figure 2.7 highlights a significant challenge in maintaining road surface markings, illustrating the types of deterioration that the Linear Regression Filter (LRF) method seeks to mitigate. The image shows a road marking with severe wear and damage, evident from the pronounced cracks and fading of the white stripes. These defects not only reduce the visibility of the markings but also compromise their functional effectiveness, which is critical for guiding drivers and ensuring road safety. The LRF method is designed to tackle such degradation by accurately reconstructing the original appearance of the markings.

$$I(d, \lambda) = \frac{P}{d^2} \cdot e^{-\alpha(\lambda) \cdot d} \tag{2.1}$$

$$R = \frac{I_r}{I_i} \tag{2.2}$$

$$S = \max_{x,y} \left(\frac{1}{N} \sum_{i,j} T(i,j) \cdot I(x+i, y+j) \right) \tag{2.3}$$

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.4}$$

$I(d, \lambda)$ represents Intensity of the laser at distance d and wavelength λ. P is the power of the laser source, and $\alpha(\lambda)$ Attenuation coefficient, which depends on the wavelength λ and describes how the medium absorbs the laser light. S is the score representing the best match of the template in the image, x, y are the position coordinates in the image where the match is calculated, N is the number of elements in template T, $T(i, j)$ is the template used for matching indexed at (i, j), and $I(x + i, y + j)$ is the intensity at position offset by (i, j) from (x, y). TP and FP stand for true positive and false positive respectively

2.2.2 Data Fusion and Unsupervised Learning in Drivable Area Detection

The paper titled "Detecting Drivable Area for Self-driving Cars: An Unsupervised Approach" [16] delves into a critical aspect of autonomous driving - the detection of drivable areas. At the very core of self-driving technology lies the capability to accurately discern where a vehicle can safely navigate. Traditional methods predominantly relied on lane markings for determining these drivable areas. However, such an approach poses limitations, especially when roads lack clear lane markings, which is often the case in many inner-city and rural areas.

The unique challenge presented by these scenarios is the high variability of traffic scenes and lighting conditions. While there have been significant strides in the detection of well-marked roads, the task becomes complex for unlabeled roads. Drawing inspiration from human driving behavior, where the primary instinct is to identify drivable areas[16] before making further navigational decisions, the researchers proposed an unsupervised solution.

Their innovative approach hinges on the fusion of image data from a monocular camera and point cloud[30][31][32][33] data from a 3D-LIDAR scanner[34][35][36]. The integration of these

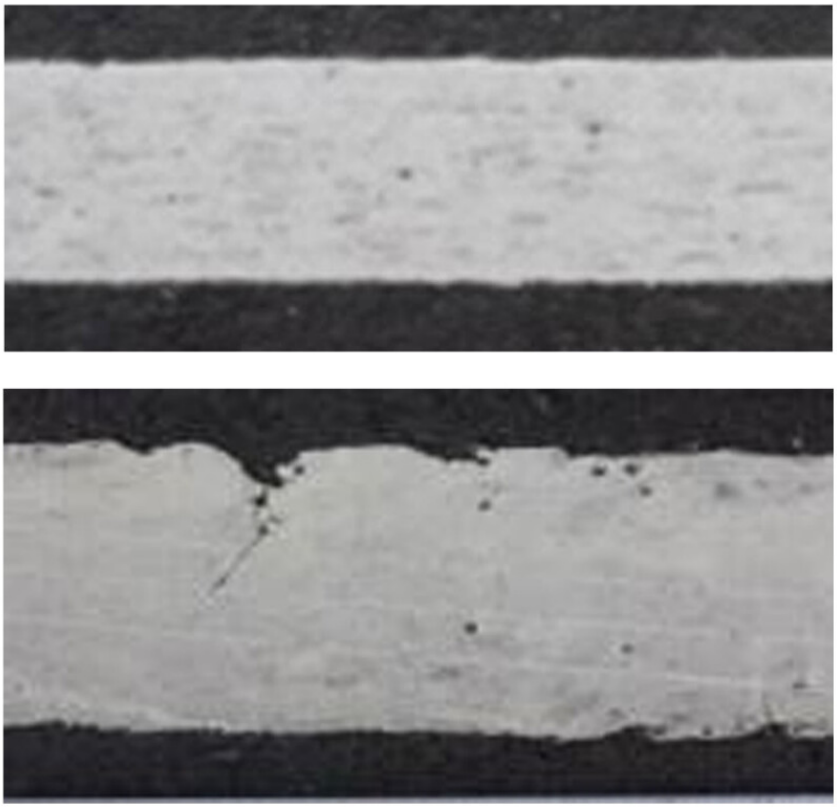


Figure 2.6: Deterioration Challenge Addressed by LRF Method (1)[12]

two data sources facilitates the creation of a "direction ray map," which serves as a foundational element to locate initial drivable areas. A crucial component of their methodology is the fusion at the feature level, enhancing the robustness of their solution. This fusion problem is ingeniously addressed through a Markov network[37][38][39], with a belief propagation algorithm aiding in model inference.

In the broader context of road surface marking detection and evaluation, this paper's methodology underscores the potential of data fusion. While the primary focus is on drivable area detection, the techniques employed are pertinent to the field of road surface marking. The fusion of image and LIDAR data offers a richer and more comprehensive dataset, enhancing detection accuracy. Such an approach can be extended or adapted for detecting road surface markings, especially in complex environments where ambient conditions might obscure these markings.

Furthermore, the paper's emphasis on an unsupervised approach eliminates the need for extensive training datasets, which is a significant advantage. As road conditions and markings vary across regions, an unsupervised method offers flexibility and adaptability. When applied to road surface marking detection and evaluation, such a methodology can pave the way for more accurate, real-time assessments, vital for both autonomous driving systems and urban infrastructure maintenance.

Figure 2.7: Deterioration Challenge Addressed by LRF Method (2)[12]

2.2.3 Convolutional Neural Networks for Road Surface Marking Analysis

The paper titled "Road Surface Marking Recognition and Road Surface Quality Evaluation Using Convolution Neural Network"[40] delves into the application of deep learning, specifically convolution neural networks (CNNs)[41][42][43][44][45], for the nuanced task of road surface marking detection and evaluation.

The existing challenges in the domain of road surface defects classification are explored, underscoring the gaps and limitations in traditional methodologies. Against this backdrop, the paper introduces an automated solution using CNNs, recognized for their prowess in image data processing, offering potential advantages in accuracy and consistency.

The primary focus of the research is the development and fine-tuning of a convolution neural network model tailored for recognizing road surface quality and detecting markings. This CNN-based approach is chosen for its inherent capabilities in efficiently processing and analyzing image data, making it well-suited for the task at hand.

A rigorous assessment of the developed CNN model ensures its proficiency in recognizing road markings and evaluating their quality with high precision. Such an automated methodology is presented as a transformative solution for road organizations, facilitating a more proactive and efficient approach to road surface monitoring.

Incorporating digital image processing with deep learning, the paper highlights the adaptability of CNNs in road surface analysis. This integrated approach promises an autonomous system capable of recognizing objects and assessing their quality, presenting a significant advancement in the field of road surface marking detection and evaluation.

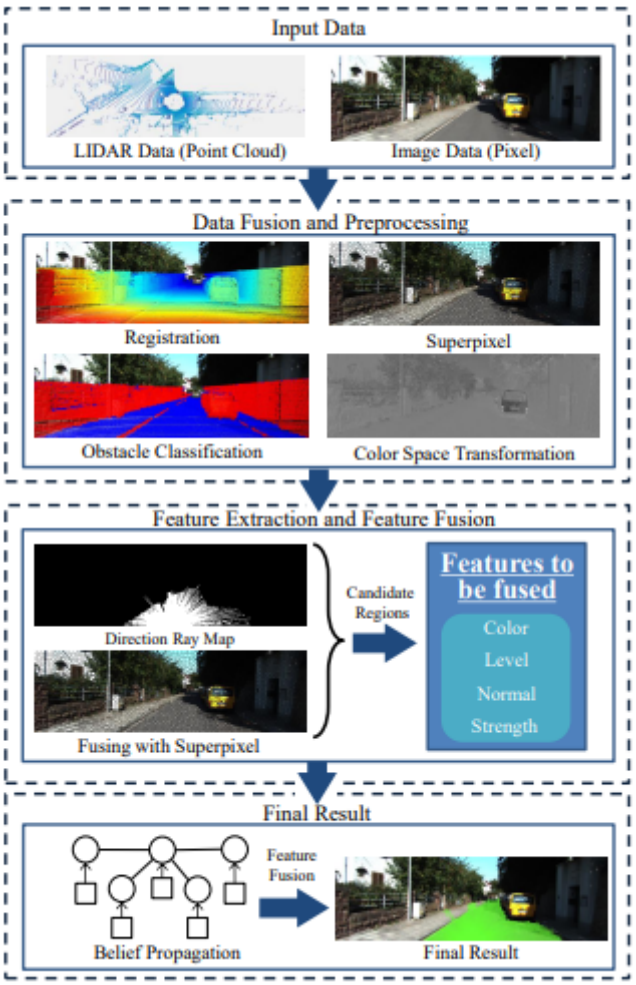


Figure 2.8: Framework Proposed by Data Fusion Approach and Unsupervised Learning in Drivable Area Detection[16]

2.2.4 Machine Vision Technologies for Assessing Road Marking Quality

In the paper titled "An Innovative Road Marking Quality Assessment Mechanism Using Computer Vision"[46], the research delves into the aesthetic quality acceptance of road markings, traditionally evaluated through subjective visual inspections. Recognizing the pitfalls of such subjective evaluations, which can lead to inconsistencies and biases, the study introduces a cutting-edge road marking quality assessment mechanism empowered by machine vision technologies. Here, edge smoothness stands out as a primary quantitative aesthetic indicator. The mechanism operates by capturing and analyzing digital images of the finished road marking surface to discern its geometric characteristics. Based on these characteristics, a scoring system is formulated, which offers a more objective and reliable measure of road marking quality. This research, anchored in machine vision technology, finds its foundation in the technology's successes across various sectors, from manufacturing to construction. In particular, machine vision's ability to provide automated, accurate, and rapid quality control evaluations has proven invaluable in diverse applications, from semiconductor manufacturing to construction automation and intelligent transportation systems.

Meanwhile, in the paper titled "Evaluation of Machine Vision Collected Pavement Marking

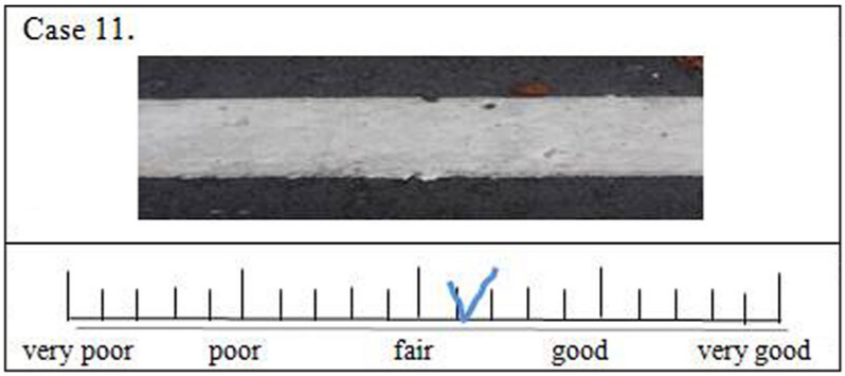


Figure 2.9: Sample of Manual Evaluation Performed by Road Marking Experts [46]

Quality Data for Use in Transportation Asset Management"[47], the authors confront the challenges faced by transportation departments in effectively managing the vast asset of pavement markings. One significant hurdle is the sheer volume of markings coupled with the absence of standardized data collection methods. The paper suggests machine vision technology, especially as integrated within Advanced Driver Assistance Systems (ADAS)[48][49][50][51][52], as a promising solution. These systems, equipped with machine vision cameras, can autonomously assign quality ratings to pavement markings, collecting vast datasets without requiring constant human intervention. This data, once processed, can significantly inform asset management decisions. However, for this data to be truly instrumental, the study investigates the reliability of the quality scores assigned by machine vision under varying conditions and their correlation with established pavement marking evaluation metrics like retro reflectivity, luminance, and contrast. The findings underscore the potential of ADAS machine vision in this realm but also emphasize the need for further research to solidify its applicability across diverse conditions.

2.2.5 AI-driven Assessment for Road Marking Quality

In "AI Driven Road Maintenance Inspection"[53] the focus is directed towards revolutionizing the traditional, labor-intensive road infrastructure maintenance by leveraging advanced artificial intelligence (AI) and computer vision (CV) techniques. Recognizing the substantial costs and delays associated with manual inspections, the paper underscores the potential of AI to automate and streamline the process, consequently reducing the human intervention required for road maintenance checks.

A significant highlight of this work is the application of state-of-the-art CV techniques, specifically object detection and semantic segmentation. These techniques are deployed to detect and evaluate various primary road structures, with a notable emphasis on road surface markings. Such markings play a pivotal role in ensuring road safety and directing traffic. The capability of the AI models to detect and evaluate the quality and state of these markings can significantly impact decision-making processes related to their maintenance and repair.

The methodology proposed is rooted in AI models trained on extensive, commercially viable datasets, further enhanced with proprietary data to ensure real-world applicability. This dual approach ensures that the AI models are both comprehensive in their understanding and precise in their application. Specifically, these models are adept at identifying road surface damages and their respective markings, providing a granular breakdown of their type, extent, and precise geographic locations.

In essence, this research showcases how AI can be a game-changer for road surface marking detection and evaluation. The automation brought about by AI not only promises cost savings but also a heightened accuracy in detecting markings' wear and tear. This advanced detection capability, when compared to traditional manual inspections, offers higher recall rates, ensuring that deteriorated markings are identified promptly and accurately, paving the way for timely maintenance and enhanced road safety.

2.3 Segmentation Models Review

Image segmentation is a foundational and indispensable technique within the realm of computer vision, holding paramount importance in various applications ranging from medical imaging[54][55][56][57][58][59][60][61][62][63] to autonomous vehicles[64][65][66][67][68][69][70][71][72][73]. At its core, the process of image segmentation involves dissecting a digital image into distinct segments, often visualized as "sets of pixels". Each of these individual segments is intended to represent a specific object or a distinct part of an object present within the image.

The rationale behind employing image segmentation is multifaceted. Firstly, it serves to transform the intricate details of an image by breaking it down into more manageable and identifiable regions. This transformation aids in distilling the essence of an image, making it more comprehensible for subsequent analysis. By partitioning an image into these segments, it becomes considerably simpler to focus on specific regions of interest, thereby eliminating potential noise or irrelevant details that might otherwise obfuscate the primary objects or features within the image.

Moreover, the overarching objective of image segmentation is not just partitioning in itself but to metamorphose the visual representation of an image into a format that is both semantically richer and contextually more significant. This enhanced representation proves invaluable for a plethora of applications, as it paves the way for more nuanced interpretations and analyses. For instance, in medical diagnostics, the accurate segmentation of an MRI scan can delineate between healthy tissue and potential tumors, guiding doctors in their diagnoses and treatment plans.

In essence, image segmentation acts as a bridge, translating the raw pixel data of an image into a structured and organized format. This transformation, by compartmentalizing the image into discernible segments, facilitates a deeper understanding and provides a springboard for further computational and analytical tasks in the domain of computer vision and beyond.

2.3.1 Types of Segmentation in Computer Vision

There exists a spectrum of segmentation techniques, each tailored to capture different levels of detail and cater to diverse application requirements. The choice of a specific segmentation approach often hinges on the intricacies of the task at hand.

Semantic Segmentation

Semantic segmentation[74][75][76][77] focuses on assigning a specific class label to every individual pixel in the image. The primary characteristic of semantic segmentation is its lack of distinction between individual instances of the same object within an image. Essentially, it paints a broad brush, categorizing pixels into overarching classes without differentiating between separate occurrences of the same class.

Consider an urban street scene image that has multiple cars, pedestrians, and buildings. Through semantic segmentation, all pixels corresponding to every car in that image would be labeled simply as "car". Similarly, all pixels forming the shape of pedestrians would be labeled "pedestrian", and so on. The process does not distinguish between two different cars or two separate pedestrians; they all fall under the same label umbrella.

Semantic segmentation is often used in scenarios where the primary concern is understanding the kind of objects present in an image rather than their individual instances. It's frequently applied in satellite image analysis[78][79][80][81][82], land cover classification, and some medical imaging tasks.

Instance Segmentation

This segmentation[83][84][85][86][87] method goes a step beyond semantic segmentation. While it assigns class labels to pixels, it also differentiates between individual instances of the same class. Instance segmentation is characterized by its ability to provide a more granular breakdown of images. Not only does it label pixels based on their class, but it also delineates between separate occurrences of the same class. Taking the same urban street scene, instance segmentation would label each car distinctly. So, if there are three cars, it might label them as "car1", "car2", and "car3", allowing for individual identification of each car.

Instance segmentation is particularly beneficial in scenarios where understanding individual instances of objects is crucial. It's commonly used in autonomous driving systems, where distinguishing between different vehicles or pedestrians can be vital for decision-making.

Panoptic Segmentation

Panoptic segmentation[88][89][90][91][92] marries the concepts of semantic and instance segmentation to offer a comprehensive segmentation solution. It provides a holistic view by ensuring every pixel is assigned a class label while also distinguishing between individual object instances. In our urban street scene, panoptic segmentation would label all cars and pedestrians with their respective class labels, but would also distinctly identify each car and pedestrian instance, much like instance segmentation.

Given its comprehensive nature, panoptic segmentation is well-suited for tasks that require both a broad understanding of the scene and detailed insights into individual object instances. It's being increasingly adopted in advanced computer vision applications, including detailed scene understanding and advanced surveillance systems. Figure 2.10 shows an example of Panoptic segmentation output.

2.3.2 Fully Convolutional Network (FCN)

Fully Convolutional Networks[94], often abbreviated as FCN, represent a significant paradigm shift in the approach to image segmentation tasks using deep learning. Before the advent of FCN, most image classification architectures, such as AlexNet or VGG, relied heavily on fully connected layers, which are adept at classifying entire images but struggle with pixel-wise classification essential for segmentation.

FCN ingeniously adapted these classification-centric architectures for the segmentation task by transforming the fully connected layers into convolutional layers, enabling the network to operate on images of varying sizes. This modification was not just a simple replacement but

Figure 2.10: Example of Panoptic Segmentation[93]

a transformative change that endowed the network with the capability to generate spatial heatmaps corresponding to object presence at different locations in the image.

One of the hallmark features of FCN is its upsampling mechanism. After the image undergoes a series of convolutional and pooling layers, reducing its spatial dimensions, FCN employs a deconvolution process to upscale, or 'deconvolve', these feature maps back to the original image size. This process ensures that the output maintains spatial coherence, producing a segmented map that aligns well with the input image.

To further refine this upscaled output and capture intricate details lost during the downsampling process, FCN introduces skip connections. These connections merge feature maps from earlier convolutional layers with the upsampled output. The integration of features from different resolution scales enables the network to make more informed decisions, combining both high-level contextual information and fine-grained details.

FCN's design is versatile, allowing it to be integrated with various backbone architectures. For instance, one can utilize the VGG[95] or ResNet architectures as the foundational structure and append the FCN upsampling layers to achieve segmentation. This modularity has facilitated the adoption and extension of FCN across various domains and applications.

In the grand tapestry of image segmentation, FCN has been a transformative force. It bridged the gap between image classification and segmentation, providing a blueprint for many subsequent architectures. By showcasing how conventional networks can be repurposed and augmented for new tasks, FCN not only enriched the domain of image segmentation but also exemplified the potential of innovative architectural adaptations in deep learning. The general

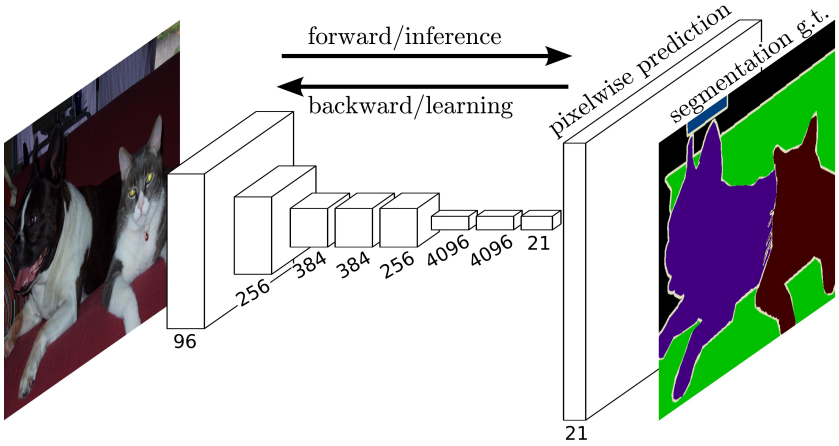


Figure 2.11: Diagram of FCN Model[94]

structure of the FCN model is shown in Figure 2.11.

These equations below serve as the mathematical foundation for FCNs in image segmentation tasks. In Equation (2.5), the transformation of fully connected layers to convolutional layers allows the network to maintain spatial information across the image, which is essential for segmentation. The deconvolution process in Equation (2.6) is pivotal for upsampling, ensuring that the network's output matches the original image dimensions. Lastly, Equation (2.7) highlights how integrating multi-scale contextual information via skip connections enhances the detail and accuracy of the segmentation output. Together, these adaptations facilitate the FCN's ability to effectively perform image segmentation, demonstrating the network's flexibility and robustness in handling varying image sizes and complex segmentation tasks.

$$\mathbf{y} = \mathbf{W} * \mathbf{x} + \mathbf{b} \tag{2.5}$$

$$\mathbf{x}_{\text{upsampled}} = \mathbf{W}_{\text{deconv}} * \mathbf{x} \tag{2.6}$$

$$\mathbf{x}_{\text{refined}} = \mathbf{x}_{\text{upsampled}} + \mathbf{x}_{\text{skip}} \tag{2.7}$$

where $\mathbf{W}$ represents the weights, $\mathbf{x}$ the input, $\mathbf{b}$ the bias, and $*$ denotes the convolution operation.$\mathbf{W}_{\text{deconv}}$ represents the weights of the deconvolution filter, and $\mathbf{x}$ is the input feature map from a lower resolution. And $\mathbf{x}$skip is the feature map from an earlier layer added to the upsampled output $\mathbf{x}$upsampled.

2.3.3 SegNet

SegNet[96] is a robust architecture for handling image segmentation tasks, particularly those involving scene parsing. Originating from the University of Cambridge, SegNet builds upon the established principles of convolutional neural networks (CNNs) but introduces certain unique elements to make it particularly suited for segmentation.

At its core, SegNet is built upon the VGG16 architecture, a well-regarded deep learning model for image classification. However, instead of using the fully connected layers from VGG16, SegNet capitalizes solely on its convolutional layers. This design choice is fundamental to ensuring the network remains spatially aware, a prerequisite for successful segmentation.

The architecture of SegNet can be thought of as having two main components: an encoder and a decoder. The encoder progressively captures the context in the image, while the decoder refines this contextual information to generate a finely segmented output.

In the encoder phase, the input image undergoes a series of convolutions and pooling operations. These operations reduce the spatial dimensions of the image while increasing the depth, encapsulating more intricate and abstract features. As the image progresses through the encoder, it transforms from a high-resolution representation with basic features to a low-resolution one with complex features.

The decoder, on the other hand, is responsible for transforming this low-resolution, high-depth feature map back into a high-resolution segmented map. To achieve this, SegNet employs a series of up-sampling operations. However, instead of using traditional up-sampling techniques, SegNet innovatively uses the pooling indices saved from the encoder's max-pooling steps. By reusing these indices, SegNet ensures that the spatial information lost during the encoding phase is effectively restored during decoding. This results in sharper segmented outputs that align well with the original image's structures.

Another merit of SegNet lies in its efficiency. By leveraging pooling indices for up-sampling and forgoing the need for fully connected layers, SegNet manages to drastically reduce the number of trainable parameters. This makes the network both memory-efficient and computationally economical, allowing it to be deployed even on devices with constrained resources.

In the world of image segmentation, SegNet has established itself as a reliable and efficient choice. Its ingenious use of pooling indices for up-sampling, combined with its foundation on the VGG16 architecture, ensures that it can generate precise segmentation maps while remaining computationally economical. Whether it's for autonomous driving, robotics, or any other domain where understanding the scene at a pixel level is crucial, SegNet stands out as a formidable tool for the job. The general structure of the SegNet model is shown in Figure 2.12.

Equations (2.8), (2.9), and (2.10) elucidate the fundamental operations. In Equation (2.8), convolution layers, augmented by ReLU activation, help in extracting detailed features while preserving non-linearity. The innovative use of pooling indices in Equations (2.9) and (2.10) allows SegNet to efficiently map low-resolution feature data back to high-resolution output without losing critical spatial information. This method not only enhances the quality of the segmentation but also ensures that the network remains lightweight and efficient, suitable for applications where computational resources are limited. These equations underpin the efficacy of SegNet in handling various segmentation tasks with reduced computational overhead.

$$\mathbf{y} = \text{ReLU}(\mathbf{W} * \mathbf{x} + \mathbf{b}) \tag{2.8}$$

$$(\mathbf{y}, \mathbf{I}) = \text{max\_pool}(\mathbf{x}) \tag{2.9}$$

$$\mathbf{x}_{\text{upsampled}} = \text{upsample}(\mathbf{y}, \mathbf{I}) \tag{2.10}$$

In the equations, $\mathbf{W}$ represents the convolutional weights, $\mathbf{x}$ the input to the layer, $\mathbf{b}$ the bias, and ReLU is the activation function that introduces non-linearity, $\mathbf{y}$ is the pooled output, $\mathbf{x}$ is the input feature map, and $\mathbf{I}$ stores the indices of the max values, crucial for later reconstruction in the decoder, $\mathbf{y}$ is the low-resolution feature map from the encoder, and $\mathbf{I}$ are the indices used to guide the placement of these features during upsampling.

2.3.4 Pyramid Scene Parsing Network (PSPNet)

PSPNet[97], which stands for Pyramid Scene Parsing Network, is a sophisticated deep learning architecture tailored for semantic segmentation tasks. It emerged from the recognition that objects and regions in real-world scenes can be vastly different in scale. To address this,

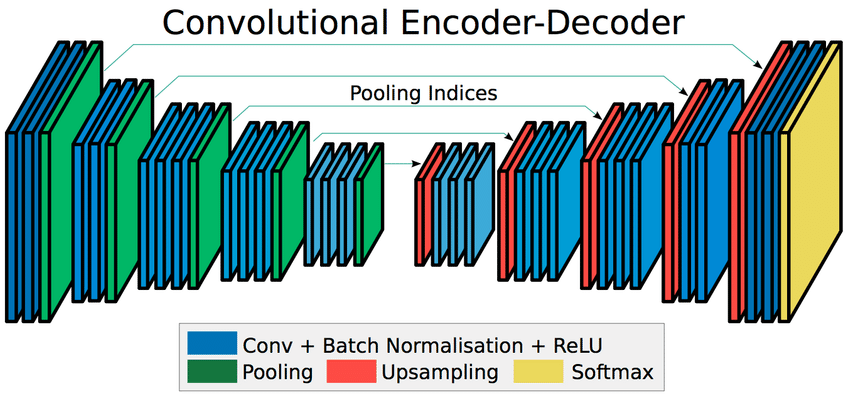


Figure 2.12: Outline of Seg-Net Model[96]

PSPNet introduces a pyramid parsing module to capture different scale information, ensuring a comprehensive understanding of complex environments.

The architecture's foundation is built upon a backbone network, typically a deep convolutional neural network like ResNet, which extracts rich feature maps from input images. While this backbone captures various features, the challenge remains to interpret objects and regions of different sizes accurately.

To tackle this challenge, PSPNet employs a unique pyramid pooling module. This module divides the feature map obtained from the backbone network into different regions, ranging from coarse to fine levels. Each of these regions undergoes pooling operations to capture contextual information at various scales. By pooling over regions of multiple sizes, the network can recognize objects that might be small and detailed, as well as larger, more dominant structures.

Once the pyramid pooling is completed, the output from each level is up-sampled back to the original size and then concatenated. This merged map holds multi-level contextual information, providing a comprehensive understanding of the scene's structure. To generate the final segmentation map, a convolution operation is applied to this concatenated feature map.

PSPNet's effectiveness is also bolstered by an auxiliary loss. During training, an intermediate layer's output is used to compute an auxiliary loss, which is then added to the final loss. This auxiliary pathway acts as a form of deep supervision, helping in stabilizing the training process and enhancing the discriminative capability of the feature maps.

One of the standout achievements of PSPNet is its ability to produce detailed segmentation maps that can discern intricate structures and patterns in images. This finesse is largely attributed to its pyramid pooling module, which ensures that contextual information from various scales is effectively integrated.

In various benchmark tests, PSPNet has consistently showcased superior performance, underlining its prowess in handling semantic segmentation challenges. Whether it's for applications like autonomous driving, where understanding every detail of a scene is paramount, or for medical image analysis, where minute structures can hold significant importance, PSPNet has proven to be a potent tool, adeptly parsing scenes and delivering high-resolution segmentation outputs. The general structure of the PSPNet model is shown in Figure 2.13.

The equations (2.11), (2.12), (2.13), and (2.14) highlight the core processes. Equation (2.11)

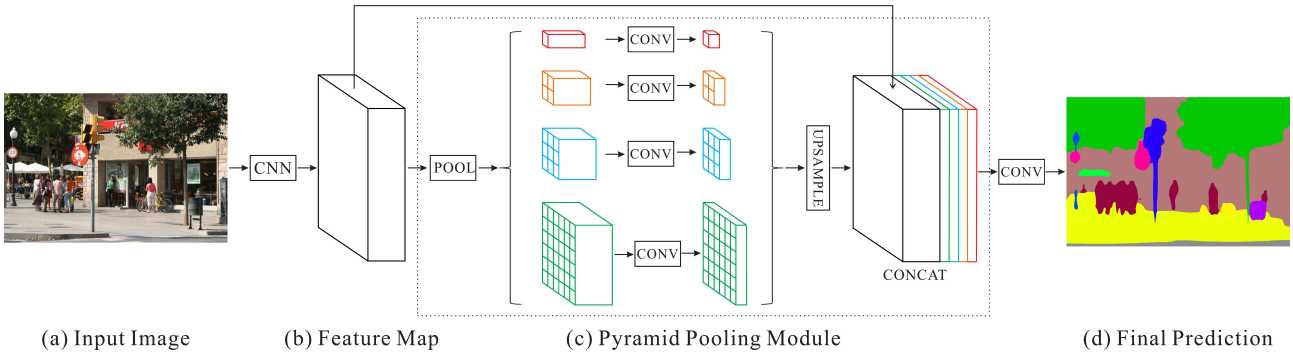


Figure 2.13: Architecture of PSPNet Model[97]

details how deep features are extracted, critical for understanding the content of the image. The pyramid pooling module, represented in Equation (2.12), tackles the challenge of varying object scales by processing the features at multiple resolutions, ensuring that both small and large objects are accurately perceived. The merging of these multi-scale processed features in Equation (2.13) enables the network to make informed segmentation decisions by utilizing comprehensive contextual information. Finally, Equation (2.14) translates these rich features into a precise segmentation map, demonstrating PSPNet's effectiveness in producing detailed and accurate segmentation results essential for applications where fine-grained detail is important.

$$\mathbf{F} = \text{ResNet}(\mathbf{x}) \tag{2.11}$$

$$\mathbf{P}_k = \text{pool}(\mathbf{F}, \text{scale}_k) \tag{2.12}$$

$$\mathbf{C} = \text{concat}(\text{upsample}(\mathbf{P}_1), \text{upsample}(\mathbf{P}_2), \ldots, \text{upsample}(\mathbf{P}_K)) \tag{2.13}$$

$$\mathbf{S} = \text{Conv}(\mathbf{C}) \tag{2.14}$$

In the above equations, $\mathbf{x}$ is the input image and $\mathbf{F}$ is the resulting feature map from the backbone network, $\mathbf{P}_k$ is the pooled feature map at scale k, and scale_k denotes the region size for the k-th level pooling operation, $\mathbf{C}$ is the concatenated feature map that integrates information from all scales, $\mathbf{S}$ represents the final segmentation map.

2.3.5 DeepLabV3 and DeepLabV3+

DeepLabV3[98] and DeepLabV3+[99] are advanced neural network architectures designed for the task of semantic segmentation, aiming to categorize every pixel in an image into predefined classes. Both architectures have built upon their predecessors in the DeepLab series, introducing novel components to enhance segmentation accuracy, particularly in regions with intricate details.

DeepLabV3's primary innovation over its predecessors is the introduction of atrous spatial pyramid pooling (ASPP)[100]. The ASPP module employs parallel dilated convolutions with different dilation rates to capture multi-scale information. By doing this, it can extract features from different scales without losing resolution. This means that objects and regions of varying sizes within the image can be captured and understood more effectively. DeepLabV3 also integrates image-level features into ASPP to capture longer-range information, thereby achieving better object segmentation at multiple scales.

DeepLabV3+ builds upon DeepLabV3 by incorporating an encoder-decoder structure to refine the object boundaries further. The encoder extracts rich semantic features from the input image, while the decoder upsamples these features to produce a dense pixel-wise output, ensuring precise boundary localization. The combination of ASPP with the encoder-decoder structure in DeepLabV3+ makes it adept at capturing both semantic information and detailed boundaries.

The decoder in DeepLabV3+ specifically addresses the challenge of recovering object boundaries, which can often be blurred in the up-sampling process. By introducing skip connections from the lower-level features of the encoder, the decoder can access high-resolution features that are crucial for delineating object boundaries. This results in sharper, more accurate segmentation maps.

In terms of applications, both DeepLabV3 and DeepLabV3+ have shown impressive performance in various domains. From segmenting objects in street scenes for autonomous driving applications to parsing medical images for diagnostic purposes, these architectures have consistently produced state-of-the-art results. Their ability to handle varying scales, combined with precise boundary detection, makes them go-to choices for tasks that require detailed and accurate semantic segmentation. The general structure of the PSPNet model is shown in Figure 2.14.

$$\mathbf{F}_{\text{ASPP}} = \sum_{r \in R} \text{Conv}_{\text{dilated}}(\mathbf{F}, r) \tag{2.15}$$

$$\mathbf{S} = \text{Decoder}(\text{Encoder}(\mathbf{F})) \tag{2.16}$$

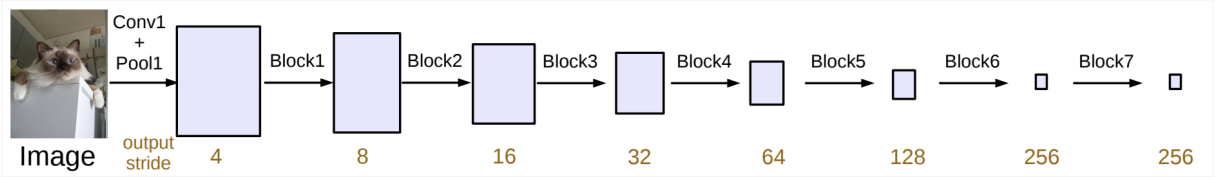
$$\mathbf{S} = \text{Decoder}(\mathbf{F}_{\text{low}} \oplus \mathbf{F}_{\text{high}}) \tag{2.17}$$

In the equations, $\mathbf{F}$ is the input feature map, r is the dilation rate, and R is the set of dilation rates used in ASPP. $\text{Conv}_{\text{dilated}}(\mathbf{F}, r)$ represents the dilated convolution operation with dilation rate r, $\mathbf{S}$ is the output segmentation map, and $\text{Encoder}(\mathbf{F})$ and $\text{Decoder}(\cdot)$ represent the encoder and decoder functions, respectively, $\mathbf{F}$low and $\mathbf{F}$high represent the low and high-resolution feature maps from the encoder, and $\oplus$ denotes the concatenation operation used to merge these features before decoding.

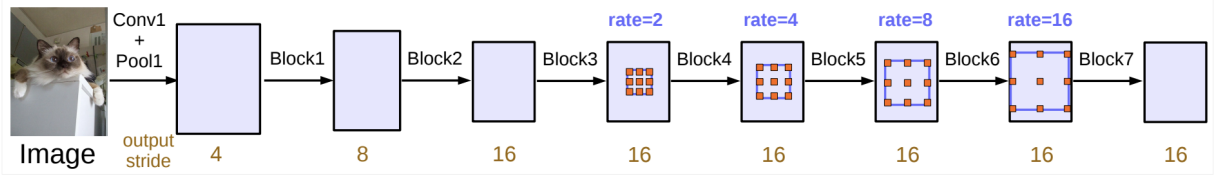
These equations, (2.15), (2.16), and (2.17), encapsulate the core enhancements introduced in the DeepLabV3 and DeepLabV3+ architectures. The atrous convolution in Equation (2.15) allows the model to capture context at multiple scales without increasing the computational burden significantly. Equation (2.16) highlights the refinement process that ensures detailed segmentation by combining deep semantic information with boundary precision. Finally, Equation (2.17) reflects the integration of fine-grained details through skip connections, vital for achieving high-quality segmentation outputs, particularly around the edges of objects. These advancements make DeepLabV3 and DeepLabV3+ highly effective for tasks requiring nuanced understanding of images, exemplifying how architectural innovations can push the boundaries of what's possible in semantic segmentation.

2.3.6 Mask R-CNN

Mask R-CNN[101] is a groundbreaking deep learning model developed for both object detection and instance segmentation. As an extension of the Faster R-CNN framework, which excels at object detection, Mask R-CNN adds a branch for predicting an object mask in parallel with



(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

Figure 2.14: Architecture of DeepLabV3+ Model[99]

the existing branch for bounding box recognition. This architecture allows it to predict both the object's class and its precise pixel-wise mask simultaneously.

The key innovation in Mask R-CNN is the "RoIAlign" layer. In traditional Region of Interest (RoI) pooling, which is used in Faster R-CNN, there's often a misalignment between the extracted features and their original positions due to the quantization of the RoI into discrete spatial bins. This misalignment is fine for bounding box prediction but problematic for pixel-wise mask prediction where spatial precision is crucial. The RoIAlign layer addresses this issue by removing the harsh quantization, maintaining the exact spatial locations, and using bilinear interpolation to compute the exact values of the input features for each spatial position in the output feature map. This ensures that the spatial locations in the output feature maps align perfectly with the input, allowing for more accurate mask predictions.

Another noteworthy aspect of Mask R-CNN is its multi-task loss function. The model is trained using a combined loss that takes into account the classification, bounding box, and mask predictions. This unified approach means that the network learns to optimize all tasks simultaneously, leading to improved performance across the board.

In practice, Mask R-CNN has demonstrated superior performance on a variety of tasks, especially in the domain of instance segmentation. Its ability to detect objects and generate high-quality segmentation masks for each detected object makes it a popular choice in applications that require detailed object understanding. From medical imaging to autonomous driving and even video analysis, Mask R-CNN's combination of precise object detection and instance-level segmentation has set new benchmarks and opened up new possibilities in the realm of computer vision.

$$\mathbf{F}_{\text{roi}} = \text{RoIAlign}(\mathbf{F}, \mathbf{b}) \tag{2.18}$$

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}} \tag{2.19}$$

In the above, $\mathbf{F}$ represents the feature map extracted from the backbone network, and $\mathbf{b}$ is the bounding box coordinates associated with the RoI. RoIAlign$(\cdot)$ performs bilinear interpolation to compute the feature values, preserving the exact spatial locations, L_{cls}, L_{box}, and L_{mask} are the losses for classification, bounding box regression, and mask prediction, respectively. This combined loss helps optimize the network for all tasks concurrently, enhancing overall performance.

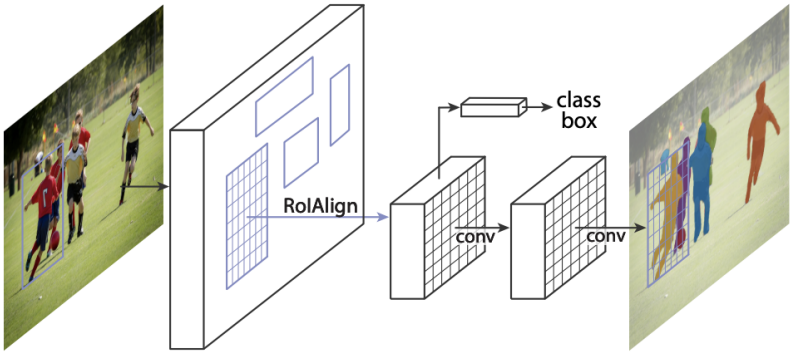


Figure 2.15: General Pipeline of Mask-RCNN Model[101]

These equations, (2.18) and (2.19), encapsulate the operational enhancements introduced in Mask R-CNN that enable precise instance segmentation along with object detection. The RoIAlign layer (Equation (2.18)) corrects the limitations of traditional RoI pooling by aligning the extracted features exactly with the input, crucial for accurate mask generation. The multi-task loss function (Equation (2.19)) exemplifies the integrated training approach that optimizes classification, localization, and segmentation simultaneously, demonstrating how Mask R-CNN harmonizes these tasks to achieve superior segmentation results. This methodology underscores Mask R-CNN's pivotal role in advancing the fields of medical imaging, autonomous driving, and other applications requiring nuanced object segmentation. A depiction of the general pipeline of the Mask-RCNN model is shown in in Figure 2.15.

2.3.7 RefineNet

RefineNet[102] is a multi-path refinement network designed for high-resolution semantic segmentation. It seeks to harness the capabilities of deep convolutional neural networks, which are known for their prowess in handling intricate structures and details. However, due to the pooling or striding operations in these networks, spatial resolution often gets compromised, leading to a loss in detail. RefineNet addresses this challenge by creating a robust architecture that progressively recovers the spatial resolution of the feature maps, ensuring the segmented output maintains fine details.

The architecture of RefineNet is constructed around the idea of linking together the high-level and low-level features at different stages in the network. By doing this, it captures both the semantic information from deeper layers and the fine-grained details from earlier layers. This fusion of information helps in producing a richer representation of the image, which is crucial for segmentation tasks.

A defining characteristic of RefineNet is its "chained residual pooling," which captures background context from a large receptive field. It's achieved by applying multiple pooling operations with different window sizes and fusing the results. This operation ensures that the network considers both local and global contextual information, making it capable of handling complex scenes and varied object scales.

Another intriguing feature of RefineNet is its multi-branch fusion. This involves merging the feature maps from different paths, enhancing the network's ability to integrate multi-scale features. This fusion process is iteratively refined, ensuring that the network learns to focus on

crucial details at every stage.

RefineNet's approach to semantic segmentation offers a fine balance between capturing high-level semantics and retaining low-level details. Its unique architecture and the innovative fusion techniques make it a competitive choice for tasks that require high-resolution segmentation, such as medical image analysis, aerial imagery interpretation, and scene understanding. By effectively bridging the gap between deep semantic understanding and detailed spatial information, RefineNet has set a notable standard in the world of semantic segmentation models. A depiction of the RefineNet model can be seen in Figure 2.16.

$$\mathbf{F}_{\text{crp}} = \sum_{k=1}^{K} \text{Pool}_k(\mathbf{F}) * \mathbf{W}_k \tag{2.20}$$

$$\mathbf{F}_{\text{mbf}} = \sum_{i=1}^{N} \mathbf{F}_i * \mathbf{W}_i \tag{2.21}$$

In the equations, $\mathbf{F}$ is the feature map input to the pooling layer, $\text{Pool}_k(\cdot)$ denotes a pooling operation with a predefined window size at the k^{th} stage, and $\mathbf{W}_k$ are the weights of the convolutional layer applied after pooling, $\mathbf{F}_i$ represents the feature maps from different network paths or scales, and $\mathbf{W}_i$ are the weights associated with the feature maps, facilitating the integration of information across scales.

Equations (2.20) and (2.21) encapsulate the core functionality of RefineNet in refining the segmentation process. The chained residual pooling (Equation (2.20)) allows the network to incorporate a wide range of contextual information by applying successive pooling operations, enhancing the ability to process complex backgrounds and varied object scales. The multi-branch fusion (Equation (2.21)) optimizes the integration of multi-scale features, ensuring that the high-resolution details are preserved while incorporating high-level semantic information. These features collectively enable RefineNet to achieve detailed and precise segmentation, making it suitable for applications requiring fine-grained image analysis such as medical imaging and aerial photo interpretation.

2.3.8 Image Cascade Network (ICNet)

ICNet[103], or Image Cascade Network, is an innovative framework designed to address real-time semantic segmentation tasks, especially catering to the needs of applications that require rapid processing, such as autonomous driving. The principal challenge in real-time segmentation is finding the right balance between speed and accuracy, as deep neural networks, while providing high accuracy, often come at the cost of computational efficiency.

ICNet tackles this dilemma by employing a multi-resolution, cascade structure. The idea is to process the input image at multiple resolutions concurrently, ensuring that each resolution captures features at a different scale. This multi-scale approach allows ICNet to extract both high-level semantic information and fine-grained spatial details from the image.

One of the distinctive features of ICNet is its strategic use of computational resources. The network processes the low-resolution version of the image with more layers to extract high-level semantic features, while the high-resolution version goes through fewer layers to preserve detailed spatial information. By doing so, ICNet ensures that it doesn't expend unnecessary computational power on extracting fine details from deep layers.

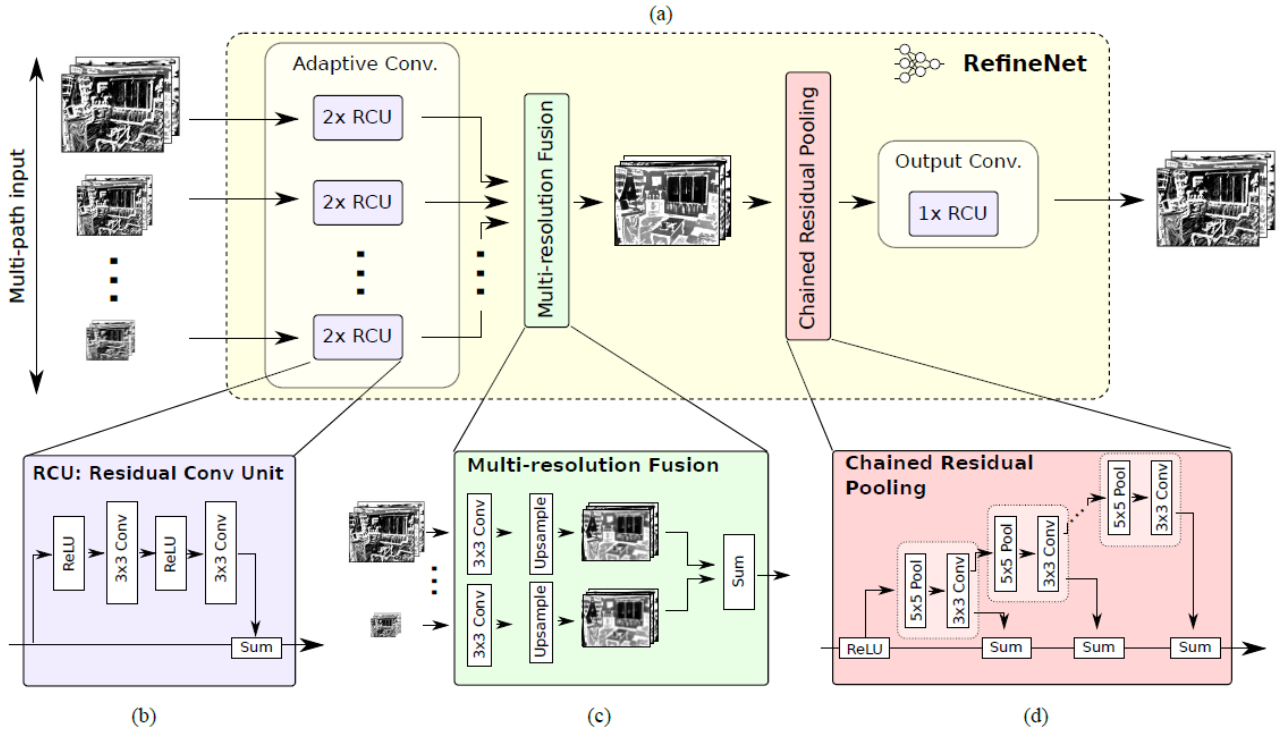


Figure 2.16: RefineNet's architecture uses residual connections with identity mappings for effective gradient propagation, enabling efficient end-to-end training[102].

After processing the image at different resolutions, ICNet fuses the multi-scale feature maps through a cascade feature fusion unit. This fusion process ensures that the final feature representation benefits from the rich semantics of the low-resolution path and the detailed spatial context of the high-resolution path. The cascading mechanism allows for efficient integration of these heterogeneous features, leading to a more comprehensive understanding of the image.

Another advantage of ICNet is its adaptability. It's designed to be flexible, allowing for adjustments based on the computational resources available. This means that ICNet can be scaled down for applications on edge devices with limited processing power or scaled up for tasks that demand higher accuracy and are executed on more powerful systems.

In summary, ICNet's strength lies in its ability to deliver real-time semantic segmentation without compromising significantly on accuracy. Its multi-resolution, cascading approach ensures efficient utilization of computational resources while capturing both semantic and spatial details. This balance between speed and performance makes ICNet an ideal choice for applications that require instant image segmentation, particularly in dynamic environments. Figure 2.17 shows a depiction of the ICNet model.

$$\mathbf{F}_{\text{res}} = \bigoplus_{i=1}^{3} \mathcal{D}_i(\mathbf{I}_i) \tag{2.22}$$

In the above equation, $\mathbf{I}_i$ represents the input image at the i^{th} resolution, $\mathcal{D}_i$ denotes the deep convolutional processing applied to each resolution, and $\bigoplus$ symbolizes the concatenation of features extracted at each level.

$$\mathbf{F}_{\text{fusion}} = \sum_{i=1}^{3} \alpha_i \cdot \mathcal{U}_i(\mathbf{F}_i) \tag{2.23}$$

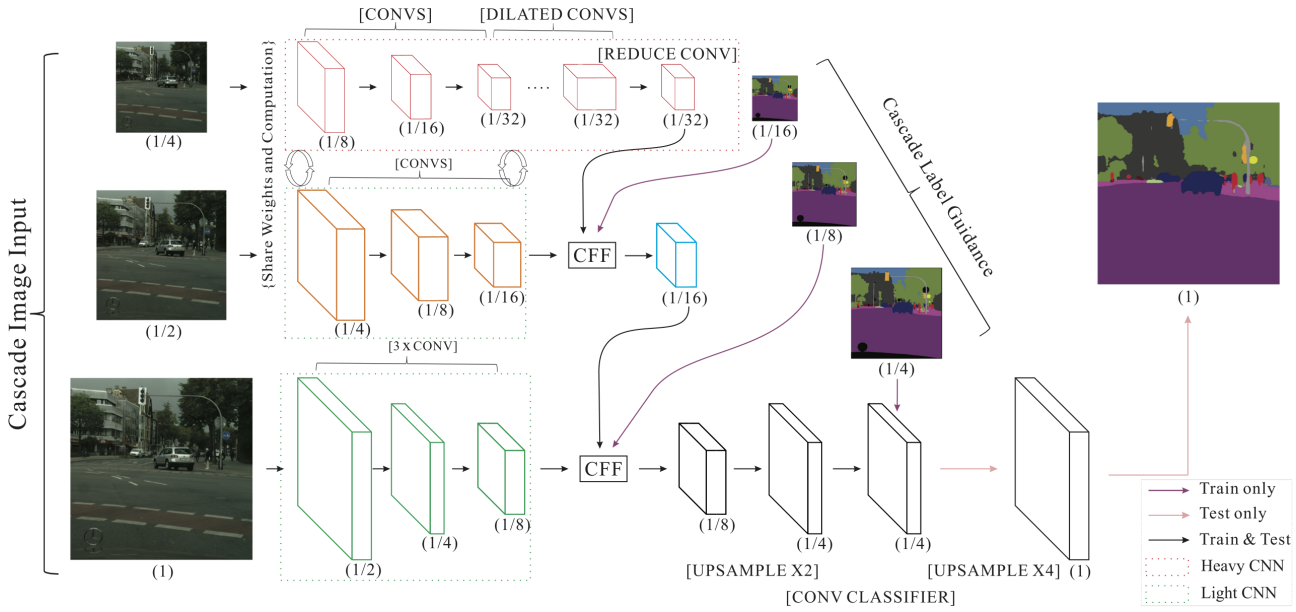


Figure 2.17: ICNet's architecture includes cascade feature fusion (CFF) and uses $\times 4$ upsampling in the bottom branch only during testing, with feature map sizes and operations specified [103].

$\mathbf{F}_i$ are the feature maps from each resolution, $\mathcal{U}_i$ represents up-sampling operations to match the feature dimensions, and α_i are weighting coefficients that adjust the influence of each resolution's features in the final output.

These equations (2.22) and (2.23) capture the essence of ICNet's architecture. The multi-resolution processing (Equation (2.22)) enables ICNet to effectively handle both high-level semantic and low-level spatial details by processing images at varying resolutions. This approach not only enhances the accuracy of the segmentation but also maintains computational efficiency. The cascade feature fusion mechanism (Equation (2.23)) ensures that the segmented output leverages both coarse and fine details, leading to a more accurate and visually coherent segmentation. This method of fusing multi-scale features allows ICNet to adapt to various computational constraints, making it suitable for real-time applications like autonomous driving where rapid processing is crucial.

2.3.9 High-Resolution Network (HRNet)

HRNet[104], or High-Resolution Network, represents a paradigm shift in how convolutional neural networks (CNNs) handle spatial resolutions, especially when it comes to tasks like semantic segmentation. Traditionally, CNNs for segmentation and other visual tasks would downsample an image to extract high-level semantic information and then gradually upsample to make predictions at the original resolution. However, this approach often leads to a loss of fine-grained spatial details, which are crucial for many vision tasks.

Contrary to this standard practice, HRNet maintains high-resolution representations through the entirety of the network rather than downsampling and then later upsampling. The core idea behind HRNet is to construct high-to-low resolution subnetworks simultaneously, capturing multi-resolution features at each level. Instead of degrading the spatial resolution during the initial layers, HRNet keeps the high-resolution representation and extracts features in parallel across various resolutions.

As the network progresses, HRNet employs a series of cross-resolution fusion modules. These modules allow for the exchange of information between different resolutions, ensuring that each parallel branch benefits from features captured at other scales. This continuous multi-resolution fusion is key to HRNet's success, as it means the high-resolution branch can leverage the semantic richness of the lower-resolution branches and vice versa.

The outcome is that, by the end of the network, HRNet produces a high-resolution output that has been continually enriched by multi-scale features throughout the network's depth. This approach ensures that the final feature representation is both semantically rich and spatially detailed.

One of the main advantages of HRNet is its ability to produce more precise and detailed segmentation masks, making it particularly well-suited for tasks where fine-grained spatial accuracy is essential. Additionally, the architecture is versatile and can be applied to a variety of vision tasks beyond segmentation, such as object detection and pose estimation.

In essence, HRNet challenges the conventional wisdom in deep learning architectures by maintaining high-resolution pathways throughout the network. This unique approach, combined with continual cross-resolution fusion, allows HRNet to achieve state-of-the-art performance in tasks that demand a delicate balance between semantic understanding and spatial precision.

$$\mathbf{F}_{\text{high}}, \mathbf{F}_{\text{med}}, \mathbf{F}_{\text{low}} = \text{HRNetSubnets}(\mathbf{I}) \tag{2.24}$$

In the equation, $\mathbf{I}$ is the input image, $\mathbf{F}$high, $\mathbf{F}$med, and $\mathbf{F}_{\text{low}}$ are the feature maps at high, medium, and low resolutions, respectively, produced by separate subnetworks within HRNet.

$$\mathbf{F}_{\text{fusion}} = \sum_{i=\text{high, med, low}} \beta_i \cdot \mathcal{F}_{\text{cross}}(\mathbf{F}_i) \tag{2.25}$$

$\mathcal{F}_{\text{cross}}$ denotes the cross-resolution fusion operation, and β_i are coefficients that adjust the contribution of each resolution's features in the final fusion output.

These equations (2.24) and (2.25) encapsulate HRNet's innovative approach to handling resolutions. The network structure (Equation (2.24)) avoids the common pitfalls of downsampling by establishing parallel streams for different resolutions, thus preserving spatial details while capturing multi-scale semantic information. The fusion process (Equation (2.25)) ensures that features from all resolutions are combined effectively, enhancing the final segmentation output with both detail and context.

In the context of HRNet, $\mathbf{F}$high, $\mathbf{F}$med, $\mathbf{F}_{\text{low}}$ represent feature sets at varying resolutions, crucial for capturing detailed to broad semantic features. β_i coefficients in (2.25) optimize the influence of each resolution's contribution, ensuring a balanced feature integration that supports precise and detailed segmentation outputs. This methodical fusion of multi-scale features positions HRNet as an exceptional model for complex segmentation tasks where both high-level semantics and fine-grained spatial accuracy are required.

2.3.10 Fast-SCNN

Fast-SCNN[105], short for Fast Semantic Segmentation Convolutional Neural Network, is a deep learning architecture tailored for efficient semantic segmentation. While many state-of-the-art segmentation models offer impressive accuracy, they often come with a computational cost,

making real-time applications on edge devices challenging. Fast-SCNN addresses this challenge by delivering a fast and efficient segmentation solution without significantly compromising on performance.

The architecture of Fast-SCNN is designed with three main modules: the Learning-to-Downsample module, the Global Feature Extractor, and the Feature Fusion module.

The Learning-to-Downsample module is the initial stage of the network and is responsible for reducing the resolution of the input image. Instead of using conventional pooling layers that might lead to information loss, this module employs convolutional layers with strides to downsample the image, capturing essential features in the process.

The Global Feature Extractor is the heart of the Fast-SCNN. It utilizes depthwise separable convolutions, which are computationally efficient compared to standard convolutions. This module captures high-level semantic information from the downsampled image. The depthwise separable convolutions, along with bottleneck layers and pyramid pooling, allow the network to extract rich features with fewer parameters, speeding up the computation.

Lastly, the Feature Fusion module merges the high-resolution features from the Learning-to-Downsample module with the semantically rich features from the Global Feature Extractor. This fusion ensures that the final segmentation map retains both the spatial details and the semantic context. To achieve this, the module employs a combination of depthwise convolutions and pointwise convolutions, further contributing to the model's efficiency.

One of the standout features of Fast-SCNN is its speed, especially when it comes to inference. The model's design choices, such as the use of depthwise separable convolutions and the efficient fusion of multi-resolution features, allow it to operate much faster than many other segmentation networks with similar accuracy. This makes Fast-SCNN particularly suitable for real-time applications on devices with limited computational resources.

In summary, Fast-SCNN offers a harmonious blend of speed and accuracy in the realm of semantic segmentation. Its modular architecture, combined with efficient convolution techniques and feature fusion, allows it to deliver high-quality segmentation results in real-time, opening doors for a plethora of applications, especially on edge devices. A depiction of the Fast-SCNN model is seen in Figure 2.18.

$$\mathbf{F}_{\text{downsample}} = \mathcal{C}_{\text{stride}}(\mathbf{I}, W, b) \tag{2.26}$$

$\mathbf{I}$ in the eauation is the input image, $\mathcal{C}_{\text{stride}}$ denotes a convolutional operation with a stride greater than 1 (to reduce dimensionality), and W and b represent the weights and biases of the convolutional filters.

$$\mathbf{F}_{\text{global}} = \mathcal{D}(\mathbf{F}_{\text{downsample}}, W_{\text{depth}}, W_{\text{point}}) \tag{2.27}$$

$\mathcal{D}$ represents the depthwise separable convolution operation, W_{depth} is the depthwise convolutional weights, and W_{point} is the pointwise convolutional weights.

$$\mathbf{F}_{\text{fusion}} = \mathcal{F}_{\text{fusion}}(\mathbf{F}_{\text{downsample}}, \mathbf{F}_{\text{global}}) \tag{2.28}$$

$\mathcal{F}_{\text{fusion}}$ denotes the feature fusion operation that integrates high-resolution and semantically rich features.

Equations (2.26), (2.27), and (2.28) articulate the principal operations within Fast-SCNN, showcasing its efficient design strategy. Equation (2.26) reduces the spatial resolution while

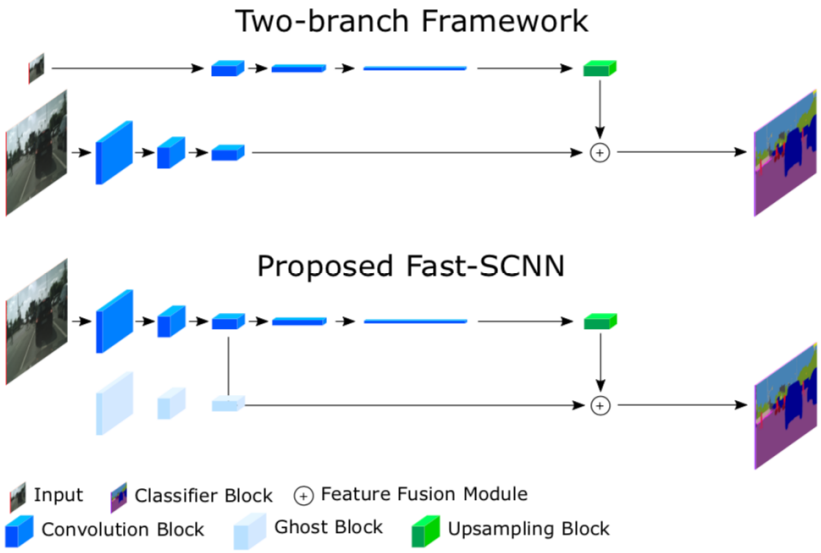


Figure 2.18: Schematic comparison of Fast-SCNN with encoder-decoder and two-branch architectures highlights distinctive approaches. The encoder-decoder utilizes multiple skip connections across various resolutions, typically emerging from deep convolutional blocks. In contrast, two-branch methods integrate global features from low-resolution inputs along with shallow spatial details. Fast-SCNN simultaneously captures spatial details and the initial layers of global context through the learning-to-downsample module [105].

preserving essential details through strided convolutions. Equation (2.27) enhances the semantic richness of the features with minimal computational resources by employing depthwise separable convolutions. Lastly, Equation (2.28) merges detailed and high-level semantic features to produce a comprehensive output.

In the Fast-SCNN's structure, **F**downsample and **F**global represent low and high-level feature sets, crucial for capturing basic to complex image features. The fusion process (Equation (2.28)) optimizes the use of these features, ensuring that the final segmentation maps are both accurate and detailed, making Fast-SCNN an ideal choice for real-time applications where both efficiency and performance are paramount.

2.4 Uncertainty Aware Regression

Uncertainty-aware regression[106][107][108] is an advanced approach in machine learning that acknowledges and quantifies the inherent uncertainties in predictions. Traditional regression models focus on predicting a single output value for each input. However, these models often overlook the uncertainty associated with their predictions. Uncertainty-aware regression addresses this by providing not just a prediction but also a measure of confidence in that prediction.

2.4.1 Types of Uncertainty

Aleatoric Uncertainty[109][110][111][112][113]: This type of uncertainty arises from the randomness inherent in the data itself. For instance, noise in the data or uncontrolled experimental conditions can lead to aleatoric uncertainty. It's inherent and irreducible.

Epistemic Uncertainty[114][115][116][117][118]: This is due to the limitations in our knowledge or the model's structure. It arises from a lack of data or an overly simplistic model and can be reduced by collecting more data or using a more complex model.

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{2.29}$$

In Equation (2.29), y is the observed output, $f(x)$ is the true function, and ϵ represents normally distributed noise with variance σ^2.

$$p(y|x, \mathcal{D}) = \int p(y|x, \theta)p(\theta|\mathcal{D})d\theta \tag{2.30}$$

$p(y|x, \theta)$ is the likelihood of y given x and model parameters θ, and $p(\theta|\mathcal{D})$ is the posterior distribution of θ given the data $\mathcal{D}$.

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta) \tag{2.31}$$

$p(\mathcal{D}|\theta)$ is the likelihood of the data given the parameters and $p(\theta)$ is the prior distribution of the parameters.

$$p(y|x, \mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^{T} p(y|x, \hat{\theta}_t) \tag{2.32}$$

$\hat{\theta}_t$ represents the parameters at the t-th forward pass with dropout, and T is the total number of stochastic forward passes.

$$y_*|x_*, \mathcal{D} \sim \mathcal{N}(\mu(x_*), \sigma^2(x_*)) \tag{2.33}$$

y_* is the predicted output for a new input x_*, and $\mu(x_*)$ and $\sigma^2(x_*)$ are the mean and variance predicted by the GP.

In the following equations provided:

- $H[y|x, \mathcal{D}]$ represents the entropy of the predictive distribution, providing a measure of uncertainty in predictions for input x given the dataset $\mathcal{D}$. $p(y|x, \mathcal{D})$ is the predictive probability distribution of the output y given input x and dataset $\mathcal{D}$.
- $\text{Var}[y|x, \mathcal{D}]$ is the variance of the predictive distribution which quantifies the spread of the predicted values around the mean, indicating the level of uncertainty. $\mu(x)$ is the mean of the predictive distribution for input x.
- $I(\theta, y|x, \mathcal{D})$ is the mutual information between the model parameters θ and the output y, which quantifies the reduction in uncertainty about y obtained by knowing θ. $E_{p(\theta|\mathcal{D})}[H[y|x, \theta]]$ is the expected entropy of y given x and parameters θ, averaged over the posterior distribution of θ.
- σ_{pred} is the predictive standard deviation, providing a measure of the expected error or uncertainty in the model's predictions, calculated as the square root of the predictive variance.
- CI represents the confidence interval, providing a range within which the true value of y is expected to fall with a certain level of confidence (typically 95% in this context).

The interval is calculated using $\mu(x)$, the mean prediction, and σ_{pred}, the predictive standard deviation, scaled by 1.96 which corresponds to the 95% confidence level under the assumption of normal distribution.

These parameters ($\mu(x)$, σ_{pred}, etc.) are fundamental in understanding how uncertainty is quantified and managed within predictive models, enhancing their reliability and usefulness in practical applications where decision-making under uncertainty is critical.

$$H[y|x, \mathcal{D}] = -\int p(y|x, \mathcal{D}) \log p(y|x, \mathcal{D}) dy \tag{2.34}$$

$$\text{Var}[y|x, \mathcal{D}] = \int (y - \mu(x))^2 p(y|x, \mathcal{D}) dy \tag{2.35}$$

$$I(\theta, y|x, \mathcal{D}) = H[y|x, \mathcal{D}] - E_{p(\theta|\mathcal{D})}[H[y|x, \theta]] \tag{2.36}$$

$$\sigma_{\text{pred}} = \sqrt{\text{Var}[y|x, \mathcal{D}]} \tag{2.37}$$

$$CI = [\mu(x) - 1.96 \cdot \sigma_{\text{pred}}, \mu(x) + 1.96 \cdot \sigma_{\text{pred}}] \tag{2.38}$$

Equations ((2.29) to (2.38)) cover the mathematical foundation of uncertainty-aware regression, detailing how both aleatoric and epistemic uncertainties can be quantified and incorporated into predictive models. They provide a comprehensive toolkit for addressing the inherent uncertainties in data and model predictions, enhancing the reliability and interpretability of machine learning models in critical applications.

Importance in Predictive Modeling

In predictive modeling, acknowledging uncertainty is crucial. It allows for more reliable and trustworthy predictions, especially in critical applications like medical diagnosis, financial forecasting, and autonomous vehicles. By quantifying uncertainty, models can indicate when their predictions are less reliable, guiding users to proceed with caution.

2.4.2 Implementing Uncertainty-Aware Regression

Bayesian methods are a cornerstone in implementing uncertainty-aware regression. They involve updating the belief about the model's parameters based on prior knowledge and observed data. Bayesian Neural Networks (BNNs), for example, offer a probabilistic interpretation of neural networks, where weights are treated as distributions rather than fixed values.

Deep learning models, particularly those incorporating dropout layers, can be adapted for uncertainty estimation. Techniques like Monte Carlo Dropout allow models to express uncertainty by performing multiple forward passes with dropout enabled during inference, leading to a distribution of outputs.

Gaussian Processes (GPs) offer a robust way to model uncertainty. They provide a probabilistic framework where predictions are made in the form of probability distributions rather than point estimates. This is especially useful in scenarios where data is sparse, and the model's confidence in predictions varies across the input space.

2.4.3 Real-World Applications

In medical diagnosis, uncertainty-aware models can indicate the confidence level in identifying pathologies, aiding clinicians in decision-making[119][120][121][122][123]. In finance, models that quantify uncertainty in market predictions can be valuable for risk assessment[124] [125] [126] [127] [128]. For self-driving cars, understanding the uncertainty in object detection and trajectory prediction is critical for safe navigation[129][130][131][132][133].

2.4.4 Challenges and Future Directions

Implementing these models, especially Bayesian approaches, can be computationally intensive. Optimizing algorithms for efficiency is an ongoing challenge. In domains with limited data, accurately estimating uncertainty becomes challenging. Making the uncertainty estimates interpretable and actionable for end-users is a key area of research.

Finally, this study should make clear that Uncertainty-aware regression represents a significant advancement in the realm of predictive analytics. By integrating the quantification of uncertainty into regression models, one does not obtain only predictions but insights into the reliability of these predictions. This approach is essential in making informed decisions in various high-stakes domains. The ongoing research and development in this field promise to enhance the robustness and reliability of predictive models, contributing significantly to numerous sectors relying on data-driven decision-making.

Chapter 3

Semantic Segmentation of Traffic Landmarks Using Classical Computer Vision and U-Net Model

3.1 Introduction

This chapter delves into the initial exploratory phase of this research, where the primary objective was to establish a robust detection and segmentation framework for road markings using advanced computational techniques. The challenge was twofold: not only did we need to accurately detect the road markings but also segment them effectively from their surrounding environments in high-resolution images. This task was approached by integrating classical computer vision methods with cutting-edge deep learning techniques, specifically Convolutional Neural Networks (CNNs), which are renowned for their efficacy in handling complex image analysis tasks.

The application of CNNs in this context was driven by their superior performance in image segmentation tasks, significantly surpassing the capabilities of traditional computer vision algorithms. CNNs excel in extracting and learning features from images automatically, which is crucial for distinguishing subtle differences in texture and color that characterize different objects within an image. In the approach, this study employed a method focusing on the 'Smoothness' of color variations—a key attribute of road markings which tend to have less color variability compared to their immediate surroundings. This attribute was critical in developing an algorithm that could isolate these markings from other elements in the images effectively.

To facilitate the development and training of our CNN model, specifically a U-Net architecture with a VGG-16 backbone, we encountered the challenge of the absence of a readily available dataset specifically tailored for road marking detection. To overcome this, we innovated a semi-assisted labeling tool, leveraging the algorithm developed to segment road markings initially. This tool enabled us to efficiently annotate a set of 182 high-resolution images, which formed the basis for training our model.

The process of annotating these images was meticulous and involved manually adjusting the labels to ensure high accuracy and relevance of the training data. This dataset not only provided a solid foundation for our initial experiments but also served as a crucial testbed for refining

Figure 3.1: Example of Eroded Road Marking

our segmentation model. The performance of the U-Net model, assessed through the Dice coefficient—a statistical tool that measures the similarity between the model output and the ground truth labels—reached an encouraging 78.90% on the validation set.

These promising results not only demonstrated the potential of using CNNs for the task of road surface marking detection but also set the stage for subsequent phases of the research. The insights gained from this initial phase underscored the feasibility of using deep learning models to automate and enhance the precision of road surface marking evaluations, thereby contributing to safer driving conditions and more effective traffic management systems. This chapter will expand further on the specific methodologies employed, the challenges encountered, and the strategic innovations that were developed to address the complexities of this task.

3.2 Dataset

The foundation of any effective road sign detection algorithm hinges on the availability and quality of the dataset used for training and testing the models. In this study, we were fortunate to collaborate with the Mie prefecture local authorities who provided us with a substantial dataset comprising approximately 13,000 high-resolution images. These images, as depicted in Figures 3.1 and 3.2, encompass a wide geographical range, capturing both urban environments and less populated outskirts, thus ensuring a diverse representation of road scenarios.

The variability in the dataset is crucial for developing robust models capable of functioning under varied real-world conditions. Not only do these images exhibit a range of road markings in differing states of wear and tear—some of which are significantly deteriorated as shown in Figure 3.1 but they also present these landmarks in varying lighting and weather conditions. This diversity is essential for training algorithms to be reliable under different operational circumstances. For instance, some images capture the challenging glare effect caused by sunlight reflecting off the road surface, as illustrated in Figure 3.2. This specific condition is particularly challenging for vision-based systems due to the high contrast and potential for obscuring visible road signs.

Despite the comprehensive nature of the dataset in terms of image diversity, it initially lacked pixel-wise annotations necessary for training semantic segmentation models. Semantic segmentation requires detailed annotations that outline the exact pixels of each class within an

Figure 3.2: Example of Glare Effect on Roads

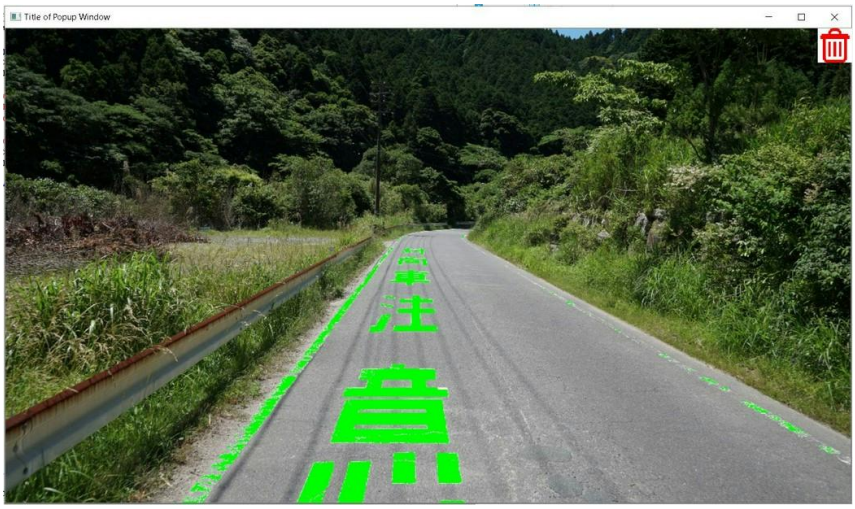


Figure 3.3: Example of Data Sample Opened with Labeling Software

image— in this case, the specific road signs. To address this gap, we embarked on a semi-assisted labeling process. This process was facilitated by a computer vision method developed specifically for this project, which was capable of generating approximate annotations automatically. These preliminary annotations provided a baseline from which further refinements could be made manually, ensuring higher accuracy.

The semi-assisted labeling process began with the application of the developed computer vision method to initially identify and delineate road signs from the surrounding environment in the images. This preliminary step was crucial as it reduced the manual effort required and significantly sped up the annotation process. Once these initial annotations were in place, they were meticulously reviewed and adjusted by human annotators to correct any inaccuracies and to add finer details that the automated method might have missed. This hybrid approach leveraged the speed and efficiency of automated systems while maintaining the accuracy and reliability of human oversight. Figure 3.3 shows the labeling tool used. Furthermore, each image in the dataset was annotated not just for the presence of road signs but also included information about the number, type, and quality of these signs among other variables. This enriched dataset provided a multidimensional view of the road signs, offering insights not just into their appearance but also their condition and categorization. This detailed level of annotation is

invaluable for training sophisticated models that need to understand and evaluate the quality of road signs, not just detect their presence.

The preparation and refinement of such a dataset are critical undertakings in the development of machine learning models for real-world applications. By addressing both the variety of conditions under which road signs appear and the detailed annotation of these signs, this dataset serves as a foundational pillar for building and testing algorithms that can reliably perform road sign detection and segmentation under diverse environmental conditions. This robust dataset preparation thus sets the stage for the subsequent phases of model training and evaluation, which are geared towards achieving high accuracy and reliability in road sign detection—an essential step towards enhancing road safety and traffic management through automated systems.

3.3 Computer Vision Approach

Detecting road surface markings efficiently necessitated a nuanced approach that integrated a variety of computer vision techniques, each selected for its ability to address specific challenges presented by the dataset. Initially, our methods were grounded in traditional computer vision techniques, which are adept at extracting distinct features from images but can struggle with variability in environmental conditions.

One of the first techniques we employed was pixel-wise K-means clustering, a method particularly useful for images where color was the predominant feature distinguishing the road markings. K-means clustering works by partitioning the image pixels into clusters based on color similarity, thereby grouping similar pixels together. This method proved effective in environments with uniform lighting but faced challenges under more complex lighting conditions, such as varying shadows or when other objects in the image had colors similar to the markings. These limitations are demonstrated in Figures 3.5 and 3.4, where the effectiveness of K-means clustering diminishes in non-uniform lighting conditions.

Figure 3.4 showcases a successful application of the K-means clustering algorithm for detecting road markings. In this example, the algorithm has accurately segmented the road markings, which are clearly distinguished from the surrounding environment. The white lines, arrows, and pedestrian crossing are correctly identified as part of the cluster with the highest mean, highlighting the road markings with precision. This successful detection demonstrates the potential of K-means clustering in identifying road-related features when the algorithm parameters and preprocessing steps are appropriately configured. The high-contrast output effectively separates the markings from the background, ensuring that the essential road guidance elements are prominently visible.

Figure 3.5 depicts the result of using the K-means clustering algorithm for segmenting an urban street scene, with the intention of detecting road markings. However, this method has shown only limited success in this context. While K-means has identified various regions within the image, it has also mistakenly classified many non-road marking objects as road markings. This misclassification includes buildings, vehicles, and other urban features, leading to a significant number of false positives. The high-contrast segmentation was intended to isolate the road markings, but the algorithm's inability to distinguish between similar visual features has compromised its accuracy.

For images where the shape or geometry of the markings was more distinct, we employed the

Figure 3.4: Output of K-means and Pixels Selection of Cluster with Highest Mean

Figure 3.5: Example of Segmentation Result Using K-means

Figure 3.6: Removing Top Half of each Image

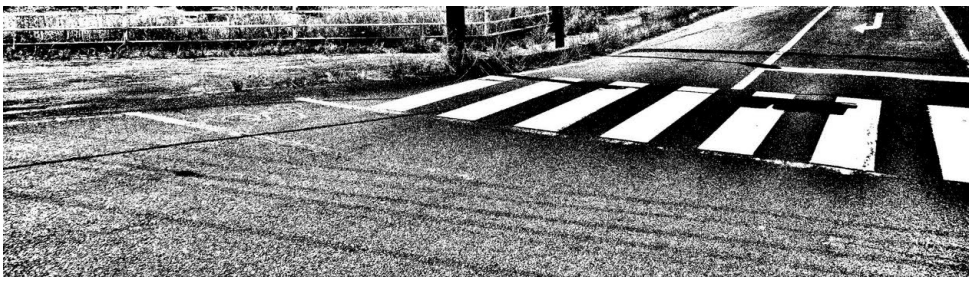


Figure 3.7: The result after applying a small blurring effect followed by a Gaussian thresholding filter, which still left a lot of unwanted noise.

Hough Lines algorithm. This technique is particularly useful for detecting linear patterns and was instrumental in identifying and delineating the linear aspects of road markings. The Hough Lines method transforms points in a Cartesian plane to a parameter space, where straight lines can be more easily identified through accumulative analysis. However, like K-means clustering, Hough Lines has its limitations, especially in cluttered scenes or where the linearity of the road markings is interrupted by cracks, wear, or occlusion.

To create a more robust detection system, we integrated these techniques within a broader strategy that also incorporated a novel parameter we named "Smoothness." This parameter was pivotal in distinguishing road markings from the background based on the consistency of the color distribution within specific areas of the image. Our approach began with preprocessing steps to focus on the relevant sections of the image and reduce noise. Recognizing that road markings are primarily located in the lower half of road images, we removed the top half of each image to reduce processing redundancy and to concentrate our algorithms on the most likely regions where markings would be found. This process is shown in Figure 3.6.

Following the cropping of the images, we applied basic noise reduction filters. These filters helped to smooth out the image and reduce the impact of granular noise that could complicate the detection process. With a cleaner image, we then applied Jarvis's wrapping algorithm, a technique designed to encapsulate distinct regions within an image. This algorithm was particularly effective at isolating potential road marking areas from the less relevant sections of the image.

After isolating these key regions, we computed the Smoothness parameter. This metric evaluated the uniformity and variation of color distribution within each delineated area. By setting an experimentally determined threshold for Smoothness, we were able to effectively separate the regions of interest—those likely containing road markings—from the surrounding background. This step was critical as it allowed for the accurate and efficient detection and segmentation of road markings, setting the stage for subsequent semantic segmentation and quality analysis. This process is shown in Figure 3.7 to Figure 3.10. The integration of these varied noise

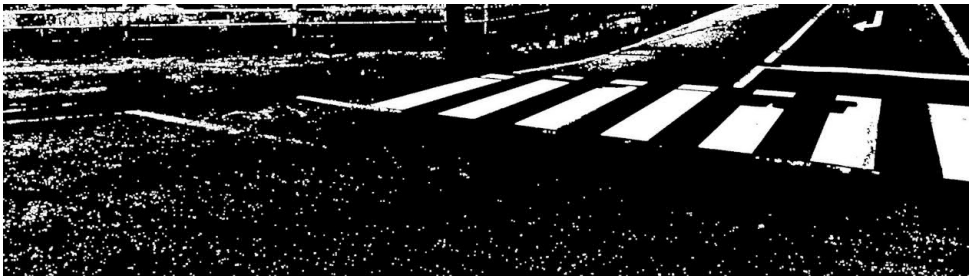


Figure 3.8: The results obtained after applying erosion and dilation, brings us close to obtaining a final result in this example, but it would work as a general solution.

Figure 3.9: Applying the Jarvis algorithm outputs' contours which are sets of pixels delimiting a certain cluster of pixels in the image, for each contour we will compute the "Smoothness".

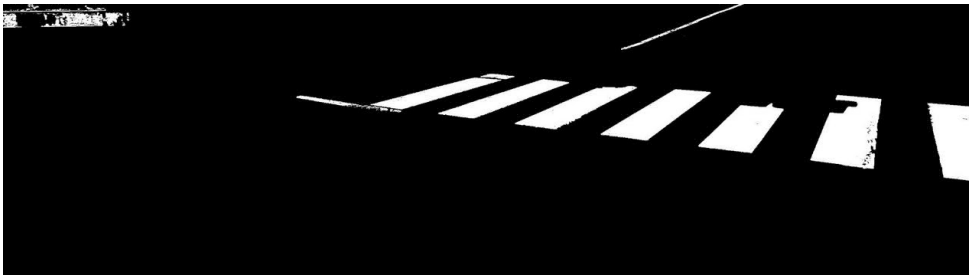


Figure 3.10: After computing the "Smoothness" parameter and filtering out the results to keep only the contours that scored the highest, we obtain this final result.

filtering techniques, and the novel Smoothness parameter—into a cohesive detection strategy underscores the complexity and necessity of using a multi-faceted approach in the automated detection of road surface markings. By adapting and combining these methods, we were able to address the diverse challenges posed by the dataset, from variable lighting conditions to the presence of similar colors and complex shapes within the images. This comprehensive approach not only enhanced the accuracy of our detection system but also demonstrated the potential for advanced computer vision techniques to significantly improve the reliability and efficiency of road surface markings analysis in real-world conditions.

3.3.1 Noise Minimization

To ensure the precise detection of road surface markings, an initial step was to minimize the inclusion of irrelevant visual information within the analysis framework. A significant observation from the dataset was that the road surface, which contains the relevant markings, was consistently visible in the lower half of the images. This allowed us to strategically focus our processing efforts by cropping out the upper half of the images, where unrelated features such as clouds often appeared. These clouds, displaying various shades of grey and white, could easily be mistaken for road markings due to their color similarities, potentially confusing the detection algorithms. This similarity risked false detections and was a crucial factor to mitigate, as shown in Figure 3.6 where such challenges are visually exemplified.

Following this crucial cropping step, further refinement of the image was necessary to enhance the detection accuracy. The first intervention involved the application of a blurring filter. This filter smoothed the image by reducing the sharpness and granularity of the noise, which helped in homogenizing the visual data and reducing detail that could distract from the main features of interest. Subsequent to the blurring, a Gaussian thresholding filter was applied, a process depicted in Figure 3.7. Gaussian thresholding is a technique that adjusts the pixel values in an image such that only those within a certain intensity range are retained. This step was instrumental in enhancing the contrast between the road markings and the surrounding pavement, making the markings more discernible.

To further refine the detection accuracy, we employed erosion and dilation operations, two morphological processes that modify the shapes within an image. Erosion removes pixels on object boundaries, effectively reducing the size of the objects within the image. This was particularly useful in eliminating fringes and small artifacts that could be misinterpreted as part of the road markings. Following erosion, dilation was applied to expand the remaining features back to their original size. This process not only restored the size of the road markings but also helped in emphasizing them against the background. The combination of erosion and dilation ensured that any remaining noise was minimized and that the markings were prominently displayed. The application of these operations is detailed in Figure 3.8, which shows how they contribute to creating a binary image where the road markings are distinctly isolated and highlighted.

Through these stages—cropping, blurring, thresholding, erosion, and dilation—a refined image was produced where the road surface markings were clearly defined and other distracting elements were effectively suppressed. Each step played a critical role in enhancing the clarity and accuracy of the segmentation process, ensuring that the computer vision algorithms could perform optimally without being misled by extraneous visual information. This meticulous approach to image processing ensured that the subsequent detection and analysis phases could proceed with a high degree of precision, focusing solely on the accurately highlighted road

markings.

3.3.2 Region Separation

After successfully processing the initial images, our project advanced into a more nuanced phase that involved the implementation of the Jarvis wrapping algorithm. This sophisticated algorithm addresses the Convex-Hull problem, which is central to many image processing tasks, particularly those involving shape analysis. In essence, the Convex-Hull problem involves calculating the smallest convex boundary that can enclose a group of points or shapes within an image. In our application, this meant identifying the outermost boundaries that define the clusters of features within the binary images derived from our previous image processing steps.

When we applied the Jarvis wrapping algorithm to our data, it efficiently computed the contours of various clusters within these binary images. The visualization of these contours, as depicted in Figure 3.9, revealed clear and distinct demarcations between different regions within the images. Notably, it became visually apparent that certain regions, which contained road surface markings, were distinctly isolated from other clusters within the images. This visual separation was a crucial step forward because it allowed us to see the potential of the algorithm in isolating regions of interest—specifically, road surface markings—from less relevant areas of the image.

Despite the success in visual demarcation provided by the Jarvis wrapping algorithm, a significant challenge remained. We needed to convert this visual distinction into a computational one, which is a non-trivial step in the development of automated systems for road marking analysis. The primary goal here was to develop a computational strategy that could not only recognize but also systematically differentiate the target regions—those containing road surface markings—from all other regions within the image.

To address this challenge, our approach was to develop a set of computational criteria based on the properties of the contours identified by the Jarvis wrapping algorithm. These criteria included aspects such as the geometric shape, the area covered by the contour, and perhaps most importantly, the degree of uniformity or 'smoothness' within the contour, which is a typical characteristic of road surface markings due to their consistent and standardized application on roads.

This differentiation process is critical as it forms the basis for the subsequent steps in our methodology. By ensuring that only the regions containing road surface markings are selected for further analysis, we significantly enhance the accuracy and efficiency of our subsequent assessments, such as condition evaluation and deterioration tracking. This systematic differentiation not only streamlines the process of marking evaluation but also lays a robust foundation for deploying these techniques in real-world scenarios where automation in road maintenance checks is increasingly becoming a necessity.

3.3.3 Road Sign Detection

Building upon the refined output of the initial image processing, the subsequent task in our methodology involved the intricate detection of road signs by further analyzing the preserved contours. This task began with the strategic elimination of smaller, less significant contours that did not meet a predefined point count threshold. By focusing only on significant contours, we reduced the noise further, enhancing the precision of our detection algorithm.

Once the irrelevant noise was removed, we were left with well-defined, somewhat uniform areas

within the images. These areas, characterized by their distinct shapes and sizes, were considered potential target regions that possibly contained road surface markings. To effectively process these regions, each was isolated into a separate frame to enable detailed analysis. This isolation was crucial as it allowed for the application of specialized image processing techniques on a per-region basis, thus avoiding the influence of adjacent regions.

One of the critical techniques employed at this stage was the Differential of Gaussian (DoG) computation. The DoG is a refined technique for edge detection, which involves the subtraction of one blurred version of an image from another, less blurred version of the same image. This method effectively highlights regions of the image with high frequency, which correspond to edges or transitions in intensity.

To compute the DoG value for each isolated frame, we first applied a Gaussian filter to the image. The Gaussian filter is a type of image-blurring filter that reduces detail and noise, governed by the standard deviation of the Gaussian distribution. By applying two such filters with different standard deviations to the image and subtracting one resulting image from the other, we highlighted areas with significant color variations—often indicative of road markings.

The results of the DoG computation were then scaled relative to the area of each potential target region. This scaling, referred to as "Smoothness," assesses how uniform a given area is. In our context, the smoother a region, the more likely it is to be a road surface marking, as these markings typically exhibit less color variation compared to their surrounding areas.

Based on the computed "Smoothness" value, we set a threshold to distinguish between likely road marking regions and other regions. This thresholding was a crucial decision point in our algorithm, separating target regions from background areas and thus enabling focused analysis on potential road markings.

The final segmentation of the traffic landmarks was achieved by setting appropriate thresholds for the "Smoothness" parameter. This process was not only systematic but also grounded in quantitative analysis, ensuring that each step was reproducible and verifiable. The Python implementation involved using discrete Gaussian filters to smooth the image at two different levels and subtracting these to compute the DoG, thereby operationalizing the theory into a practical tool for road sign detection as seen in Figure 3.10. This approach ensured that our segmentation was both accurate and robust, providing a reliable basis for further quality evaluation of the detected road surface markings.

In the given equations:

- Equation 3.1 defines the Gaussian function $G(x, y)$, where x and y are the spatial coordinates and s is the standard deviation of the Gaussian distribution. This function is used to apply a Gaussian blur to an image, which smooths out the image and reduces noise and detail.
- Equation 3.2 and Equation 3.3 describe the Differential of Gaussian (DoG), which is computed by differentiating the Gaussian-blurred image $f * G_s$ with respect to the scale parameter s. This operation highlights edges and transitions in the image by subtracting one blurred image from another, less blurred image.
- Equation 3.4 defines the "Smoothness" S, which is the ratio of the DoG value to the area of the region being analyzed. This measure helps determine how uniform the area is, with higher values indicating less uniformity and more potential for being a road marking due to the presence of edges and transitions.

$$G(x,y) = \frac{1}{2\pi s^2} e^{-\frac{x^2+y^2}{2s^2}} \tag{3.1}$$

$$\text{DoG} = \frac{\partial}{\partial s}(f * G_s) \tag{3.2}$$

$$\text{DoG} = \frac{\partial}{\partial s}(f * G_s) = f * \frac{\partial G_s}{\partial s} \tag{3.3}$$

$$S = \frac{\text{DoG}}{\text{Area}} \tag{3.4}$$

$$D = \frac{2\sum_{i=1}^{N} g_i p_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2} \tag{3.5}$$

$$\text{CrossEntropy}(p,y) = \begin{cases} -log(p) & \text{if } y = 1 \\ -log(1-p) & \text{otherwise} \end{cases} \tag{3.6}$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{3.7}$$

$$FocalLoss(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t) \tag{3.8}$$

3.4 Deep Learning Approach

The transition to a deep learning approach marked a significant advancement in our quest to enhance the detection and segmentation of road surface markings. This methodological pivot was prompted by the limitations observed with traditional computer vision techniques, which sometimes misclassified non-target objects such as sidewalks, tactile paving, and white cars as road markings. These misclassifications underscored the need for a more robust technique capable of handling the complexities of real-world data without the reliance on hand-engineered features.

Given the complexities associated with manual feature engineering, we opted for U-Net, a model renowned for its effectiveness in image segmentation tasks. U-Net's architecture, particularly its ability to learn from data directly, presented an opportunity to overcome the challenges of manually crafted features. However, a significant hurdle was the lack of pixelwise annotations in our dataset, which are crucial for training segmentation models.

To address the annotation challenge efficiently, we developed a semi-automated labeling tool, leveraging the computer vision techniques previously devised. This tool enabled the rapid annotation of a large volume of images by providing initial guesses of annotations based on the "Smoothness" method, which users could easily adjust. This process was visualized in Figure 3.3, where the tool displayed a data sample alongside suggested annotations. Users could refine these by selecting the average color in a region with a right-click, effectively segmenting the image into foreground and background based on this selection. This intuitive interface

significantly reduced the time and effort required for manual labeling and allowed for quick corrections through a simple delete function.

The subsequent pre-processing and training phases were streamlined due to the manual oversight of the labeling process, minimizing the need for extensive pre-processing. Nevertheless, we employed erosion and dilation operations to refine the labels further, removing residual background noise and enhancing the clarity of the annotations. The images were resized to 1600x1600 pixels to ensure they were well-suited for processing by the U-Net model.

Recognizing the challenge of class imbalance—a common issue in pixelwise segmentation where most pixels typically represent the background—we carefully selected our loss function. We employed a combination of Dice loss and Focal loss. Dice loss was particularly valuable for its focus on boundary precision, essential for the accurate delineation of road markings. Focal loss, on the other hand, addressed the imbalance by modifying the model's focus towards harder-to-classify instances, thereby ensuring that less frequent but crucial features such as road markings were not overlooked during the learning process. In the equations presented:

- Equation 3.5 represents the Dice coefficient, denoted by D. Here, g_i refers to the ground truth values, and p_i are the predicted values, with i indexing over all N pixels in the output segmentation map. This metric evaluates the overlap between the predicted segmentation and the ground truth, emphasizing the accurate delineation of the boundaries of road markings.
- Equation 3.6 outlines the Cross-Entropy loss function. In this equation, p is the predicted probability of the target class, and y is the binary indicator (0 or 1) if the class label is the correct classification for the observation. This loss function is essential for classification tasks, penalizing the divergence between the predicted probabilities and the actual binary outcomes.
- Equation 3.7 defines p_t, the transformed prediction, which adjusts p, the original prediction, based on the ground truth label y. If the true label y is 1, p_t equals p; otherwise, it equals $1 - p$. This transformation is crucial for calculating the Focal Loss, focusing the training on hard-to-classify examples.
- Equation 3.8 describes the Focal Loss, where α_t is a weighting factor for the class (typically inverse class frequency), and γ is a focusing parameter that adjusts the rate at which easy examples are down-weighted. $(1 - p_t)^\gamma$ scales the loss at each level of confidence, which helps in addressing class imbalance by focusing more on difficult, misclassified cases.

These parameters collectively enhance the U-Net model's capability to effectively learn from the training data, addressing both the accuracy in boundary detection and the challenge of class imbalance in pixel-wise segmentation.

Training was conducted in a GPU-accelerated environment utilizing the PyTorch framework, which facilitated efficient computation and iterative testing. The training sessions demonstrated continuous improvement in the model's performance, culminating in a Dice score of 89% on the training set—an indicator of the model's proficiency in capturing the essential boundaries of road markings. In Table 3.1 we can see the different models performances on the dataset presented in [5]. In Table 3.2 we can see the resuls of our experiments.

For validation, we subjected our model to a set of tests to assess its practical effectiveness. The model achieved a Dice score of 78.90% on the validation set, affirming its capability to generalize well to new, unseen data. This performance was not only a testament to the model's accuracy

Table 3.1: Performance of Different Segmentation Models across Various Locations

Method	**Austin**		**Chicago**		**Kitsap**		**West Tyrol**		**Vienna**		**Overall**	
	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc
FCN-8s	50.28	92.30	53.89	87.24	32.09	98.52	56.40	95.84	62.75	88.30	56.19	92.44
U-Net	78.62	96.89	70.39	92.89	66.26	99.27	70.93	97.71	78.28	93.85	74.79	96.12
SegNet	70.60	94.74	64.81	89.72	60.55	98.89	71.41	97.28	74.97	91.79	70.10	94.48
DeepLab	76.65	96.56	69.39	92.56	65.78	99.24	75.01	97.98	79.24	94.06	74.86	96.08
PSPNet	71.69	95.73	66.67	91.62	63.08	99.18	72.07	97.72	79.49	93.12	71.67	95.47

Table 3.2: Performance Comparison of Segmentation Approaches

Method	**Dice Score**	**Accuracy**
U-Net Model	78.90%	99.34%
Classical Computer Vision Approach	31.96%	93.27%

in detecting road marking boundaries but also highlighted the comparative advantages over the initial computer vision approaches. The results, both qualitative and quantitative, were documented comprehensively, providing a clear depiction of the model's reliability and the effectiveness of integrating advanced deep learning methods into the road marking segmentation process. we can see results in Figures 3.11 to 3.13.

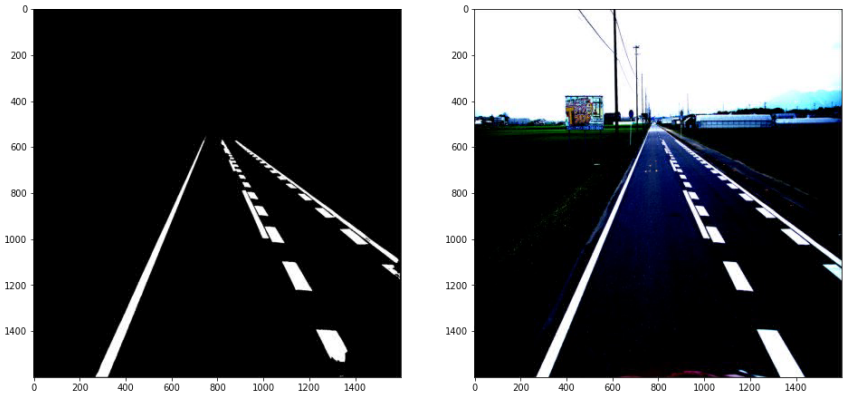


Figure 3.11: Example of Successful Detection on Validation Set Sample (1)

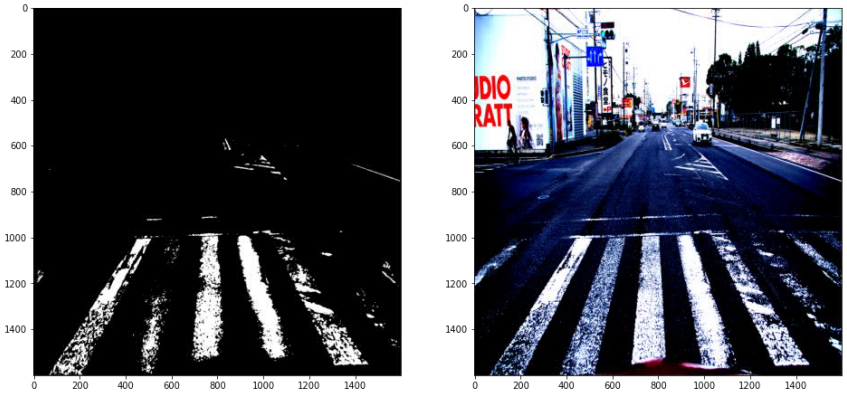


Figure 3.12: Example of Successful Detection on Validation Set Sample (2)

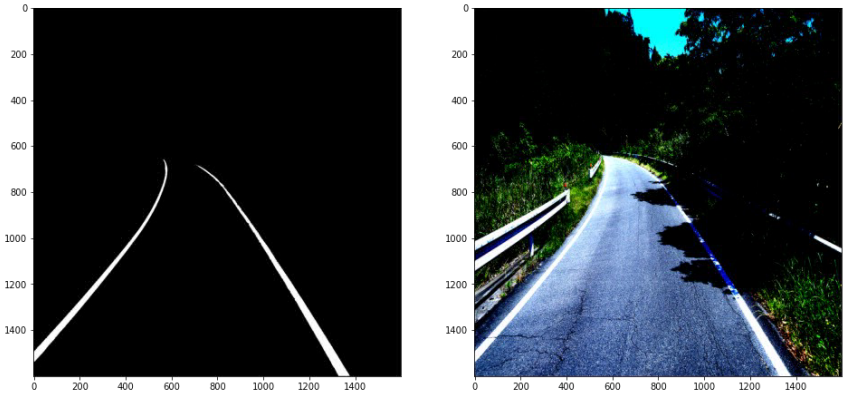


Figure 3.13: Example of Successful Detection on Validation Set Sample (3)

Chapter 4

Road Surface Marking Quality Evaluation Using Efficient VGG-16 model

4.1 Introduction

In this chapter, we delve into the advancements made in our road surface marking evaluation pipeline through the development and deployment of the "Efficient VGG-16" model. This innovative approach represents a significant enhancement over traditional VGG-16 based models, specifically tailored to address the unique challenges of road surface marking deterioration—an area that has seen limited exploration despite its critical importance for road safety and urban planning.

The collaboration with the local government facilities of Mie prefecture in Japan provided access to a valuable dataset of road images, which played a pivotal role in the training and fine-tuning of our model. This dataset, enriched with binary masks and quality labels derived from a previously trained segmentation model, allowed for a nuanced understanding and characterization of road marking conditions ranging from pristine to severely deteriorated.

Central to this study is the application of the "Efficient VGG-16" model to this dataset, alongside comparisons with standard VGG-16 and ResNet-18 architectures. By adapting the original VGG-16 framework to better suit the specific needs of road surface analysis, we introduced several optimizations aimed at enhancing processing efficiency and predictive accuracy. These modifications were critical given the complexity and variability of the dataset which included diverse lighting conditions, varying degrees of marking degradation, and different environmental settings that typically challenge computer vision algorithms.

The optimization process involved not only architectural tweaks but also a comprehensive reevaluation of the training process to better accommodate the unique dataset characteristics. By employing advanced training techniques and leveraging powerful GPUs, the model achieved a Mean Squared Error (MSE) of 3.62%, indicating a promising ability to accurately assess road marking quality. This performance underscores the model's potential to significantly impact road safety measures by providing reliable, automated assessments of road marking quality.

Moreover, the implications of this research extend beyond mere technical achievement. By

Figure 4.1: Slightly Deteriorated Marking

integrating such advanced monitoring systems, local governments and urban planners can better manage and maintain traffic infrastructure, which is essential for the development of smarter, safer urban environments. Additionally, the findings from this research offer valuable insights for the ongoing development of autonomous driving technologies and advanced driver-assistance systems, which rely heavily on high-quality road markings for navigation and safety.

As we progress through this chapter, we will explore the specific architectural details of the "Efficient VGG-16" model, the tailored training regimen that was developed to maximize its performance, and a detailed analysis of its evaluation results. This discussion will not only highlight the technical aspects of the model but also its practical applications, setting the stage for future innovations in the field of road surface monitoring and maintenance.

4.2 Dataset

In the course of developing the "Efficient VGG-16" model for assessing the quality of road surface markings, a critical aspect was the preparation and utilization of the initial dataset. This dataset comprised high-resolution RGB images as previously discussed, shown in Figures 4.1 and 4.2, each annotated with quality labels reflecting the condition of various traffic landmarks captured within these images. The quality labels were assigned on a scale from 1 to 4, with '1' indicating no deterioration and '4' denoting complete deterioration, providing a graded assessment of the condition of the road surface markings. The quality labels are shown in Figure 4.3.

To facilitate a more focused analysis and enhance the model's learning efficiency, the dataset underwent a preprocessing phase where 800 selected images were transformed into binary masks. These masks, stripped of their color information, highlight the structural integrity of the road markings, excluding irrelevant background elements like the sky, vehicles, or vegetation that could potentially skew the model's learning process. Figures 4.4 and 4.5 illustrate these binary masks, which serve as a cleaner, more direct input for training deep learning models.

Recognizing the need to enrich the dataset and introduce variability that mirrors real-world conditions, extensive data augmentation techniques were employed. Initially, the binary masks underwent a 30-degree rotation, effectively doubling the dataset size. Subsequent mirroring of these rotated images further expanded the dataset, culminating in a comprehensive set of

Figure 4.2: Road Marking in Good Condition

M	N	O	P	Q	R
剥離度（区画線１）	剥離度（区画線２）	剥離度（区画線３）	剥離度（区画線４）	剥離度（区画線５）	剥離度最大値
4	4	-	-	-	4
4	2	-	-	-	4
4	2	-	-	-	4
4	2	-	-	-	4
4	2	-	-	-	4
4	2	-	-	-	4
1	1	-	-	-	1
1	1	-	-	-	1
1	1	-	-	-	1
1	1	-	-	-	1
1	1	-	-	-	1
1	1	-	-	-	1
1	1	-	-	-	1
1	1	-	-	-	1

Figure 4.3: Snapshot of Quality Annotations

Figure 4.4: Example of Binary Mask by Segmentation Model

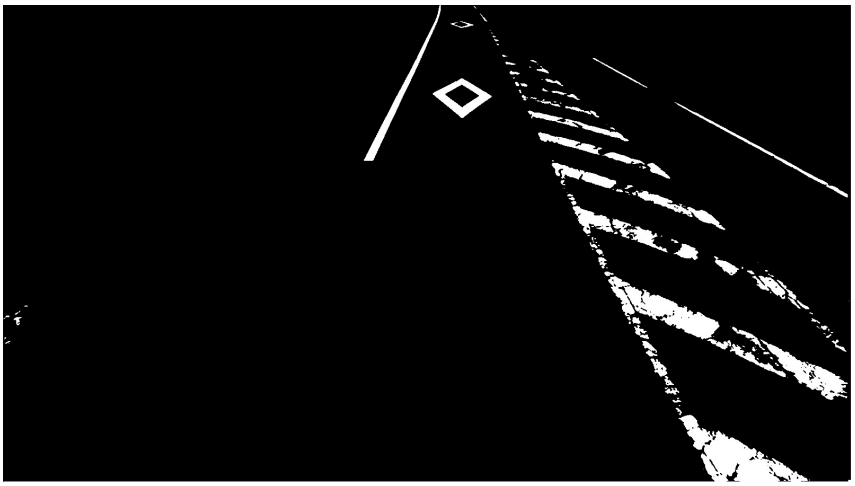


Figure 4.5: Training Data (The top half of the image was cropped.)

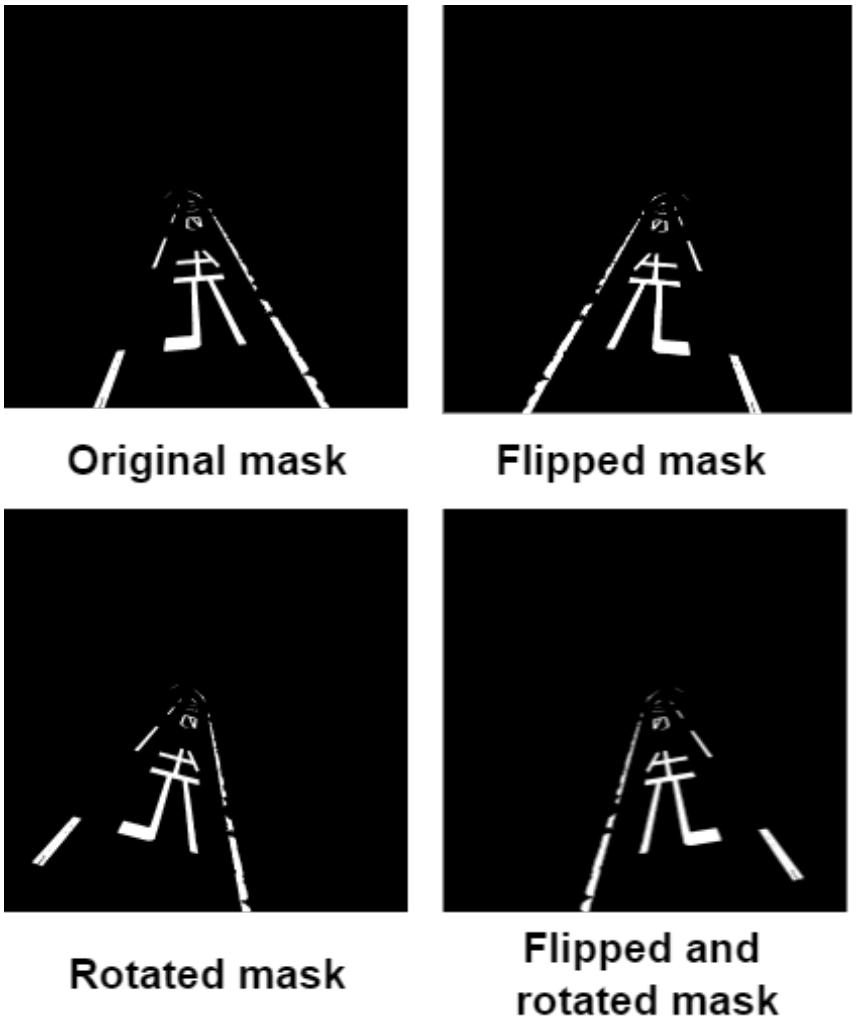


Figure 4.6: Examples of Data Augmentation (All of the masks shown would have the same quality evaluation, since quality of landmarks does not change by flipping or rotating the image.)

3,200 binary masks. Each mask was then resized to 800px by 400px, focusing exclusively on the lower half where road markings are predominantly located, as the upper half generally does not contain pertinent information for this specific task. Figure 4.6 shows the augmentation of a sample from the dataset.

This approach to data preparation not only streamlined the input for the deep learning models but also addressed the challenges faced by the models when training directly on RGB images. In previous attempts, models trained on RGB data struggled to converge, likely confused by the extraneous details present in full-color images. By transitioning to binary masks, the model could concentrate on discerning the textural and structural nuances pertinent to the road surface markings' quality without the distraction of unrelated visual information.

Given the variable number of road marking types that could appear in an image and the complexity this introduces in predicting a vector of quality evaluations, a simplification was made. Rather than dealing with a multi-dimensional output, the quality labels for each image were averaged, reducing the output to a single quality score per image. This simplification allowed for a more streamlined model architecture and training process, making it feasible to

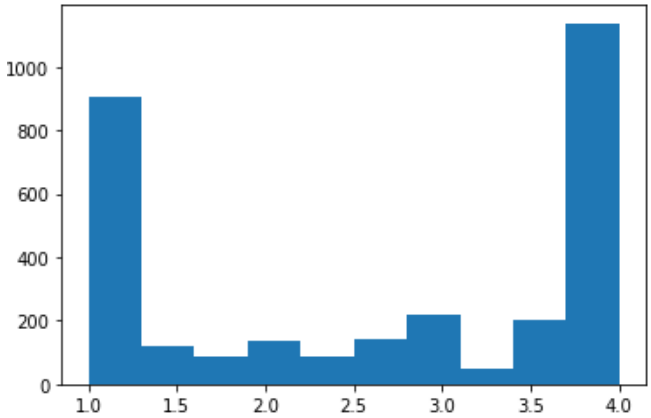


Figure 4.7: Distribution of Labels in Dataset

train a model that accurately predicts the average quality of road markings across a diverse set of images.

The final dataset, now enriched and appropriately formatted, presented a new challenge—imbalance in the quality label distribution, as evidenced in Figure 4.7. This imbalance could potentially bias the model towards more frequently represented classes. To mitigate this, regularization techniques were integrated into the training process to ensure a balanced sensitivity to all categories of road surface quality. This section of the thesis details the methodical steps taken to prepare the dataset for effective training of the "Efficient VGG-16" model. It underscores the thoughtful considerations and adaptations required to harness the power of deep learning for the practical and impactful task of road surface marking quality evaluation. As we proceed, the focus will shift to the architectural specifics of the "Efficient VGG-16" model, the innovative training strategies employed, and the rigorous evaluation process that underscores the model's potential in transforming road safety measures through advanced image processing techniques.

4.3 Models

4.3.1 VGG-16

VGG-16, which stands for Visual Geometry Group with 16 weighted layers, is a convolutional neural network model. This model, developed by researchers from the University of Oxford, has become one of the primary reference points in the field of deep learning, especially when discussing image analysis and computer vision.

At its core, the VGG-16 model's architecture is characterized by its simplicity, using only 3x3 convolutional layers stacked on top of each other in increasing depth. Convolutional layers, in the realm of deep learning, refer to the layers of the neural network where the convolution operation takes place. This operation involves the use of a filter or kernel which scans over the input data (such as an image) to produce a feature map, effectively transforming the input data into a form that makes it easier for the network to understand.

The depth of the network, which refers to the number of layers, is a significant factor in the VGG-16 model. A deeper network allows for the extraction of more complex and high-level features from the input image. In the VGG-16 model, depth is achieved with 13 convolutional layers, interspersed with max-pooling layers for down-sampling, followed by three fully

connected layers. Down-sampling, performed by the max-pooling layers, reduces the spatial dimensions of the input, making the network less computationally intensive and reducing the chances of overfitting. Overfitting is a common problem in machine learning where a model performs exceptionally well on training data but poorly on unseen or new data.

Another hallmark of the VGG-16 model is its use of a large number of filters. Filters in convolutional layers help detect specific features like edges, textures, and patterns. VGG-16 starts with 64 filters in the first layer and doubles the number as it goes deeper into the network, reaching up to 512 filters.

Despite its depth and complexity, one of the advantages of VGG-16 is its uniform architecture. This consistency makes it easier to implement and modify, which has contributed to its popularity in the research community.

Activation functions play a critical role in neural networks, introducing non-linearity to the model, which allows it to learn from the error and make adjustments, a process known as back-propagation. VGG-16 uses the Rectified Linear Unit (ReLU) activation function throughout its architecture. ReLU replaces all negative pixel values in the feature map with zero and has been found to train deep learning models faster than other traditional activation functions.

Towards the end of the network, after several convolutional and max-pooling layers, the VGG-16 has three fully connected layers. Fully connected layers are traditional layers where each input node is connected to each output node. The final layer has a softmax activation function that classifies the input image into one of the 1,000 predefined classes.

In summary, the VGG-16 model stands as a testament to the power of deep convolutional neural networks in image recognition tasks. Its depth, combined with its simple and consistent architecture, allows it to extract a wide range of features from input images, making it one of the top-performing models in image classification challenges. Its design principles have influenced many subsequent deep learning architectures for visual recognition tasks.

4.3.2 VGG-16 as Regression Model

VGG-16, as initially proposed, was designed for image classification tasks. However, the adaptability and effectiveness of convolutional neural networks (CNNs) like VGG-16 make them versatile tools, extendable beyond just classification. One such application is regression, a type of predictive modeling technique where the goal is to predict a continuous value instead of a discrete label.

The relationship between VGG-16 and regression stems from the underlying architecture of the model. At its essence, VGG-16 extracts a plethora of features from input images through its multiple convolutional layers. These features range from simple edge detections in the initial layers to more complex patterns and structures in the deeper layers. Once these features are extracted, they serve as a rich representation of the input data.

For regression tasks, the primary adaptation involves the final layers of the VGG-16 model. In classification, the last fully connected layer typically employs a softmax activation function to distribute the probabilities across multiple classes. For regression, however, this softmax layer is replaced. Instead of predicting class probabilities, the network is modified to predict one or more continuous values. This can be achieved by using a linear activation function in the final layer, ensuring that the output can range across real-valued numbers. Several instances highlight the use of VGG-16 for regression tasks:

Facial Key Points Detection

In tasks where the goal is to identify specific points or landmarks on faces (like the corners of eyes, tip of the nose, etc.), VGG-16 can be adapted for regression. Here, instead of classifying different faces, the network predicts the x and y coordinates of these landmarks.

Age Estimation from Facial Images

Predicting a continuous value like age from facial images is another regression task. By training a VGG-16 model on a dataset of faces with known ages, the network can be fine-tuned to estimate ages from new, unseen images.

Gesture Recognition

In some advanced human-computer interaction scenarios, the precise angle or pose of a hand or finger might be required. VGG-16 can be employed to regressively predict these angles based on the image of the hand.

Object Localization

While object detection typically involves classifying and drawing bounding boxes around objects in images, the exact coordinates of these bounding boxes can be predicted using regression. VGG-16 can be adapted to predict the x and y coordinates of the top-left corner, along with the height and width of the bounding box.

Medical Imaging

In healthcare, VGG-16 can be used for tasks like predicting the size or growth rate of tumors from radiology images.

The adaptability of VGG-16 for regression tasks is further enhanced by transfer learning. Given its pre-training on large datasets like ImageNet, the model already possesses a wealth of feature detectors. By leveraging this pre-training, one can fine-tune the model on a smaller, task-specific dataset, making it suitable for regression while benefiting from the knowledge it has already acquired.

In conclusion, while VGG-16's inception was rooted in image classification, its flexibility and robust feature extraction capabilities make it a potent tool for regression tasks. By making subtle architectural changes, especially in the final layers, and employing transfer learning, VGG-16 can be tailored for a wide array of regression-based applications, showcasing its versatility in the realm of deep learning.

$$L(p, y) = (p - y)^2 \tag{4.1}$$

$$L_\sigma(p, y) = \begin{cases} \frac{1}{2}(p - y)^2 & \text{if } |p - y| < \sigma \\ \sigma|p - y| - \frac{1}{2}\sigma^2 & \text{otherwise} \end{cases} \tag{4.2}$$

$$L(p, y) = log(cosh(p - y)) \tag{4.3}$$

In the regression context using VGG-16, the equations provided facilitate the understanding of various loss functions applied during training: Equation 5.1 represents the mean squared error

loss, $L(p, y) = (p-y)^2$, where p is the predicted value and y is the actual value. This loss function is commonly used in regression tasks to minimize the difference between the predicted and true values, emphasizing the penalty on larger errors. Equation 5.2 defines a smoothed L1 loss, often referred to as Huber loss. This loss function is less sensitive to outliers than the squared error loss. It behaves like a mean squared error when the error is small (less than a threshold σ) and as a linear function when the error is large, combining the benefits of both L1 and L2 regularization. Here, σ acts as a threshold determining the switch between quadratic and linear behavior. Equation 5.3 outlines the logarithm of the hyperbolic cosine of the prediction error, $L(p, y) = \log(\cosh(p - y))$. This function gradually approaches a linear form at higher values of error, reducing the influence of outliers in the predictions, while maintaining differentiability and smoothness of the loss landscape.

These loss functions are critical in fine-tuning the VGG-16 model adapted for regression tasks. They ensure that the model is robust against various types of errors and is sensitive to the nuances of continuous output values. By employing these sophisticated loss measures, VGG-16's utility extends beyond classification, providing precise and reliable predictions for continuous outcomes in diverse applications such as medical imaging and facial landmark detection.

4.3.3 Res-Net

Res-Net, short for Residual Networks, introduced a groundbreaking concept that changed the landscape of deep learning, especially in the domain of computer vision. Before the inception of Res-Net, one of the prevailing notions in neural network design was that deeper networks, with more layers, would naturally result in better performance. However, training very deep networks presented a unique set of challenges, notably the vanishing gradient problem. This issue occurs during the backpropagation step of training, where gradients of the loss function can become extremely small. As a result, weights in the initial layers of the network barely get updated, leading to poor convergence.

The inventors of Res-Net tackled this problem with an innovative solution: residual blocks. The central idea behind a residual block is quite intuitive. Instead of trying to learn an underlying function directly, why not focus on learning the residual (or difference) between the input and the desired output? In other words, if certain layers can already represent the underlying function reasonably well, additional layers should be geared to learn only the difference between the current representation and the desired one.

Each residual block in a Res-Net contains one or more convolutional layers followed by batch normalization and a ReLU activation function. The output from these layers is then added to the original input, creating a shortcut or skip connection. This skip connection ensures that even if the weights of the convolutional layers become very small, the original input can still be passed directly to subsequent layers, preserving the information and mitigating the vanishing gradient problem.

The beauty of Res-Net lies in its scalability. Models with varying depths, such as ResNet-18, ResNet-34, ResNet-50, and even ResNet-152, have been successfully trained, setting new performance benchmarks on several datasets.

Another advantage of Res-Net is that it generalizes well across different tasks. While it was initially designed for image classification, Res-Net architectures have been employed for object detection, image segmentation, and even some natural language processing tasks. The residual

blocks help the network adapt to different complexities and data distributions, ensuring that it doesn't overfit to a specific training set.

Furthermore, the concept of residual learning introduced by Res-Net has inspired several subsequent innovations in neural network design. Variants like DenseNet expanded on the idea by introducing dense connections where each layer receives input from all preceding layers. Another variant, ResNeXt, incorporates grouped convolutions into the residual blocks for more efficient learning.

In essence, Res-Net revolutionized the way we perceive and design deep neural networks. By introducing the simple yet powerful concept of residual learning, it not only alleviated the challenges of training deep networks but also paved the way for even deeper and more efficient architectures. Whether you're dealing with image data, text, or even audio, the principles behind Res-Net offer a robust foundation for building state-of-the-art models in various domains.

4.3.4 Efficient-Net

EfficientNet emerged as a compelling response to a crucial question in deep learning: How can we scale up convolutional networks in a manner that ensures better performance without an exponential increase in computational demand? Typically, when one thinks of scaling neural networks, it's either about making the network deeper (adding more layers), wider (increasing the number of channels or neurons), or increasing the resolution of the input images. But figuring out the right balance between these dimensions can be tricky, and naively scaling up any of these dimensions can lead to suboptimal results or excessive computational costs.

EfficientNet's brilliance lies in its systematic approach to this scaling dilemma. Instead of independently scaling each dimension, EfficientNet introduced a compound scaling method. This method uses a fixed set of scaling coefficients for depth, width, and resolution that are determined based on the available computational budget. The idea is to scale all dimensions of the network in a balanced way, ensuring that each part of the network scales in proportion to the others.

The journey of EfficientNet began with the development of a baseline model named EfficientNet-B0. This model was discovered through a neural architecture search, ensuring that it was already optimized for performance. Once this baseline was established, it was scaled up to create a series of models (from EfficientNet-B1 to EfficientNet-B7) using the compound scaling method.

At the heart of the EfficientNet architecture is the use of mobile inverted bottleneck blocks (MBConv), a design element borrowed from another influential model, MobileNetV2. These blocks are incredibly efficient and are particularly suitable for mobile and edge devices due to their reduced computational requirements. The inclusion of squeeze-and-excitation blocks within the MBConv further improves the model's capacity to focus on the most informative channels.

One of the standout features of EfficientNet is its incredible efficiency in terms of parameters and operations. For instance, EfficientNet-B7 achieves state-of-the-art performance on benchmarks like ImageNet but does so with significantly fewer parameters and operations compared to other models of similar capacity. This efficiency is a testament to the power of balanced scaling and the architectural choices made in the design of EfficientNet.

Besides image classification, the principles and architecture of EfficientNet have been adapted

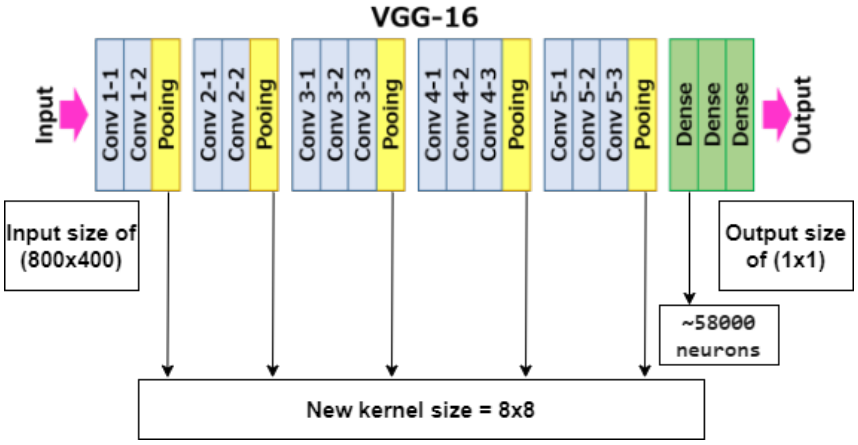


Figure 4.8: Changes of Original VGG-16 Model for Efficient VGG-16

for various other tasks. Transfer learning with EfficientNet, where a model is pre-trained on a large dataset and then fine-tuned on a smaller, task-specific dataset, has shown exceptional performance across a multitude of tasks, from object detection to medical imaging.

In summary, EfficientNet represents a paradigm shift in how we approach the scaling of neural networks. By balancing the scaling of depth, width, and resolution and by leveraging efficient architectural components, EfficientNet delivers state-of-the-art performance while maintaining computational efficiency. It's a testament to the fact that with thoughtful design and scaling, it's possible to build models that are both powerful and efficient, making them suitable for a wide range of applications, from cloud-based solutions to on-device deployments.

4.3.5 Efficient VGG-16

The Efficient VGG-16 is a new version of the well-known VGG-16 model, which is commonly used in deep learning. The main idea behind creating this new model was to make it work faster and use fewer resources without losing the ability to identify the quality of road markings. Figure 4.8 and Figure 4.9 show the modifications made to the VGG-16 model in order to obtain the new model. In the regular VGG-16 model, it uses a lot of details and calculations, which makes it slow and heavy. This is because it has a high number of parameters, which are the settings it needs to check and adjust to learn from images. The more parameters, the more the model needs to compute, and sometimes, having too many can even make the model learn wrong patterns from the data.

To solve this, the Efficient VGG-16 changes the way the model looks at images. Instead of shrinking the image little by little using a small max-pooling filter of 2x2, it uses a bigger max-pooling filter of 8x8. This means it looks at bigger parts of the image at once and reduces the image size faster. For an image that is 800x400 pixels in size, the new model creates a summary of the image that is 512x6x19 in size. This smaller size means it needs fewer settings or parameters, around 270 million in total.

When compared to the regular VGG-16 using the same image size, the regular model creates a bigger summary of 512x12x25 and needs around 661 million parameters. This shows that the Efficient VGG-16 is much simpler and lighter.

To make sure the Efficient VGG-16 is actually better and not just simpler, we tested it against other versions of the VGG-16 model with different image sizes. The results showed that the

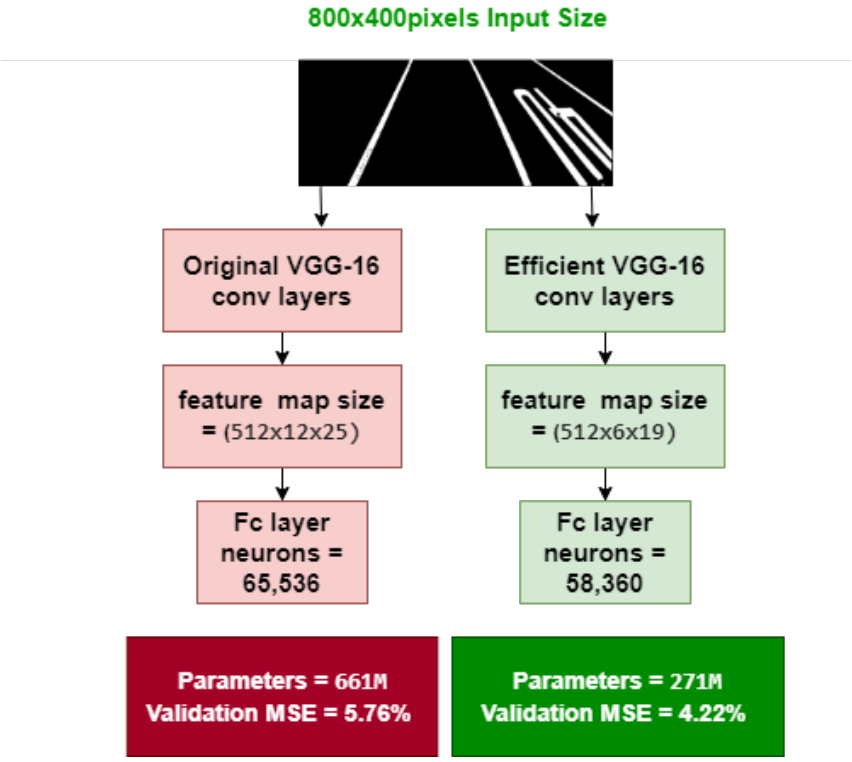


Figure 4.9: Comparison of Performance of VGG-16 and Efficient VGG-16 Models

Efficient VGG-16 did a good job, even with fewer parameters.

In short, the Efficient VGG-16 takes the best parts of the VGG-16 model and makes them simpler and faster. This makes it a great choice for looking at images and understanding road markings, and it could be useful for many other tasks in the future.

The Efficient VGG-16 model introduces several key optimizations to the classic VGG-16 architecture to improve efficiency and performance. Below are some equations that describe these enhancements:

$$\text{Params}_{\text{new}} = \frac{H \times W}{s^2} \times D \times N \tag{4.4}$$

Equation 4.4 calculates the number of parameters in the Efficient VGG-16, where H and W are the height and width of the input image, s is the size of the new larger maxpooling stride, D is the depth of the feature maps, and N is the number of convolution filters. This shows how increasing the maxpooling size reduces the spatial dimensions of the feature maps, thereby reducing the total number of parameters.

$$\text{Output Size} = \left(\frac{W}{s}, \frac{H}{s}\right) \tag{4.5}$$

Equation 4.5 demonstrates how the spatial dimensions of the output feature maps are reduced by the larger maxpooling size s. This reduction is directly proportional to the stride, enhancing

processing speed by decreasing the computational load.

$$\text{Efficiency Ratio} = \frac{\text{Params}_{\text{original}}}{\text{Params}_{\text{new}}} \tag{4.6}$$

Equation 4.6 defines the efficiency ratio, comparing the number of parameters in the original VGG-16 model to the Efficient VGG-16. This ratio quantifies the improvement in parameter efficiency, which contributes to faster processing and reduced memory usage.

$$T = \frac{1}{\sum_{i=1}^{n} \frac{1}{T_i}} \tag{4.7}$$

Equation 4.7 calculates the effective processing time T for the Efficient VGG-16, where T_i represents the processing time per layer. The harmonic mean provides a better average when dealing with rates, highlighting the speed improvement due to the reduced number of operations per layer.

These equations and their parameters highlight the technical underpinnings of the Efficient VGG-16 model, demonstrating how changes in architecture lead to gains in computational efficiency while maintaining effective learning and prediction capabilities.

4.4 Experimental Results and Analysis

The training process of our models is an intricate journey, where we ensure the models learn effectively from the data they're provided. As shown in the flowchart in Figure **??**, our data was organized into different batches - 73 samples for validation, 145 for testing, and 582 for training. Before applying data augmentation, a technique used to artificially increase the amount of data by slightly altering the original images, we ensured that the data was separated using a 5-fold cross-validation technique. This technique ensures the models are exposed to different subsets of the data throughout the training, promoting better generalization when they encounter new data.

The models' input is structured as 800x400 pixel binary masks, each having a label representing the average quality of the road markings. For the traditional VGG-16 model, however, the input size was tweaked to 510x255 pixels. These models were trained over a span of 1000 steps, after which they were evaluated based on how well they performed on the test set. The results are shown in Table 7.1 and Table 5.2.

A pivotal part of training deep learning models is the choice of the loss function. It's essentially the guiding star for the model, telling it how far off its predictions are from the actual values. We explored three different loss functions - Square loss, Huber loss, and Log-cosh loss. Each of these comes with its own set of advantages and challenges.

The Square loss, a staple in regression tasks, calculates the squared difference between the actual and predicted values. It's known for its quick convergence due to the way it's structured. When the error is high, the square loss function penalizes the model heavily, making it adjust more aggressively. However, this also means it's very sensitive to outliers, potentially leading the model astray.

Huber loss, on the other hand, tries to strike a balance. It combines the features of both squared loss and absolute loss functions. Depending on the error's magnitude, it decides which function

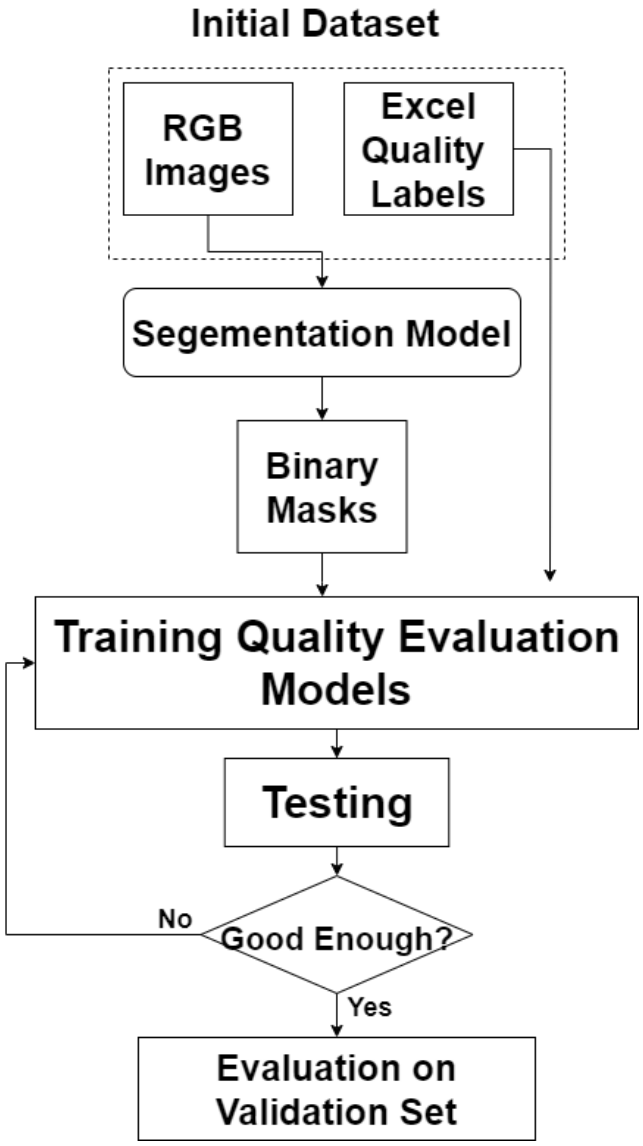


Figure 4.10: Flowchart Summarizing Process for Training each Model

to use. It's like a safety net, ensuring the model doesn't get too penalized by outliers, which is an issue with the square loss. But it's a bit more complicated, potentially making the training process a tad more involved.

Lastly, the Log-cosh loss is an interesting one. It shares similarities with Huber's loss, including being differentiable everywhere, which is a nice property when optimizing. However, it tends to plateau for large errors, meaning the model might not correct itself as aggressively as one might want.

In essence, the choice of loss function plays a fundamental role in shaping the model's learning journey. By comparing these functions, we aim to understand their effects on the training process, ensuring we choose the best one for our specific problem. Some qualitative results are shown in Figure 4.11.

Table 4.1: Performance Results on Test Set (NVIDIA GeForce RTX 3090 GPU)

Test Set Evaluation						
Model	Corr	MSE%	R2	Acc%	Number of parameters	Time per image
Efficient VGG-16	0.95	3.04%	0.91	84.44%	271M	12.50ms
VGG-16 (800x400p)	0.94	3.71%	0.89	80.10%	661M	13.78ms
VGG-16 (520x260p)	0.96	3.10%	0.91	83.77%	300M	11.00ms
VGG-16 (510x255p)	0.93	4.52%	0.87	78.92%	252M	9.79ms
Efficient-Net	0.07	33.41%	0.00	13.87%	4M	-
Res-Net-18	0.742	15.49%	0.53	44.08%	-	-
Efficient VGG-16 Huber loss	0.94	100.42%	-2.07	0.64%	271M	-
Efficient VGG-16 cosh loss	-0.011	114.59%	-2.50	28.20%	271M	-

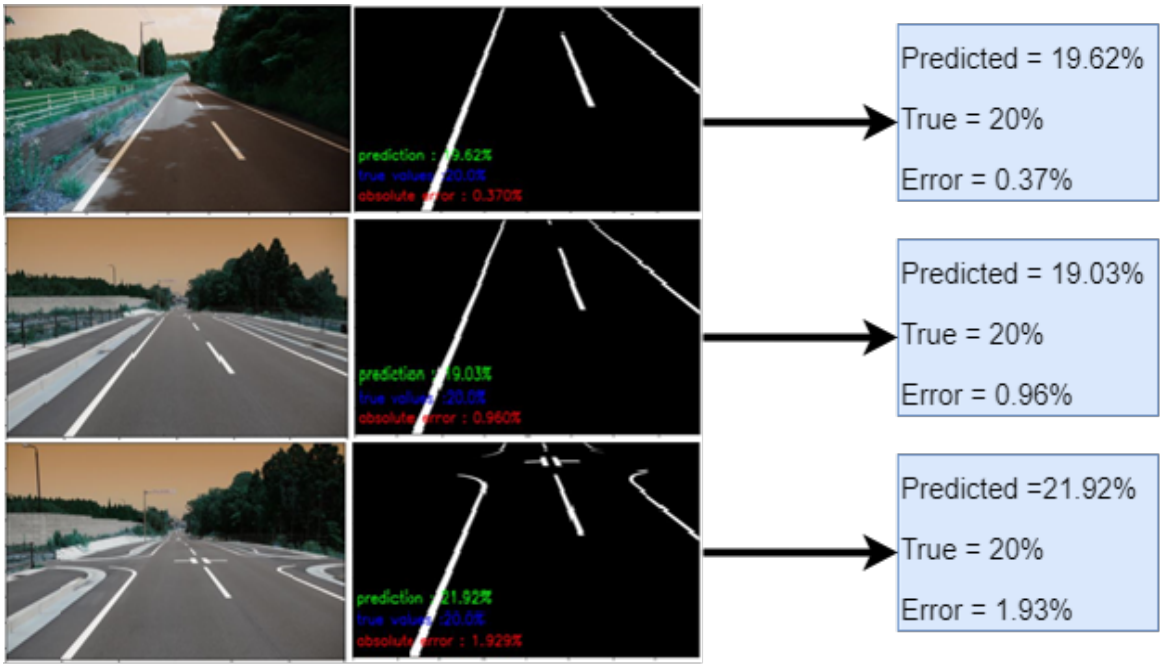


Figure 4.11: Example of Qualitative Result (Pprediction of Deterioration Rate)

Table 4.2: Performance Results on Validation Set (NVIDIA GeForce RTX 3090 GPU)

Validation Set Evaluation						
Model	Corr	MSE%	R2	Acc%	Number of parameters	Time per image
Efficient VGG-16	0.94	3.62%	0.89	82.87%	271M	-
VGG-16 (800x400p)	0.93	4.70%	0.86	77.73%	661M	-
VGG-16 (520x260p)	0.94	3.78%	0.88	82.87%	300M	-
VGG-16 (510x255p)	0.91	6.04%	0.81	72.60%	252M	-
Efficient-Net	0.09	32.57%	0.00	17.46%	4M	-
Res-Net-18	0.73	16.35%	0.51	44.73%	-	-
Efficient VGG-16 Huber loss	0.92	98.14%	-1.90	02.04%	271M	-
Efficient VGG-16 cosh loss	0.05	110.98%	-2.28	30.70%	271M	-

Chapter 5

Building a Large Binary Mask Dataset and Surveying Segmentation Models

5.1 Introduction

Embarking on the journey of developing a segmentation solution, as detailed in the preceding chapter, unveiled a panorama of possibilities that convolutional neural networks (CNNs) offer for solving complex segmentation tasks. The realization that numerous segmentation models exist, each with its unique capabilities and idiosyncrasies, propelled us into conducting an in-depth exploration of these models. Our aim was to not just understand the breadth of segmentation CNNs available but to dissect their functionalities, strengths, and limitations through a meticulous comparative analysis. This endeavor was not merely academic but a foundational step toward refining our segmentation approach, ensuring that we leverage the most effective model tailored to our specific needs in road surface marking detection.

Understanding the intricacies of these models necessitated a structured approach to evaluation. To this end, we embarked on survey research, an analytical journey that required a comprehensive dataset of labeled images as its cornerstone. These labels, or binary masks, serve as indicators of road surface markings within the images, distinguishing them from the surrounding environment. Creating such a dataset posed a significant challenge, as it required precise and accurate labeling that could accurately represent the real-world scenarios depicted in the images.

Addressing this need led to the development of a specialized software tool, an innovation born out of necessity. As discussed previously, this tool was designed from the ground up to facilitate the manual labeling of images, allowing us to create binary masks with a high degree of accuracy. Utilizing this software, we meticulously labeled 400 images, a task that, while labor-intensive, was crucial for laying the groundwork for our segmentation model training. This initial set of labeled images then served as the training data for a U-Net model, chosen for its proven effectiveness in similar tasks.

The trained U-Net model became a pivotal tool in our arsenal, enabling us to perform inference on the original, larger dataset. This step was instrumental in pre-filtering the dataset, identifying and segregating images where the segmentation results did not meet our standards of accuracy. Recognizing the limitations of relying solely on automated segmentation for dataset preparation, we developed a new software tool. This tool was designed with a specific purpose:

to streamline the review process, allowing us to efficiently filter out incorrectly segmented images and refine the dataset further.

This iterative process of manual labeling, model training, automated inference, and rigorous filtering culminated in the creation of a robust dataset. This dataset, enriched with accurately labeled images of road surface markings, became the foundation of our survey research. It not only enabled us to conduct a comprehensive comparative analysis of various segmentation models but also ensured that our findings were grounded in real-world applicability and relevance.

Our survey research transcended a mere academic exercise. It was a quest for optimization, a search for the segmentation model that not only promised theoretical excellence but also demonstrated practical efficacy in detecting road surface markings. Through this survey, we delved into the architecture, performance metrics, and application scenarios of each model, comparing them against the backdrop of our specific use case. This meticulous evaluation aimed to distill insights that could guide the future direction of segmentation in road surface marking detection, potentially setting new benchmarks for accuracy and efficiency.

As we venture into the details of this survey in the following sections, we not only aim to share our findings but also to illuminate the path for future research in this domain. The development of the dataset and the ensuing comparative analysis of segmentation models represent pivotal milestones in our ongoing exploration of computer vision's potential to revolutionize traffic management and road safety. This chapter, therefore, is not just a recounting of our methodological journey but a testament to the transformative power of targeted research and technological innovation in addressing real-world challenges.

5.2 Building Dataset

The foundation of any robust machine learning project is a well-constructed dataset. For our research, we were fortunate to have access to a comprehensive collection of 13,000 high-resolution images (3000x1600 pixels) provided by the local government facilities of Mie Prefecture. These images present a rich tapestry of the region's road network, capturing a diverse array of scenarios that include bustling urban intersections, serene rural roads, and challenging off-road segments. This variety ensures that our dataset is not only extensive but also reflective of real-world conditions.

These images were not uniform in their composition; they depicted varied lighting conditions and a range of complexities, such as areas with significant glare, shadowy patches, deteriorated road sections, and visible traffic signs. Each of these elements introduces particular challenges for segmentation models, making the dataset an excellent test bed for evaluating the robustness of different algorithms. The diversity within the dataset is critical, as it pushes the boundaries of what segmentation models must handle and helps in developing solutions that are adaptable to a wide range of real-world scenarios.

The crux of our segmentation task lies in the accurate identification of road surface markings, which necessitates detailed pixel-wise labeling of each image. These labels, or binary masks, are created such that each pixel is assigned a value of 1 if it corresponds to a road surface marking, and 0 otherwise. This meticulous process of labeling is pivotal as it directly influences the training effectiveness and eventual performance of our segmentation models.

In our initial efforts, as documented in our first publication, we developed a specialized anno-

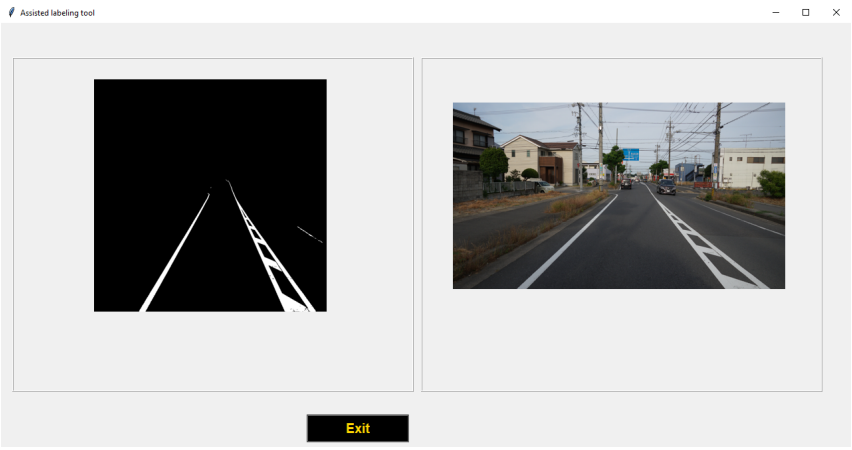


Figure 5.1: The user interface of Labeling Tool (This figure shows the primary model prediction alongside the original image allowing the user to judge if it can be used as a valid label.)

tation tool tailored for this task. With this tool, we managed to manually label approximately 400 images. These labeled images were instrumental in training a U-Net model, which subsequently achieved a Dice score of 78.90% on our designated validation set. However, having only 400 labeled images meant we were tapping into merely about 3% of the potential insights the full dataset could offer.

To leverage the entire dataset effectively, we needed to scale our labeling efforts without compromising the accuracy and reliability of the labels. We approached this by employing the U-Net model we had trained to perform inference across all 13,000 images. This step was crucial as it automated the initial phase of labeling, identifying areas likely to be road surface markings based on learned patterns. To refine this process, we then designed an assisted labeling tool that displayed these preliminary results alongside the original images. This setup allowed users to review and adjust the automated labels where necessary, enhancing the accuracy of the final labels. The labeling tool discussed is shown in figure5.1.

This semi-automated approach to labeling proved highly efficient. It allowed us to expand our dataset to include 12,000 accurately labeled images. During this process, approximately 1,000 images were discarded as they were deemed unsuitable for accurate annotations due to their complexity or poor image quality, which could lead to ambiguous or incorrect labels. This rigorous filtering process ensured that the quality of our dataset was maintained, providing a solid foundation for subsequent analyses and model training.

The expanded and refined dataset not only enriches our research but also sets a benchmark for future studies in road surface marking segmentation. By meticulously assembling and curating this dataset, we have created a valuable resource that will aid in the development of more advanced computer vision models, capable of performing with high accuracy and reliability in diverse and challenging real-world conditions. The process of dataset creation, therefore, is not just a preliminary step but a cornerstone that supports the overarching goals of enhancing traffic management systems through improved road surface marking detection and segmentation.

5.3 Overview of Surveyed Segmentation Models

The core objective of segmentation models in the realm of computer vision is to generate a detailed segmentation map as discussed in previous chapters. This map delineates class

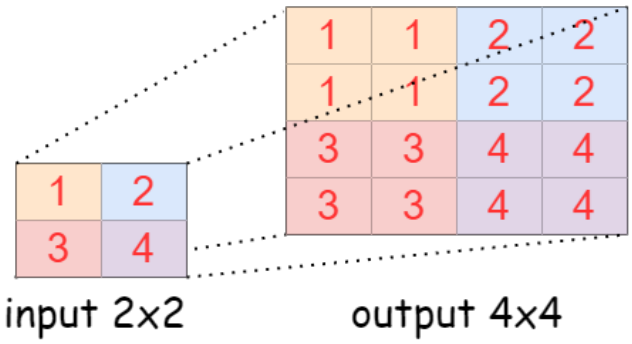


Figure 5.2: Color-coded Up-pool Operation

labels for each pixel within a given input image, effectively partitioning the image according to predefined criteria. In typical implementations, the output of these segmentation networks employs one-hot encoding, where each class label is represented by a unique binary code in the output channels. This results in the number of output channels being equivalent to the number of classes being identified. For our specific application, where the focus is singularly on road surface markings, the output from each of the models we trained is a two-dimensional binary mask, indicating the presence or absence of the marking.

Segmentation models are generally structured around an encoder/decoder framework. Initially, the input image undergoes a series of downsampling operations through convolutional layers, which reduce its resolution while extracting and condensing the feature information. This low-resolution feature map encapsulates the essential details needed to understand and segment the image but at a reduced scale.

The next phase involves upsampling these compressed features back to the original resolution of the input image, forming the precise segmentation map. This is where the models diverge significantly from typical convolutional operations.

Upsampling in segmentation networks is a pivotal operation, reintroducing spatial dimensions that were compressed during downsampling. It allows the network to project the learned abstract features back onto the high-resolution grid necessary for pixel-level classification. Two primary techniques for upsampling are prevalent among the models we studied.

Up-pooling increases the spatial dimensions of the input by scaling up each value to cover a larger area on the output feature map. One common variant of up-pooling is "Nearest Neighbor" upsampling, which replicates the input values across the expanded area, as illustrated in our referenced Figure 5.2 and Figure 5.3. This method, while straightforward, can sometimes lead to a blocky, less precise restoration of image details.

Transpose Convolution, More sophisticated than up-pooling, transpose convolution, also known as deconvolution, involves reversing the forward convolution process. Unlike standard convolution that aggregates input data through a filter to produce a single output at each location, transpose convolution takes a single input from the compressed feature map, applies the filter, and spreads the result across a larger area in the output. This process is depicted in 2D form in Figure 5.4 and simplified further in a 1D representation in Figure 5.5.

While transpose convolution is generally preferred for its ability to learn optimal upsampling from the data itself, it is not without drawbacks. One notable issue is the potential for "checkerboard artifacts," where the overlapping of deconvolved outputs can lead to uneven or striated

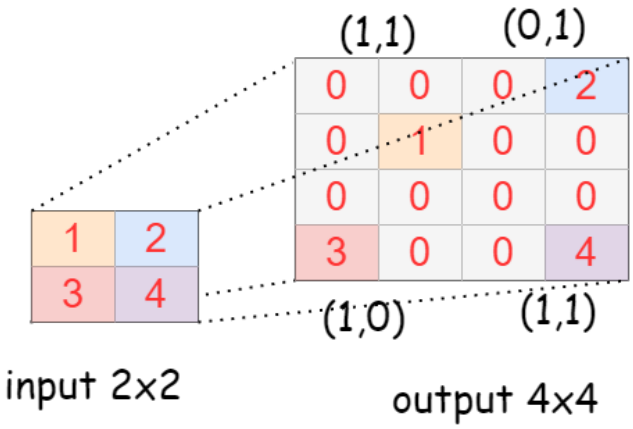


Figure 5.3: Up-pool Operation in Decoder Part of Seg-Net

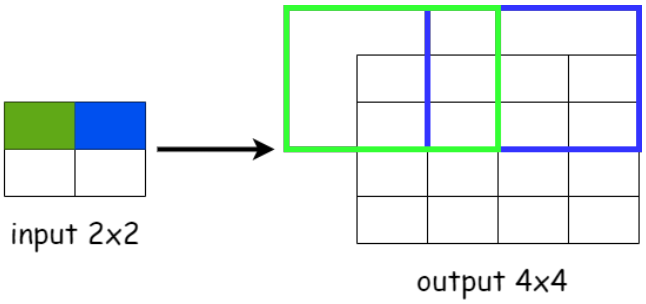


Figure 5.4: Transpose Convolution Operation (Ppadding=1, Stride=2)

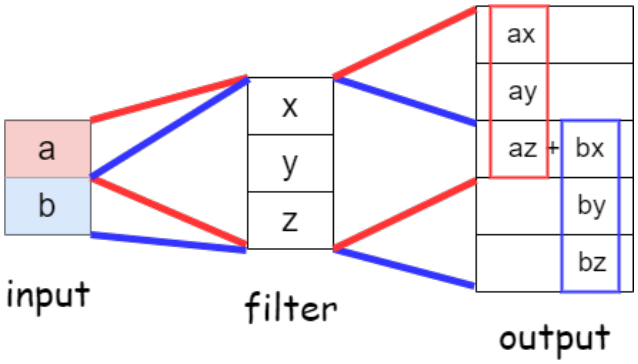


Figure 5.5: Operation of 1D Transpose Convolution for Purpose of Clarification

patterns in the reconstructed image. This artifact arises when the spread regions from adjacent activations overlap, a situation exacerbated by certain stride and kernel size configurations in the transpose convolution layers.

To mitigate issues like checkerboard artifacts, careful consideration is needed when configuring the parameters of transpose convolutions. Ensuring that the outputs do not overlap or adjusting the stride and padding settings can help in producing smoother, more uniform segmentation maps.

As we transition into the subsequent sections that detail the training and testing of these models, we will explore how each model handles the upsampling challenges and their efficacy in providing high-fidelity segmentation results. Our comparative analysis will focus not only on the architectural strengths and weaknesses of each model but also on their practical performance in segmenting road surface markings from diverse urban and rural scenes. This discussion will culminate in a robust understanding of which segmentation strategies are most effective and under what conditions, providing valuable insights for future applications in automated image segmentation.

5.4 Training and Testing

The training of segmentation models requires significant computational resources, especially when working with high-resolution datasets like ours, which includes 12,000 images. Managing these resources efficiently is crucial to the success of the project. We divided the dataset into several parts to facilitate a balanced approach to training and evaluation. Specifically, we allocated 10,000 images for training, which allows the models to learn from a diverse set of data, representing various road conditions and scenarios. We reserved 1,000 images for validation and another 1,000 for testing. This separation ensures that we can periodically assess the model's performance against data it hasn't seen during training, which is critical for tuning the models and preventing overfitting. Figure 5.6 shows the process of training each segmentation model.

Training segmentation models is computationally expensive, particularly due to the high resolution of our images. Each image is resized to 1600x1600 pixels, requiring significant GPU resources to process. We conducted our training on a GeForce RTX 3090 GPU using Pytorch, which provided the necessary computational power to handle the dataset efficiently.

During the training process, we implemented an early stopping mechanism to optimize computational efforts. If a model showed no improvement on the test set after five consecutive epochs, we halted further training. This approach helps in conserving resources and avoiding unnecessary computations that do not yield better results. For the Mask-RCNN model, training was specifically stopped once we observed that the training loss plateaued, indicating that continuing training would likely not result in further gains.

The loss function used for training is a critical component of our setup. It is composed of a weighted sum of Dice loss and Focal loss. This combination was chosen based on its proven effectiveness in previous work where it was beneficial for datasets with imbalanced classes. In our case, the dataset contains significantly more background pixels than target pixels, which classifies the target class of road surface markings as a "hard" class due to its rarity. The Focal Loss is particularly useful in such scenarios as it helps to focus more on hard-to-classify examples by adjusting the focus parameter, thus improving model performance on minority classes. The Dice Loss, on the other hand, is crucial for ensuring good boundary detection, a

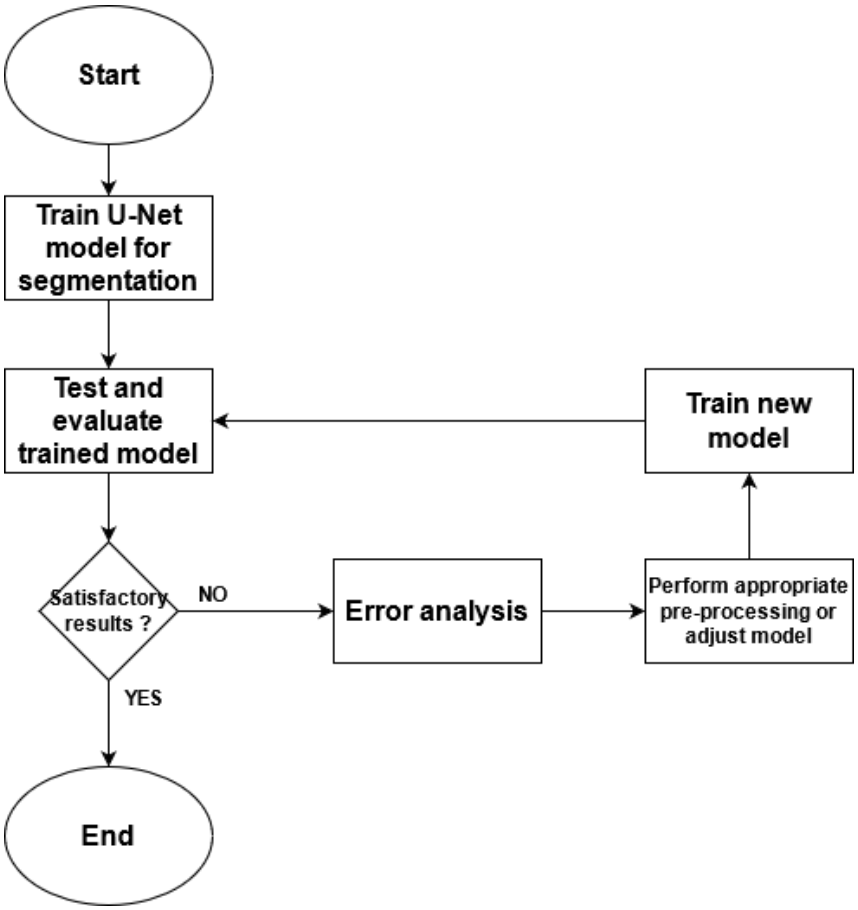


Figure 5.6: Process of Training Segmentation Models

key factor in the quality evaluation of road surface markings.

The formulas for the Dice Loss and Focal Loss are critical to understanding how they influence model training: Dice Loss is calculated using the formula provided in Equation 5.1, where pi represents the predicted pixel values, gi is the ground truth, and N is the total number of pixels. Focal Loss is detailed from Equation 5.2 to Equation 5.5, with p representing the predicted pixel values, y the true pixel values, and α, γ are hyperparameters set to 2 and 10, respectively, to adjust the model's sensitivity to the target class.

$$D = \frac{2\sum_i^N g_i p_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \tag{5.1}$$

$$CrrossEntropy(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases} \tag{5.2}$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{5.3}$$

$$CrrossEntropy(p, y) = CrrossEntropy(p_t) = -\log(p_t) \tag{5.4}$$

$$FocalLoss(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \tag{5.5}$$

Through this detailed setup and careful monitoring, we aim to develop robust models capable of accurately segmenting road surface markings from diverse images, thereby enhancing the reliability and efficiency of traffic management systems.

5.5 Results and Analysis

Following a rigorous training and testing regime, the performance of our segmentation models was carefully analyzed across both the test and validation sets. We employed a variety of metrics to gain a comprehensive understanding of each model's strengths and weaknesses. The findings from this analysis were systematically presented in table 7.1 and table 5.2, providing a clear comparative perspective.

Table 5.1: Performance Results (Test Set)

Encoder type	mean IoU score	Dice score	Accuracy score	Number of parameters
U-Net models				
VGG-16	38.80%	88.24%	99.14%	23M
VGG-19	38.78%	88.25%	99.14%	29M
Inception-v4	40.58%	88.97%	99.15%	48M
Resnet-18	38.91%	87.84%	99.13%	14M
PSPNet models				
VGG-16	42.97%	85.83%	99.10%	15M
VGG-19	41.91%	85.78%	99.10%	20M
Inception-v4	40.43%	84.84%	99.09%	41M
Resnet-18	38.79%	83.61%	99.07%	11M
DeepLab models				
Resnet-18	39.29%	85.50%	97.76%	15M
Resnet-34	40.63%	85.95%	97.80%	26M
FCN models				
Resnet-50	39.17%	84.20%	99.05%	32M
Mask RCNN				
VGG-16	30.16%	39.53%	96.16%	44M
Seg-Net				
VGG-16	38.88%	84.68%	99.09%	29M
Link-Net				
Resnet-34	38.75%	87.73%	99.13%	21M

Consistency Between Testing and Validation Interestingly, our analysis revealed that the performance disparities between the test results and the validation results were minimal. This consistency is indicative of the robustness of the training regimen, which included monitoring the Dice score at each epoch on the test set and halting training when improvements in the Dice score ceased. This approach proved effective in mitigating the risk of overfitting, which is a common challenge in machine learning models.

Challenges with Mask-RCNN In the case of the Mask-RCNN model, however, the outcomes were less favorable. This model struggled to converge as effectively as others and exhibited a noticeable decline in Dice score performance when transitioning from the test set to the

Table 5.2: Performance Results (Validation Set)

Encoder type	mean IoU score	Dice score	Accuracy score	Number of parameters
U-Net models				
VGG-16	39.60%	88.43%	99.10%	23M
VGG-19	39.56%	88.17%	99.15%	29M
Inception-v4	41.46%	88.91%	99.15%	48M
Resnet-18	39.71%	87.56%	99.13%	14M
PSPNet models				
VGG-16	43.59%	85.73%	99.11%	15M
VGG-19	42.58%	85.71%	99.11%	20M
Inception-v4	41.16%	84.86%	99.10%	41M
Resnet-18	39.58%	83.58%	99.07%	11M
DeepLab models				
Resnet-18	40.07%	85.57%	97.79%	15M
Resnet-34	41.28%	85.90%	97.81%	26M
FCN models				
Resnet-50	39.96%	83.98%	99.05%	32M
Mask RCNN				
VGG-16	31.44%	27.93%	96.25%	44M
Seg-Net				
VGG-16	39.70%	84.89%	99.09%	29M
Link-Net				
Resnet-34	39.53%	87.50%	99.13%	21M

validation set. This drop suggests that the training strategy, which focused predominantly on tracking training loss, may have inadvertently led to overfitting. Such a scenario underscores the need for a more nuanced approach to training loss monitoring and model evaluation.

Superiority of U-Net in Dice Performance The U-Net models demonstrated a significant advantage in terms of the Dice score, reaffirming the efficacy of the concatenation method used in these models to preserve information through the encoder and decoder segments of the network. This method minimizes information loss and enhances the model's ability to accurately segment images based on learned features.

PSP-Net and Spatial Pooling Efficiency On another front, PSP-Net models excelled in achieving higher mean Intersection over Union (IoU) scores. This metric is particularly telling of a model's ability to precisely delineate the area of overlap between the predicted segmentation and the ground truth. The success of PSP-Net in this area can be attributed to its pyramid spatial pooling technique, which effectively captures contextual information at various scales, thus enhancing segmentation accuracy.

Link-Net's Efficiency and Simplicity Link-Net also emerged as a noteworthy model, surpassing the 87% Dice score threshold with only 21 million parameters. This result is particularly impressive, showcasing Link-Net's ability to offer a favorable balance between model simplicity and segmentation efficacy, compared to more complex models like Seg-Net, DeepLab, and PSP-Net.

Limitations of Accuracy as a Metric Despite the high accuracy scores (above 98%) achieved

by all models, this metric proved to be a less reliable indicator of model performance for our specific segmentation task. Due to the predominance of background pixels over target pixels in our dataset, a model predicting all pixels as background could still achieve high accuracy while failing at meaningful segmentation of road surface markings.

The results across different encoder types in the U-Net models show VGG-16 outperforming VGG-19 with a Dice score of 88.43% and an accuracy of 99.10% while only using 23M parameters. This suggests a better efficiency of VGG-16 in utilizing fewer parameters to achieve slightly better segmentation results. On the other hand, VGG-19, with 29M parameters, showed a minimal decrease in performance which raises questions about the cost-benefit of the additional complexity.

Inceptionv4 used in U-Net models stands out with a higher mean IoU score of 41.46% and a Dice score of 88.91%, making it one of the best performers in terms of balancing segmentation accuracy and boundary precision, albeit at a higher computational cost given its 48M parameters.

Resnet-18, while having the least parameters among the U-Net models at 14M, shows commendable performance with a Dice score of 87.56% and an accuracy of 99.13% . This highlights its efficiency in handling segmentation tasks with fewer resources, though it falls slightly behind Inceptionv4 and VGG-16 in Dice performance.

For the PSPNet models, the VGG-16 encoder leads with a mean IoU of 43.59% and a Dice score of 85.73%, which is notable for its higher spatial overlap accuracy. This model, with only 15M parameters, provides a strong argument for its use in applications where IoU is a critical metric. Conversely, the VGG-19 encoder doesn't significantly improve on this performance despite having more parameters.

Inceptionv4 in PSPNet models doesn't perform as well as it does in U-Net configurations, which could suggest that the network architecture may not synergize as well with the PSP pooling strategies, reflected in both lower IoU and Dice scores.

Resnet-18 in PSPNet shows the lowest performance among its group, which might indicate limitations in its ability to upscale detailed features necessary for precise segmentation in complex pooling structures like those in PSPNet.

DeepLab models using Resnet as an encoder show a moderate performance with Resnet-34 slightly leading over Resnet-18, suggesting that increased depth can help in capturing more complex features beneficial for segmentation tasks, but the difference isn't stark, indicating a potential area of diminishing returns when adding depth.

FCN models with Resnet-50 didn't achieve as high performance as some of the simpler U-Net configurations, which might be due to the inherent challenges in upscaling in FCN architectures which can dilute detailed features necessary for high-quality segmentation.

Mask RCNN with VGG-16 showed significantly lower performance across all metrics, which may reflect challenges in adapting the Mask RCNN architecture for this particular type of segmentation task, possibly due to its typical use in instance segmentation which focuses on object detection rather than pixel-wise classification.

Seg-Net using VGG-16 and Link-Net with Resnet-34 show performances that are comparable to simpler U-Net models, indicating that while these models are capable, they do not necessarily provide a clear advantage over the U-Net with a VGG-16 encoder in terms of efficiency and

efficacy for this dataset.

These results contribute valuable insights into the trade-offs between model complexity, parameter count, and performance across various metrics. They underscore the importance of selecting the right model architecture tailored to specific needs of the segmentation task at hand, balancing precision, efficiency, and computational demands.

This survey not only serves as a detailed reference for the research community interested in segmentation models but also highlights our efforts to refine solutions for road surface marking detection. The insights derived from this study pave the way for future endeavors in traffic landmark quality evaluation, crucial for enhancing road safety and maintenance protocols.

Chapter 6

Enhancing Road Surface Marking Reconstruction Through Synthetic Noise and Autoencoder Techniques

6.1 Introduction

Building upon the established necessity of maintaining visible and effective road surface markings, this chapter introduces a novel autoencoder-based solution designed to reconstruct these markings from a state of degradation. This approach marks a significant advancement in our ongoing efforts to automate and enhance the accuracy of road marking assessments. We can see some examples of road markings in their original state in Figure 6.1 which is the state we will try to restore the deteriorated markings to. Some examples of deteriorated markings can be seen in Figure 6.2.

Previous methods, while effective within their specific contexts, often relied heavily on subjective assessments or were constrained by the granularity of data derived from traditional imaging techniques. To address these limitations, this research leverages a more objective, data-driven approach through the use of synthetic noise data, simulating the natural deterioration processes of road markings. This technique allows for a controlled replication of various degradation conditions, providing a robust training environment for our models.

The introduction of an autoencoder model in this study is predicated on its ability to learn efficient data codings in an unsupervised manner, specifically for the task of predicting the degradation state of road markings. Autoencoders are particularly suited for this kind of application because they excel in capturing and reconstructing the underlying patterns in the data, even when the data has been corrupted or is incomplete.

In the deployment of this model, we first utilize a pre-existing segmentation model to isolate the road markings from their backgrounds effectively, as shown in Figure 6.3. Following this segmentation, synthetic noise is introduced to the isolated markings as seen in Figure 6.4, mimicking real-world deterioration effects such as fading, cracking, and wear. The autoencoder is then trained not only to filter out this noise and reconstruct the road markings to their near-original state but also to refine the model's ability to generalize from synthetic to real-world conditions. This chapter details the process of developing the autoencoder architecture, the training regimen adopted, and the synthetic noise models used. It discusses the comprehensive

Figure 6.1: Example of Road Surface Markings (good Condition)

Figure 6.2: Example of Road Surface Markings (Deteriorated Condition)

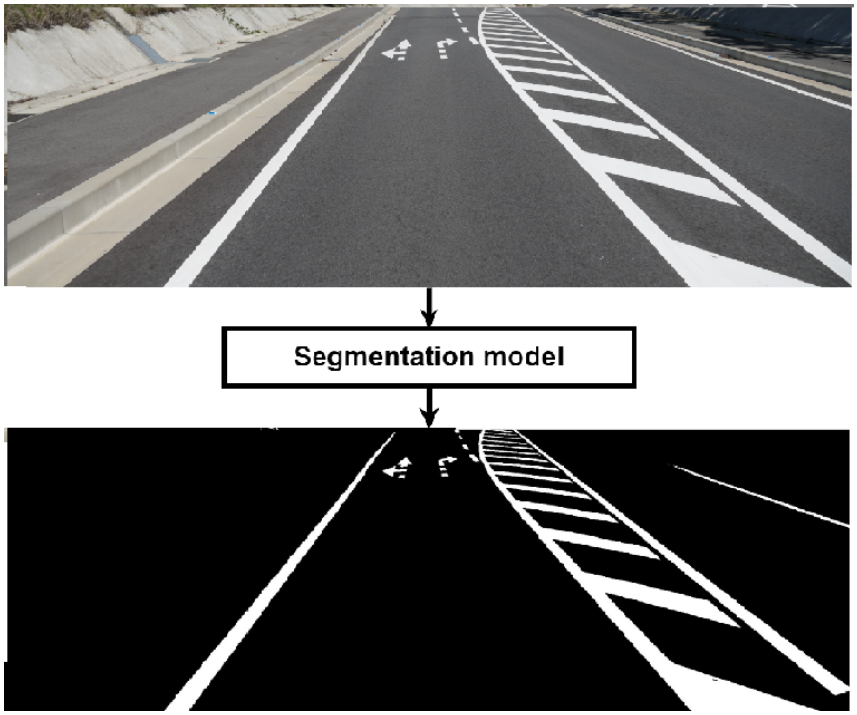


Figure 6.3: Example of Segmentation Step Performed on Original Images

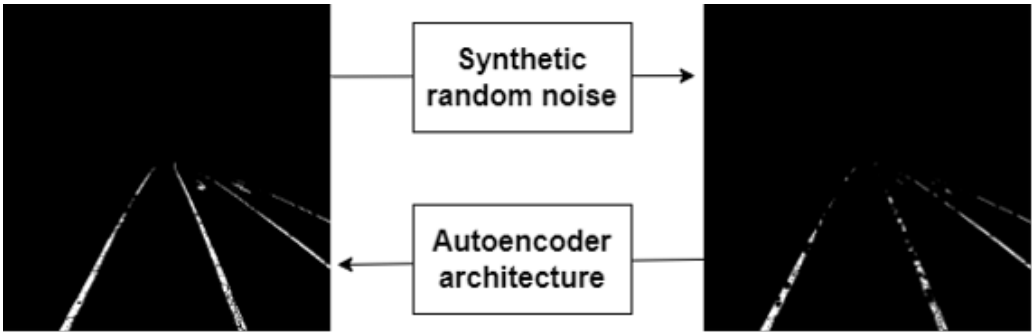


Figure 6.4: General Structure of Pipeline Proposed by Autoencoder Approach

testing and validation of the model, emphasizing the challenges encountered when applying the trained model to real-world data. The outcomes of these experiments are crucial in illustrating the potential and limitations of using deep learning techniques for the reconstruction and ongoing assessment of road surface markings.

By advancing this autoencoder approach, this research contributes to the broader field of traffic management by providing a scalable, efficient tool for monitoring and restoring the clarity and visibility of road markings, thereby enhancing road safety and traffic flow efficiency.

6.2 Autoencoder Model for Road Marking Reconstruction

Building on the foundational dataset described earlier, the focus of this section of the thesis is on a novel autoencoder model designed specifically for the reconstruction of road surface markings. This model uses a sophisticated combination of U-Net and Pyramid Scene Parsing Network architectures to handle synthetic noise data simulating real-world degradation of road markings, ensuring the process approaches real-life conditions as closely as possible.

6.2.1 Circle Noise

In the development of an autoencoder aimed at reconstructing road surface markings from synthetic noise-augmented data, we employed a strategic approach to simulate the real-world deterioration of these markings. This step was crucial in preparing the models to handle actual road conditions effectively. Deterioration in road markings can stem from various factors including weather exposure, the constant stress of vehicular traffic, and environmental effects, all of which degrade the quality and visibility of these crucial navigational aids.

To introduce realistic challenges into our training dataset, we implemented a method of synthetic noise generation that mimics these common forms of deterioration. Specifically, we started by adding random black circles to approximately half of the dataset images. These circles serve as a proxy for common physical damages such as holes, cracks, and large chips that are frequently caused by the heavy wear and frequent impact from vehicles. Such defects not only compromise the functionality of road markings but also pose safety risks by reducing the clarity of road layouts for drivers.

The choice to simulate such imperfections was informed by observations and studies indicating that these types of physical deteriorations are among the most common and impactful on road safety. By integrating these simulated defects into our training images, the autoencoder is tasked with identifying and reconstructing markings as they might appear post-damage, thus enhancing its capability to generalize from synthetic training scenarios to real-world conditions.

This synthetic deterioration approach helps the model to better understand the varied textures and contrasts that characterize damaged road surfaces. Training on this enhanced dataset ensures that the autoencoder develops robust pattern recognition abilities, crucial for the accurate reconstruction of degraded markings. The logic behind using random placement and sizing of the black circles was to ensure that the model does not overfit to specific patterns of damage, but rather learns to recognize and react to a wide range of imperfection scenarios.

Furthermore, this method of introducing synthetic noise is beneficial for testing the model's effectiveness in differentiating between actual road markings and similar-looking anomalies.

This differentiation is vital as it directly affects the model's practical application in real-world settings, where a multitude of extraneous visual elements can potentially be mistaken for road markings by less discerning models.

The introduction of synthetic noise to mimic deterioration is a nuanced process, requiring careful balance to ensure that the generated images remain realistic and represent a broad spectrum of potential real-world conditions. The success of this approach hinges on the ability of the model to learn from these complexities and still perform effectively when confronted with actual degraded road markings. This training methodology not only enhances the model's accuracy but also its applicability in diverse operational environments where road markings vary significantly in design, color, and condition.

By rigorously training the autoencoder with these synthetically altered images, we aim to equip it with the necessary skills to accurately identify and reconstruct road markings from varied states of decay, thus supporting efforts to maintain clear and reliable road signage. This is particularly crucial for the safety of all road users and the efficiency of traffic management systems, especially as we move towards more automated and assisted driving technologies. An example of the application of this noise can be seen in Figure 6.5.

6.2.2 Erosion and Gaussian Noise

The integration of erosion and Gaussian noise into the dataset of road surface markings encapsulates a deliberate effort to simulate more nuanced forms of degradation. Erosion, in this context, replicates the wear and tear inflicted by continuous vehicular movement and natural wear factors. The erosion effect is implemented algorithmically to mimic the gradual loss of marking material, which can result from various factors including friction from tires, water erosion due to rain, and even the slow degradation from chemicals used in road maintenance.

Gaussian noise, on the other hand, introduces a random, variable distortion that simulates environmental noise—such as dust, fog, or spray from other vehicles—that can obscure or alter the appearance of road markings. This type of noise is characterized by its bell-shaped probability distribution, which effectively introduces variations in pixel intensity across the image the range of which can be seen in Figure 6.6, thus emulating the random visual noise found in real-world conditions. The combined application of erosion and Gaussian noise creates a richly textured simulation of road markings that have been subjected to a diverse array of degrading influences, enhancing the realism of the training set. An example of the application of this noise can be seen in Figure 6.7.

This sophisticated approach to synthetic data generation not only broadens the model's exposure to various degradation scenarios but also significantly bolsters its ability to generalize from training data to real-world applications. As the autoencoder is trained on these artificially deteriorated images, it learns to pinpoint and reverse these changes, honing a set of capabilities essential for real-time applications in road maintenance and safety. The model is thus trained not only to recognize intact and slightly worn markings but also to detect and interpret severely compromised road markings that might otherwise be overlooked.

Training the autoencoder on this diversified dataset ensures that it learns to recognize and rectify a wide spectrum of degradation states. This comprehensive understanding allows the model to apply its learned capabilities to actual road conditions with greater accuracy and effectiveness. The ultimate goal of this training approach is to create an automated system

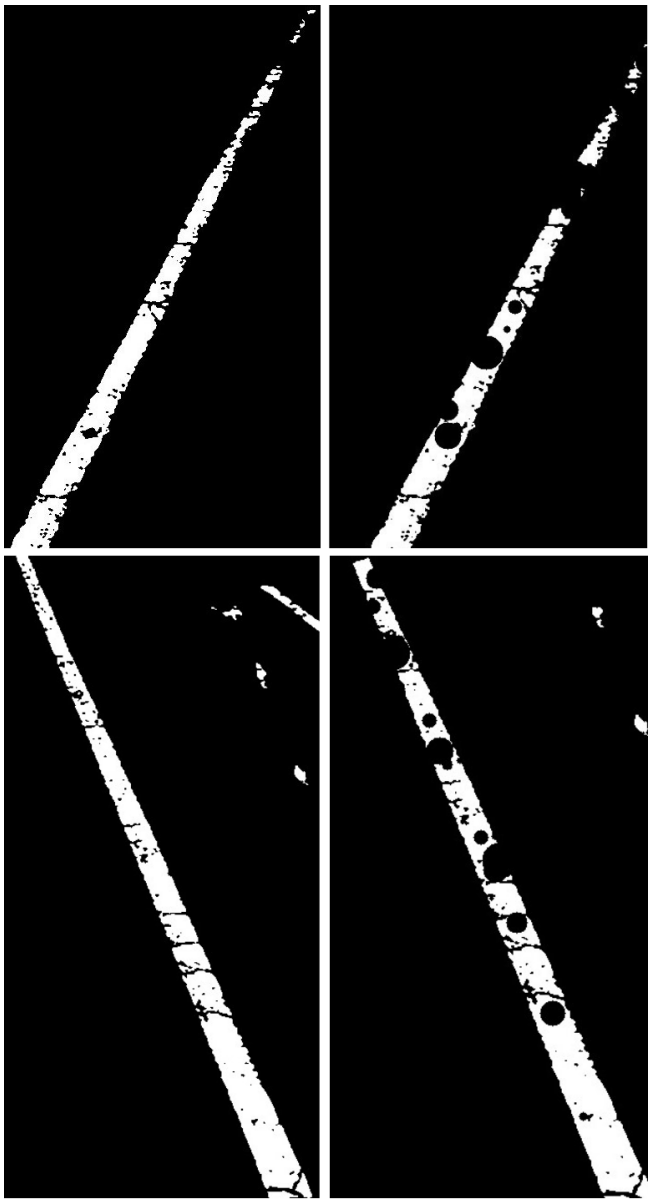


Figure 6.5: Example of Introducing Random Circles Noise for Synthetic Dataset (The left figure is the original segment of the road marking and the right one is noised image.)

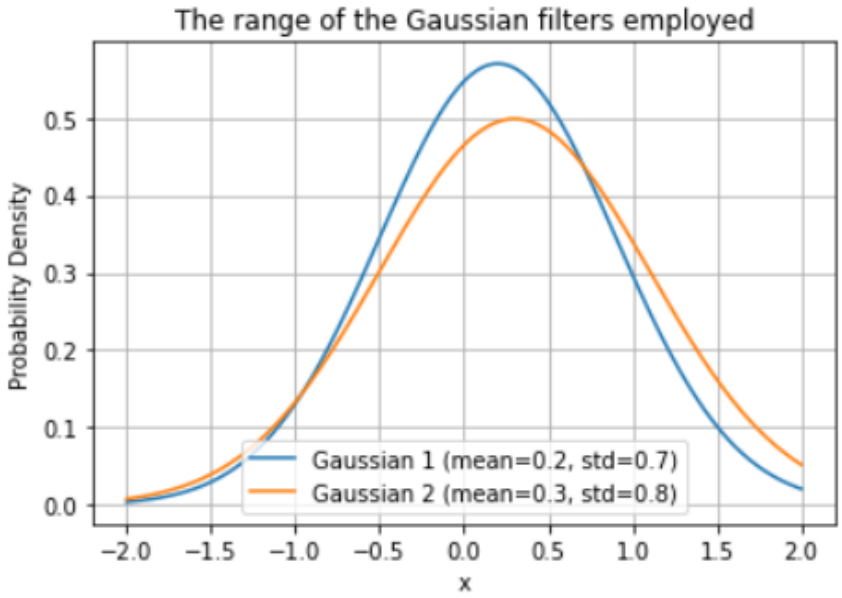


Figure 6.6: Plot of Two Limits of Range of Gaussians (The outside of this range were observed to produce noise that was too unrealistic.)

that can reliably analyze and restore clarity to road markings, thereby contributing significantly to road safety and the effectiveness of traffic management systems.

In essence, this methodical training strategy is not merely about teaching the autoencoder to recognize patterns but about empowering it to apply its learned insights to actively improve real-world conditions, thus closing the gap between theoretical models and practical applications. The ability of the system to adapt to the varied and unpredictable conditions of real-world road environments marks a significant step forward in the application of deep learning technologies to the field of public infrastructure maintenance.

$$M(x,y) = \begin{cases} 1 & \text{if } I(x,y) = \text{white} \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

where $M(x,y)$ is a binary mask indicating the position of potential noise addition, $I(x,y)$ is the pixel intensity at coordinates (x,y) in the input mask, and "white" represents the target color for noise addition.

$$C(x,y,r) = \begin{cases} 1 & \text{if } \sqrt{(x-x_c)^2 + (y-y_c)^2} \leq r \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

where $C(x,y,r)$ represents a circle of radius r centered at (x_c, y_c), and the condition inside checks if a point (x,y) lies within the circle.

$$r \sim \text{Uniform}(r_{\min}, r_{\max}) \tag{6.3}$$

where r is the radius of a circle, and it is chosen randomly from a uniform distribution between $r_{\min}$ and $r_{\max}$, defining the range of possible circle sizes.

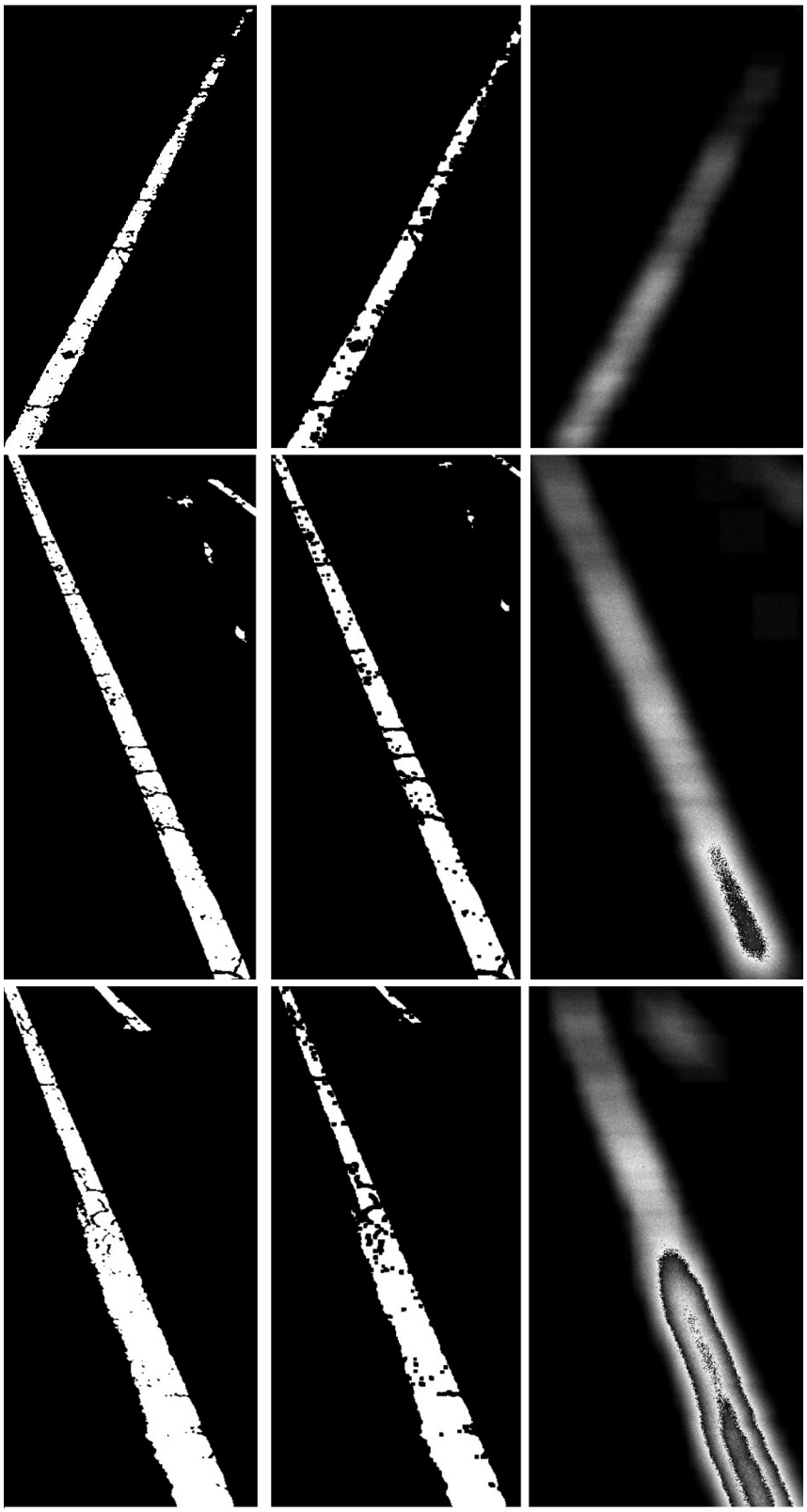


Figure 6.7: Examples from the Second Dataset (The left images are the original images used as labels for reconstruction, the middle ones are the results of erosion with a random sized filter, and the right ones show the result of introducing Gaussian noise to the erosion result, both the erosion result and the Gaussian result are used in the dataset.)

$$I_{\text{noisy}}(x,y) = \begin{cases} 0 & \text{if } M(x,y) \cdot C(x,y,r) = 1 \\ I(x,y) & \text{otherwise} \end{cases} \tag{6.4}$$

where $I_{\text{noisy}}(x,y)$ is the resulting image after applying the noise. This equation applies a black pixel where the mask and circle overlap, simulating the effect of a hole or crack.

These equations effectively describe the process of adding synthetic noise in the form of black circles to white areas of an input mask, capturing a method to simulate physical damage on road markings. The randomness in circle placement and size adds variability, mimicking real-world deterioration more closely.

$$I_{\text{norm}} = \frac{I - \mu}{\sigma} \tag{6.5}$$

where I_{norm} represents the normalized input image, I is the original input image, μ is the mean pixel intensity of the training dataset, and σ is the standard deviation of pixel intensities. This normalization step ensures the model processes data within a similar range, enhancing the training stability.

$$I_{\text{noisy}} = I_{\text{norm}} + N(0, \sigma_n^2) \tag{6.6}$$

where I_{noisy} is the image after adding synthetic noise, and $N(0, \sigma_n^2)$ represents Gaussian noise with a mean of 0 and a variance of σ_n^2. This equation simulates environmental noise factors affecting the road markings.

$$\hat{I} = f_{\text{AE}}(I_{\text{noisy}}) \tag{6.7}$$

where $\hat{I}$ is the reconstructed image from the autoencoder f_{AE}, aiming to denoise and restore the original road marking features from the noisy input I_{noisy}.

These equations collectively form the mathematical foundation of the autoencoder model's implementation, providing a structured approach to training and evaluating the model's ability to reconstruct degraded road markings. The normalization and noise addition prepare the model for real-world scenarios, while the loss function and PSNR offer quantitative metrics to assess model performance.

6.2.3 Training and Testing

The rigorous training regimen adopted for these models was meticulously crafted to maximize their efficiency and accuracy in segmenting and reconstructing road surface markings. By allocating 80% of the dataset to training, the models were exposed to a diverse array of scenarios, encompassing various stages of road marking degradation. This extensive training set was crucial for the models to learn a wide range of features and deterioration patterns, ensuring comprehensive learning coverage. The remaining 20% of the dataset, used for testing, served as a new, unseen set of data to rigorously assess the models' generalization capabilities. This division was strategic, aimed at balancing the depth of learning with the necessity of validation against unbiased data.

Monitoring the models' training progress was executed with precision, utilizing the Dice score at each epoch's conclusion. The Dice score, a statistical tool measuring the overlap between

the predicted segmentation and the actual annotation, was pivotal in evaluating the models' segmentation effectiveness. This metric is particularly valuable in scenarios where the exact delineation of the object (in this case, road markings) is critical for the model's operational success.

To mitigate the risk of overfitting—a common challenge in neural network training where models learn to perform well on training data but poorly on any new data—training cessation protocols were implemented. Specifically, if the model's performance did not improve after five consecutive epochs, training was halted. This precaution was essential to ensure that the model did not simply memorize the training data's details but developed a robust capability to generalize across different datasets.

The choice of loss functions was pivotal in refining the model's training. The composite loss function, combining Dice loss and Focal loss, was strategically selected to enhance model performance on two fronts. Dice loss was crucial for its ability to promote precise boundary detection, enhancing the model's capability to delineate road markings accurately from their surroundings. Accurate boundary detection is vital for effective road marking recognition, directly influencing the quality of the reconstructions produced by the autoencoder.

Focal loss, on the other hand, addressed the challenge of class imbalance prevalent in the dataset. In road marking datasets, the majority of pixels represent the background rather than the road markings themselves, which could lead models to prioritize the more frequent background class at the expense of the critical, less frequent road marking class. Focal loss adjusts the focus towards harder-to-classify instances, ensuring that these crucial but less frequent features are not overshadowed during the learning process.

The entire computational workload, including the training of these sophisticated models, was managed using the PyTorch framework, renowned for its flexibility and efficiency in handling deep learning applications. The use of NVIDIA's GeForce RTX 3090 GPU was instrumental in managing the substantial computational demands of this project. The GPU's robust processing capabilities allowed for rapid handling of large datasets and complex model architectures, significantly reducing training time and enhancing the iterative testing process. This hardware setup not only provided the raw power needed to process extensive data volumes but also supported the advanced computational tasks required to train deep learning models effectively.

This detailed approach to training setup, model monitoring, and computational handling forms a comprehensive strategy aimed at developing highly effective autoencoder models capable of recognizing and reconstructing deteriorated road markings with high accuracy, thereby pushing the boundaries of what can be achieved with automated systems in road safety and infrastructure maintenance. A diagram illustrating the steps for training our models can be seen in Figure 6.8. In summary, this methodology leverages advanced synthetic techniques and deep learning architectures to develop a robust system for road marking detection and reconstruction. By simulating real-world degradation in a controlled environment and training models to correct these imperfections, the project aims to significantly enhance the clarity and visibility of road markings, thus contributing to safer and more efficient traffic systems.

6.3 Experimental Results and Analysis

In this results and discussion section, we delve into the performance evaluations of our autoencoder models, specifically the U-Net and PSPNet, which were trained on two distinct synthetic

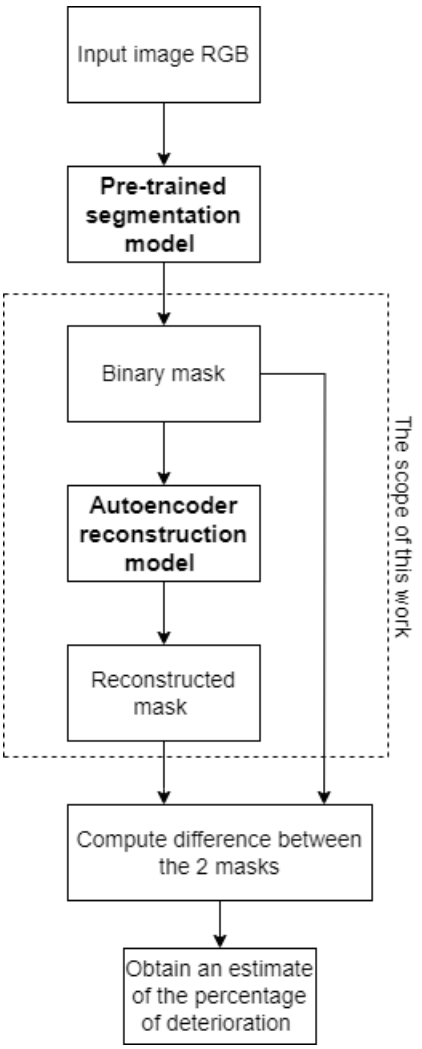


Figure 6.8: Segmentation Step Performed on Original Set

Figure 6.9: Result of PSP-Net Model on Circle Noise Dataset (1)

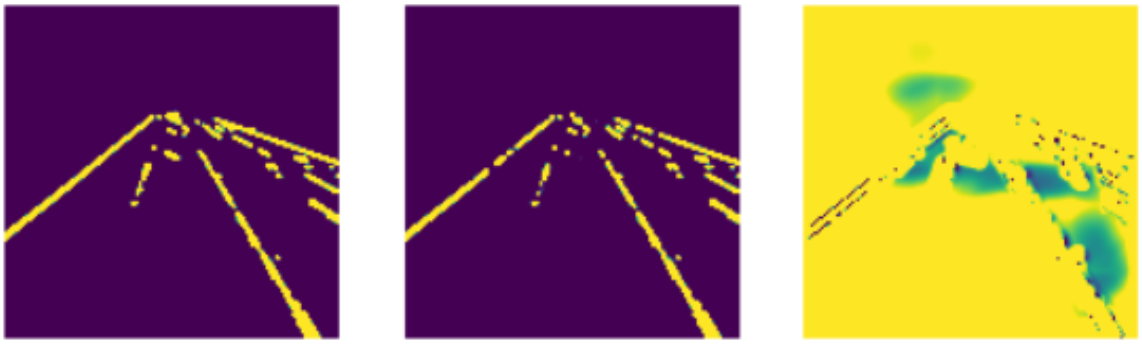


Figure 6.10: Result of PSP-Net Model on Circle Noise Dataset (2)

noise datasets: Circle Noise and Erosion and Gaussian Noise. The insights gleaned from these evaluations are critical in understanding the models' capabilities and limitations, particularly in relation to their application in real-world scenarios. Figures 6.9 to 6.14 show some qualitative results from the test set.

In our exploration of optimal training methods for the autoencoder models, the Circle Noise dataset emerged as a particularly effective tool for enhancing model performance. The addition of random circles to simulate straightforward physical damage, such as holes or general wear, on the road markings, created a set of conditions that was somewhat simplified yet sufficiently challenging to improve the robustness of the models. The controlled nature of this synthetic damage allowed the models, specifically U-Net and PSPNet, to focus on recognizing and repairing clear-cut defects in road markings, which are commonly caused by physical wear and

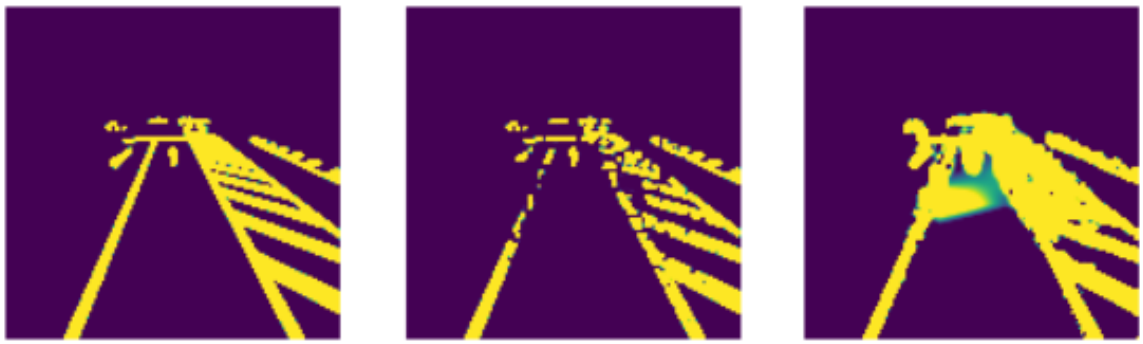


Figure 6.11: Result of PSP-Net Model on Circle Noise Dataset (3)

Figure 6.12: Result of U-Net Model on Circle Noise Dataset (1)

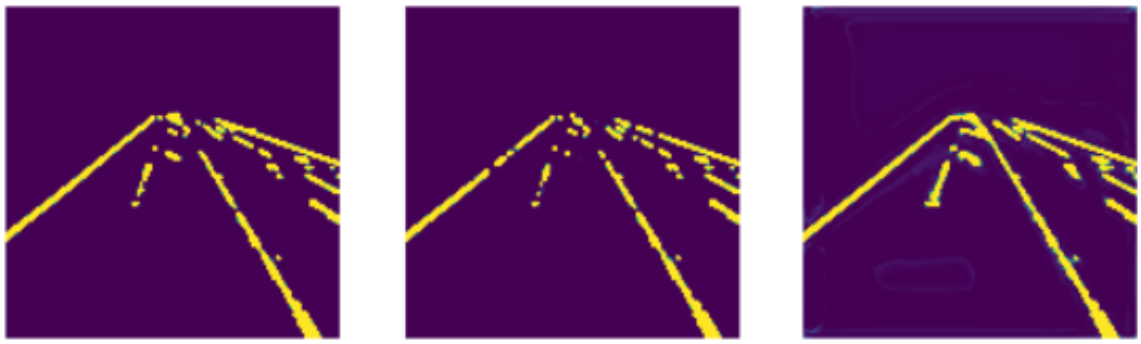


Figure 6.13: Result of U-Net Model on Circle Noise Dataset (2)

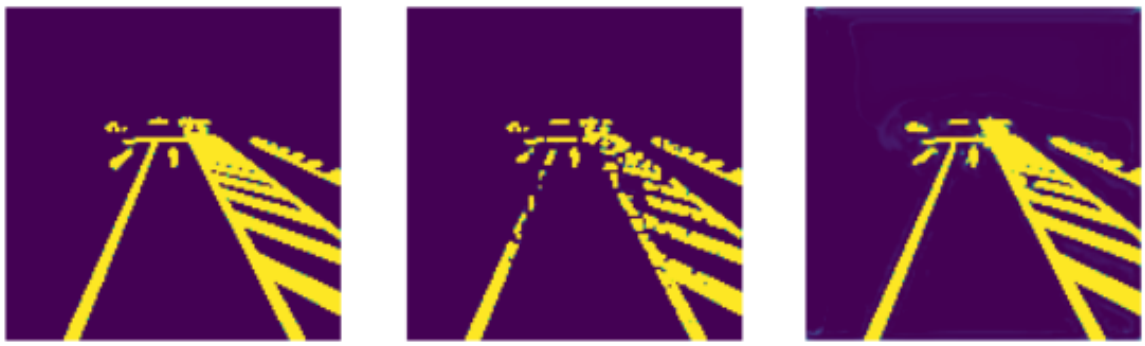


Figure 6.14: Result of U-Net Model on Circle Noise Dataset (3)

tear over time. In figures 6.15 to 6.18 we can see some results of reconstruction performed on real-world samples where the deterioration is due to natural factors. The models trained on the Circle Noise dataset consistently demonstrated superior performance across a variety of metrics when compared to those trained on the Erosion and Gaussian Noise dataset. This latter dataset was designed to replicate a more complex type of degradation. It included effects such as chemical erosion from environmental exposure and the granular noise that might result from scattered debris or water damage. These elements introduce a higher level of randomness and complexity, which challenges the models' ability to accurately reconstruct road markings by requiring them to differentiate between a wider range of degradation patterns and background noise.

Interestingly, while the Erosion and Gaussian Noise dataset provided a robust test of the models' capabilities in handling complex scenarios, the models trained on this dataset tended to achieve higher mean Intersection over Union (mIoU) scores. The mIoU metric is crucial as it evaluates the extent of overlap between the predicted segmentation and the actual markings, regardless of the precision of the boundaries. This suggests that while these models were generally good at identifying the areas where road markings existed, they were not as precise in delineating the exact edges of those markings.

On the other hand, the Dice score, which is a direct measure of the accuracy of the reconstructions against the actual markings, was consistently higher for models trained with the Circle Noise dataset. This indicates that these models were particularly adept at precisely reconstructing the shape and boundaries of road markings, resulting in outputs that were not only more accurate but also visually closer to the actual conditions of the road markings prior to degradation.

Qualitatively, the outputs from models trained on the Circle Noise dataset were visually more appealing and accurate. These reconstructions successfully restored the road markings to a semblance of their original state before deterioration, offering a clear visual confirmation of the models' effectiveness. The fidelity of these reconstructions was evident when comparing the output images from the models against actual photos of road conditions, where the restored markings often appeared nearly indistinguishable from their pre-degraded states.

The findings from this comparative analysis underscore the importance of dataset selection in training models for specific tasks. The simpler, more controlled degradation represented in the Circle Noise dataset allowed for more focused learning and refinement of the models' capabilities in restoring road markings. Meanwhile, the more varied and complex degradation patterns in the Erosion and Gaussian Noise dataset, while providing a rigorous challenge, highlighted the need for models to evolve further to handle such diverse environmental effects effectively.

The comparative analysis between the U-Net and PSPNet models revealed a significant disparity in their performance, particularly when evaluated on the synthetic Circle Noise dataset. U-Net's superior performance in terms of reconstruction accuracy and quality stems largely from its architecture, which is specifically tuned for detailed image segmentation tasks. The U-Net architecture is designed to capture fine details through its symmetric expanding path which recovers the spatial resolution that might be lost during the contracting path. This design allows U-Net to excel in tasks where precise delineation of features is critical, such as the reconstruction of road surface markings where every detail can be crucial for clarity and recognition.

In contrast, PSPNet, known for its strength in integrating contextual information over large

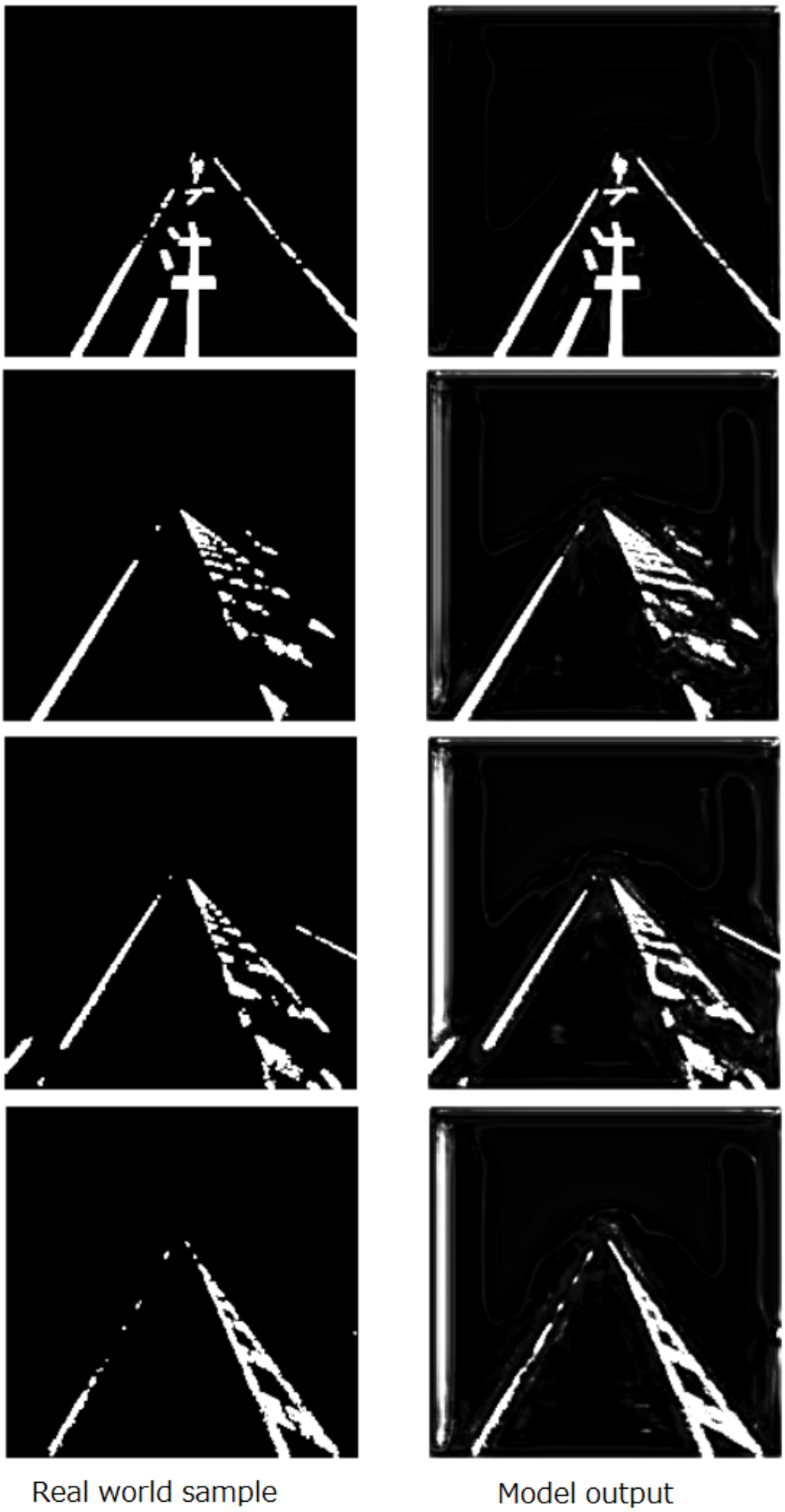


Figure 6.15: Results of Circle Noise (U-Net Model on Real-world Examples)

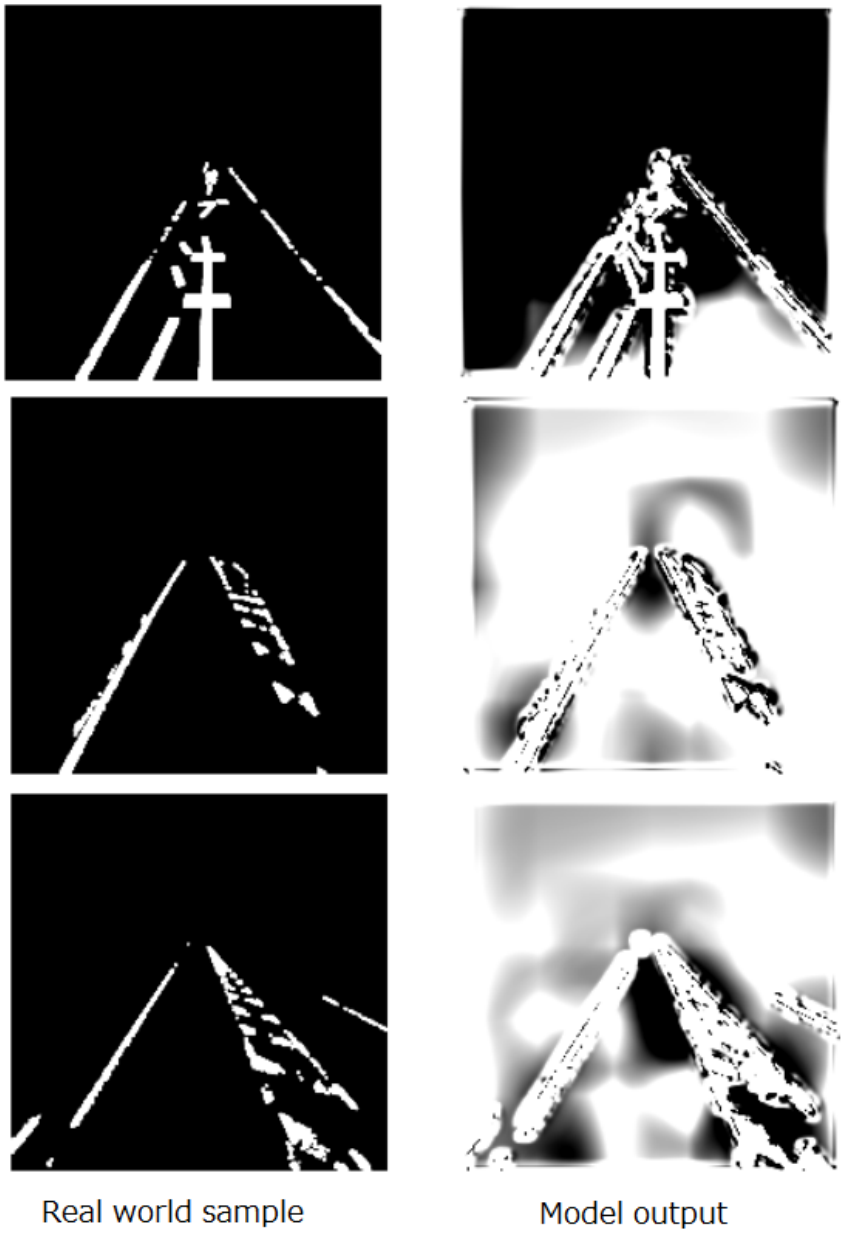


Figure 6.16: Results of Circle Noise (PSP-Net Model on Real-world Examples)

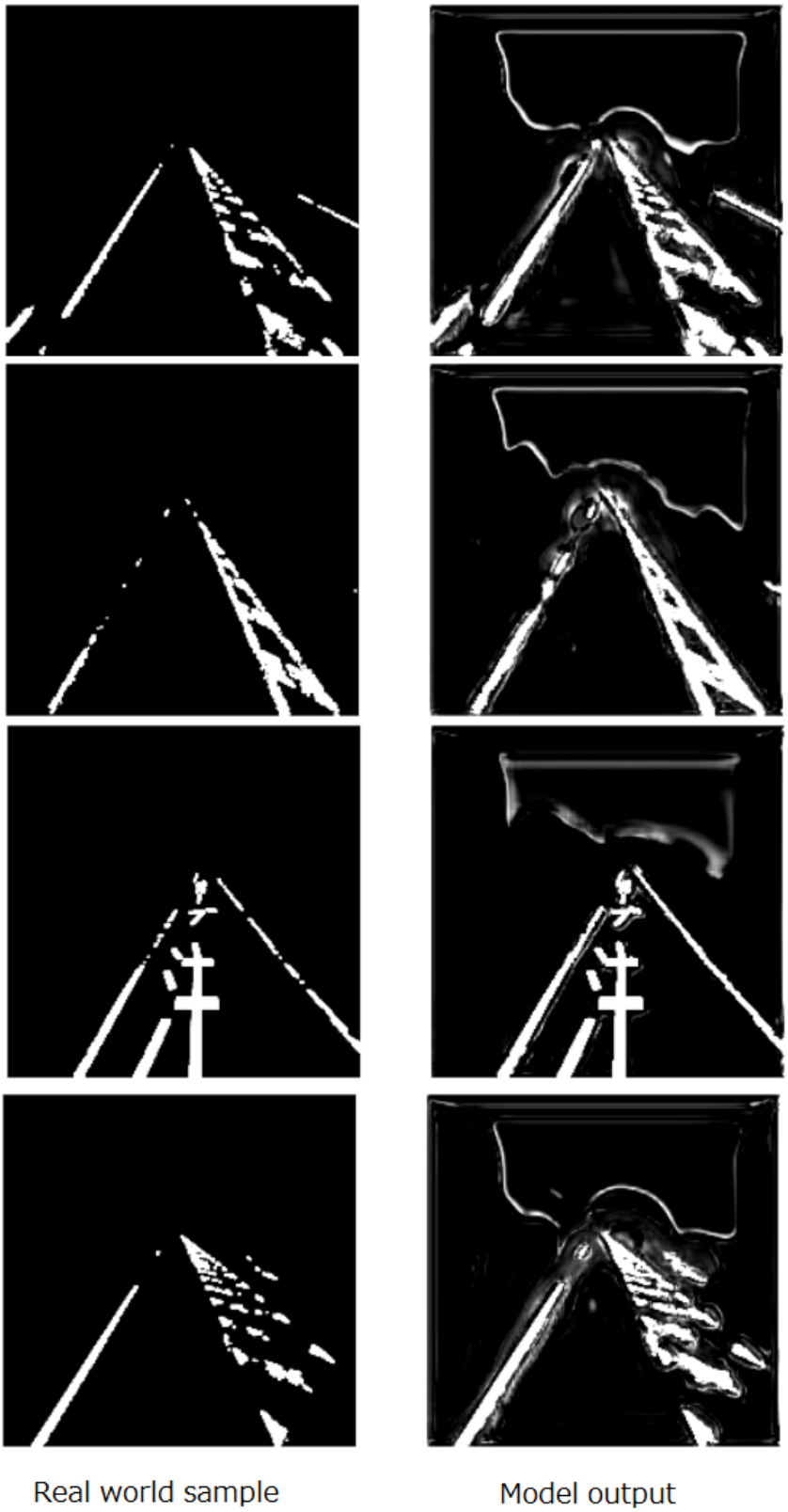


Figure 6.17: Example of Qualitative Results of Erosion & Gaussian Noise (U-Net Model on Real-world Examples)

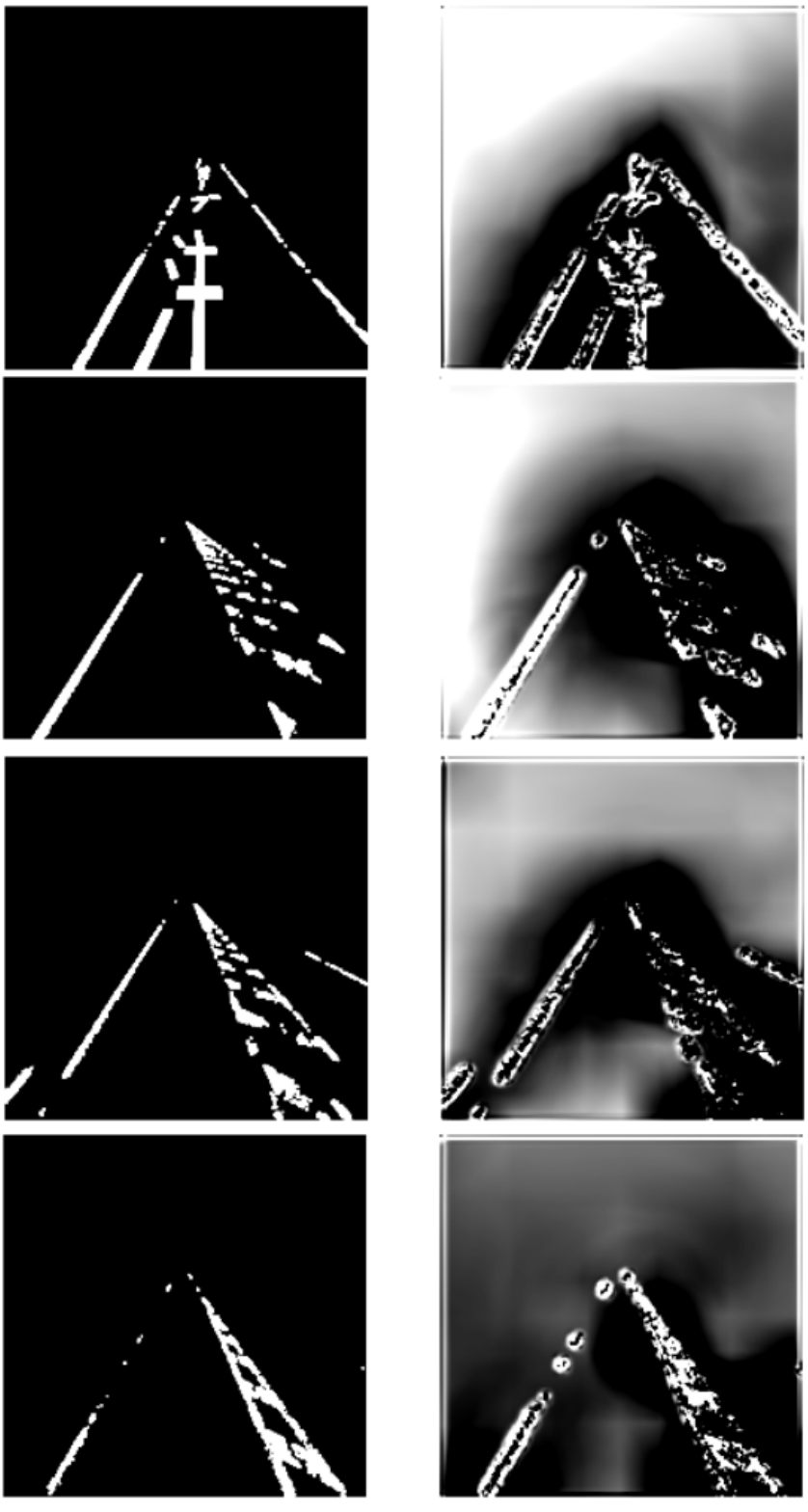


Figure 6.18: Results of Erosion & Gaussian Noise (PSP-Net Model on Real-world Examples)

areas, tends to introduce additional noise into the reconstructions. This model leverages a pyramid pooling module to aggregate global context, which theoretically helps in understanding the broader scene. However, this characteristic, while beneficial in large-scale scene parsing tasks, proved to be less advantageous for the specific needs of road marking restoration, where precision in small-scale features is more significant. The additional noise introduced by PSPNet can result in a less clean, more cluttered image output, which obscures the finer details essential for high-quality road marking restoration.

The shift to real-world application provided a stern test for both models, which both saw a decline in their effectiveness. This decline in performance highlights the challenge of domain shift, a common issue in the deployment of machine learning models. Domain shift occurs when the distribution of data on which the model is trained (synthetic data in this case) does not adequately reflect the distribution of data it encounters in a real-world setting. Factors contributing to this shift in the context of road surface marking include variations in environmental conditions such as weather effects, physical wear and tear from traffic, and fluctuations in lighting and visibility.

These real-world conditions introduce complexities that the training datasets may not fully capture. For instance, real road markings may be faded not just by physical abrasion but also by chemical reactions with vehicular pollutants or obscured by dirt and debris. Additionally, lighting conditions can vary widely in the real world, affecting the visibility and appearance of road markings. The synthetic datasets, while carefully constructed, could not replicate these nuanced variations completely.

This gap between training environments and actual deployment scenarios underscores the importance of constructing training sets that mirror the full spectrum of real-world conditions as closely as possible. Enhancing the diversity of training datasets to include a broader range of deterioration types, environmental conditions, and lighting scenarios would likely lead to improved model robustness and reliability when applied in real-world settings.

Moreover, this scenario illustrates the broader challenge in machine learning of ensuring that models are not only trained to perform well on their training data but are also adaptable and robust enough to handle the unpredictable variability of real-world data. As machine learning continues to make inroads into practical applications, particularly in critical areas like road safety, the ability to generalize across different domains remains a paramount concern. Addressing this issue may involve not only diversifying training data but also exploring advanced techniques in domain adaptation and continual learning to help models adjust to new data without needing exhaustive retraining.

Chapter 7

Quality Evaluation of Road Surface Markings with Uncertainty Aware Regression and Progressive Pretraining

7.1 Introduction

In advancing the methodologies for automated quality assessment of road surface markings, this chapter introduces an innovative approach utilizing Uncertainty Aware (UA) regression coupled with a progressive pretraining (PPT) strategy. This method refines the evaluation process by integrating advanced machine learning techniques to enhance precision and reliability, moving beyond the limitations of existing automated systems.

in this study we leverage the dataset previously used in quality evaluation using "Efficient VGG-16" comprised of binary masks derived from RGB images of road markings. This dataset, enriched through sophisticated data augmentation techniques, serves as the foundation for training state-of-the-art convolutional neural network (CNN) models. By transforming complex real-world images into a format more amenable to automated analysis, we ensure that our models are both robust and adaptable to diverse operational scenarios.

The core of our methodological innovation lies in the application of the uncertainty aware regression and the progressive pretraining strategy. This approach begins with a baseline model, which undergoes successive refinements to incorporate more complex architectures capable of accounting for uncertainty. This progression is not merely incremental; it is strategic, enhancing the model's ability to deliver reliable predictions under varied and unpredictable conditions. The CNN architectures employed, namely VGG-16 and ResNeXt, are adapted through this training regimen to better predict quality metrics of road markings with enhanced accuracy.

To quantify the effectiveness of our approach, we conducted extensive evaluations comparing the performance of our UA-enhanced models against a traditional baseline. The metrics used for this assessment include Mean Absolute Error (MAE) and accuracy, which provide insights into the precision and reliability of the models. These performance metrics underscore the potential of integrating uncertainty into the regression framework, highlighting improvements over traditional methods that do not account for the inherent variability and noise present in real-world data.

This chapter will detail the training process, from the initial data preparation through to the advanced stages of model refinement. It will also discuss the specific challenges encountered when applying these models to real-world data, such as the domain shift problem, and how our approach seeks to mitigate these issues through innovative training strategies and model architectures.

By addressing these sophisticated challenges with cutting-edge machine learning strategies, this research not only pushes forward the capabilities of infrastructure quality assessment but also sets a foundation for further advancements in the application of deep learning to real-world problems. The insights gained here extend beyond road surface marking evaluation, offering potential applications in other domains where quality assessment is critical and where environmental variability plays a significant role.

7.2 Methodology

7.2.1 Dataset and Baseline Models

For this task, we used the dataset described previously. We processed 800 RGB images to create binary masks. These masks were augmented by rotating and applying mirror imaging techniques, resulting in 3,200 enhanced masks. Each mask is linked to a quality score averaged from initial human assessments, providing a spectrum of quality from 1 to 4. These scores, critical for our regression models, were adjusted to address class imbalances in the dataset.

The first baseline model used was VGG-16. VGG-16 is a widely-used deep learning model, originally created by researchers at the University of Oxford. The model consists of 16 layers, of which 13 are specialized in detecting features such as edges and textures in images. The remaining three layers assist in making final decisions based on these features. What makes VGG-16 unique is its use of small filters in its layers, allowing it to learn intricate image features. However, because of its

The second model we used was ResNeXt. ResNeXt is a more recent type of deep learning model that is specifically designed for tasks related to image recognition. It represents an evolution of earlier models and aims to improve their performance. One of the key features of ResNeXt is the introduction of "cardinality," which refers to the number of parallel paths or transformations that the model uses. This is different from older models that mostly relied on increasing the depth (adding more layers) or width (adding more units to each layer) for better performance.

7.2.2 The Baseline Model

In the initial phase of our methodology, we employed a conventional regression approach to establish a baseline for our subsequent experiments. This initial model featured a simplistic architecture with a single output neuron, designed to predict the quality score of road surface markings from input images. The purpose of this baseline model was to provide a straightforward, deterministic prediction of quality scores, where each image's assessment was boiled down to a single numerical value representing the quality of the road markings. Figure 7.1 shows the pipeline of this approach.

The regression model we used is a form of supervised learning where the goal is to predict a continuous outcome—the quality score—based on input features extracted from the images. In

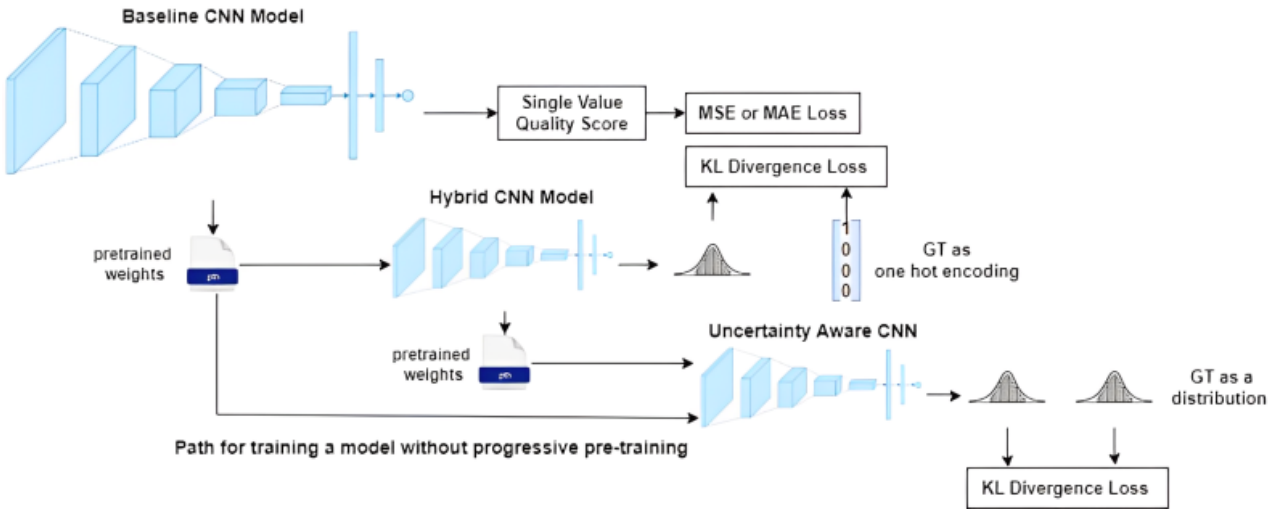


Figure 7.1: A depiction of the training pipeline for the different models we used. The baseline CNN model in the top left represents the original network trained as a regression model with a single output neuron that is supposed to predict the quality score. The hybrid CNN model in the middle is initialized with weights from the previous model, it outputs a probability distribution, and the ground truth on which it was trained is a 1-hot encoding of the quality score on a vector of 40 entries. The UA CNN on the right is trained to predict a distribution, and the ground truth presented to it is also encoded as a distribution which accounts for the uncertainty. When this model inherits the weights from the hybrid model we call it "progressive pretraining." For comparison purposes, we also train a model with no "progressive pretraining," which inherits the weights of the baseline CNN only.

our case, these features might include aspects of texture, color, and edge information that are indicative of the condition of the road markings. The single output neuron makes this a simple linear regression task, where the model learns to fit a line through the data points mapped in a high-dimensional feature space.

Using a traditional regression model like this has several advantages, including simplicity and computational efficiency. It provides a clear, quantifiable baseline against which more complex models can be evaluated. However, this approach also comes with significant limitations, particularly in the context of road marking quality evaluation, where the quality assessment is inherently subjective and influenced by numerous factors that a single score cannot fully encapsulate.

For instance, traditional regression models do not account for the uncertainty inherent in the quality assessment of road markings. Road markings can degrade in a variety of ways—fading colors, chipping, or becoming obscured by dirt—and different observers might judge these conditions differently. By relying solely on a deterministic output, the baseline model assumes that there is an absolute truth to be discovered in the data, which simplifies the real-world complexity of the task.

To enhance our approach and address these shortcomings, we later incorporate uncertainty-aware regression techniques. These techniques allow our models not only to predict a quality score but also to estimate the confidence in their predictions. This is crucial for practical applications where decisions need to be made based on these predictions, such as prioritizing road maintenance tasks based on the predicted quality of road markings.

The progression from a simple baseline regression model to more sophisticated uncertainty-aware models illustrates an evolution in our methodology. It reflects a deeper understanding of the challenges inherent in automated road marking quality evaluation and represents an effort to develop tools that can provide more reliable and nuanced insights into the condition of road infrastructure. This approach not only improves the accuracy of the predictions but also provides a probabilistic framework that captures the uncertainty, offering a more comprehensive tool for infrastructure management and planning.

7.2.3 The Hybrid Model

Following the development and validation of our baseline regression model, we embarked on a more advanced phase of our methodology by transitioning the learned weights to a hybrid model as seen in Figure 7.1. This transition marks a significant enhancement in our approach to evaluating road surface marking quality, by introducing the capability to predict a range of scores instead of a single value. The hybrid model utilizes a softmax function at the output layer, which allows it to predict a probability distribution over possible quality scores rather than a single deterministic output. This method reflects a more nuanced understanding of the variability and uncertainty inherent in the assessment of road surface marking quality.

The softmax function is a critical component in this setup. It is typically used in classification tasks where the outputs are mutually exclusive classes. Here, however, we adapt it to our regression framework by treating each potential quality score as a class. The softmax function converts the raw logits—outputs of the last neural network layer before the softmax—into probabilities by exponentiating and normalizing them. This means that each output represents the model's confidence in assigning a particular quality score to the input image, allowing for a probabilistic interpretation of the model's predictions.

To effectively implement this approach, we utilized one-hot encoded vectors for the labels of our training data. Each label vector contains 40 entries, corresponding to the range of scores from 1 to 4, subdivided into decimal increments (for instance, 1.0, 1.1, ..., 4.0). This granularity in scoring allows for a more precise training and evaluation process, where the model learns to associate specific image features with subtle variations in quality. Each entry in the vector represents the presence or absence of a particular score, with a '1' indicating the true score for a given image and '0's indicating all other possible scores.

The process of transferring weights from the simpler baseline model to this more complex hybrid model is a key aspect of our Progressive Pretraining (PPT) strategy. By starting with a trained baseline model, the hybrid model inherits a foundational knowledge of the task, which can improve the efficiency and effectiveness of further training. This technique leverages the already learned features that are relevant for quality prediction and builds upon them to refine the model's predictions under a probabilistic framework.

Moreover, this step illustrates a practical application of knowledge transfer within deep learning, where a model developed for one task is adapted for another related task or a more complex version of the same task. In our case, the hybrid model does not start its learning process from scratch but begins with a nuanced understanding inherited from the baseline model. This not only accelerates the learning process but also helps in overcoming challenges associated with training deep models from zero knowledge, such as convergence issues and sensitivity to the initial random weights.

This phase of transferring and transforming model capabilities is pivotal in our approach, bridg-

ing the gap between simple regression and a fully uncertainty-aware framework. It represents an intermediate yet crucial step towards developing a robust model that not only predicts the quality of road surface markings more accurately but also provides insights into the confidence of its predictions. This enhancement is instrumental in paving the way towards employing advanced uncertainty quantification techniques in subsequent phases, further refining the model's ability to handle real-world variability and ambiguity in road surface marking quality assessments.

7.2.4 The Uncertainty Aware Model

In the final phase of our methodology, we significantly advanced our model by incorporating an Uncertainty Aware (UA) regression approach. This innovative step involved transitioning the knowledge and parameters from the previously developed hybrid model into a new, more sophisticated framework. This new UA model is designed not just to predict a static range of quality scores but to dynamically express these scores as a probability distribution. Such a distribution encompasses 40 discrete values, reflecting the granularity of quality evaluation possible by human experts. This probability distribution is centered around the values provided by human evaluators, thus capturing the inherent uncertainty in manual quality assessments of road surface markings. A diagram illustrating the implementation steps of our approach is seen in Figure 7.2.

The move to an UA approach allowed our model to account for real-world variability and the subjective nature of quality assessment. By doing so, it more accurately mirrors human judgment, which inherently includes a degree of uncertainty. This adaptation is visually represented in Figure 7.3, where the distribution of scores illustrates the variability and uncertainty captured by the model.

The progressive pretraining (PPT) technique utilized across our model development phases ensures that each subsequent model phase builds on the refined understanding and weights from its predecessors. This layered learning approach is especially beneficial in the domain of road surface marking quality evaluation, a field where subjective assessments can vary widely from one evaluator to another. By progressively training our models to anticipate and accommodate these variations, the UA model learns to output a range of potential quality scores, each associated with a calculated probability. This method offers a more nuanced and flexible approach to predicting road marking quality, which is far superior to fixed-score predictions.

To train and evaluate this sophisticated model architecture effectively, we partitioned our dataset into an 80% training set and a 20% testing set. This allocation allows for comprehensive training while still reserving a significant portion of data for rigorous testing of the model's efficacy. For our baseline models, such as the VGG-16 and ResNeXt, we employed traditional loss functions, L1 and L2 respectively, to optimize the learning process. These functions facilitate the adjustment of model parameters based on the difference between predicted outputs and actual quality scores.

Further refining our approach for the hybrid and UA models, we implemented the Kullback–Leibler (KL) divergence[134] as the loss function. The KL divergence is particularly suited for scenarios where the model outputs and the target values are probability distributions, as it quantifies how one probability distribution diverges from a second, expected probability distribution. In our application, this involves comparing the predicted distribution of road surface quality scores against the actual distribution derived from expert evaluations.

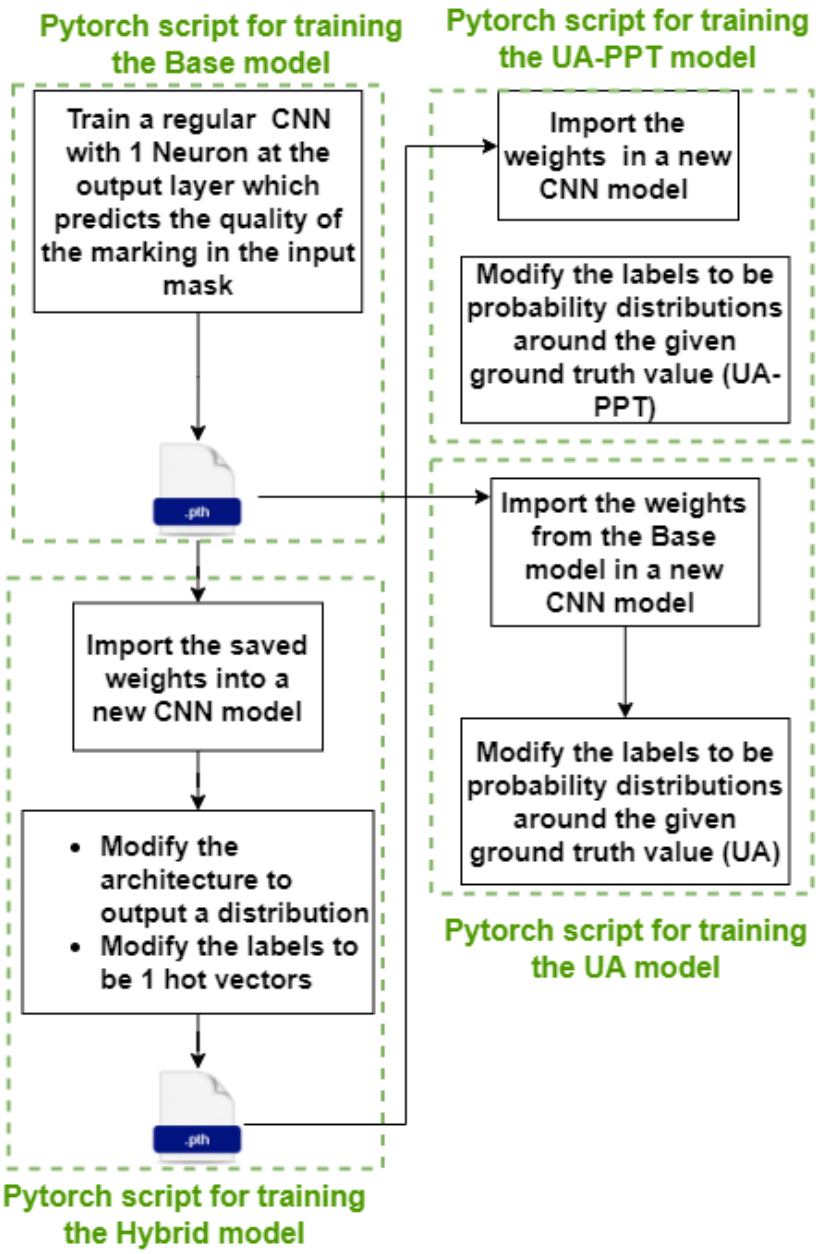


Figure 7.2: Diagram that describes the specific implementation process of the proposed quality evaluation of road surface markings scheme.

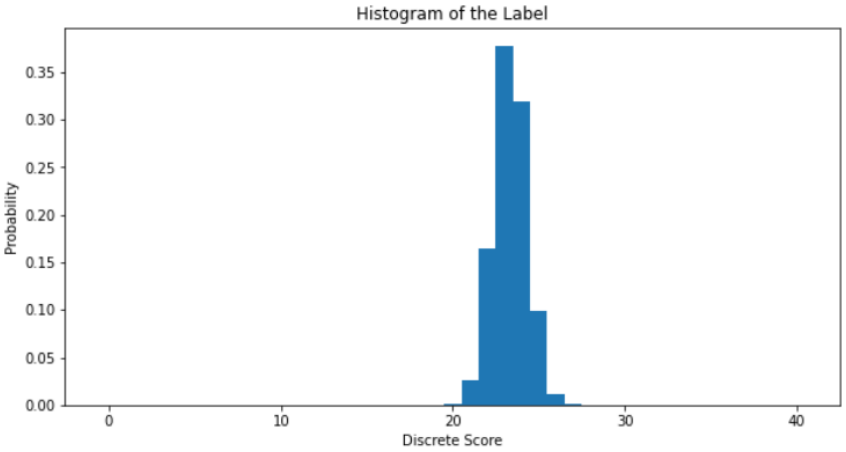


Figure 7.3: An example of a label for the UA model as a probability distribution around the true value across 40 discrete values representing our score range.

$$KL(\hat{y}||y) = \sum_{c=1}^{M} \hat{y}_c \log \frac{\hat{y}_c}{y_c} \tag{7.1}$$

Here, $KL(\hat{y}||y)$ measures the divergence between the predicted probability distribution $\hat{y}$ and the true distribution y, over M discrete classes. $\hat{y}_c$ and y_c are the predicted and true probabilities of class c, respectively.

$$\sum_{i=1}^{D} |x_i - y_i| \tag{7.2}$$

MSE is the average of the squares of the errors between the predicted values x_i and the actual values y_i, providing a measure of the variance of the predictions.

$$\sum_{i=1}^{D} (x_i - y_i)^2 \tag{7.3}$$

This equation represents a Gaussian probability density function where c is the variable, s is the mean of the distribution, and σ^2 is the variance.

$$g(c) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-(c-s)^2}{2\sigma^2}\right) \tag{7.4}$$

This equation represents a Gaussian probability density function where c is the variable, s is the mean of the distribution, and σ^2 is the variance.

$$\mathbf{g_c} = [g(c_1), g(c_2), \ldots, g(c_m)] \tag{7.5}$$

$\mathbf{g_c}$ is a vector consisting of Gaussian distribution values computed for different classes $c_1, c_2, \ldots, c_m$.

$$p(c_i) = \frac{g(c_i)}{\sum_{j=1}^{m} g(c_j)}, \quad i = 1, 2, \ldots, m \tag{7.6}$$

$p(c_i)$ is the normalized probability for class c_i, calculated by dividing the Gaussian value for c_i by the sum of Gaussian values for all classes, ensuring that the probabilities sum to 1. For computational efficiency and to handle the intensive demands of training such sophisticated

models, we utilized a high-performance NVIDIA GeForce RTX 3090 GPU. This hardware choice ensures that our models train swiftly and can handle the complex calculations required by our advanced neural network architectures.

The following equations together summerize the progression of model training and sophistication from simple regression to handling complex, uncertainty-laden outputs in a structured and methodical way, which parallels our description of employing a progressive pretraining (PPT) approach.

$$y = f(x; \theta) \tag{7.7}$$

Equation 7.7 represents the baseline regression model where y is the predicted quality score for input x and θ represents the parameters of the model.

$$\theta_{\text{hybrid}} = \theta + \Delta\theta \tag{7.8}$$

In Equation 7.8, θ_{hybrid} denotes the parameters of the hybrid model, initially set to the learned weights θ from the baseline model, adjusted by $\Delta\theta$, which represents modifications tailored to handle a probability distribution output.

$$P(y|x; \theta_{\text{hybrid}}) = \text{softmax}(g(x; \theta_{\text{hybrid}})) \tag{7.9}$$

Here, $P(y|x; \theta_{\text{hybrid}})$ in Equation 7.9 is the probability distribution over possible quality scores, computed using a softmax function applied to the output of function g, which is parameterized by θ_{hybrid}.

$$\theta_{\text{UA}} = \text{transfer}(\theta_{\text{hybrid}}) \tag{7.10}$$

In Equation 7.10, θ_{UA} represents the parameters of the UA model, obtained by adapting the θ_{hybrid} through a transfer function that further aligns the model with the uncertainty representation requirements.

$$Q(y|x; \theta_{\text{UA}}) = \text{sample}(\text{softmax}(g(x; \theta_{\text{UA}}))) \tag{7.11}$$

Equation 7.11 calculates the prediction as a sampling from the softmax distribution provided by the UA model, emphasizing the probabilistic nature of the output which encapsulates uncertainty.

To guard against overfitting a common challenge in deep learning where models learn the training data too well and fail to generalize to new data we introduced specific stopping criteria. For instance, training for the VGG-16 models would cease if there was no improvement in the error metric after five consecutive evaluations, with each evaluation occurring every 1,000 training steps. For the ResNeXt models, a more lenient ten-check threshold was set.

The application of these rigorous and methodical training procedures, along with strategic use of advanced regression techniques and computational resources, allows our models not only to predict road marking quality with high accuracy but also to reflect the uncertainty inherent in such real-world assessments. This dual capability significantly enhances the practical utility and reliability of our automated road surface marking quality evaluation system.

7.2.5 Model Evaluation

To rigorously evaluate the effectiveness and reliability of our regression models in assessing infrastructure quality, we employed a suite of statistical metrics designed to capture various aspects of model performance comprehensively.

Accuracy was the primary metric we used to gauge how closely the model's predictions approximated the actual values. Specifically, this metric assessed whether the model's predictions fell within a tolerance of 0.5 units from the true values. This narrow margin is crucial for practical applications where precision in categorizing infrastructure quality can have significant implications, such as in maintenance scheduling and safety assessments.

Mean Squared Error (MSE) served as another critical metric in our analysis. Unlike simple accuracy, MSE provides a more nuanced view by squaring the errors before averaging them, thus giving more weight to larger errors. This characteristic of MSE makes it particularly useful in situations where avoiding large deviations from the true values is more critical than smaller deviations, which is often the case in quality control processes in infrastructure management.

Mean Absolute Error (MAE) was also utilized, offering a straightforward interpretation by averaging the absolute differences between predicted and actual values. MAE is especially valuable in providing a clear measure of the average error magnitude a user might expect when using the model in real-world scenarios. This metric can directly inform stakeholders of the expected reliability of the model in operational settings, helping to manage expectations and plan accordingly.

We further enriched our evaluation by calculating the R-squared ('2) value, a statistic that measures the proportion of variance in the dependent variable that can be predicted from the independent variables. In our context, the R-squared value is particularly telling, as it quantifies how well our regression model, with its inputs as observed values, can explain the variability in the quality scores of road markings. An R-squared value close to 1 indicates that our model accounts for a significant proportion of the variance, suggesting high explanatory power and effectiveness. Conversely, a lower R-squared value would indicate that our model lacks predictive accuracy, highlighting areas for potential improvement. Equation 7.12 shows how to compute the R^2 score.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{7.12}$$

Correlation analysis was the final step in our evaluation process. By measuring the strength and direction of the relationship between the predicted and actual values, we could assess the degree of linear correspondence between the two sets of data. A high correlation coefficient would indicate that increases or decreases in predicted values consistently match those in actual values, reinforcing the model's validity. This statistical tool provides an additional layer of validation by confirming whether the model's outputs meaningfully relate to real-world observations, an essential factor in practical applications.

Each of these metrics contributes a distinct perspective on the model's performance, offering a rounded assessment that encompasses accuracy, error magnitude, explanatory power, and correlation with actual outcomes. By applying these comprehensive evaluations, we ensure that our models are not only statistically validated but also practically applicable in real-world scenarios, providing reliable, actionable insights for infrastructure quality management.

7.3 Experimental Results and Discussion

In the evaluation of our model's performance, we organized our results in a comprehensive manner to effectively compare the various models we implemented. This comparison was particularly insightful as it showcased the varying effectiveness of our models under different training paradigms. The results are recorded in table 7.1 and table 5.2.

Table 7.1: Results of Various Models (Test Data)

	MSE	MAE	Corr	R2	Acc
VGG-16 (Baseline)	5.57%	37.57%	91.38%	0.83	73.91%
VGG-16 (Hybrid)	5.92%	**24.49%**	**91.82%**	0.82	**82.60%**
VGG-16 (UA-PPT)	**5.83%**	**24.38%**	**91.66%**	**0.83**	**81.27%**
VGG-16 (UA)	13.95%	41.63%	83.04%	0.59	72.57%
ResNeXt (Baseline)	19.44%	79.19%	78.19%	0.42	31.43%
ResNeXt (Hybrid)	25.75%	**61.54%**	69.86%	0.23	**67.22%**
ResNeXt (UA-PPT)	26.03%	**61.77%**	**71.13%**	0.23	**66.72%**
ResNeXt (UA)	71.78%	-	-	-	-
Efficient VGG-16	2.39%	-	0.96%	0.93	88.33%

Table 7.2: Results Performance for Different Values of σ (VGG-16(UA-PPT) Model)

σ	MSE	MAE	Corr	R2	Acc
$\sigma = 0.1$	5.83%	24.38%	91.66%	0.83	81.27%
$\sigma = 0.15$	6.14%	24.29%	91.21%	0.82	81.77%
$\sigma = 0.2$	6.85%	25.10%	90.02%	0.80	83.61%
$\sigma = 0.25$	**5.23%**	**22.24%**	**92.29%**	**0.84**	**84.44%**

We categorized our models into several types for clarity and ease of understanding. The original models, which served as our baseline, were compared against the more advanced hybrid and UA (Uncertainty Aware) models. These comparisons were particularly important for identifying improvements in our newer models over the baseline, highlighting advancements in machine learning techniques that could better handle the complexities of road surface marking quality evaluation.

The VGG-based models, known for their robustness in handling image data, showed notable improvements across several key performance metrics. The Mean Absolute Error (MAE), correlation, and accuracy metrics all saw enhancements, demonstrating the models' refined ability to predict and evaluate road marking quality accurately. The UA-PPT model, which integrated both the uncertainty aware approach and progressive pretraining (PPT), matched the

baseline model's R-squared score, indicating its effectiveness in explaining the variance in the data similar to the baseline but with the added benefits of uncertainty modeling.

However, despite these advancements, the baseline model still held an edge in the Mean Squared Error (MSE) metric. This could suggest that while the new models are better at handling average cases, they might still struggle with outliers or more extreme cases, which are more heavily penalized by MSE.

In contrast, the ResNeXt-based models, which are typically praised for their scalable performance across different domains, only outperformed the baseline in terms of MAE and accuracy. This shows that while some newer models offer advantages in specific areas, they may not uniformly outperform older models in all aspects.

Direct comparisons between the hybrid and UA-PPT models under the VGG architecture revealed that the UA model provided better outcomes in terms of MSE, MAE, and R-squared scores, highlighting the benefits of integrating uncertainty into the model's predictions. This suggests a more nuanced approach to handling data variability and improving model reliability, especially in tasks involving quality assessment where precision is crucial.

The progressive pretraining approach significantly boosted the performance of the UA models, demonstrating its effectiveness in enhancing model capabilities through a step-wise learning process. This was particularly evident in the stark necessity of PPT for the ResNeXt-based UA models, where without this approach, the models failed to converge.

The superior performance of VGG-based models over ResNeXt models across several metrics underscores the suitability of VGG architectures for this specific application, possibly due to their architectural characteristics which may align better with the demands of image-based quality evaluation tasks.

Moreover, we observed the impact of the hyperparameter σ, which represents the standard deviation in the label distribution and quantifies the amount of uncertainty in our ground truth data. An optimal value of $\sigma = 0.25$ was found to yield the best performance, surpassing even the baseline in MSE and other metrics. This not only reinforces the effectiveness of our UA and PPT strategies but also underscores the importance of accurately setting parameters that reflect real-world uncertainty. The graphical representations in our results, particularly the Bland–Altman plots seen in Figures 7.4 and 7.5 and distribution overlays, visually supported these findings, showcasing the improved alignment and consistency of the UA-PPT model predictions with true values, compared to the baseline.

These insights not only validate the novel approaches employed in this study but also highlight areas for future enhancements. The need for further refinement in our models is evident, especially in terms of efficiency and handling outliers, which continue to pose challenges as indicated by the persistence of MSE advantages in the baseline model. This suggests a continuing evolution in our approach, aiming to not only match but surpass existing methods in all aspects of performance, thereby pushing the boundaries of what is possible in automated road surface marking evaluation.

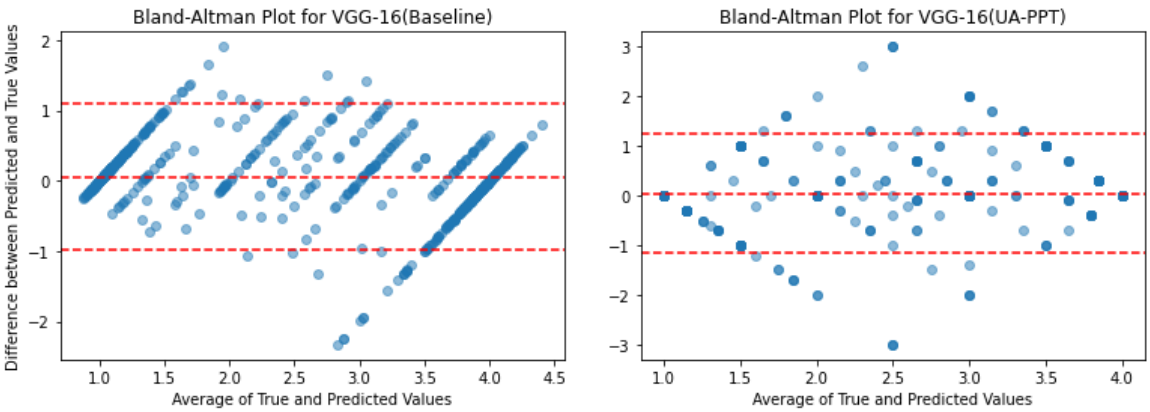


Figure 7.4: The Bland–Altman plot for the VGG-16 baseline model on the left and the Bland–Altman plot for the VGG-16 UAPPT model on the right.

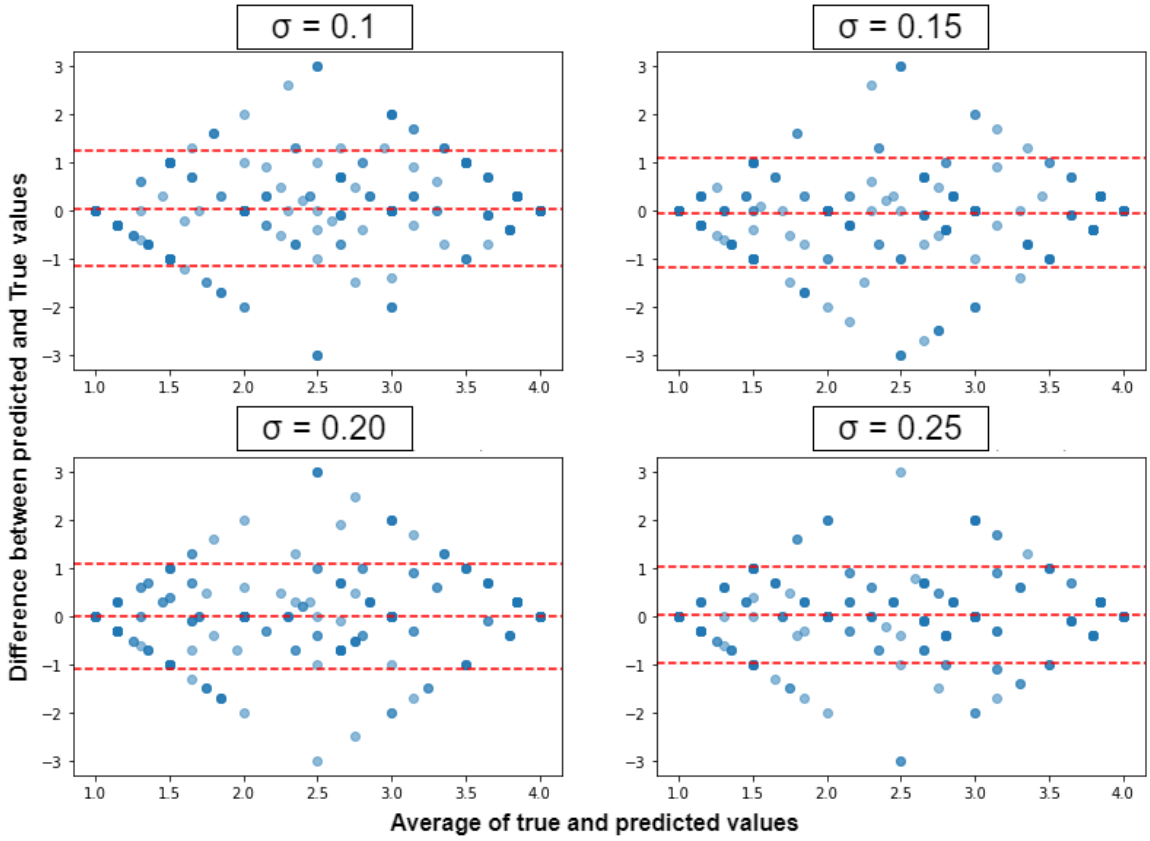


Figure 7.5: Plot of Bland–Altman for different values of f for the VGG-16 UA-PPT model.

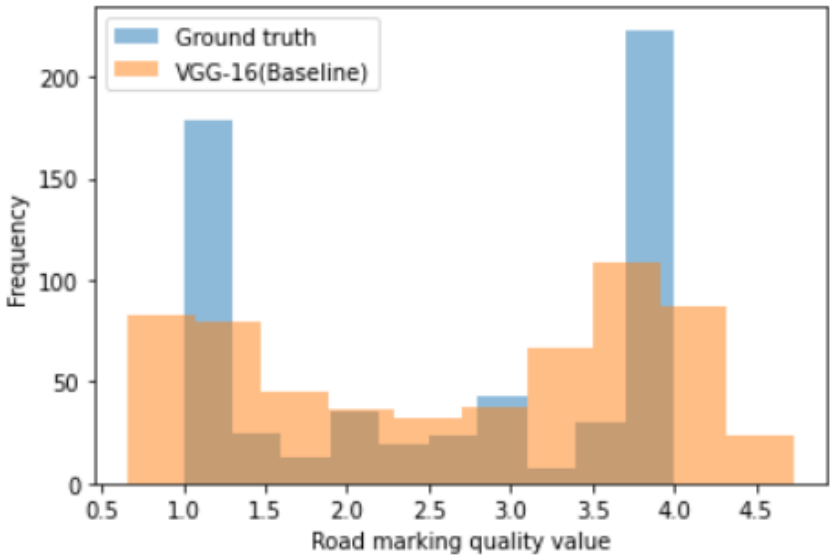


Figure 7.6: A histogram depicting the distribution of the predictions made by the baseline VGG-16 model on the test set contrasted with the distribution of the ground truth values.

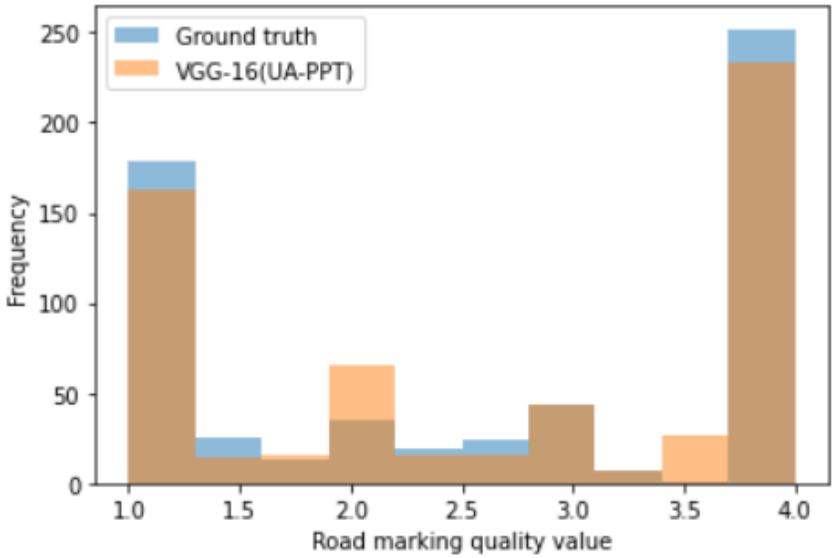


Figure 7.7: A histogram depicting the distribution of the predictions made by VGG-16 UA-PPT model on the test set contrasted with the distribution of the ground truth values.

Chapter 8

Conclusion

The field of road surface marking analysis stands at the forefront of technological innovation, offering promising avenues for transformative research with far-reaching practical implications. Our study represents a pioneering endeavor, venturing into uncharted territory by introducing novel methodologies tailored specifically to the segmentation and evaluation of road surface markings. Through meticulous experimentation with an array of models and techniques, we have uncovered insights that not only enrich our understanding of this complex domain but also present tangible solutions to real-world challenges.

Central to our investigation is the implementation of cutting-edge models, notably the "Efficient VGG-16" model, which has demonstrated remarkable efficacy in extracting pertinent features and patterns from road markings. Leveraging advanced techniques such as Uncertainty Aware Regression with Progressive Pre-Training, we have pushed the boundaries of conventional methodologies, setting a new standard for precision and accuracy in this field. The robustness of our findings not only validates the viability of these models but also underscores their potential to revolutionize existing practices within the transportation sector.

Our results serve as more than just a testament to the effectiveness of our chosen methodologies; they represent a rallying cry for the broader research community to embrace and build upon this foundation. By showcasing the tangible benefits of our approach in real-world applications, we aim to inspire further exploration and innovation in this burgeoning field. Future endeavors should heed the call to delve deeper into the potential of unsupervised methods, as exemplified by the seminal work cited in [28], which promises to enhance the accuracy of assessments while mitigating the inherent subjectivity associated with human judgment.

Moreover, the integration of generative adversarial networks (GANs) offers a tantalizing prospect for augmenting data augmentation techniques, as elucidated in [29]. By harnessing the power of GANs to generate synthetic data, researchers can potentially overcome limitations posed by data scarcity and variability, thereby fortifying the robustness and generalizability of predictive models. This symbiotic fusion of cutting-edge methodologies holds the key to unlocking new frontiers in road surface marking analysis, heralding a paradigm shift in how we perceive and interact with our urban environments.

In summation, our study stands as a beacon of innovation, illuminating the path towards a future where the analysis of road surface markings transcends mere functionality to become a cornerstone of intelligent transportation systems. As we chart a course towards this vision, we invite fellow researchers to join us on this exhilarating journey of discovery and transformation.

Together, we can harness the power of technology to forge a safer, more efficient, and more sustainable future for all.

Bibliography

[1] M. V. Medvedev and V. I. Pavlov, "Road surface marking recognition and road surface quality evaluation using convolution neural network," in *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 2020, pp. 1–3. DOI: 10.1109/FarEastCon50210.2020.9271368.

[2] K.-L. Lin, T.-C. Wu, and Y.-R. Wang, "An innovative road marking quality assessment mechanism using computer vision," *Advances in Mechanical Engineering*, vol. 8, no. 6, 2016. DOI: 10.1177/1687814016654043.

[3] A. R. Stacy, "Evaluation of machine vision collected pavement marking quality data for use in transportation asset management," The Office of Graduate and Professional Studies of Texas AM University, Tech. Rep., 2019.

[4] B. Li, D. Song, H. Li, A. Pike, and P. Carlson, "Lane marking quality assessment for autonomous driving," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9. DOI: 10.1109/IROS.2018.8593855.

[5] M. Boudissa, H. Kawanaka, and T. Wakabayashi, "Semantic segmentation of traffic landmarks using classical computer vision and u-net model," in *International Conference of Engineering Technology 2021 (ICET)*, 2021.

[6] M. Boudissa, H. Kawanaka, and T. Wakabayashi, "Traffic landmark quality evaluation using efficient vgg-16 model," in *2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS-ISIS)*, 2022, pp. 1–5. DOI: 10.1109/SCIS-ISIS55246.2022.10002145.

[7] F. Hamilton, "Implementing urban resilience in urban planning: A comprehensive framework," *ScienceDirect*, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0197397509000891.

[8] S. Fainstein, "The landscape and evolution of urban planning science," *ScienceDirect*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169204621000868.

[9] P. Davidoff and T. A. Reiner, "Sustainable urban planning and making sustainable cities," *SpringerLink*, 1962. [Online]. Available: https://link.springer.com/article/10.1007/s10901-016-9510-2.

[10] V. Cristie and M. Berger, "An exploratory approach for urban data visualization and spatial analysis with a game engine," *SpringerLink*, 2016. [Online]. Available: https://link.springer.com/article/10.1007/s11042-015-3087-1.

[11] L. Hou, T. Shi, and Q. Gui, "A review of urban planning research for climate change," *MDPI Sustainability*, 2017. [Online]. Available: https://www.mdpi.com/2071-1050/9/12/2224.

[12] S. OHKAWA, H. Date, and Y. TAKITA, "Detection method of road marking using lrf intensity of surface," in *The Proceedings of the Transportation and Logistics Conference*, vol. 22, 2013, pp. 249–252. DOI: 10.1299/jsmetld.2013.22.249.

[13] T. Lee, Y. Yoon, C. Chun, and S. Ryu, "Cnn-based road-surface crack detection model that responds to brightness changes," *Electronics*, vol. 10, no. 12, p. 1402, 2021. DOI: 10.3390/electronics10121402.

[14] F. Asdrubali, C. Buratti, E. Moretti, F. D'Alessandro, and S. Schiavoni, "Assessment of the performance of road markings in urban areas: The outcomes of the civitas renaissance project," CIRIAF, Interuniversity Centre of Research on Pollution by Physical Agents, University of Perugia, Via G. Duranti, snc, 06125 Perugia, Italy, Tech. Rep., 2013.

[15] R. Mukherjee, H. Iqbal, S. Marzban, *et al.*, "Ai driven road maintenance inspection," 27th ITS World Congress, Hamburg, Germany, Tech. Rep., 2021.

[16] Z. Liu, S. Yu, X. Wang, and N. Zheng, "Detecting drivable area for self-driving cars: An unsupervised approach," *arXiv preprint arXiv:1705.00451v1*, 2017.

[17] Various, "Semantic image segmentation: Two decades of research," *arXiv preprint arXiv:2302.06378*, 2023. [Online]. Available: https://arxiv.org/abs/2302.06378.

[18] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "Image segmentation using deep learning: A survey," *arXiv preprint arXiv:2001.05566*, 2020. [Online]. Available: https://arxiv.org/abs/2001.05566.

[19] M. K. Kassis, P. Clough, and V. Dimitrova, "Historical document image segmentation with lda-initialized deep neural networks," *arXiv preprint arXiv:1710.07363*, 2017. [Online]. Available: https://arxiv.org/abs/1710.07363.

[20] A. from The Visual Computer, "3d lidar point-cloud projection operator and transfer machine learning for effective road surface features detection and segmentation," *The Visual Computer*, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s00371-022-02458-0.

[21] L. Ma, Y. Li, J. Li, Z. Zhong, and M. Chapman, "Simultaneous road edge and road surface markings detection using convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 5, pp. 1572–1586, 2019. DOI: 10.1109/JSTARS.2019.2904514.

[22] R.-C. Chen, Y.-C. Zhuang, and H. J. Christanto, "Yolov5 series algorithm for road marking sign identification," *Big Data and Cognitive Computing*, vol. 6, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2504-2289/6/4/149.

[23] A. from Computational Urban Science, "Automating intersection marking data collection and condition assessment at scale with an artificial intelligence-powered system," *Computational Urban Science*, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s43762-022-00019-4.

[24] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "Image segmentation using deep learning: A survey," *arXiv preprint arXiv:2001.05566*, 2020.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "A survey on object instance segmentation," *SN Computer Science*, 2021.

[26] S. Ghosh and A. Dubey, "A comprehensive survey of image segmentation: Clustering methods, performance parameters, and benchmark datasets," *Multimedia Tools and Applications*, 2021.

[27] X. Li, X. Liang, Y. Wei, Y. Xu, J. Feng, and S. Yan, "Dfanet: Deep feature aggregation for real-time semantic segmentation," *arXiv preprint arXiv:1904.02216*, 2019.

[28] M. Boudissa, H. Kawanaka, and T. Wakabayashi, "Surveying semantic segmentation models using traffic landmark dataset," in *Proceedings of the 2022 Joint 12th Interna-*

tional Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCISISIS2022), 2022, F-1-D–1.

[29] M. Cordts, M. Omran, S. Ramos, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] M. Wang, G. Yue, J. Xiong, and S. Tian, "Intelligent point cloud processing, sensing, and understanding," *Sensors*, vol. 24, no. 1, p. 283, 2024. DOI: 10.3390/s24010283.

[31] D. Mari and S. Milani, "Recent advancements in learning algorithms for point clouds: An updated overview," *Sensors*, vol. 22, no. 4, p. 1357, 2022. DOI: 10.3390/s22041357.

[32] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 645–13 654. [Online]. Available: https://paperswithcode.com/paper/pointasnl-robust-point-clouds-processing.

[33] Y. Peng and Z. Mao, "Recent advances and perspectives in deep learning techniques for 3d point cloud data processing," *Robotics*, vol. 12, no. 4, p. 100, 2023. DOI: 10.3390/robotics12040100.

[34] A. from arXiv, "See beyond seeing: Robust 3d object detection from point clouds via cross-modal hallucination," *arXiv preprint arXiv:2309.17336*, 2023. [Online]. Available: https://ar5iv.org/abs/2309.17336.

[35] A. from arXiv, "Machine learning in lidar 3d point clouds," *arXiv preprint arXiv:2101.09318*, 2021. [Online]. Available: https://ar5iv.org/abs/2101.09318.

[36] A. K. Aijazi and P. Checchin, "Non-repetitive scanning lidar sensor for robust 3d point cloud registration in localization and mapping applications," *Sensors*, vol. 24, no. 2, p. 378, 2024. DOI: 10.3390/s24020378. [Online]. Available: https://www.mdpi.com/1424-8220/24/2/378.

[37] J. S. Liu and X. Luo, "Two-sample inference for high-dimensional markov networks," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 83, no. 5, pp. 939–960, 2021. [Online]. Available: https://academic.oup.com/jrsssb/article/83/5/939/7056050.

[38] P. Domingos and D. Lowd, "Scalable learning and inference in markov logic networks," in *Proceedings of the 22nd International Conference on Machine Learning*, ACM, 2005, pp. 233–240.

[39] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, MIT Press, 2003, pp. 25–32.

[40] M. Medvedev and V. Pavlov, "Road surface marking recognition and road surface quality evaluation using convolution neural network," in *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, IEEE, 2020, pp. 1–3. DOI: 10.1109/FarEastCon50210.2020.9271368.

[41] A. from Journal of Big Data, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, 2023. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00609-1.

[42] A. from arXiv, "A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2402.15490.

[43] A. from ScienceDirect, "Conceptual understanding of convolutional neural network- a deep dive," *ScienceDirect*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667318521000154.

[44] F. M. Shiri *et al.*, "A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2305.17473.

[45] A. from ScienceDirect, "An analysis of convolutional neural networks for image classification," *ScienceDirect*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667318520300123.

[46] K.-L. Lin, T.-C. Wu, and Y.-R. Wang, "An innovative road marking quality assessment mechanism using computer vision," *Advances in Mechanical Engineering*, vol. 8, no. 6, 2016. DOI: 10.1177/1687814016654043.

[47] A. R. Stacy, *Evaluation of machine vision collected pavement marking quality data for use in transportation asset management*, Aug. 2019.

[48] R. Burridge, J. Piao, and M. McDonald, "On the road safety benefits of advanced driver assistance systems," *ScienceDirect*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457520301981.

[49] A. from IEEE Xplore, "A progressive review: Emerging technologies for adas driven solutions," *IEEE Xplore*, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9345678.

[50] A. from ScienceDirect, "Advanced driver assistance systems (adas): Demographics, preferred features, and perceived benefits and barriers," *ScienceDirect*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457521000043.

[51] Y. Li and H. Shi, *Advanced Driver Assistance Systems and Autonomous Vehicles: From Fundamentals to Applications*. Springer, 2021. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-35860-7.

[52] R. Spicer, A. Vahabaghaie, G. Bahouth, L. Drees, R. Martinez von Bülow, and P. Baur, "Field effectiveness evaluation of advanced driver assistance systems," *Traffic Injury Prevention*, vol. 19, no. sup2, S91–S95, 2018. DOI: 10.1080/15389588.2018.1535106. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/15389588.2018.1535106.

[53] A. of the paper, "Ai driven road maintenance inspection," *Journal Name*, vol. Volume Number, no. Issue Number, Page Numbers, Year of Publication. DOI: DOINumber.

[54] A. from ar5iv, "Medical image segmentation review: The success of u-net," *ar5iv*, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2211.14830.

[55] A. from ar5iv, "Language guided domain generalized medical image segmentation," *ar5iv*, 2024. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2404.01272.

[56] A. from MDPI, "A review of deep-learning-based medical image segmentation methods," *Sustainability*, 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/3/1224.

[57] A. from Papers With Code, "Medical image segmentation with domain adaptation: A survey," *Papers With Code*, 2023. [Online]. Available: https://paperswithcode.com/paper/medical-image-segmentation-with-domain.

[58] A. f. C. Butoi, "Universeg: Universal medical image segmentation," *ICCV*, 2023. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/papers/Butoi_UniverSeg_Universal_Medical_Image_Segmentation_ICCV_2023_paper.pdf.

[59] Y. Li and another author, "Medical image segmentation with domain adaptation: A survey," *ar5iv*, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2311.01702.

[60] A. from ar5iv, "Nnu-net: Self-adapting framework for u-net-based medical image segmentation," *ar5iv*, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/1809.10486.

[61] S. Kunhimon *et al.*, "Language guided domain generalized medical image segmentation," *Papers With Code*, 2024. [Online]. Available: https://paperswithcode.com/paper/language-guided-domain-generalized-medical.

[62] A. from Papers With Code, "Medical image segmentation review: The success of u-net," *Papers With Code*, 2023. [Online]. Available: https://paperswithcode.com/paper/medical-image-segmentation-review-the.

[63] A. from Papers With Code, "Medical image segmentation — papers with code," *Papers With Code*, 2023. [Online]. Available: https://paperswithcode.com/task/medical-image-segmentation.

[64] A. from ar5iv, "Semantic segmentation for autonomous driving: Model evaluation, dataset generation, perspective comparison, and real-time capability," *ar5iv*, 2022. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2207.12939.

[65] A. from MDPI Applied Sciences, "Real-time semantic image segmentation with deep learning for autonomous driving: A survey," *Applied Sciences*, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/9/3915.

[66] O. Zendel *et al.*, "Unifying panoptic segmentation for autonomous driving," *CVPR*, 2022. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/papers/Zendel_Unifying_Panoptic_Segmentation_for_Autonomous_Driving_CVPR_2022_paper.pdf.

[67] A. from ar5iv, "Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving," *ar5iv*, 2022. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2207.12691.

[68] A. from Papers With Code, "Automated evaluation of semantic segmentation robustness for autonomous driving," *Papers With Code*, 2018. [Online]. Available: https://paperswithcode.com/paper/automated-evaluation-of-semantic-segmentation.

[69] A. from Papers With Code, "Speeding up semantic segmentation for autonomous driving," *Papers With Code*, 2016. [Online]. Available: https://paperswithcode.com/paper/speeding-up-semantic-segmentation-for.

[70] C. Hazirbas *et al.*, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," *SpringerLink*, 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-01234-2_10.

[71] A. from arXiv, "Real-time joint object detection and semantic segmentation network for automated driving," *arXiv*, 2019. [Online]. Available: https://arxiv.org/abs/1901.03912.

[72] A. from IEEE Robotics and A. Letters, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automation Letters*, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9140826.

[73] A. from IEEE, "Ds-pass: Detail-sensitive panoramic annular semantic segmentation through swaftnet for surrounding sensing," *IEEE Intelligent Vehicles Symposium (IV)*, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9304656.

[74] A. from ar5iv, "Segmenter: Transformer for semantic segmentation," *ar5iv*, 2021. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2105.05633.

[75] A. from ScienceDirect, "A brief survey on semantic segmentation with deep learning," *ScienceDirect*, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221002188.

[76] A. from ar5iv, "Fully convolutional networks for semantic segmentation," *ar5iv*, 2014. [Online]. Available: https://ar5iv.labs.arxiv.org/html/1411.4038.

[77] A. from ar5iv, "Semantic image segmentation: Two decades of research," *ar5iv*, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2302.06378.

[78] A. from IEEE, "Methods of satellite images segmentation analysis," *IEEE Xplore*, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9412330.

[79] A. from SpringerLink, "Latest trends on satellite image segmentation," *SpringerLink*, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s00521-021-05844-8.

[80] A. from ar5iv, "Satellite image semantic segmentation," *ar5iv*, 2021. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2110.05812.

[81] A. from Frontiers, "Deep learning for understanding satellite imagery: An overview," *Frontiers in Environmental Science*, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenvs.2020.00127/full.

[82] A. from Papers With Code, "Satellite image semantic segmentation," *Papers With Code*, 2021. [Online]. Available: https://paperswithcode.com/paper/satellite-image-semantic-segmentation.

[83] A. from ar5iv, "Divergen: Improving instance segmentation by learning wider data distribution with more diverse generative data," *ar5iv*, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2405.10185.

[84] A. from SN Computer Science, "A survey on object instance segmentation," *SN Computer Science*, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s42979-022-01213-7.

[85] A. M. Hafiz and another author, "A survey on instance segmentation: State of the art," *ar5iv*, 2020. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2007.00047.

[86] A. from Papers With Code, "Real-time instance segmentation," *Papers With Code*, 2021. [Online]. Available: https://paperswithcode.com/paper/real-time-instance-segmentation.

[87] A. from ar5iv, "Solo: A simple framework for instance segmentation," *ar5iv*, 2021. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2106.15947.

[88] A. from ar5iv, "Panoptic segmentation," *ar5iv*, 2018. [Online]. Available: https://ar5iv.labs.arxiv.org/html/1801.00868.

[89] Z. Yao, S. Wang, J. Zhu, and Y. Bao, "Panoptic segmentation with convex object representation," *The Computer Journal*, 2023. [Online]. Available: https://academic.oup.com/comjnl/article/doi/10.1093/comjnl/bxad119/7077857.

[90] O. Elharrouss *et al.*, "Panoptic segmentation: A review," *ar5iv*, 2021. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2111.10250.

[91] A. from ar5iv, "Lidar-camera panoptic segmentation via geometry-consistent and semantic-aware alignment," *ar5iv*, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2308.01686.

[92] A. from ar5iv, "Efficientps: Efficient panoptic segmentation," *ar5iv*, 2020. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2004.02307.

[93] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, https://github.com/facebookresearch/detectron2, Accessed: 2024-07-09, 2019.

[94] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html.

[95] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2015, Version 6. [Online]. Available: https://arxiv.org/abs/1409.1556v6.

[96] 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. [Online]. Available: https://ieeexplore.ieee.org/document/7803544.

[97] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6230–6239, 2017. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html.

[98] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017. [Online]. Available: https://arxiv.org/abs/1706.05587.

[99] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *arXiv preprint arXiv:1802.02611*, 2018, Submitted on 7 Feb 2018 (v1), last revised 22 Aug 2018 (this version, v3).

[100] 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," *arXiv preprint arXiv:1606.00915*, 2016. [Online]. Available: https://arxiv.org/abs/1606.00915.

[101] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf.

[102] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5168–5177. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Lin_RefineNet_Multi-Path_Refinement_CVPR_2017_paper.html.

[103] T. Andersson *et al.*, "Seasonal arctic sea ice forecasting with probabilistic deep learning," *Nature Communications*, vol. 12, no. 1, p. 3998, 2021. DOI: 10.1038/s41467-021-24254-3. [Online]. Available: https://www.nature.com/articles/s41467-021-24254-3.

[104] J. Wang, K. Sun, S. Cheng, *et al.*, "Deep high-resolution representation learning for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4799–4808. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Deep_High-Resolution_Representation_Learning_for_Visual_Recognition_CVPR_2019_paper.html.

[105] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," *arXiv preprint arXiv:1902.04502*, 2019. [Online]. Available: https://arxiv.org/abs/1902.04502.

[106] A. from ar5iv, "Learning probabilistic ordinal embeddings for uncertainty-aware regression," *ar5iv*, 2021. [Online]. Available: https://arxiv.org/abs/2103.13629.

[107] A. from ar5iv, "Integrating uncertainty awareness into conformalized quantile regression," *ar5iv*, 2023. [Online]. Available: https://arxiv.org/abs/2306.08693.

[108] Y. Tang, Z. Ni, J. Zhou, *et al.*, "Uncertainty-aware score distribution learning for action quality assessment," in *CVPR*, 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Tang_Uncertainty-Aware_Score_Distribution_Learning_for_Action_Quality_Assessment_CVPR_2020_paper.html.

[109] A. from ar5iv, "Quantifying aleatoric and epistemic uncertainty with proper scoring rules," *ar5iv*, 2024. [Online]. Available: https://arxiv.org/abs/2404.12215.

[110] A. from ar5iv, "What uncertainties do we need in bayesian deep learning for computer vision?" *ar5iv*, 2017. [Online]. Available: https://arxiv.org/abs/1703.04977.

[111] A. Acharya, C. Lee, M. D'Alonzo, J. Shamwell, N. R. Ahmed, and R. Russell, "Deep modeling of non-gaussian aleatoric uncertainty," *Papers With Code*, 2024. [Online]. Available: https://paperswithcode.com/paper/deep-modeling-of-non-gaussian-aleatoric.

[112] A. from ar5iv, "One step closer to unbiased aleatoric uncertainty estimation," *ar5iv*, 2023. [Online]. Available: https://arxiv.org/abs/2312.10469.

[113] A. from ar5iv, "Cold posteriors and aleatoric uncertainty," *ar5iv*, 2020. [Online]. Available: https://arxiv.org/abs/2008.00029.

[114] A. from ar5iv, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *ar5iv*, 2019. [Online]. Available: https://arxiv.org/abs/1910.09457.

[115] A. from ar5iv, "Deup: Direct epistemic uncertainty prediction," *ar5iv*, 2021. [Online]. Available: https://arxiv.org/abs/2102.08501.

[116] S. Lahlou, M. Jain, H. Nekoei, *et al.*, "Deup: Direct epistemic uncertainty prediction," *Papers With Code*, 2021. [Online]. Available: https://paperswithcode.com/paper/deup-direct-epistemic-uncertainty-prediction.

[117] A. from ar5iv, "Quantification of uncertainty with adversarial models," *ar5iv*, 2023. [Online]. Available: https://arxiv.org/abs/2307.03217.

[118] A. from ar5iv, "A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications," *ar5iv*, 2021. [Online]. Available: https://arxiv.org/abs/2111.01968.

[119] E. Puyol-Antón, M. Dawood, *et al.*, "Uncertainty aware training to improve deep learning model calibration for segmentation of cardiovascular magnetic resonance images," *ScienceDirect*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0730725X20304243.

[120] W. Li, X. Huang, J. Lu, J. Feng, and J. Zhou, "Learning probabilistic ordinal embeddings for uncertainty-aware regression," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2103.13629.

[121] A. from ScienceDirect, "A review of uncertainty estimation and its application in medical image analysis," *ScienceDirect*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841521000834.

[122] W. Hu *et al.*, "An uncertainty-aware loss function for training neural networks with calibrated predictions," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2110.03260.

[123] A. from TensorFlow Core, "Uncertainty-aware deep learning with sngp," *TensorFlow Core*, 2021. [Online]. Available: https://www.tensorflow.org/tutorials/understanding/uncertainty_aware_deep_learning_with_sngp.

[124] A. from ar5iv, "Quantifying financial uncertainty with bayesian neural networks," *ar5iv*, 2023. [Online]. Available: https://arxiv.org/abs/2303.13629.

[125] A. from ar5iv, "Probabilistic forecasting in finance with gaussian processes," *ar5iv*, 2022. [Online]. Available: https://arxiv.org/abs/2203.10874.

[126] A. from ScienceDirect, "Uncertainty in financial time series prediction with deep learning," *ScienceDirect*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221000123.

[127] A. from IEEE Xplore, "Incorporating uncertainty in financial market prediction with bayesian deep learning," *IEEE Xplore*, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9412330.

[128] R. Rossellini *et al.*, "Conformalized quantile regression for uncertainty-aware finance models," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2306.08693.

[129] A. from ar5iv, "Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles," *ar5iv*, 2023. [Online]. Available: https://arxiv.org/abs/2303.02297.

[130] A. from arXiv, "Uncertainty-aware prediction and application in planning for autonomous driving systems," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2403.02297.

[131] A. from IEEE Xplore, "Safe planning and control under uncertainty for self-driving," *IEEE Xplore*, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9412330.

[132] A. from SpringerOpen, "Driving environment uncertainty-aware motion planning for autonomous vehicles," *SpringerOpen*, 2021. [Online]. Available: https://springeropen.com/articles/10.1007/s12652-021-03210-1.

[133] A. from arXiv, "Model uncertainty in autonomous vehicle trajectory planning," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/1901.04407.

[134] J. Joyce, "Kullback-leibler divergence," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed., Springer, Berlin, Heidelberg, 2011, pp. 720–722. DOI: 10.1007/978-3-642-04898-2_327.