

https://dspace.org/

Semantic segmentation for South African Railway detection

Bagratee, Avrishka

2021
https://hdl.handle.net/10539/34690

Downloaded from a DSpace repository. https://dspace.org/

UNIVERSITY OF THE WITWATERSRAND



WITS
UNIVERSITY

School of Computer Science and Applied Mathematics
Faculty of Science
MASTERS RESEARCH REPORT

Semantic Segmentation for South African Railway Detection

Avrishka Bagratee

2288629

Supervised by Dr. Hairong Bau

Johannesburg, 2021

A research report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science.

Declaration
University of the Witwatersrand, Johannesburg
School of Computer Science and Applied Mathematics
SENATE PLAGIARISM POLICY

I, Avrishka Bagratee, (Student number: 2288629) am a student registered for SCA00 - Master of Science (Coursework and Research Report) in the year 2020.

I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that ALL the work submitted for assessment for the above course is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this in not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Signature: 

Signed on 23 day of October , 2021 in Johannesburg.

Abstract

Railway detection in images is a catalyst required to advance the railway industry towards autonomous trains; to enable automatic remote inspections of the railway from video footage; and to create an advance warning system for railway obstacles. Despite the benefits of railway detection, it is relatively unexplored territory and has only recently garnered attention.

This research aims to apply semantic segmentation models to South African railway images to achieve railway detection. As a prerequisite to the application, a dataset of 1000 South African railway images were collected and labelled.

Current railway detection methods are documented in the following papers: RailSem19, Railway Intrusion Detection and RailNet. These were analysed to determine the most suitable network for implementation. RailNet was found to achieve the highest mean IoU score of 89.8% amongst the networks reviewed and consequently selected to emulate.

The collected dataset was used to train the RailNet network as well as the SegNet and UNet network architectures in a configuration of 600 training images; 200 validation images and 200 test images. The SegNet, UNet and RailNet networks scored a mean IoU score of 90.32%, 92.93% and 92.99% respectively.

A total of 600 images from a publicly available dataset (RailSem19) were selected to supplement the South African training dataset to determine the effect of the foreign data on the performance of the networks. The additional data had an overall positive effect on the performance of the networks, increasing the baseline mean IoU values of SegNet, UNet and RailNet to 96.26%, 96.36% and 96.81% respectively.

The test set images were divided into seven (7) categories to determine the relationship between the networks performance and the terrain, luminosity or number of rails in the image. SegNet and UNet achieved the highest mean IoU score for the '$miscellaneous$' and '$rocky$' category respectively, while RailNet dominated the remaining five (5) categories: bright, dark, single rail, double rail and vegetation.

The application, Labelme, was used to create pixel-wise labels, segmenting the railways from the background in the South African images.

All networks in this research were implemented in Python, using the Keras Library with a Tensorflow backend and trained using Google Colab, making use of the GPU capabilities available.

The contributions of this research are: a labelled dataset of 1000 South African images for semantic segmentation of railways; determining the performance of RailNet (the current best performing rail detection network) on the South African dataset; comparing this performance with SegNet and UNet network architectures to establish the benchmark performance; determining the effect on the performance resulting from supplementing the South African dataset with foreign data; and investigating the relationship between the networks performance in relation to the the terrain, luminosity or number of rails in the image.

Acknowledgements

This journey would not have been possible without the overwhelming support of my parents whose nurturing has allowed me to pursue my ambitions. Thank you for your wisdom and sympathetic ear, the constant sustenance and warm beverages with generous servings of encouragement. To my brother, Kiran Bagratee, thank you for stimulating my mind with ideas and success stories that inspire me to be great.

I would like to thank my fiancé, Nikesh Poolchand, for his calming presence, reassurance and his general enthusiasm towards proof reading and labelling data when there were so many others things he could do with his time.

Throughout the duration of this degree, I have received a great deal of guidance, patience, and support both technically and emotionally from my supervisor, Dr Hairong Bau, of whom I am indebted to.

I would also like to thank my employer and sponsor, Transnet, for the financial support as well as accommodating the time commitment required to complete this degree.

Lastly, I would like to thank Dr. Andy Mabaso, for planting the initial seed of interest in machine learning and making me believe I could flourish.

To quote Sir Isaac Newton, ”If I have seen further than others, it is by standing upon the shoulders of giants”.

Contents

Declaration . . . i
Abstract . . . ii
Acknowledgements . . . iii

List of Figures **v**

List of Tables **viii**

1 Introduction **1**
1.1 Problem Statement . . . 2
1.2 Motivation . . . 2
1.3 Research Objectives . . . 3
1.4 Research Aims . . . 3
1.5 Research Questions . . . 4
1.6 Report Outline . . . 4

2 Background and Related Work **5**
2.1 Introduction . . . 5
2.2 Railway Environment . . . 5
2.3 Definition . . . 6
2.4 Related Work . . . 6
2.4.1 Current Road Detection . . . 6
2.4.2 Current Railway Detection . . . 7
2.4.2.1 RailSem19 . . . 7
2.4.2.2 Railway Intrusion Detection System . . . 7
2.4.2.3 RailNet . . . 8
2.4.3 Convolutional Neural Networks (CNNs) . . . 8
2.4.4 Semantic Segmentation . . . 9
2.4.4.1 Fully Convolutional Network (FCN) . . . 9
2.4.4.2 UNet . . . 10
2.4.4.3 SegNet . . . 11
2.4.5 Residual Networks . . . 11
2.4.6 Benchmark Dataset . . . 12
2.5 Conclusion . . . 12

3 Research Methodology **13**
3.1 Introduction . . . 13
3.2 Labelled Dataset . . . 14
3.2.1 Data Acquisition . . . 14

3.2.2 Data Pre-processing . . . 14
3.2.3 Labelling Data . . . 15
3.2.4 Resources . . . 16
3.2.5 Transfer Learning . . . 16
3.2.6 Potential Risks . . . 16
3.3 RailNet Network . . . 17
3.3.1 Feature Extraction . . . 17
3.3.2 Segmentation Network . . . 19
3.3.3 Dataset . . . 21
3.3.4 Implementation Details . . . 21
3.4 Selected Semantic Segmentation Models . . . 21
3.4.1 Dataset . . . 22
3.4.2 UNet . . . 22
3.4.3 SegNet . . . 22
3.5 RailSem19 Dataset . . . 23
3.5.1 Access to the Dataset . . . 23
3.5.2 Processing the RailSem19 Dataset . . . 23
3.5.3 Modifying the Label . . . 24
3.6 Experimental Setup . . . 24
3.6.1 Dataset . . . 24
3.6.2 Training . . . 25
3.6.3 Evaluation . . . 26
3.6.3.1 Overall . . . 26
3.6.3.2 Image Categories . . . 27
3.6.3.3 Reduced Test Set . . . 28
3.7 Conclusion . . . 28

4 Experimental Results **29**
4.1 Introduction . . . 29
4.2 SegNet . . . 29
4.3 UNet . . . 34
4.4 RailNet . . . 37
4.5 Conclusion . . . 41

5 Discussion **42**
5.1 Introduction . . . 42
5.2 Image Category Analysis . . . 42
5.3 Overall Mean IoU Score . . . 44
5.4 Comparison to the Literature . . . 46
5.5 Research Objectives . . . 47
5.6 Limitations . . . 48
5.6.1 Conclusion . . . 49

6 Conclusion **50**

Bibliography **52**

List of Figures

2.1 High-level railway components and cross-section schematic . . . 5
2.2 Cross-section of the rail indicating the "railway" in red . . . 6
2.3 Results of semantic segmentation models tested on the road-class of the Cityscapes dataset [Zhu et al. 2019] . . . 7
2.4 Architecture of LeNet-5, a Convolutional Neural Network [LeCun et al. 1998] . . . 9
2.5 Fully convolutional network [Long et al. 2015] . . . 9
2.6 UNet architecture [Ronneberger et al. 2015] . . . 10
2.7 SegNet architecture [Badrinarayanan et al. 2017] . . . 11
2.8 Residual learning building block [He et al. 2016] . . . 11

3.1 Collected dataset (a) Partial shadow, (b) Horizontal light source, (c) Full shadow, (d) Vegetation background, (e) Rock background . . . 14
3.2 Examples of images taken inside the tunnel as well as unfocused images that were removed from the dataset . . . 15
3.3 Labelling process (a) RGB image, (b) Labelled mask, (c) Mask overlaid onto the image . . . 15
3.4 The high-level structure of the RailNet network [Wang et al. 2019a] . . . 17
3.5 ResNet50 architecture with stage 2 exploded view . . . 18
3.6 Feature extraction network with stage 2 exploded view . . . 19
3.7 RailNet implementation . . . 20
3.8 Sample of RGB images and annotated labels . . . 23
3.9 Examples of the images that were removed from the dataset . . . 24
3.10 Modification of RailSem19 labels (a) RGB labels overlaid on RGB image, (b) Extracted mask, (c) Mask overlaid onto the image . . . 24
3.11 Dataset visualisation of the training sets . . . 25
3.12 Mean IoU metric code implementation [Tiu [2019]] . . . 27
3.13 Analysis of the test set . . . 27
3.14 Test set representation images with ground truth . . . 28

4.1 Semantic segmentation masks for SegNet on each dataset . . . 30
4.2 Semantic segmentation masks for SegNet600 on double rail images . . . 31
4.3 Visualisation of mean IoU Scores for the test set, for each dataset tested with SegNet . . . 33
4.4 Semantic segmentation masks for UNet on each dataset . . . 34
4.5 Test set: Image 127 and UNet predictions . . . 35
4.6 Visualisation of mean IoU scores for the test set, for each dataset tested with UNet . . . 36
4.7 Semantic segmentation masks for RailNet on each dataset . . . 38
4.8 Test set: Image 127 and RailNet prediction . . . 39
4.9 Visualisation of mean IoU scores for the test set, for each dataset tested with RailNet . . . 40

5.1 Comparison of the best performing network of each network architecture 43
5.2 Comparison of RailNet, UNet and SegNet on each dataset 45
5.3 A focused comparison of RailNet, UNet and SegNet on each dataset 46

List of Tables

3.1 The seven (7) training datasets and the contribution of images from each data source 25
3.2 Table to correspond the training datasets with the network architectures 26

4.1 Table of mean IoU scores for SegNet on each dataset 31
4.2 Table of mean IoU scores for each category of images trained on SegNet 32
4.3 Table of mean IoU scores for UNet on each dataset 35
4.4 Table of mean IoU scores for each category of images trained on UNet 37
4.5 Table of mean IoU scores for RailNet on each dataset 39
4.6 Table of mean IoU scores for each category of images trained on RailNet 41

5.1 Table of mean IoU scores for each category of images trained on SegNet 200, Unet 200 and RailNet 500 44
5.2 Table of mean IoU scores for RailNet, UNet and SegNet on each dataset 44
5.3 Table of mean IoU scores for the literature and research implementation 47

Chapter 1

Introduction

Railway detection in images is a catalyst required to advance the railway industry towards autonomous trains; to enable automatic remote inspections of the railway from video footage; and to create an advance warning system for railway obstacles. Despite the benefits of railway detection, this field of research is relatively unexplored territory and has only recently garnered attention.

Research contributions in 2019 from Australia (Zendel et al. [2019]), and China (Wang et al. [2019ab]) explore the use of deep learning models to achieve railway detection with the use of semantic segmentation. In order to replicate these semantic segmentation models or any semantic segmentation model for South African railways, a labelled dataset of railway images is first required. At the time of this research, only one global dataset for railways is in existence but does not include South African images. There are also no publicly available South African railway datasets. Furthermore, each of the network architectures mentioned above have been trained and tested on its own collection of data and no benchmark dataset for railways has been established.

As a prerequisite to the implementation of semantic segmentation models, this research documents the process to collect and label a dataset of 1000 images extracted from video footage taken from the driver perspective, during a locomotive test run in the Ondervalle station, situated in Mpumalanga, South Africa.

With an established dataset in hand, this report analyses the current railway detection methods documented in the following papers: RailSem19 (Zendel et al. [2019]), Railway Intrusion Detection (Wang et al. [2019b]) and RailNet (Wang et al. [2019a]). RailNet achieved the highest mean IoU score of 89.8% amongst the networks reviewed and was selected for implementation together with out-of-the-box networks (UNet and SegNet) to determine a benchmark performance on a common dataset.

For each of the implemented network architectures, the input is a driver-view image of a South African railway and the output is a binary mask indicating the railway in the image. When trained on the collected dataset in a configuration of 600 training images; 200 validation images and 200 test images, SegNet, UNet and RailNet scored a mean IoU score of 90.32%, 92.93% and 92.99% respectively.

To utilise the access to the global RailSem19 dataset, this research further investigates the effect of supplementing the collected South African dataset with the RailSem19 images to determine the effect that foreign data has on the performance of each network. A total of 600 images are selected to supplement the South African training dataset in increments of 100. The additional data had an overall positive effect on the performance of the network, increasing the baseline mean IoU values of SegNet, UNet and RailNet to 96.26%, 96.36% and 96.81% respectively.

Lastly, the test set images are divided into seven (7) categories to determine if the luminosity of the image, or the terrain or the number of rails in the image contributed to the performance of the networks. SegNet and UNet achieved the highest mean IoU score for the '$miscellaneous$' and '$rocky$' category respectively, while RailNet dominated the remaining 5 categories: bright, dark, single rail, double rail and vegetation.

The contributions of this research are: a labelled dataset of 1000 South African images for semantic segmentation of railways; determining the performance of RailNet (the current best performing rail detection network) on the South African dataset; comparing this performance with SegNet and UNet network architectures to establish the benchmark performance; determining the effect on the performance resulting from supplementing the South African dataset with foreign data; and investigating the relationship between the networks performance in relation to the terrain, luminosity or number of rails in the image.

1.1 Problem Statement

The main aim of this research is to determine the viability of applying deep learning models to detect railways in South African images but at the time of conducting this research, there are no publicly available datasets for semantic scene understanding of South African railway images.

As part of the foundation for this research, hand crafted, image processing methods were investigated to isolate the railway lines in images as a precursor to segment the railway. These included the applications of Clustering and Watershed methods as well as applying Canny Edge Detection. Other methods explored were based on separating the image into the various colour channels (H, S, V, R, G, B, Y, Cr, Cb) to determine if the pixel values for the railway could be isolated from the image with the use of a threshold. It was observed that these hand crafted methods for railway detection performed competently under specific circumstances for individual images but failed in scene changes, variations in light such as shadows, relied on the reflective property of the rails and were unable to generalise as a one-size-fits-all solution.

As railway systems around the world vary with respect to railway standards as well as the technology implemented, they tend to vary visually, as well as in the terminology used to describe components. Therefore, the global approach to semantic segmentation may not work well in the rail domain, particularly for South African railways which is a melting pot of railway technologies.

1.2 Motivation

To compete in the global market and stay ahead of the technological curve, it is necessary for the South African railway industry to advance research efforts towards autonomous trains. That said, semantic segmentation datasets are an essential first step in that direction and a prerequisite to training deep learning models to detect railways in images.

In addition, real-time detection of railway tracks could be used to enable the implementation of remote inspections of the railway from video footage as well as advance warning systems for railway obstacles. These obstacles include people, vehicles at level crossings, animals, fallen rocks, rail vehicles and even missing tracks. which will both positively impact the safety of the general public, maintenance staff and train drivers in the hazardous environment of the railway line.

1.3 Research Objectives

1.3.1 Generate a Labelled Dataset

The first objective of this research is to generate a dataset of driver perspective, South African railway images paired with labelled semantic segmentation masks for the railway region.

1.3.2 Implementation of the current best performing Segmentation Model for Railways

The collected dataset will be used to investigate the domain adaptation and performance of the current best performing segmentation network for railways on South African images.

1.3.3 Applying Transfer Learning to implement a few selected Segmentation Models

The collected dataset will be used to train a handful of out-of-the-box segmentation models with the available pre-trained weights to determine the benchmark performance on South African images. Using transfer learning greatly reduces the amount of training time as well as the use of hardware resources that are in scarce supply.

1.3.4 Supplementing the South African Railway images with a dataset from foreign countries

RailSem19 is a dataset of driver-view railway images that was collected from 38 countries (not including South Africa) and is available to the public for academic purposes on request. The objective of this research is to supplement the collected dataset with the RailSem19 dataset to determine the effect on the performance of the networks implemented as part of objectives 1.3.2 and 1.3.3 above.

1.4 Research Aims

The main aim of this research is to determine the viability of applying deep learning models to detect railways in South African images and in so doing, advance the technology used on South African railways.

This is turn, serves to increase the safety of the railway environment for humans and fauna alike while simultaneously progressing towards a future with autonomous trains.

1.5 Research Questions

The research presented in this report will address the following research questions:

- Can the best performing semantic segmentation model trained on railway data from another country achieve equal performance when trained on South African railway images?
- Which semantic segmentation model will achieve the best performance on the South African railway dataset while keeping the training and test dataset constant?
- What effect does supplementing the South African training dataset with labelled data from around the world, have on the performance of the networks tested on the South African test set?

1.6 Report Outline

Chapter 2 details the background of the rail domain and related semantic segmentation models, benchmark performances and directly related research conducted in the railway domain to date.

Chapter 3 discusses the research methodology that will be used to achieve the research objectives as well as the implementation details of each network architecture, the experimental setup and evaluation metrics.

Chapter 4 provides the experimental results acquired from implementing the selected semantic segmentation models.

Chapter 5 presents a discussion of each of the top-performing networks as well as a comparison of the effect of the supplemented data on the trained networks and the overall mean IoU score. This chapter compares the results achieved in the literature to that of the research, discusses the extent to which the research aims were achieved and finally discusses the limitations of the research.

Chapter 6 concludes the research by presenting the main results, providing answers to the research questions, presenting insights and discussing future work.

Chapter 2

Background and Related Work

2.1 Introduction

This chapter describes the fundamentals of the railway environment of a multi-aspect signalling system on South African railways. The chapter then progresses to a discussion of the related semantic segmentation models, benchmark performances and directly related research conducted in the railway domain.

2.2 Railway Environment

The high-level elements of a multi-aspect signalling system on South African railways can be described by the components in Fig. 2.1 and reflects the type of signalling system in the collected dataset.

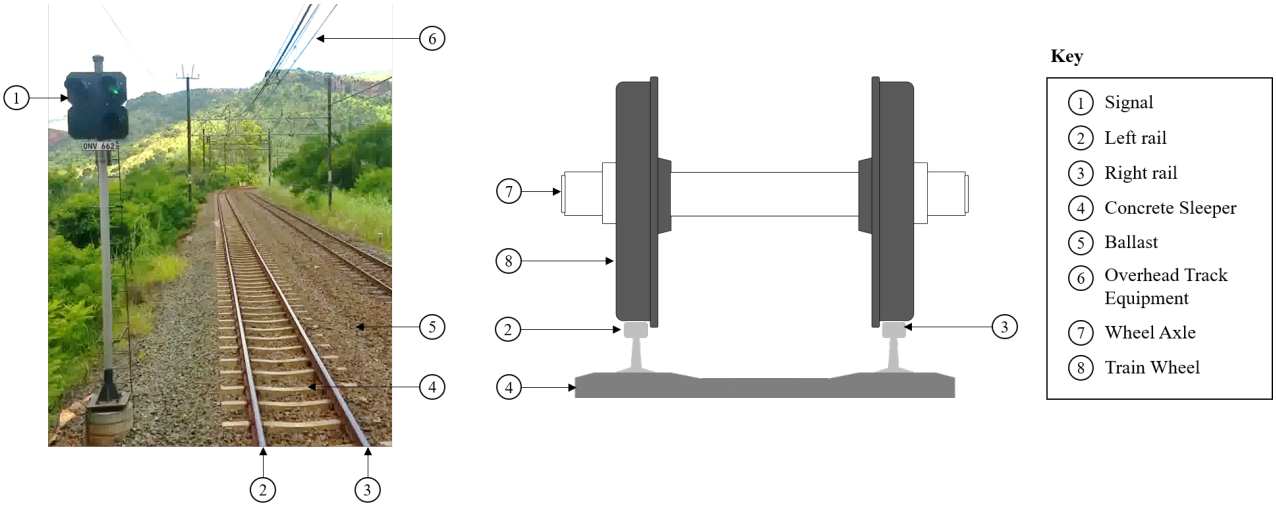


Figure 2.1: High-level railway components and cross-section schematic

Signals provide visual information to the train driver about the track ahead of the train, regarding the availability, route and safety information allowing the driver to control the speed and handle the train accordingly.

It must be noted that other variants of signalling systems exist in South Africa such as 3-aspect signalling and mechanical signalling, which are visually different, but as the focus of this research is based on rail detection, the type of signalling is irrelevant.

Fig. 2.1 also shows a schematic of the cross section of a railway line and its relation to a single set of train wheels. The crown of the left and right rail are the points of contact with the train wheels. The sleepers are connected to the rails to maintain the gauge of the rail and distribute the load of the train to the ballast. The ballast stones are packed between, below and around the sleepers to bear the load of the sleepers, facilitate drainage of water and reduce vegetation in the formation. The Overhead Track Equipment (OHTE) serves the purpose of supplying electricity to power the locomotives, moving the train along the railway line.

2.3 Definition

For the purposes of this research, the term “railway” is used to describe the crown of the left and right rail and the area in between. Fig. 2.2 shows the cross-section of the rail, indicating in red, the region of the railway and consequently the portion that is included in the railway label.



Figure 2.2: Cross-section of the rail indicating the “railway” in red

2.4 Related Work

2.4.1 Current Road Detection

Some of the best performing networks (as at 2019) for road detection on the Cityscapes dataset were the ResNet38 [Wu et al. 2019], Pyramid Scene Parsing Network (PSPNet) [Zhao et al. 2017] and DeepLabV3+ [Zhang et al. 2019], all scoring an IoU score of 98.7% according to the research conducted by [Zhu et al. 2019] with their own model scoring 98.8%. The table below (Fig. 2.3) is a snippet of the results, focusing on the road-class.

The most recent state of the art model (as at 2020) however uses hierarchical multi-scale attention for semantic segmentation and has achieved an IoU score of 99.0% for road detection according to research by [Tao et al. 2020] using an HRNet-OCR (High Resolution Network - Object Contextual Representations) as the trunk along with the proposed multi-scale attention method.

Method	split	road
Baseline	val	98.4
+ VRec with JP	val	98.0
+ Label Relaxation	val	98.5
ResNet38 [38]	test	98.7
PSPNet [44]	test	98.7
InPlaceABN [10]	test	98.4
DeepLabV3+ [14]	test	98.7
DRN-CRL [46]	test	98.8
Ours	test	**98.8**

Figure 2.3: Results of semantic segmentation models tested on the road-class of the Cityscapes dataset [Zhu et al. 2019]

2.4.2 Current Railway Detection

2.4.2.1 RailSem19

RailSem19 [Zendel et al. 2019] is a paper from the Australian Institute of Technology that details the process undertaken to collect and label a dataset for semantic rail scene understanding. The dataset consists of 8500 images from the ego perspective of trains for semantic rail scene understanding, collected from 38 countries (excluding South Africa). The paper details the use of a novel label policy for generating the labels.

Applying transfer learning, the paper presents an early prototype for pixel-wise semantic segmentation on rail scenes. It however, makes provision for 19 classes to align with the Cityscapes dataset, mostly utilising the same classes with some alterations. The classes of interest to the research proposal with respect to the Cityscapes dataset is the "rail-track" class which includes the rail and the space between the rails and regarding the RailSem19 dataset it is the combination of the "rail-track" and the "rail-raised" class to provide the same region of interest. A Full Resolution Residual Network (FRRNB) model is pre-trained on the Cityscapes dataset and fine-tuned with the RailSem19 dataset producing an overall mean IoU of 57.6% across the 19 classes with a mean IoU of 76.7% on average between the "rail-track" and "rail-raised" classes. As the RailSem19 dataset included a tram-track class as well as a rail-embedded class for tram tracks levelled with the road surface, this may have competed with the "rail-track" class and contributed to a lower mean IoU. As South Africa does not have trams, this conflict is not relevant to this research.

2.4.2.2 Railway Intrusion Detection System

A Railway Intrusion Detection System, proposed by [Wang et al. 2019b], details an adaptive segmentation algorithm to delineate the boundary of the track area with light computational burden. The proposed algorithm can be summarised into three steps.

The first, segments the image into fragmented regions. An optimal set of Gaussian kernels with adaptive directions for each specific scene is calculated using Hough transformation. This is to reduce the redundant

calculation in the evaluation of the boundary weight for generating the fragmented regions. Secondly, the fragmented regions are combined into local areas by using a new clustering rule based on the region's boundary weight and size. Finally, a classification network is used to recognise the track area among all local areas. To achieve a fast and accurate classification, a simplified CNN network is designed by using pre-trained convolution kernels and a loss function that can enhance the diversity of the feature maps.

This implementation detects a region which extends beyond the rail-track in comparison to the research conducted in this report which is restricted to the railway. The methodology of the work is however relevant with the mean IoU score for the algorithm achieving 85.23%.

2.4.2.3 RailNet

RailNet [Wang et al. 2019a] is a segmentation network for railroad detection and combines the ResNet50 backbone network with a fully convolutional network to predict a binary mask indicating the railroad. In this paper the label, "railroad", encompasses all pixels that exist between and including the railway lines and is equivalent to the definition of "railway" as defined in this research. RailNet uses the Railroad Segmentation Dataset (RSDS) generated by the authors and consists of 3000 images. This dataset however is not publicly available.

The RailNet network weights are initialised using the ResNet50 weights trained on the ImageNet Dataset. Random initialisation is used for other parameters and the network is trained on the RSDS training set for fine-tuning. The RailNet network scored a mean IoU of 89.8% which is a significant improvement on the RailSem19 results.

2.4.3 Convolutional Neural Networks (CNNs)

The architecture of Convolutional Neural Networks for digit recognition (Fig. 2.4) was introduced by [LeCun et al. 1998] and it remains to be a building block for modern day machine vision tasks. Using the convolution operation on images, allows features from the image to be extracted while retaining the spatial and temporal correlation in data. The multi-layered, hierarchical structure of a deep CNN enables it to extract low-level, mid-level, and high-level features [Khan et al. 2020].

For this reason, CNNs have had success with image classification and object recognition in images, and this is visible in the results of early network architectures such as AlexNet [Krizhevsky et al. 2012], VGGNet [Simonyan and Zisserman 2014], LeNet [LeCun et al. 2015], GoogLeNet [Szegedy et al. 2015] and ResNet [He et al. 2016] on which the most recent architectures are built around.

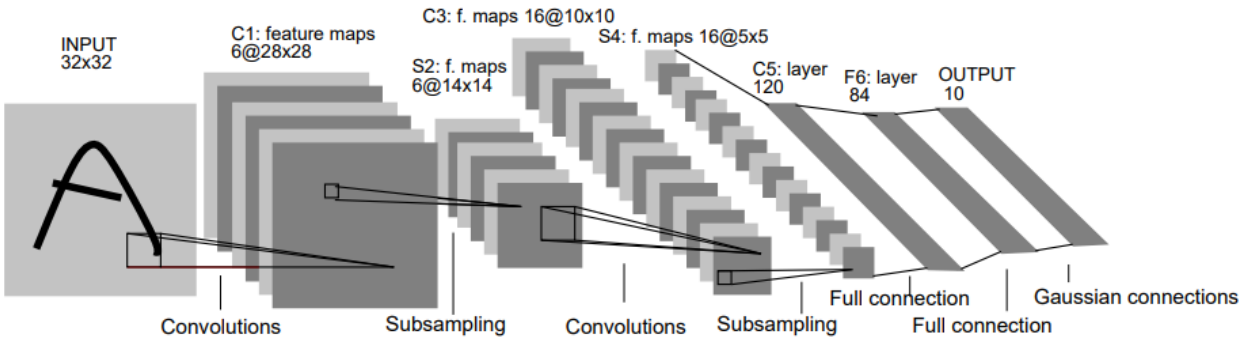


Figure 2.4: Architecture of LeNet-5, a Convolutional Neural Network [LeCun et al. 1998]

2.4.4 Semantic Segmentation

Semantic segmentation is the process of image classification at a pixel level which is why its roots are based in CNNs. The naïve approach [Krishnamurthy 2019] to achieve semantic segmentation began as mini, square patches of pixels of fixed size, fed into a CNN to classify the centre pixel of the patch as belonging to a certain class. This patch is moved across the entire image until every pixel has a classification. This process was inefficient, and the overhead related to computation increased proportionally with the image size. This approach did not allow for feature sharing among the overlapping patches and the small patches did not hold enough information about the surrounding pixels while taking larger patches increased the number of computations which resulted in a longer time to get an output.

2.4.4.1 Fully Convolutional Network (FCN)

Fully Convolutional Networks (FCN's) for semantic segmentation authored by [Long et al. 2015] adapts contemporary classification networks (AlexNet, the VGG net , and GoogLeNet) into fully convolutional networks and transfers their learned representations by fine-tuning to the segmentation task. The network architecture is shown in Fig. 2.5.

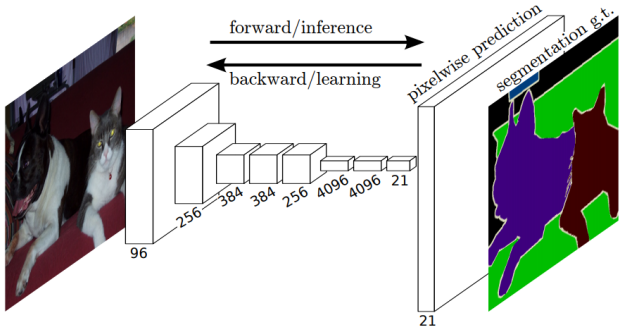


Figure 2.5: Fully convolutional network [Long et al. 2015]

The output of an FCN was found to be smaller due to subsampling which is intended to keep the filters small and computational requirements low. The outputs were also found to be coarse due to the reduction of the input size by a factor equal to the pixel stride of the receptive fields of the output units. As the probabilities are also coarse in nature, up-sampling is required to get the output to equal the input size. Simple bi-linear interpolation is inefficient, as a lot of the data is lost in the down-sampling process. To solve this problem, feature maps are picked from the intermediate layers and added to the output in an attempt to reclaim the resolution that was lost.

2.4.4.2 UNet

Developed by [Ronneberger et al. 2015], the UNet is an end-to-end encoder-decoder network originally created for biomedical semantic segmentation and designed to work with small datasets to produce precise segmentations. The architecture is a "U" shape and is symmetric around the vertical axis, consisting of an encoder (the left half of the "U"), the decoder (the right half of the "U") and the bottleneck connecting the two as shown in (Fig. 2.6). The encoder is used to down-sample and extract feature representations at each level while the decoder is used to up-sample and concatenate feature maps together with the use of skip connections from the encoder.

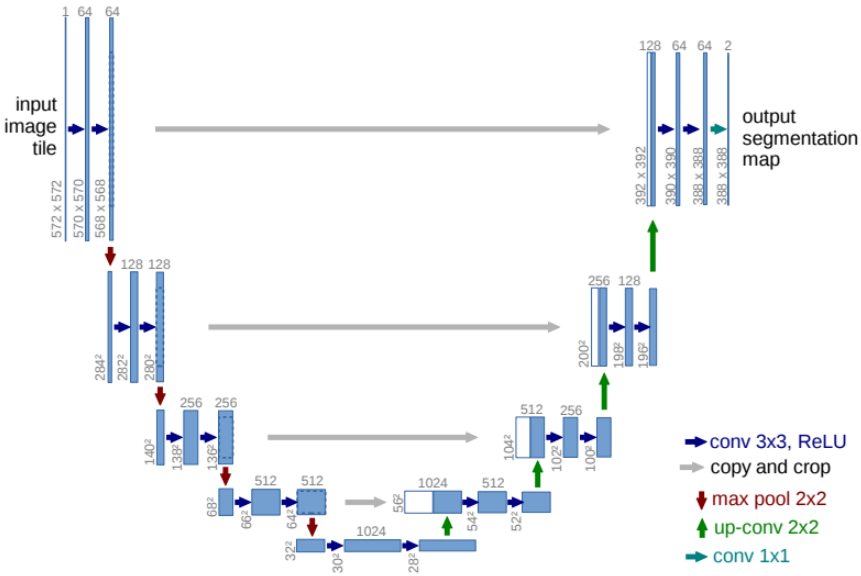


Figure 2.6: UNet architecture [Ronneberger et al. 2015]

The skip connections provide local information to global information while up-sampling the output. The final layer is a 1x1 convolution that is used to map each feature vector to the number of classes. Although the UNet network was designed to process biomedical images, this architecture has had success outside of the medical imaging field as well.

The UNet is particularly useful to this research due to the pre-trained model having success on relatively small datasets (<100 images) which is relevant under low resource requirements in both the time required to label

data as well as computing power.

2.4.4.3 SegNet

To further increase the resolution in the output image [Badrinarayanan et al. 2017], introduced a decoder network that uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear up-sampling (max unpooling). The complete architecture (Fig. 2.7) consists of a convolutional encoder network and a corresponding decoder network of which the output feature maps are fed into a soft-max classifier for pixel-wise classification.

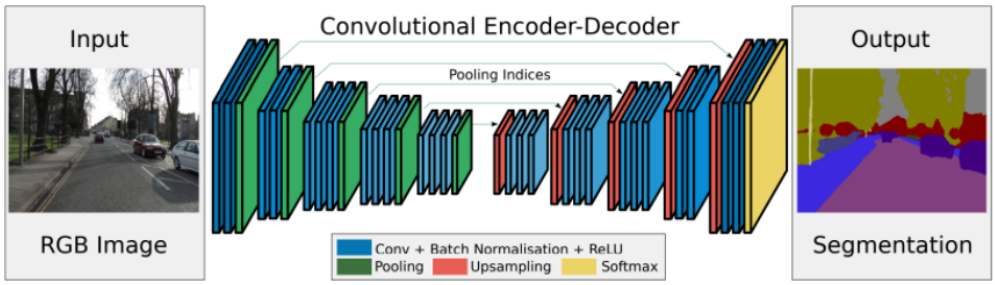


Figure 2.7: SegNet architecture [Badrinarayanan et al. 2017]

As the SegNet network is trained on the Cambridge-driving Labeled Video Database (CamVid) dataset [Brostow et al. 2008] for road scene images, it would be of value to investigate this network on rail scene images to take advantage of the similar characteristics between road and rail in terms of linearity and driver perspective as the road converges in the same way as the rail, as it approaches the horizon line.

2.4.5 Residual Networks

As neural networks became deeper in an effort to increase the training accuracy, the opposite occurred. The training accuracy was found to saturate and then degrade rapidly due to the vanishing gradient problem. Residual networks were designed by [He et al. 2016] to address this problem with an identity connection between the layers (Fig. 2.8). The identity connection originates from the input and it added to the output of the residual block.

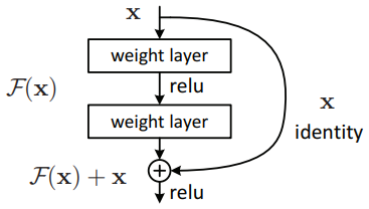


Figure 2.8: Residual learning building block [He et al. 2016]

At the core ResNet's use batch normalisation to adjust the input layer and bottleneck residual design to increase the performance of the network and mitigate the problem of covariate shift. This research will home in on the ResNet50 architecture as a feature extraction mechanism and will be discussed in further detail in Chapter 3.

2.4.6 Benchmark Dataset

As there are currently no benchmark datasets for semantic segmentation networks on a publicly available railway scene dataset, the second-best option was to review the models trained on road scene datasets to determine the current state of the art model. These datasets are the CamVid dataset [Brostow et al. 2008] and the Cityscapes dataset [Cordts et al. 2016]. Cityscapes is a dataset for visual understanding of complex urban street scenes and a large-scale dataset to train and test approaches for pixel-level semantic labelling.

Since the Cityscapes dataset is a more widely used dataset for benchmarking semantic segmentation networks, this research will determine the current state-of-the-art network based of the results achieved from training on this dataset.

This dataset contains 19 classes, of which the "road class" will be used to determine the best performing model and not the mean Intersection of Union (IoU) across the classes of the dataset. As mentioned before, this is to take advantage of the similarities between the road and rail in terms of linearity and driver perspective as the road converges in the same way as the rail, as it approaches the horizon line.

2.5 Conclusion

This chapter provided information about the key components required in order to achieve semantic segmentation of railways in South African images. The terminology used in the railway environment was described in order to define the region of interest for this research. Convolutional Neural Networks (CNNs) were discussed as the main building block to semantic segmentation as well as more recent network architectures that have had success in object detection. As part of the semantic segmentation networks, the UNet, SegNet and Residual Networks were elaborated on as these networks will be relevant to the implementation. The Cityscapes dataset was used to determine the current state-of-the-art network for semantic segmentation on the road-class. Finally the most recent implementations of rail detection using semantic segmentation networks were discussed to determine the best performing network in the rail domain.

Chapter 3

Research Methodology

3.1 Introduction

The previous chapter presented the recent work in semantic segmentation as well as applications of semantic segmentation in the rail domain. The RailSem19 paper [Zendel et al. 2019] trained an FRRNB model to predict 19 classes of pixel-wise semantic segmentation with the lowest mean IoU score of 76.7% for the region of interest for this research. The RailSem19 dataset will however be used to supplement the South African dataset. The Railway Intrusion Detection System [Wang et al. 2019b] detected a larger area than the region of interest and focused on handcrafted methods to reduce the computational burden. Finally, the RailNet paper [Wang et al. 2019a] was most aligned to the work in this proposal and also achieved the highest mean IoU score of 89.5% for railway detection on its own dataset (RSDS). The RailNet network architecture is therefore selected as the baseline model to replicate and benchmark the performance of the network on the South African dataset.

This chapter outlines the methodology that was used to achieve the research objectives and can be summarised into the following deliverables:

- Generating the Labelled Dataset
- Implementation of the RailNet Architecture
- Implementation of out-of-the-box Semantic Segmentation Models
- Modifying the RailSem19 dataset to supplement the collected dataset and re-train the models

All networks in this research were implemented in Python, using the Keras Library with a Tensorflow backend and trained using Google Colab (a Python development environment that runs in a browser using Google Cloud), making use of the GPU capabilities available.

In order to compare the performance of the networks implemented in this research on specific image types, the validation and test set images have been kept constant throughout the training of the networks. The validation and training sets consist of 200 images each, and contain images from the South African dataset only.

3.2 Labelled Dataset

The first objective of this research was to collect and label a dataset of 1000 South African railway images.

3.2.1 Data Acquisition

As part of a locomotive test run in Ondervalle and surrounding areas (situated in Mpumalanga, South Africa), video footage taken from the front of the locomotive was acquired and used to create a dataset. This video footage was converted into frames varying in lighting conditions, background scene and terrain. Samples of the dataset are shown in (Fig. 3.1).



Figure 3.1: Collected dataset (a) Partial shadow, (b) Horizontal light source, (c) Full shadow, (d) Vegetation background, (e) Rock background

As the dataset collected is limited to images from the video footage, it does not include images taken at night under artificial light.

3.2.2 Data Pre-processing

The dataset acquired was processed to remove unusable images i.e. scenes in a tunnel where the rails could not be extracted accurately from the image to be labelled and images where the camera was not focused on the railway in front of the train. Examples of the removed images are shown in (Fig. 3.2).

The frame extraction rate from the video was determined to be low enough to see a visible shift forward between the images. The images in the dataset were taken in both portrait and landscape mode with a size of 1920 x

1080 pixels, however all images were resized during training to suit the implemented network.



Figure 3.2: Examples of images taken inside the tunnel as well as unfocused images that were removed from the dataset

3.2.3 Labelling Data

The Labelme [Wada 2016] graphical image annotation tool was used to create annotations of the railway via the polygon option creating multiple points to accurately capture the pixels belonging to the railway as the ground truth label. This application provided a JSON file for the annotation that was used to create a mask at full resolution. The zoom function, brightness and contrast options greatly assisted in creating accurate masks with pixel level accuracy. This tool was selected for ease of use and can be installed easily via the terminal window. Additionally, the polygon can be edited should the mask need to be adjusted for increased precision and this feature was critical in the reviewing process elaborated on in the next section. (Fig. 3.3) shows the RGB image, the labelled mask generated from the JSON file and the mask overlaid over the original image to provide a visual indication of the accuracy of the label.



(a) (b) (c)

Figure 3.3: Labelling process (a) RGB image, (b) Labelled mask, (c) Mask overlaid onto the image

3.2.4 Resources

To achieve the objective of labelling 1000 images, additional resources were recruited to assist with the labelling. The time for labelling an image with double railways averaged to 15 minutes while a single railway on average took 4 minutes to label. To standardise the quality of the labels, a video tutorial of the labelling process was distributed to the labellers in addition to both negative and positive examples.

The following measures were taken to control the quality and progress of labelling:

- distributing the allocations of images in batches starting at 10
- reviewing the labels for accuracy
- editing the label where necessary and providing feedback before distributing the next batch
- increasing the batch size in 10's based on the confidence in the labeller
- keeping the batch size small enough to allow the labeller a sense of accomplishment but large enough to make the administrative overheads worthwhile

It was found that a final batch size of 40 images was a good balance of warranting administrative overheads while being an achievable task size for the labeller to complete.

3.2.5 Transfer Learning

The application of transfer learning was restrained to the use of pre-trained networks. To reduce the training time on the collected dataset, as well as to compensate for the size of the dataset, pre-trained networks were employed to exploit the low-level features learnt from larger datasets that are common to railways. This was achieved by initialising the weights of the implemented network to the pre-trained weights of the same network, that are as a result of training on a larger dataset such as ImageNet, CamVid or Cityscapes. In doing this, the implemented network weights are then fine-tuned to that of the collected dataset, reducing the total training time.

3.2.6 Potential Risks

As the data is sequential frames from video footage, this leads to the data not being independent and identically distributed. As there are a total of 2000 frames available from the video footage, the 600 images that will be used for the training set, will come from sequential data, while the validation and testing set, will be randomised from the remaining 1400 images to alleviate the dependence in the dataset.

3.3 RailNet Network

The RailNet architecture is split into a feature extraction portion and a segmentation portion depicted below (Fig. 3.4). The feature extraction portion is a multi-level, independent network designed outside a ResNet50 backbone network [He et al. 2016], while the segmentation portion is a fully convolutional network. This section details the implementation undertaken by the author of this research to reproduce the RailNet network. Although every effort was taken to closely align the implementation as the RailNet architects intended, due to the resource material being limited to the published paper, the implementation may vary at the discretion of the author in order to progress towards an end result.

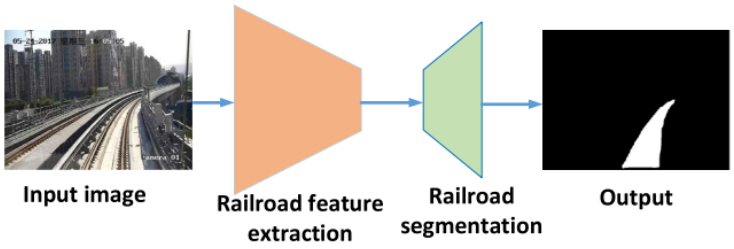


Figure 3.4: The high-level structure of the RailNet network [Wang et al. 2019a]

3.3.1 Feature Extraction

The ResNet50 network has been grouped into 5 stages. Fig. 3.5 shows the exploded view of stage 2 in the ResNet50 architecture to highlight the convolution operations that transpire within the Convolution Block and Identity Block as these outputs will be utilised for the feature extraction network.

Fig. 3.6 shows how the RailNet feature extraction network is attached to the ResNet50 architecture to produce outputs S2 – S5. The ResNet50 network weights were initialised to the pre-trained weights of the ImageNet dataset.

The average pooling layer, fully connected layer and classification layer of the ResNet50 architecture were not utilised for the feature extraction. Inside stage 2 to stage 5, the output of each convolution operation, within the Convolution Block as well as the Identity Block, is connected to another convolution operation with a kernel size of 1 x 1 and a depth of 64. The several outputs of these convolutions are added for each stage to obtain outputs S2, S3, S4 and S5. For stages 3 to 5, the sum of the outputs of the Convolution Block were up-sampled to match the dimensions of the Identity Block outputs before being added together.

Outputs S2 – S5 are then processed through a second convolution operation with a kernel size of 1 x 1 and a depth of 256. A Top-down pathway was then added to make the features have a backward propagation, with the goal to use these hybrid features P2 – P5 for the railway segmentation network.

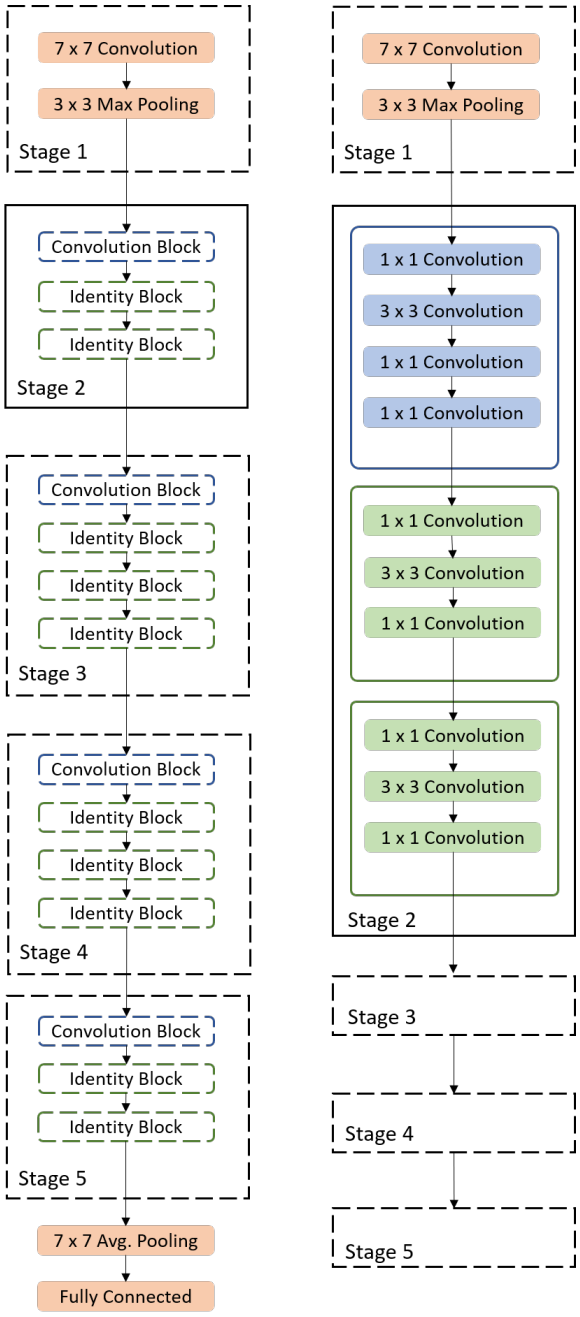


Figure 3.5: ResNet50 architecture with stage 2 exploded view

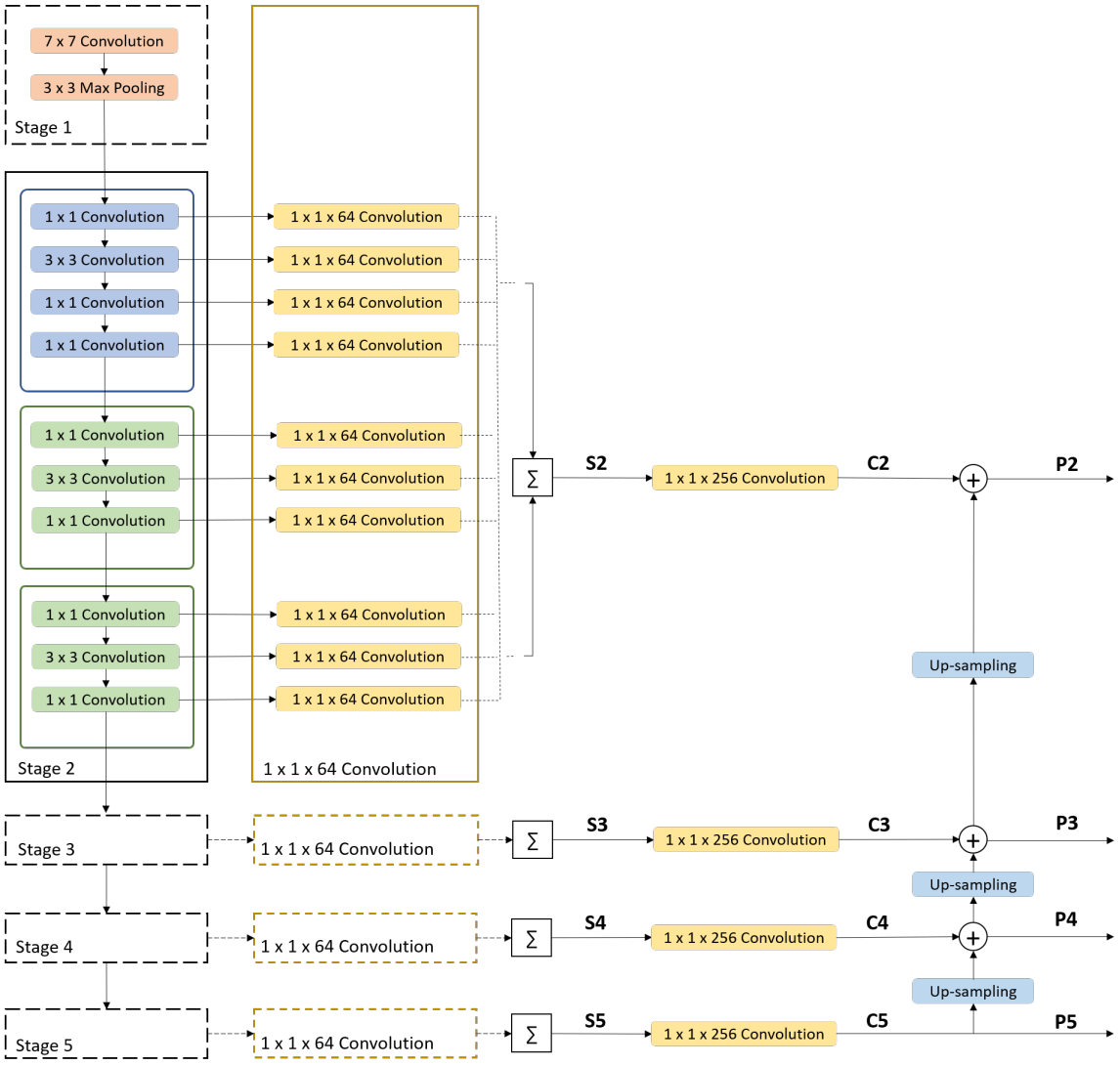


Figure 3.6: Feature extraction network with stage 2 exploded view

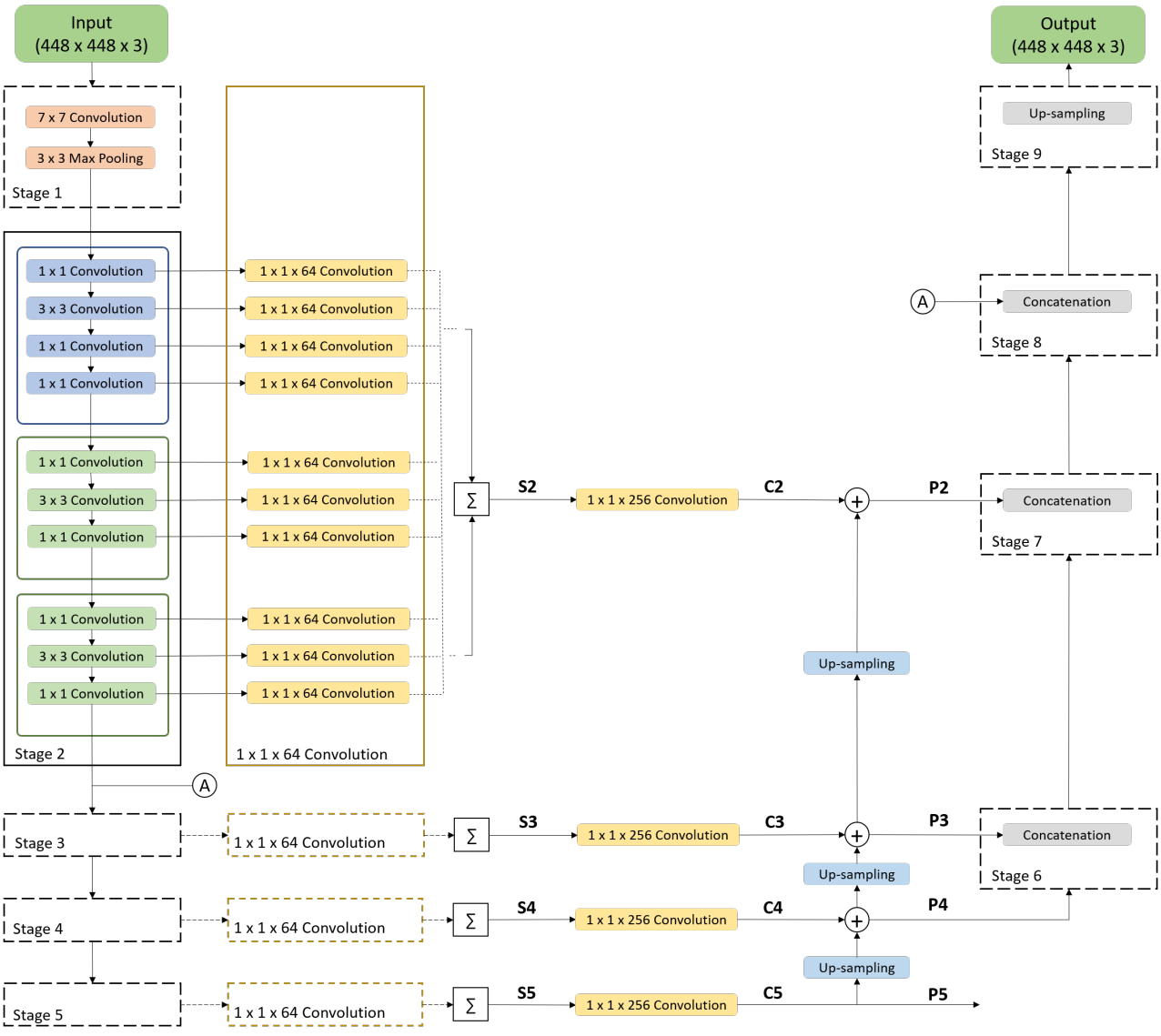
3.3.2 Segmentation Network

The segmentation portion as documented in the RailNet paper [Wang et al. 2019a] uses a fully convolutional network to compute the segmentation for the entire image. The features (P2 – P5) are converted to a fixed size and passed through several layers of convolution and deconvolution, until being reduced to a one-dimensional matrix, called Seg-map. Each value in the Seg-map represents the probability of belonging to a railway class, and the final map is generated by defining the threshold probability.

As the detail regarding the number of layers of convolution and deconvolution, the kernel sizes and the segmentation map is not elaborated on further, this has allowed for some variability in the implementation of the segmentation portion of the RailNet network.

The implementation in this report deviates from the RailNet approach in that it instead uses the ResUNet structure, focusing on the decoder portion to replace the segmentation network and complete the RailNet architecture.

Fig. 3.7 illustrates the high-level block diagram of the RailNet implementation, indicating the connection amongst the ResNet50 network that forms the basis of the encoder, the feature extraction network that produces outputs P2 to P5 and the segmentation network that results in the output image.


Input
(448 x 448 x 3)
7 x 7 Convolution
3 x 3 Max Pooling
Stage 1
1 x 1 Convolution
3 x 3 Convolution
1 x 1 Convolution
1 x 1 Convolution
Stage 2
A
Stage 3
Stage 4
Stage 5
1 x 1 x 64 Convolution
1 x 1 x 64 Convolution
1 x 1 x 64 Convolution

It can be seen that the output of Stage 2 of the ResNet50 (shown in the diagram as "A"), contributes to the concatenation step in Stage 8 of the diagram. This connection was created to accommodate the requirements of the ResUNet structure. It can also be seen that the output P5, is equivalent to C5 and therefore only contributes to the decoder via P4, as an up-sampled version.

3.3.3 Dataset

To train the network, the collected dataset of 1000, labelled, South African railway images were divided into 600 images for training, 200 images for validation and 200 images were used as a test set. This differs from the RailNet paper in that the RailNet paper used 2500 images for training, 200 images for validation and 300 images for testing. The dataset used by the RailNet paper was unfortunately not available for public use.

3.3.4 Implementation Details

As per the details in the RailNet paper, the weights of the ResNet50 network were initialised to the pre-trained weights of the network trained on the ImageNet dataset. The Adaptive Moment Estimator (ADAM) stochastic gradient descent learning algorithm was utilised during the training phase of the network. The hyper-parameters during training were set to: mini-batch size = 8, learning rate = 5e-3, scheduled decay = 0.004, $\beta 1 = 0.9$, $\beta 2 = 0.999$ and $\epsilon = 1e - 08$. The initial learning rate was set to 5e-3 and after 200 epochs of training, the learning rate was adjusted to 1e-3. The entire training was terminated after 300 epochs. The loss function was selected as the categorical cross-entropy loss function shown in Equation (3.1).

$$Loss = -\sum_{i=1}^{outputsize} y_i . \log \hat{y}_i, \tag{3.1}$$

where $\hat{y}_i$ is the i-th scalar value in the model output, y_i is the corresponding target value, and "output size" is the number of scalar values in the model output.

All input images were resized to 448 x 448 x 3, with the output predicted mask up-sampled to 448 x 448. The code used as the basis for the implementation of the RailNet network is a combination of resources available on Github (Tomar [2021], Gupta [2021]). The code has been adapted and customised for the purposes of this research.

3.4 Selected Semantic Segmentation Models

To test the performance of the collected dataset on other semantic segmentation models, two networks were selected. These networks are the UNet [Ronneberger et al. 2015], due the success of the network on small datasets

and the SegNet [Badrinarayanan et al. 2017] based on the success of detecting roads using the Cityscapes Dataset.

3.4.1 Dataset

The collected dataset for South African Railway images consisting of 1000 images will be used in the following split: a training set of 600 images, validation set of 200 images and test set of 200 images.

3.4.2 UNet

A pre-trained VGG16 network trained on the ImageNet dataset was used as the encoder for the UNet network. All input images were resized to 224 x 224 x 3 to align with the size of the images that the VGG16 network inputs were originally trained on. The ADAM stochastic gradient descent learning algorithm was utilised during the training phase of the network with a learning rate of 1e-4, with the loss function selected as the multi-class, cross-entropy loss function. For this implementation, to provide the optimiser with sufficient time to find a better path towards the minima, the technique of reducing the learning rate on plateau was utilised with the following hyperparameters: monitor = validation loss, patience = 3, factor = 0.1, verbose = 1 and minimum learning rate = 1e-6. To avoid over-fitting the training data, early stopping was also implemented with the following hyperparameters: monitor = validation loss, patience = 5 and verbose = 1. A batch size of 1 was used as the initial setting, the number of classes are set to 2 (railway class and background), and the number of epochs were set to 40. The Softmax activation function was selected to determine the probability of each class. The code used as the basis for the implementation of the UNet network is available on Github (Tomar [2021]). It has been adapted and customised for the purposes of this research.

3.4.3 SegNet

A pre-trained VGG16 network trained on the ImageNet dataset was used as the encoder for the SegNet network as well. All input images were resized to 224 x 224 x 3 to align with the size of the images that the VGG16 network inputs were originally trained on. The Keras SGD class was used to implement the stochastic gradient descent optimizer with a momentum of 0.9 as per the settings in the SegNet paper (Badrinarayanan et al. [2017]) but instead of using a fixed learning rate, a scheduled decay of 0.0005 was applied to the learning rate. The initial rate was set to 1e-4, decreasing to a minimum of 5e-5 after 60 epochs. The loss function was selected as the multi-class, cross-entropy loss function. A batch size of 8 was used as the initial setting, the number of classes are set to 2 (railway class and background), and the number of epochs were set to 100. The Softmax activation function was selected to determine the probability of each class. The code used as the basis for the implementation of the SegNet network is available on Github (Gupta [2021]). It has been adapted and customised for the purposes of this research.

3.5 RailSem19 Dataset

The final objective of this research was to supplement the training dataset with the RailSem19 dataset to analyse the effect on the South African test set. RailSem19 offers 8500 unique images taken from the ego-perspective of a rail vehicle (trains and trams). Extensive semantic annotations are provided for 19 classes, both geometry-based (rail-relevant polygons, all rails as polylines) and dense label maps with many Cityscapes-compatible road labels. An example of the image data and annotated label can be seen in Fig. 3.8.

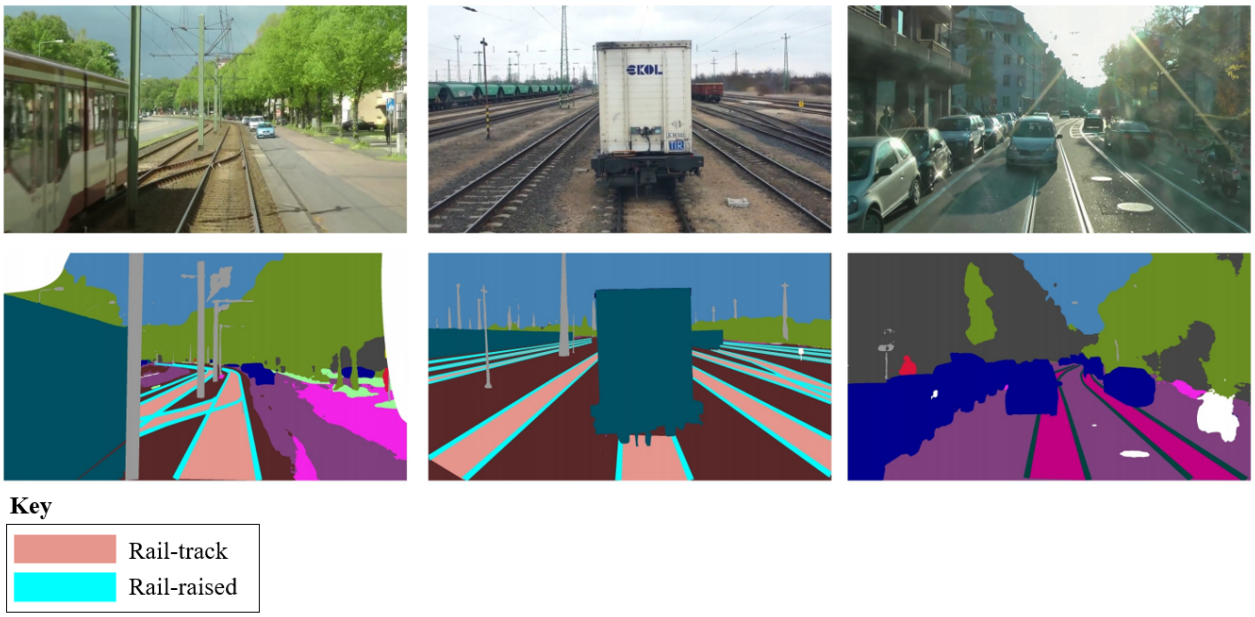


Figure 3.8: Sample of RGB images and annotated labels

3.5.1 Access to the Dataset

In order to use the RailSem19 dataset, an online application was completed and was successful. As the dataset has been collected from around the world, the author was contacted to confirm that South African railway images were not included in the dataset as part of the 38 participating countries.

3.5.2 Processing the RailSem19 Dataset

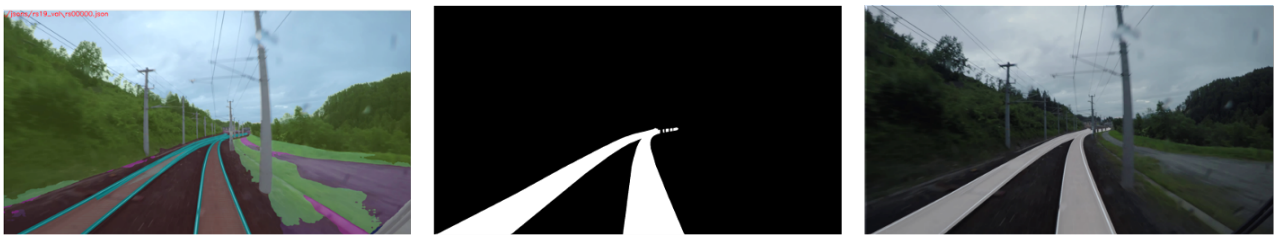
The 8500 RailSem19 images were pre-processed to find 600 images in the dataset that were most similar to the South African dataset collected. This meant excluding urban scenes with trams and tram lines (Fig. 3.9) and selecting images with vegetation and natural/ rural scenes. It was noted that the RailSem19 images had the same resolution of 1920 x 1080 pixels as the collected dataset which will assist in merging the images and processing the images alike.



Figure 3.9: Examples of the images that were removed from the dataset

3.5.3 Modifying the Label

To use the RailSem19 masks to supplement the South African collected dataset, the masks needed to be processed to conform to the region of interest of this research. As the RailSem19 masks contained 19 labels, the masks had to be reduced to binary masks to align with the labels in the collected dataset. This was executed by writing a Python script to process the selected images from the RailSem19 dataset. The script entailed the conversion of the RailSem19 masks from the RGB colour channel to greyscale, identifying the pixel value of the two classes of interest, i.e. "rail-track" and "rail-raised" classes, converting these values to 255 (as the images are using an 8-bit representation) and lastly modifying the remaining labels to be a background label with a pixel value of 0. (Fig. 3.10) shows the original RGB mask with 19 class labels overlaid on the RGB image, the binary mask as a result of combining the two classes into one and the binary mask overlaid on the RGB image.



(a) (b) (c)

Figure 3.10: Modification of RailSem19 labels (a) RGB labels overlaid on RGB image, (b) Extracted mask, (c) Mask overlaid onto the image

3.6 Experimental Setup

3.6.1 Dataset

The two sources of labelled data (South African collected dataset and RailSem19) were used to create 7 datasets that will be used to train each of the network architectures. This structure of data supports the investigation into the performance of the network on the pure South African dataset and on the supplemented dataset.

For this experiment the foreign data will be added to the South African dataset in increments of 100, up to a maximum of 600 images. While it is true that only 600 images from the RailSem19 dataset were identified to be similar to the collected dataset, it was also reasonable to cap the foreign data at 600 images (equal in number to the South African dataset) to avoid saturating the training dataset with foreign data. The table below (Table 3.1) displays the name of each dataset and the contribution of images from each data source.

Dataset	No. of SA Images	No. of RailSem19 images	Total Training Samples
Dataset 000	600	0	600
Dataset 100	600	100	700
Dataset 200	600	200	800
Dataset 300	600	300	900
Dataset 400	600	400	1000
Dataset 500	600	500	1100
Dataset 600	600	600	1200

Table 3.1: The seven (7) training datasets and the contribution of images from each data source

Fig. 3.11 illustrates the distribution of the training data from the two data sources and the relation of each training set to the other regarding the overlap of data.

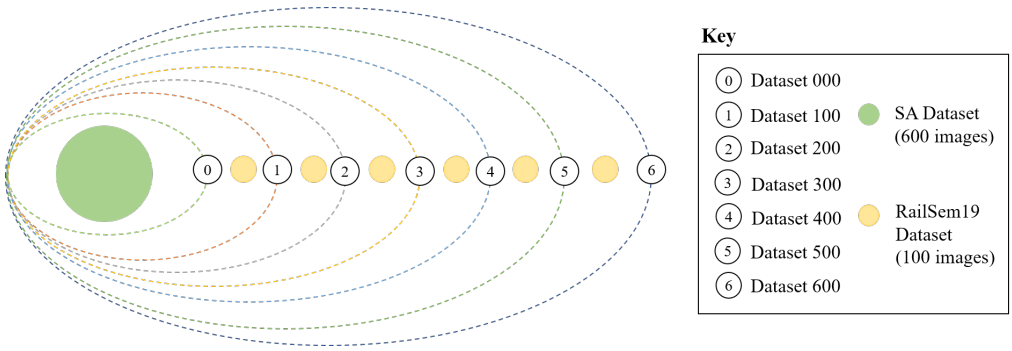


Figure 3.11: Dataset visualisation of the training sets

The test and validation dataset consist of 200 images each and come from the South African dataset only.

3.6.2 Training

A total of 3 network architectures were trained on 7 datasets. Table 3.2 provides a label to each network trained in this research, corresponding the dataset used with the network architecture.

Dataset	RailNet	UNet	SegNet
Dataset 000	RailNet000	UNet000	SegNet000
Dataset 100	RailNet100	UNet100	SegNet100
Dataset 200	RailNet200	UNet200	SegNet200
Dataset 300	RailNet300	UNet300	SegNet300
Dataset 400	RailNet400	UNet400	SegNet400
Dataset 500	RailNet500	UNet500	SegNet500
Dataset 600	RailNet600	UNet600	SegNet600

Table 3.2: Table to correspond the training datasets with the network architectures

To train each of the networks, i.e. RailNet, UNet and SegNet, all the hyperparameters as described previously in Section 3.3 and Section 3.4, are kept constant. This is to ensure that there is no advantage induced during the training, at least within the same network category.

To reduce any variability across the networks, in terms of the training images, the batches of foreign data have also been kept constant such that two network architectures trained with 200 additional foreign images, are both trained on identical datasets.

Lastly, the test and validation data is kept constant across all networks and architectures to allow a fair comparison of the results across specific image types and specific images in the test set, in addition to the overall performance and come from the South African dataset only.

3.6.3 Evaluation

3.6.3.1 Overall

The Intersection of Union (IoU) metric (3.2), also referred to as the Jaccard Index, is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth [Tiu 2019].

$$IOU(Y_T, Y_P) = \frac{|Y_T \cap Y_P|}{|Y_T \cup Y_P|}, \tag{3.2}$$

where Y_T is the set of ground truth pixels classified as either foreground or background, and Y_P is the set of predicted pixels. The metric ranges from 0 to 1 or as it is used in this research, 0 - 100%, with 0% signifying no overlap and 100% implying perfectly overlapping segmentation.

The mean IoU (mIoU) of the image is calculated by taking the IoU of each class and averaging the result. This approach is selected to evaluate the results of the predicted masks versus the ground truth masks as the mean IoU metric has been used across the researched papers and will therefore make the comparison of results possible. The implementation of the mean IoU calculation is shown in Figure 3.12 to enable replication of the

results.

```
from keras import backend as K
def iou_coef(y_true, y_pred, smooth=1):
  intersection = K.sum(K.abs(y_true * y_pred), axis=1)
  union = K.sum(y_true,1)+K.sum(y_pred,1)-intersection
  iou = K.mean((intersection + smooth) / (union + smooth), axis=1)
  return iou
```

Figure 3.12: Mean IoU metric code implementation [Tiu [2019]]

3.6.3.2 Image Categories

Further to the mean IoU Scores, a sample of images that are the best representation of the test set are selected from the 200 test set images to analyse the performance of the networks on types of images, such as well-illuminated images versus poorly lit/ partially lit images, images containing single railway lines versus double railway lines and rocky terrain versus vegetation.

The graph below (Figure 3.13) illustrates the distribution of image types in the test dataset. The mean IoU score for each category of images will be used to evaluate the network architectures in addition to the overall score.

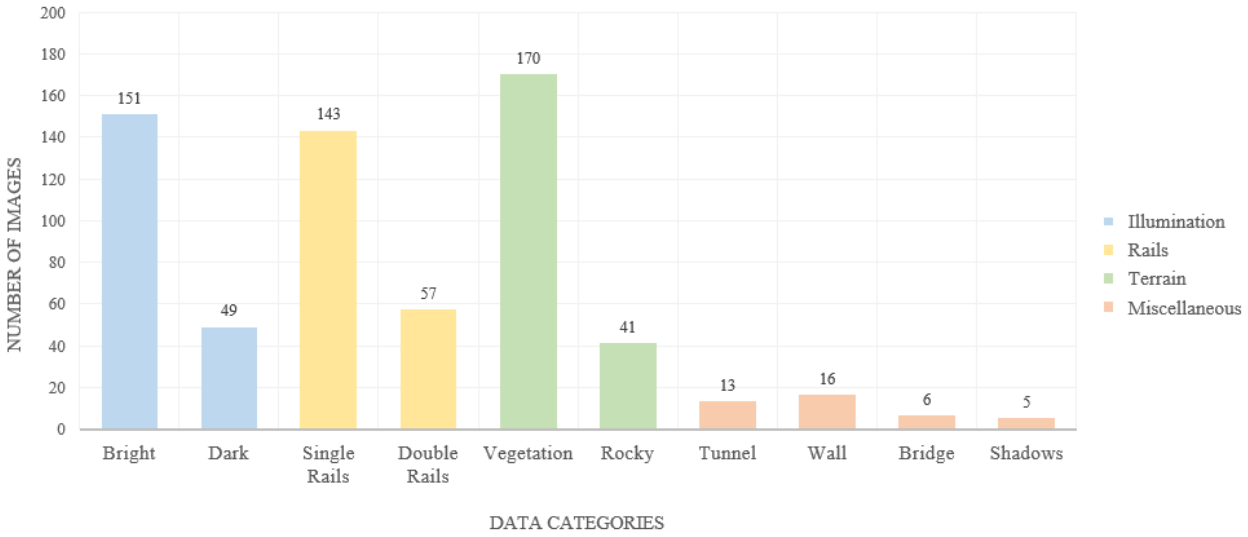


Figure 3.13: Analysis of the test set

3.6.3.3 Reduced Test Set

For the purpose of visualising the results in this report, an image from each category was selected to represent the results of the test dataset as seen in Figure 3.14.

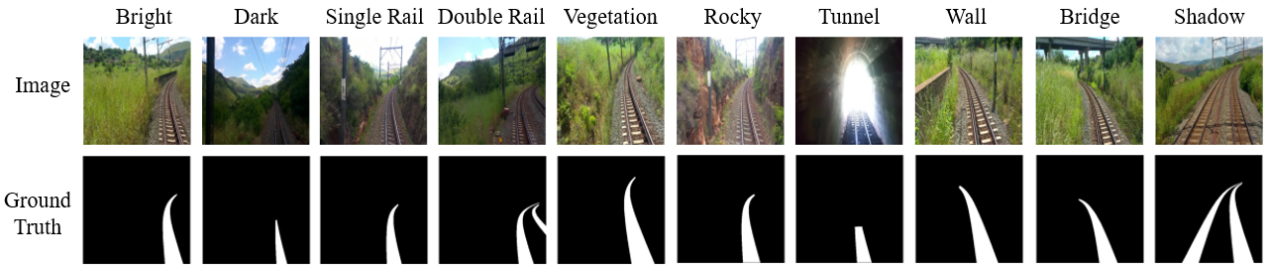


Figure 3.14: Test set representation images with ground truth

3.7 Conclusion

This chapter presented the research methodology that was used to achieve the research objectives. This included collecting and modifying the two datasets, the implementation detail of networks RailNet, SegNet and UNet, the experimental setup and evaluation methods that will be used to determine the success factor of each of the networks.

The next chapter presents the results obtained from the experiments.

Chapter 4

Experimental Results

4.1 Introduction

The previous chapter described the methodology implemented to achieve each of the research objectives outlined in this report. This entailed the process of collecting and labelling the South African dataset, modifying the RailSem19 dataset, the implementation detail of each of the network architectures (SegNet, UNet and RailNet) and lastly outlining the experimental setup that will be used to compare the performance of each of the networks and the effect on performance that arises from supplementing the collected dataset with foreign data.

This chapter will present the results from each of the network architectures when tested on each of the datasets, as well as compare the best performing network from each network architecture. In addition, this chapter will provide an analysis of the results to create information from the data presented.

4.2 SegNet

Figure 4.1 depicts the selected images from the test set, representing each category of images, together with the ground truth label and the predicted segmentation masks as a result of training the SegNet network architecture on each of the datasets. Based on visual appearance of the predicted masks, the SegNet400 network appears to produce results that most closely align to the ground truth sample.

A clear visual improvement in the predictions can be seen between the baseline (SegNet000) and SegNet100 on image samples for 'bright', 'dark', 'double rail', 'vegetation' and 'rocky'. Regarding the prediction for the 'shadow' image sample, although it improves towards the horizon line, a small amount of misclassified railway pixels are introduced on the left railway which only resolve on SegNet400.

Regarding the SegNet600 network, it can be seen that the prediction made on the double rail sample, erroneously excludes the right railway. This may be as a result of the additional 100 samples from the RailSem19 dataset (over and above the 500 used for SegNet500), now overpowering the samples of double railway images, that depict the second railway originating from the right-side of the video frame.

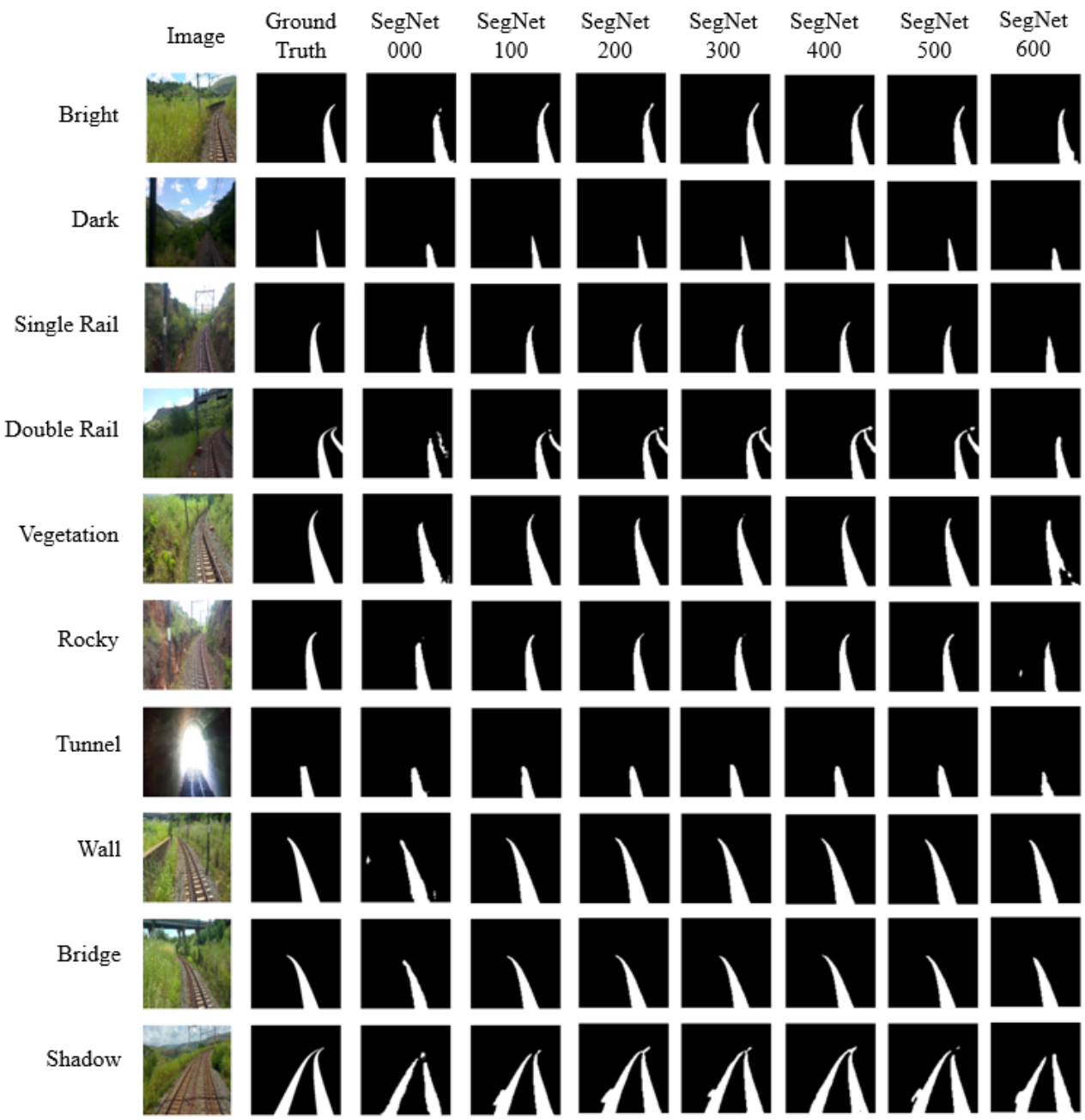


Figure 4.1: Semantic segmentation masks for SegNet on each dataset

Figure 4.2 illustrates the results from additional samples in the test set supporting the argument that this network does poorly only on this type of double rail image. The images on the left of the figure (Image 88, 92, 98, 121 and 123), in which the second railway begins on the right-side of the frame, all present a prediction with the second rail missing, whereas on the images to the right of the figure (Image 194 - 198), SegNet600 correctly predicts both rails.

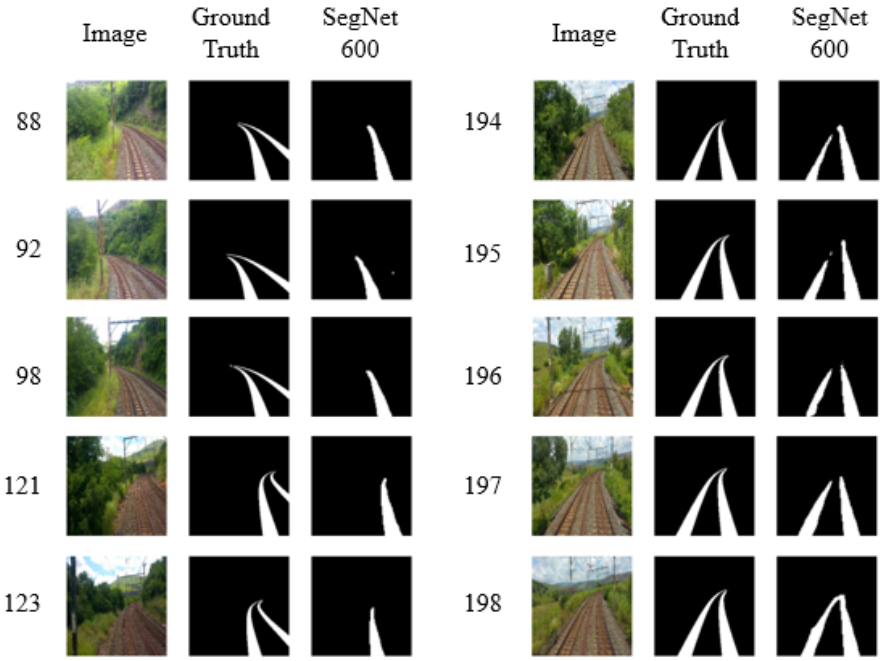


Figure 4.2: Semantic segmentation masks for SegNet600 on double rail images

The overall mean IoU Scores in Table 4.1 determine that the SegNet network performed best when trained on Dataset 200, achieving an overall IoU score of 96.26%, a 6.57% increase in accuracy over the baseline model. This is however, in contradiction to the observation that would select SegNet400 as the best performing network. This opinion is supported by research conducted by Csurka et al. [2013] that suggests the mean IoU metric does not always correspond to human qualitative judgements of good quality segmentation and that the mean IoU favours region smoothness over boundary accuracy. The result on the shadow category for SegNet 200, reaffirms that the boundary accuracy is not weighted in the evaluation. Future research, should consider combining evaluation metrics such that boundary accuracy is weighted in order to get closer to human accuracy.

SegNet 000	SegNet 100	SegNet 200	SegNet 300	SegNet 400	SegNet 500	SegNet 600
90.32	96.22	96.26	96.10	96.10	96.17	91.75

Table 4.1: Table of mean IoU scores for SegNet on each dataset

The high mean IoU scores are based on the prediction of the background class included in the calculation and occupying a larger pixel area in comparison to the railway class. To better visualise the mean IoU scores for the railway class, Figure 4.3, plots the mean IoU score for each image in the test set for the background and railway class, as well as the overall score, for each dataset that the SegNet network is trained on. These

graphs show how the foreground prediction has more impact on decreasing the overall mean IoU score, with the background mean IoU score maintaining some predictability with the exception of SegNet600. In addition, the graphs provide insight into the test dataset, illustrating an increase in the difficulty level of predicting the railway in images 50 till 135, which can be seen in each of the trained networks for the SegNet architecture.

Finally, Table 4.2 depicts the mean IoU scores for the SegNet architecture trained on each dataset for each category type.

		SegNet						
Category	No. of Images	000	100	200	300	400	500	600
Bright	151	90.23	96.34	96.38	96.19	96.19	96.26	91.70
Dark	49	90.64	95.85	95.89	95.81	95.81	95.90	91.90
Single	143	93.93	97.36	97.32	97.24	97.24	97.25	95.37
Double	57	81.27	93.36	93.60	93.23	93.23	93.45	82.65
Vegetation	170	89.40	95.87	95.93	95.73	95.73	95.81	91.05
Rocky	41	93.69	97.29	97.22	97.03	97.03	97.12	94.25
Misc.	34	92.54	96.87	97.03	96.81	96.81	96.89	93.26

Table 4.2: Table of mean IoU scores for each category of images trained on SegNet

The results in Table 4.2 shows a significant increase of 15.2% from the baseline model, SegNet000 to SegNet200 in the double-railway image category. With SegNet100 achieving the best performance for classifying railways in images that contain a single railway and/or a rocky terrain.

With the overall mean IoU for SegNet200 equating to 96.26%, this table provides insight into the three weakest categories, scoring below the average, i.e. dark, double railway images and vegetation.

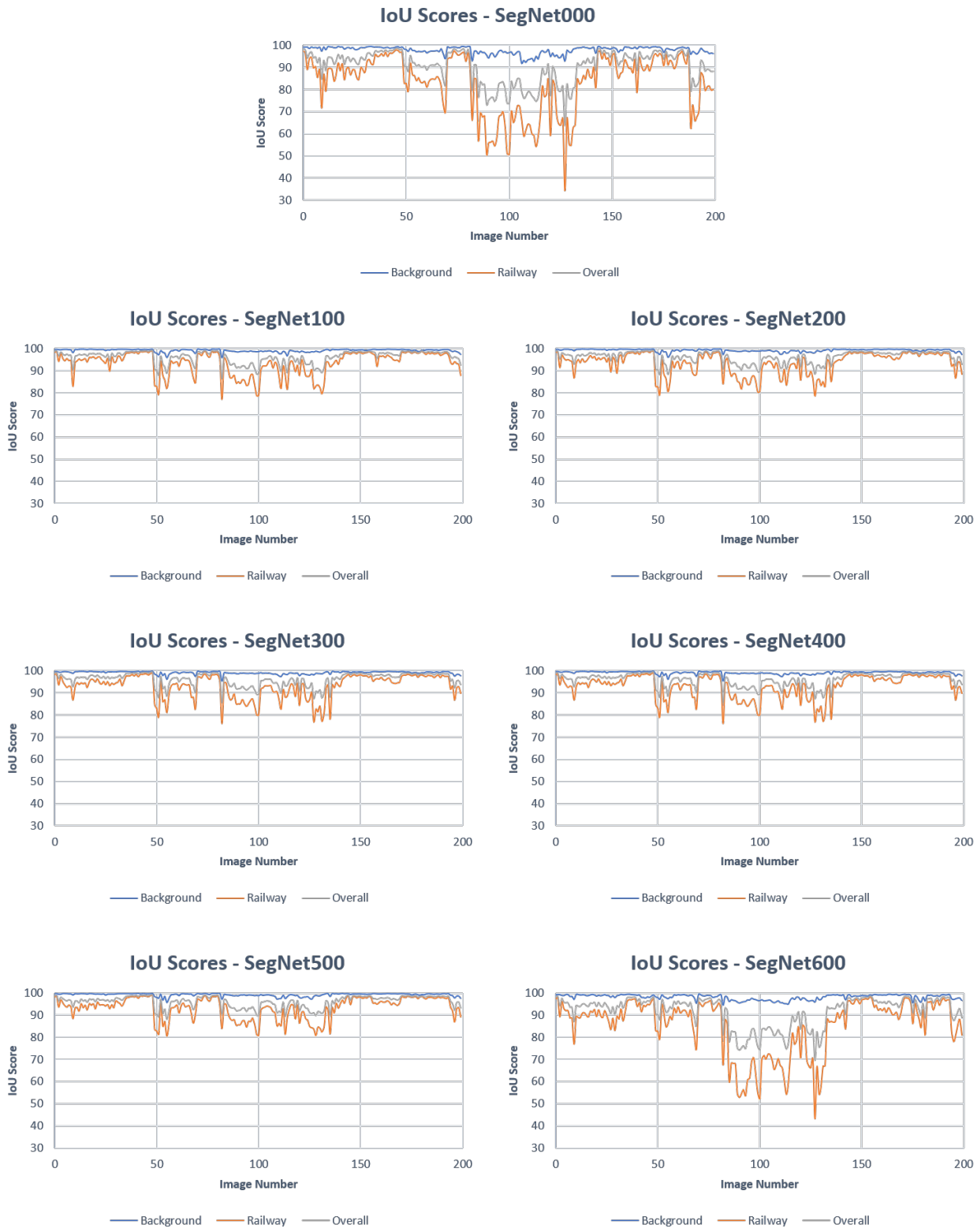


Figure 4.3: Visualisation of mean IoU Scores for the test set, for each dataset tested with SegNet

4.3 UNet

Figure 4.4 depicts the selected images from the test set, representing each category of images, together with the ground truth label and the predicted segmentation masks as a result of training the UNet network architecture on each of the datasets.

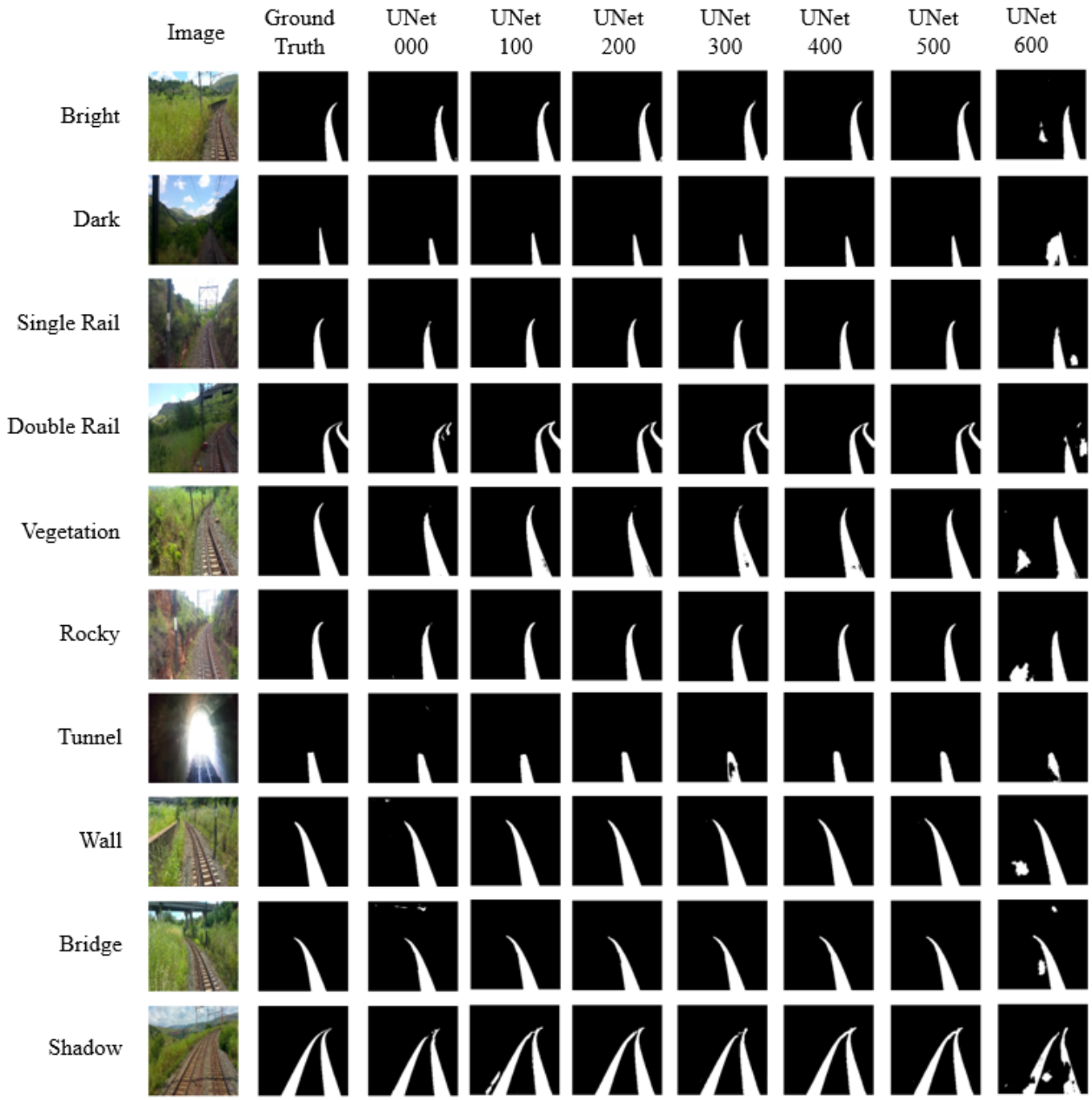


Figure 4.4: Semantic segmentation masks for UNet on each dataset

Upon visual inspection, UNet 200 appears to have the best overall railway prediction across each of the categories. UNet 600 on the other hand, is unable to make useful predictions for the railway class. The total of 600 RailSem19 images have saturated the collected dataset resulting in a negative impact on the South Africa test set.

Table 4.3 below presents the mean IoU scores for the UNet network, and confirms that UNet trained on Dataset 200, is the highest achieving network with a score of 96.36%. An improvement of 3.7% on the baseline model.

UNet 000	UNet 100	UNet 200	UNet 300	UNet 400	UNet 500	UNet 600
92.93	96.17	96.36	96.30	96.29	96.29	87.85

Table 4.3: Table of mean IoU scores for UNet on each dataset

To better visualise the performance of the UNet network on each dataset for the complete test set of 200 images, Figure 4.6 plots the mean IoU Scores for each image for the background and railway class as well as the overall mean IoU score. As with the SegNet network, there is a general increase in the difficulty of predicting the railway class between images 50 and 150, as well as the foreground mean IoU score providing a better indication of the accuracy of the prediction.

Homing in on UNet200, a spike of the mean IoU score for the railway class on Image 127 is observed. To investigate further, Figure 4.5 is depicted below, illustrating Image 127 and the predictions made by Unet on each of the datasets. It can be seen that the rails placed on either side of the railway, with the purpose of supporting the rail formation, has been misclassified by the network as a railway.

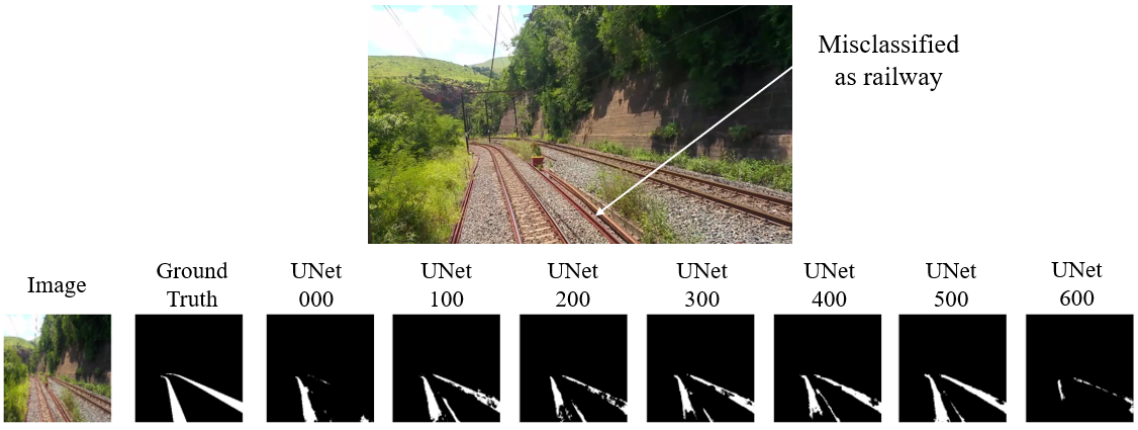


Figure 4.5: Test set: Image 127 and UNet predictions

It is noted, that the UNet network was unable to make a clear distinction between the railway and the supporting rails on all datasets for Image 127 and if the UNet network were to be implemented in the industry, this would need to be addressed.

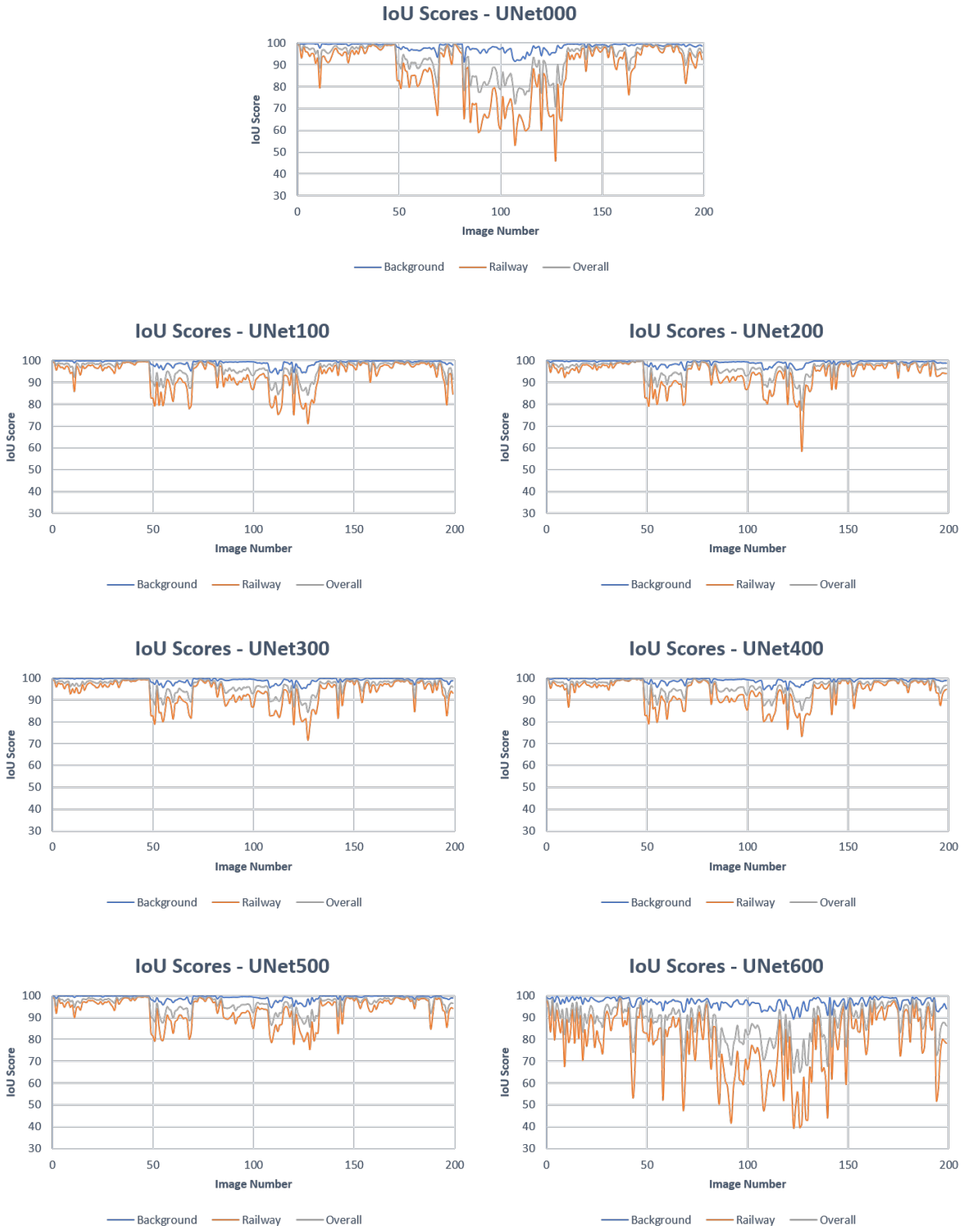


Figure 4.6: Visualisation of mean IoU scores for the test set, for each dataset tested with UNet

Finally, Table 4.4, depicts the mean IoU scores for the UNet architecture trained on each dataset for each category type.

Category	No. of Images	UNet 000	100	200	300	400	500	600
Bright	151	93.49	96.60	96.80	96.70	96.67	96.71	87.79
Dark	49	91.21	94.86	95.00	95.06	95.11	94.99	88.04
Single	143	96.27	97.51	97.37	97.37	97.45	97.38	91.14
Double	57	84.57	92.82	93.82	93.62	93.37	93.55	79.58
Vegetation	170	92.03	95.74	96.02	95.97	95.96	95.93	87.17
Rocky	41	96.98	98.03	97.97	97.90	97.75	97.84	89.33
Misc.	34	94.48	96.98	96.90	96.82	96.87	96.85	88.84

Table 4.4: Table of mean IoU scores for each category of images trained on UNet

With the exception of the double rail class for the baseline model, UNet achieves relatively high IoU scores in all categories and improves when trained on Dataset 100 and Dataset 200. As with SegNet200, the UNet200 network has the weakest predictions on the category with double railways in the images. This is possibly due to there being more single rail images in the training set as this would explain the consistency in the results of both networks.

4.4 RailNet

Figure 4.7 depicts the selected images from the test set, representing each category of images, together with the ground truth label and the predicted segmentation masks as a result of training the RailNet network architecture on each of the datasets.

Upon visual inspection of the results on the sample test set, RailNet 500 appears to be the best performing network. It is of interest that RailNet 000 and RailNet 100 are unable to predict the railway in the tunnel category and as a result, the success of this prediction in the latter networks can be solely attributed to the addition of the RailSem19 data.

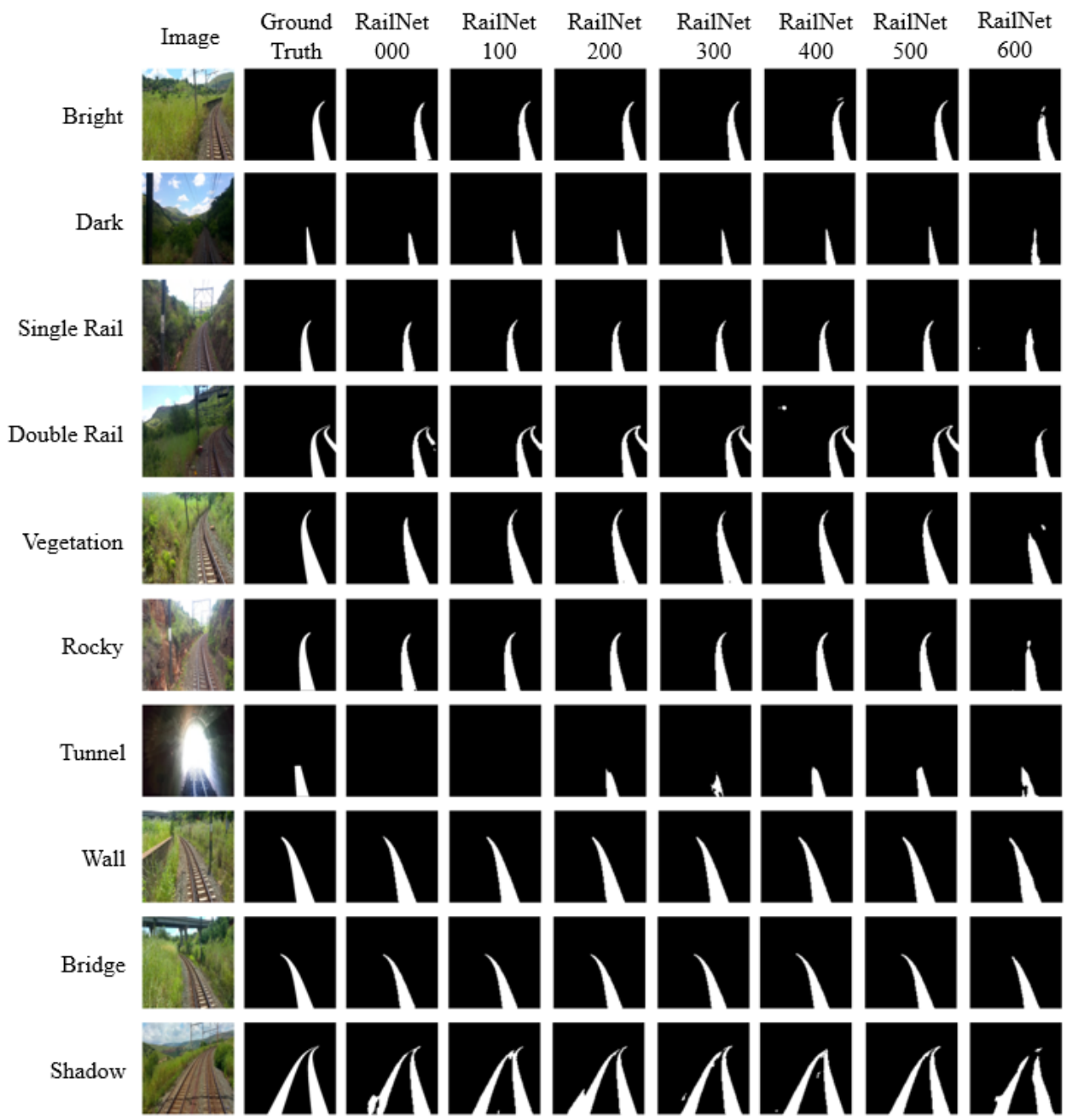


Figure 4.7: Semantic segmentation masks for RailNet on each dataset

It is seen from the shadow category, that the RailNet network was not exempt from the misclassified pixels on the left railway as seen with the SegNet network architecture, but this problem as well is resolved on latter networks, in this case, RailNet 500. It is noted that similarly to the UNet network, the predictions of RailNet trained on Dataset 600 have deteriorated in accuracy and this is validated in Table 4.5 below. RailNet 500 achieves the highest mean IoU score of 96.81%, which is a total increase of 4.1% from the baseline result.

RailNet 000	RailNet 100	RailNet 200	RailNet 300	RailNet 400	RailNet 500	RailNet 600
92.99	96.43	96.43	96.57	96.62	96.81	90.98

Table 4.5: Table of mean IoU scores for RailNet on each dataset

To better visualise the mean IoU scores for the railway class, Figure 4.9, plots the mean IoU score for each image in the test set for the background and railway class, as well as the overall score, for each dataset that the RailNet network is trained on.

As mentioned previously, the foreground mean IoU score is a truer reflection of the quality of the predictions. The mean IoU scores for the foreground class on RailNet500 indicate a spike dipping below the 80% mark. This image is Image 127, and the same image that the UNet network scored particularly poorly on. Figure 4.8 illustrates the predictions made by the RailNet network for each of the datasets.

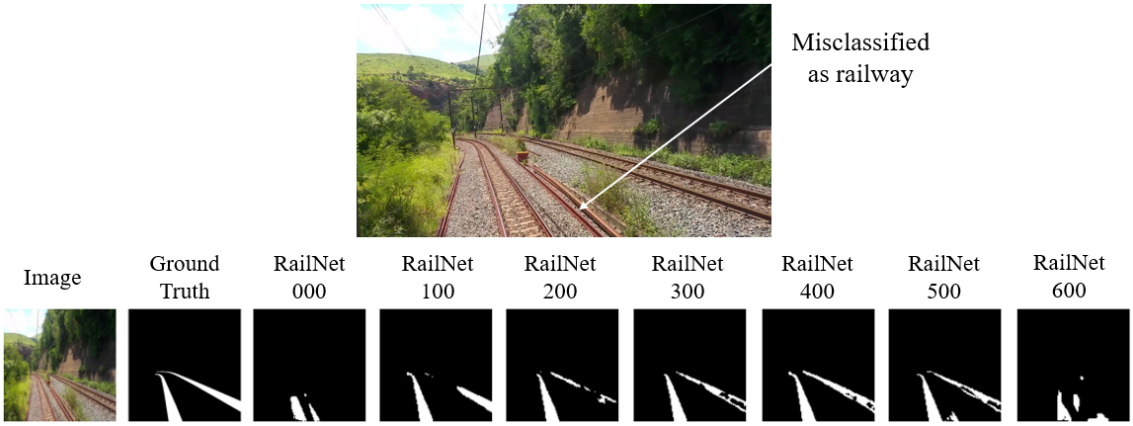


Figure 4.8: Test set: Image 127 and RailNet prediction

It can be seen that, for this example, RailNet 400 achieves a more accurate result. Therefore the additional 100 RailSem19 images that are the degree of difference between RaiNet 400 and RailNet 500 should be more carefully selected to avoid the introduction of error on this network.

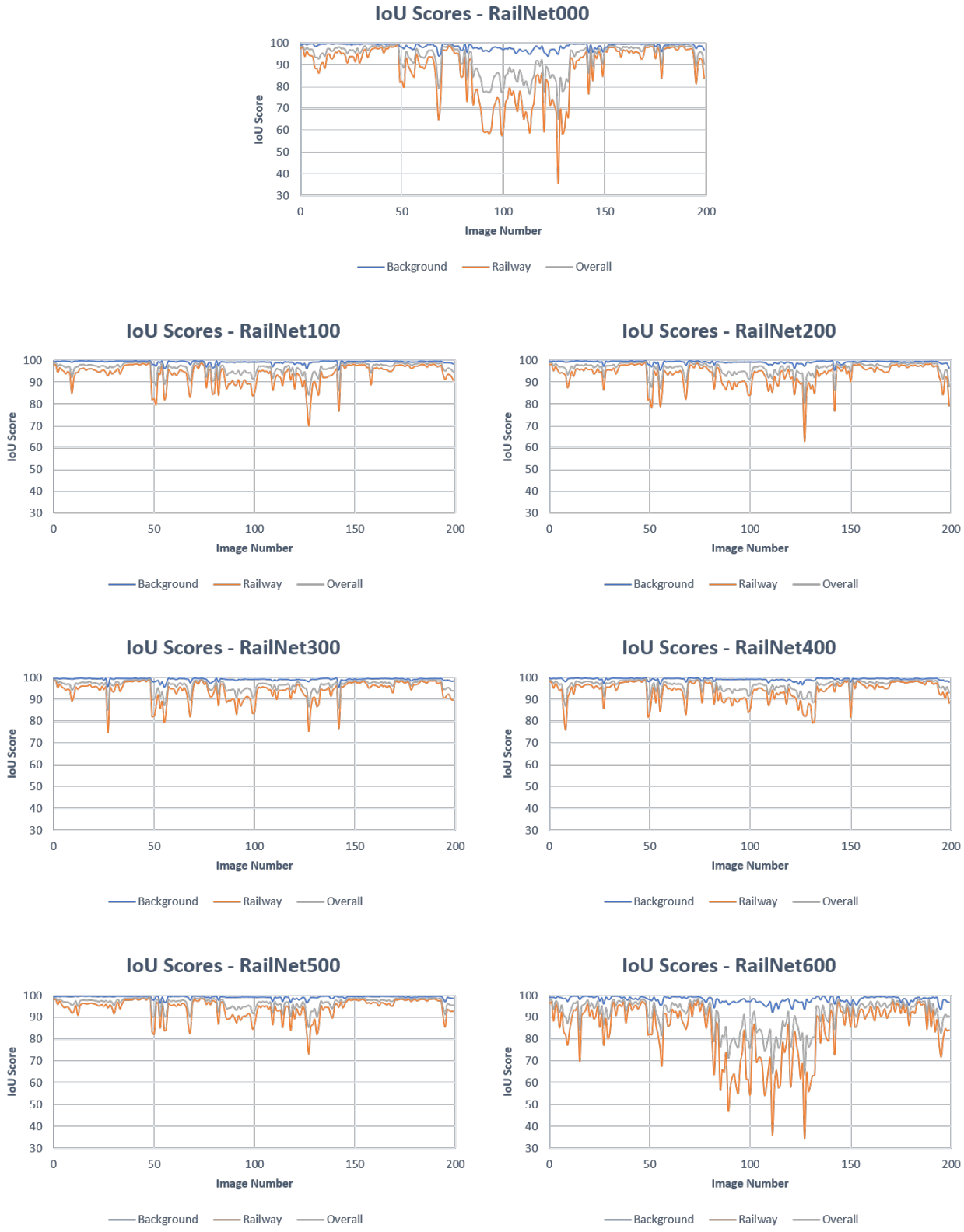


Figure 4.9: Visualisation of mean IoU scores for the test set, for each dataset tested with RailNet

Finally, Table 4.6 depicts the mean IoU scores for the RailNet architecture trained on each dataset for each category type.

Category	No. of Images	RailNet						
		000	100	200	300	400	500	600
Bright	151	93.34	96.72	96.50	96.76	96.65	96.86	90.99
Dark	49	91.92	95.54	96.19	96.00	96.52	96.67	90.93
Single	143	96.25	97.18	97.29	97.15	97.48	97.66	94.49
Double	57	84.81	94.54	94.26	95.11	94.44	94.69	82.17
Vegetation	170	92.49	96.24	96.23	96.47	96.42	96.60	90.40
Rocky	41	95.53	97.50	97.32	97.36	97.17	97.61	93.38
Misc.	34	93.76	95.78	96.18	96.21	96.91	96.88	92.33

Table 4.6: Table of mean IoU scores for each category of images trained on RailNet

From the table, it can be deduced that with the exception of the '$miscellaneous$' class, RailNet 500 has also dominated the mean IoU scores for each of the categories. This is significant and desirable as the RailNet 500 network has successfully been able to extract the railway from images despite the changes in illumination, the number of railways in the image and the terrain of the background. With regards to the miscellaneous class, a score of 96.88% is a close second, only 0.03% away from the highest mean IoU. This indicates the networks ability to generalise and adapt to the South African railway environment.

4.5 Conclusion

This chapter reported the experimental results for each of the networks investigated i.e. SegNet, UNet and RailNet. This entailed a discussion of the reduced test set predictions, the mean IoU scores of each network architecture trained on each dataset as well as the mean IoU scores for each category of each trained network. The next chapter puts forth a combined discussion of the three network architectures.

Chapter 5

Discussion

5.1 Introduction

The previous chapter presented the experimental results for each of the networks trained as well as expressing observations in the results for the individual networks. This chapter combines the results of the best performing network from each architecture to extract patterns and generalisations.

5.2 Image Category Analysis

To get an informed opinion on the performance of the networks for each image category, an image was selected for each category from the test set to visually present the results (Figure 5.1) as well as calculating the mean IoU scores of each network for these categories (Table 5.1). This analysis will focus on the best performing network trained for each of the network architectures i.e. RailNet 500, UNet 200 and SegNet 200.

With reference to Figure 5.1, visually the results appear smoother for the RailNet 500 network. The UNet 200 network achieves the intended outcome of detecting the railway as well, with minimal noticeable error. The SegNet 200 network includes a noticeable amount of misclassified pixels for the 'shadow' class on the left railway.

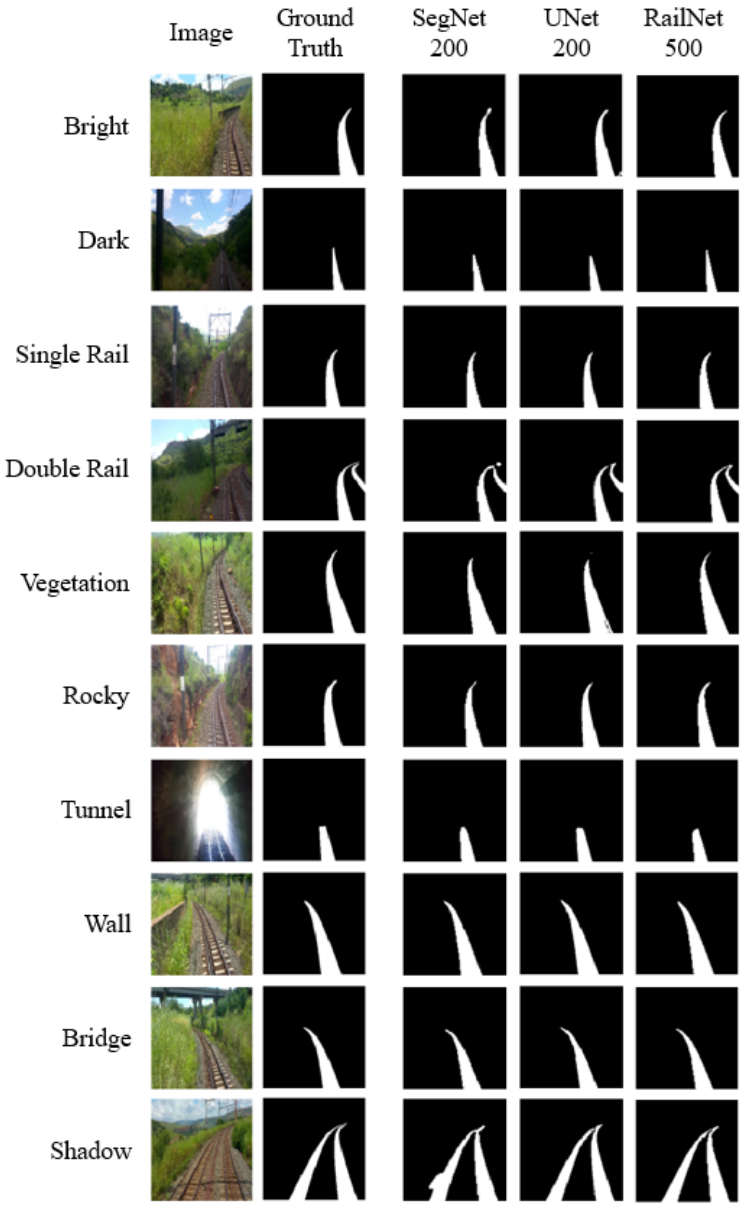


Figure 5.1: Comparison of the best performing network of each network architecture

To compare the network performance of each category, Table 5.1 reports the mean IoU scores for each category, for each network.

Category	No. of Images	Network		
		SegNet 200	UNet 200	RailNet 500
Bright	151	96.38	96.80	96.86
Dark	49	95.89	95.00	96.67
Single	143	97.32	97.37	97.66
Double	57	93.60	93.82	94.69
Vegetation	170	95.93	96.02	96.60
Rocky	41	97.22	97.97	97.61
Misc.	34	97.03	96.90	96.88

Table 5.1: Table of mean IoU scores for each category of images trained on SegNet 200, Unet 200 and RailNet 500

It can be seen that RailNet 500 dominates the first 5 categories as expected but unexpected is that UNet 200 achieves a highest result in the '$rocky$' category, with SegNet 200 also scoring highest in the '$miscellaneous$' category.

This brings to light, the need to analyse the environment of the intended deployment of a railway detection system, as it is possible that some architectures generalise better to certain environment types.

5.3 Overall Mean IoU Score

The mean IoU Score for each network trained in this research was used to determine the success factor of the network. These scores are recorded in Table 5.2.

Network	Dataset						
	000	100	200	300	400	500	600
SegNet	90.32	96.22	96.26	96.10	96.10	96.17	91.75
UNet	92.93	96.17	96.36	96.30	96.29	96.29	87.85
RailNet	92.99	96.43	96.43	96.57	96.62	96.81	90.98

Table 5.2: Table of mean IoU scores for RailNet, UNet and SegNet on each dataset

From the table, it is can be seen that the best performing network architecture based on the mean IoU score is the RailNet network, scoring 96.81% when trained on Dataset 500, with UNet in second position, scoring 96.36%, and SegNet in third, scoring 96.26%, with the latter two networks trained on Dataset 200.

For the SegNet network architecture, each of the mean IoU scores that result from supplementing the dataset are an improvement on the baseline performance (90.32%). Therefore for this network, when faced with a choice of supplementing the dataset, the additional data is recommended. The amount of improvement, is however relative to the amount of additional foreign data that is added and this research has determined that one third the amount of local data produced the optimum result for SegNet.

Excluding the results achieved from training on Dataset 600, the effect of supplementing the dataset with foreign data for the UNet and RailNet network architectures, is a positive impact on the overall performance of the network, with each of the networks improving on the mean IoU score when compared to the baseline performance seen on Dataset 000.

To visualise the relationship between the addition of foreign data and the performance, each of the mean IoU scores on the test set are plotted for each of the datasets, for each network architecture. This is illustrated in Figure 5.2.

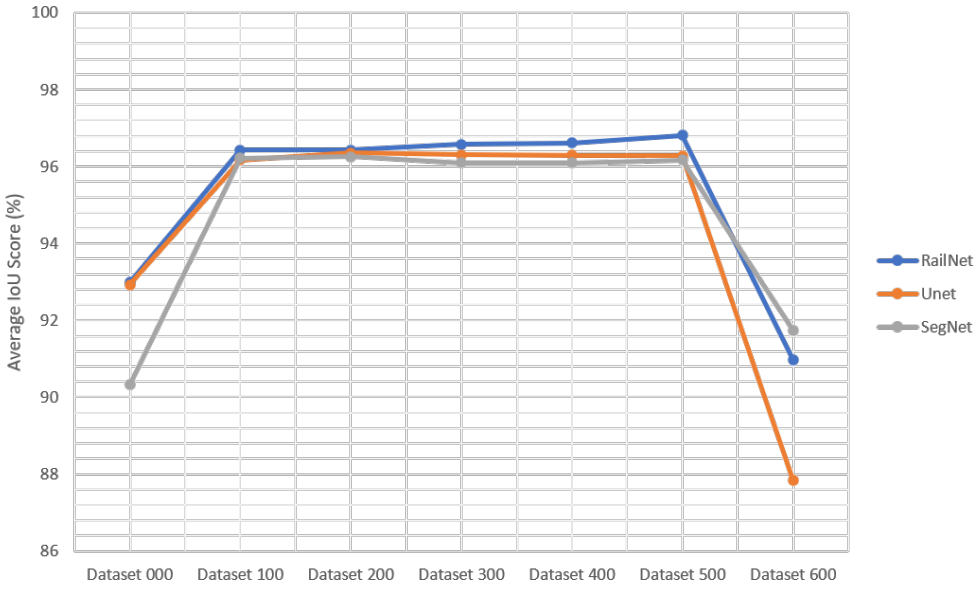


Figure 5.2: Comparison of RailNet, UNet and SegNet on each dataset

From Figure 5.2, it can be seen that the general shape for all 3 networks is an inverted ”U”. This correctly suggests a peak after which the the addition of more foreign data has a negative result on the peak performance. To see this more clearly, Figure 5.3 depicts a localised version of the graph, homing in on the mean IoU Scores between 96% to 97%.

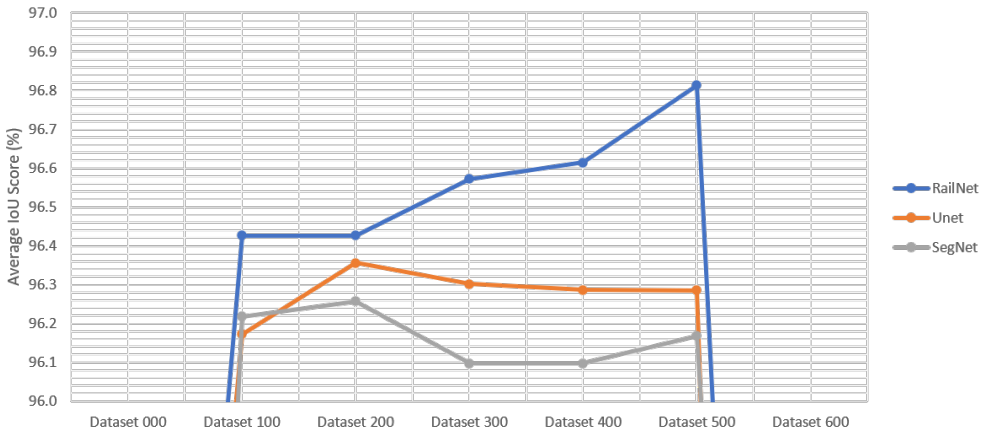


Figure 5.3: A focused comparison of RailNet, UNet and SegNet on each dataset

The RailNet network architecture (presented in blue) has an overall upward trajectory, peaking at Dataset 500, which is seen as the turning point for its decline. The UNet architecture (presented in orange) peaks early at Dataset 200, with a relatively flat slope until Dataset 500, where it too, declines rapidly. Finally SegNet (presented in grey), peaks at Dataset 200, dips between Dataset 300 and Dataset 400, recovers on Dataset 500 but meets the same demise in performance on Dataset 600, although unlike the other two architectures, it doesn't dip below the baseline result.

As the best performing networks from each architecture range from 96.26% to 96.81% according to the mean IoU score, the variability of performance is contained within 1%. Based on the complexity of the networks implemented and the overall training time, the UNet architecture is a favourable choice provided the fine grain accuracy is not required.

Of course, in a high-risk implementation such as autonomous trains, every 0.01% is critical to the safety of the system and the RailNet network architecture would be the preferable option to conduct further research on as this is also not a fail-safe solution currently.

5.4 Comparison to the Literature

Table 5.3 presents the mean IoU scores extracted from the literature review to represent the area in the image equivalent to the 'railway' as defined in this research as well as mean IoU scores for this research.

It can be seen that the networks in this research have achieved significantly high mean IoU scores in comparison to the literature and the reasons for this are as follows:

- RailSem19 was trained to distinguish 19 classes in the dataset, and therefore each pixel is classified into 1 of 19 classes, which heavily decreases the chances of predicting a correct pixel i.e. 5.2% chance of a

Network	mean IoU Score (%)
Literature	
RailSem19	79.7
Railway Intrusion Detection System	85.23
RailNet	89.8
Research Implementation	
SegNet 200	96.26
UNet 200	96.36
RailNet 500	96.81

Table 5.3: Table of mean IoU scores for the literature and research implementation

correct prediction versus a 50% chance with a binary prediction

- The Railway Intrusion Detection System is optimised for low computational burden and therefore makes use of hand-crafted feature extraction and a simplified Convolutional Neural Network
- RailNet utilises a larger dataset of 3000 images, it is therefore possible that there is more variability in the dataset that does not exist in the collected dataset, which can induce difficulty in the network predictions. As this dataset is not publicly available, this cannot be investigated further and remains speculation.
- Finally, for this research, the training, validation and test set data come from the same distribution of images, as the images are limited to those captured from the video footage. Although every effort was made to randomise the validation and test set, from a selection of 1400 frames (200 images were selected as the validation set and 200 images selected for the test set), it is not impossible that similar images exist between the two sets as the data is non-IID (Independent and Identically Distributed).

5.5 Research Objectives

Section 1.3 outlined the research objectives that this research aimed to achieve. This section discusses the extent to which the objectives were achieved.

- **Generate a Labelled Dataset**
  This objective was achieved, contributing 1000 hand-labelled, high quality South African railway images and segmentation masks which is intended to be publicly available provided the relevant permissions are granted.
- **Implementation of the current best performing Segmentation Model for Railways**
  RailNet is the current best performing model for segmenting railways, and this model was implemented as accurately as the information available would allow. Until such time, an official version of the network exists, this objective has been achieved within reason.
- **Applying Transfer Learning to implement a few selected Segmentation Models**
  This objective was achieved with the implementation of SegNet and UNet. Due to time constraints, only

two (2) models were implemented. Transfer learning was utilised to reduce the training time drastically and these networks were capable of producing competing results to that of RailNet. In an industry where real-time predictions are as critical as accuracy, less complex networks have the ability to make quicker predictions and this research has provided both options.

As part of future work in the field, it would be of interest to know how the HRNet-OCR network fairs against RailNet.

- **Supplementing the South African Railway images with a dataset from foreign countries**
  This objective was achieved extensively by creating 7 datasets with varying amounts of foreign data and analysing the performance of each of the network architectures trained on each of the datasets.

5.6 Limitations

Time was a critical resource when conducting this research and to ensure the completion of the research within the given time-frame, only two (2) additional models could be implemented and trained to compare to the performance of RailNet.

Time was additionally a limiting factor with the number of semantic segmentation labels that could be created for the dataset, as labelling 1000 images ranged between 70 - 250 hours and had to be completed prior to the training of the networks.

Furthermore, the time to train the networks played a significant role in the optimisation of hyper-parameters. Whereas the UNet and SegNet networks, completed training within approximately 1-3 hours, the RailNet network took between 6-8 hours. Considering a total of 21 networks were trained for this research, this did not allow much room for experimenting with hyper-parameters, as results could not be determined immediately and any change in the baseline model would then require re-training on each of the 6 datasets.

This research was limited by the available hardware resources required for training the networks. Although, GPU capabilities from Google Colab were used for the entirety of this research, the amount of training that could be achieved in a day was subjected to the free usage policy. It was necessary to keep the GPU usage times under 12 hours per day to avoid reaching maximum usage, as reaching the limit would result in your access to the GPU being suspended for a period of 12-24 hours. Furthermore, the process of training the networks needed to be monitored, as a 'pop-up' box would appear requesting the user to respond within a few minutes, in which case, a non response disconnects your access and forfeits the training progress made.

This research was also limited by data, in that all samples were extracted from the video footage and this was likely to affect the results. As part of future work on this research, it would be beneficial to test this network on completely independent samples of railways captured from different parts of South Africa.

Finally, the scope of this research does not extend to a complete rail detection solution but rather serves to explore the viability of using deep learning models to detect railways from images.

5.6.1 Conclusion

This chapter discussed each of the top-performing networks in terms of the performance with regards to each category of images presented in the test set as well as a comparison of the effect of the supplemented data on the trained networks and the overall mean IoU score. This chapter then went on to compare the results achieved in this research to that of the literature, discuss the extent to which the research aims were achieved and finally the limitations of this research.

The final chapter will conclude this research by presenting the main results, providing answers to the research questions, presenting insights and future work.

Chapter 6

Conclusion

This research report investigates the application of semantic segmentation on South African railway images to achieve railway detection. Part of the contribution of this research is a labelled dataset of 1000 South African railway images. The collected dataset has been trained on SegNet and Unet architectures as well as a recent, rail-specific architecture RailNet, scoring a mean IoU value of 90.32%, 92.93% and 92.99% respectively.

The RailNet implementation confirms that a semantic segmentation model trained on railway data from another country can exceed the performance when trained on South African railway images. The RailNet implementation has also proven to have the ability to adapt to the South African domain which is much different to the railway environment in China, from where this architecture originates.

A publicly available, labelled dataset, RailSem19 was modified to align with the labels of the collected dataset and used to determine the effect of supplementing the training dataset with foreign data. It was determined that this had a positive effect on the performance when added in the correct quantity, increasing the baseline mean IoU values of SegNet, UNet and RailNet to 96.26%, 96.36% and 96.81% respectively. The correct amount of images is however specific to each network architecture and was determined to be 200 images for SegNet and UNet, while 500 images contributed to the highest performance for the RailNet architecture. The results also provided some insight into the mean IoU metric favouring region smoothness over boundary accuracy which is in contradiction to human preference. To accommodate this hindsight, future work in the field should consider combining evaluation metrics to add a weighting towards the boundary accuracy component in order to align the evaluation metric to human perception.

Analysing the mean IoU score per category for each of the best performing networks, concluded that SegNet 200 achieved the highest IoU score for the '$miscellaneous$' category, the UNet 200 network achieved the highest IoU score for the '$rocky$' category, with RailNet dominating the remaining 5 categories: bright, dark, single, double and vegetation. This result suggests the possibility of some network architectures generalising better to certain environment types and this should be considered when selecting a network architecture.

The main drawback of this research is that the collected dataset was restricted to the video footage taken in the region of one location in South Africa. Had the collected dataset allowed for large variability and representation of railway images from all 9 provinces, the results presented in this research would be significantly strengthened.

Future work to extend this research would involve collecting and labelling the aforementioned dataset that includes variability and representation for varying environments in South Africa and testing the currently im-

plemented networks on this dataset. Another aspect of future work will be implementing the current best performing network on the road class of the Cityscapes dataset, HRNet-OCR, and training this network on the collected dataset and supplemented datasets. Lastly, additional future work would encompass pre-training each of the networks on the RailSem19 dataset and fine tuning the networks on the South African dataset to test for domain adaptability of each of the networks.

In conclusion, this research is presented as a contribution of the first labelled dataset for South African Railway images, and it is hoped that the database of images and research in the field continue to grow to keep the future of autonomous trains on track.

Bibliography

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

Brostow, G. J., Shotton, J., Fauqueur, J., and Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223.

Csurka, G., Larlus, D., Perronnin, F., and Meylan, F. (2013). What is a good evaluation measure for semantic segmentation?. In *BMVC*, volume 27, pages 10–5244.

Gupta, D. (2021). Image-Segmentation-Keras. `https://github.com/divamgupta/image-segmentation-keras`.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516.

Krishnamurthy, K. (2019). NPTEL-NOC IITM: Semantic Segmentation. `https://www.youtube.com/watch?v=_N7HRnBgoCw&t=322s`.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Object recognition with gradient-based learning. *Proceedings of the IEEE*, 86(11):2278–2324.

LeCun, Y. et al. (2015). Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Tao, A., Sapra, K., and Catanzaro, B. (2020). Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*.

Tiu, E. (2019). Metrics to Evaluate your Semantic Segmentation Model. `https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2`.

Tomar, N. (2021). Deep-Residual-Unet. https://github.com/nikhilroxtomar/Deep-Residual-Unet.

Wada, K. (2016). labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme.

Wang, Y., Wang, L., Hu, Y. H., and Qiu, J. (2019a). Railnet: A segmentation network for railroad detection. *IEEE Access*, 7:143772–143779.

Wang, Y., Zhu, L., Yu, Z., and Guo, B. (2019b). An adaptive track segmentation algorithm for a railway intrusion detection system. *Sensors*, 19(11):2594.

Wu, Z., Shen, C., and Van Den Hengel, A. (2019). Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133.

Zendel, O., Murschitz, M., Zeilinger, M., Steininger, D., Abbasi, S., and Beleznai, C. (2019). Railsem19: A dataset for semantic rail scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0.

Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., and Ding, E. (2019). Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6798–6807.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890.

Zhu, Y., Sapra, K., Reda, F. A., Shih, K. J., Newsam, S., Tao, A., and Catanzaro, B. (2019). Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8865.