UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Classification of Off-road Terrains

Author: Petrus Lafras Fritz

U15082645

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering

(Mechanical Engineering)

In the Faculty of

Engineering, Built Environment and Information Technology (EBIT)

at the

The University of Pretoria,

Pretoria

April 2022

Summary

Title:	Off-road terrain classification
Author:	Petrus Lafras Fritz
Supervisor:	Dr H. Hamersma
Co-supervisor:	Dr T. Botha
Department:	Mechanical and Aeronautical Engineering, University of Pretoria
Degree:	Master of Engineering (Mechanical Engineering)

In recent years, a significant amount of research has been done on the development of autonomous vehicles. An important part of autonomous vehicles is their capability to traverse the terrain they are driving over while maintaining safety and ride comfort. To adjust the suspension settings of a vehicle, the vehicle control system requires accurate input data of the terrain it is/will be driving over. In this dissertation, research was done to determine whether off-road classification is possible before a vehicle drives over a specific part of a road. The results showed that not only is off-road classification possible, but also that it can be achieved with great accuracy.

The research started with an in-depth study to determine the different methods for classifying off-road terrain. Different types of vision-based sensor were investigated, which included cameras, radar and lidar. This was followed by research to establish how current state-of-the-art solutions use these sensors. The goal was to determine which sensors are mainly used and which classification method sensors are coupled with to achieve the best possible results. The research study concluded that in this study, image data would be captured with a camera sensor coupled with a convolutional neural network (CNN) classifier using a supervised learning model. Since the literature study revealed that none of the state-of-the-art solutions used some form of standard to distinguish between different classes, it was decided to use the ISO 8608 (2016) standard to determine the different off-road terrain classes.

Chapter 3 describes the development of the classification models. Further research was conducted to determine the colour spectrum for capturing the data and how to build a CNN effectively. It was decided to capture the data in the red, green and blue colour spectrum because it has high accuracy results and is easy to use. Two different classification models were built: the first model was built from scratch and the second model was a pretrained model. The goal of building two models was first to compare the results to establish if a certain method yielded better results. Second, to ensure that the results reflected the classification capabilities of the method predicting the off-road terrain and that the method did not reflect the results of a poorly built model. A binary classifier was built to classify between specific classes.

After the models were built, the next step was capturing the data and determining the classes. The tests were performed at the Gerotek Test Facilities in South Africa. Three different off-road terrain classes were identified using the ISO 8608 (2016) standard based on the material of the profile and the roughness of the road. Two datasets consisting of a total of 15 308 images were captured of which 12 645 images were used during the actual training phase. Both downward- and forward-facing camera setups were used. The downward-facing camera setup pointed directly at the ground, whereas the forward-facing camera setup pointed between 5° and 15° downwards relative to the horizon. The forward-facing dataset contained unwanted noise. Noise was defined as obstacles that did not form part of the terrain profile, for example barriers and trees. A model was created that cropped the images and only focused on the actual road profile. The downward-facing dataset did not have this problem.

The respective CNN models achieved the following prediction results:

- Multiple-class classifier on downward-facing data – 99.65%
- Multiple-class classifier on forward-facing data – 99.71%
- Pretrained model on downward-facing data – 100%
- Pretrained model on forward-facing data – 100%

Further research was done to determine what would happen if some of the off-road terrains were omitted during the training phase and reintroduced during the prediction phase. The results were not as good as the initial tests that included all the different terrain classes. If a specific terrain was omitted but that terrain shared similarities with the other off-road terrains in that class, the prediction accuracy results were satisfactorily high. A prediction accuracy of 87.97–100% was achieved when the parallel corrugated track was omitted from class B (ride and handling track). A prediction accuracy of 98.71–100% was achieved when the fatigue track was omitted from class B. When the rough track was omitted from class H (rough track and gravel track), the results were as low as 28.91% due to the rough track having a completely different appearance than the other off-road profiles in class H. This was an indication that the models could potentially not classify based on the roughness of the terrain, but instead used the colour of the actual terrain. The decrease in classification accuracy was expected as the models were trained using a supervised learning model. By omitting the off-road terrain, the model was not able to train and extract the different features from that terrain.

A binary classification task was performed to establish whether it would be possible to classify between parallel and angled corrugated terrains. The results gave a 100% prediction accuracy for both the created and pretrained model for the forward-facing dataset. However, the models only achieved a prediction accuracy of 84% (pretrained model) and 86% (created model) for the downward-facing dataset. The downward-facing dataset camera was considerably closer to the ground and could not capture the spatial frequency of the terrain. The forward-facing camera pointed further away from the terrain and was able to capture the spatial frequency of the terrain.

The dataset captured was limited in the sense that it only consisted of the tracks captured at Gerotek. The dataset was further captured during ideal weather conditions; therefore, it is recommended to increase the quality of the dataset by capturing data points during various different weather conditions. Several recommendations are made to increase the prediction accuracy of the model when presented with new terrains. Additional sensors including a second camera and other sensors can be added to the model, which could allow the model to classify the off-road terrain at different angles and use feedback sensors to confirm the classification made. Other learning models can also be used, including semi-supervised, unsupervised or reinforcement learning models.

The study concluded that off-road terrain classification is possible and can be done with satisfactory accuracy before a vehicle drives over a specific part of a road, but only on trained classes in ideal weather conditions.

Acknowledgements

I would like to thank the following people without whom the successful completion of this dissertation would not have been possible:

- Prof. Schalk Els for his advice and guidance throughout my honours and master's degree.
- Dr Herman Hamersma for his mentorship and guidance throughout my master's degree.
- Dr Theunis Botha for his mentorship and guidance throughout my master's degree.
- My fiancé, Himne, for her unconditional love and support and for always being there for me.
- My parents for their support throughout my studies and for always being there to help and assist.
- The entire Vehicle Dynamics Group at the University of Pretoria for their eagerness to help.

Table of Contents

Summary ..... **ii**
Acknowledgements ..... **iv**
Table of Contents ..... **v**
List of Figures ..... **viii**
List of Tables ..... **xii**
Lists of Symbols and Abbreviations ..... **xiii**
Lowercase Roman symbols xiii
Uppercase Roman symbols xiii
Greek symbols xiv
Abbreviations xiv
Chapter 1: Introduction **1**
1.1 Introduction 1
Chapter 2: Literature survey **3**
2.1 Types of terrain classification 3
2.2 Different types of sensor used for vision-based terrain classification 3
2.2.1 Cameras 3
2.2.2 Radar 4
2.2.3 Lidar 5
2.3 State-of-the-art solutions 6
2.4 Ride comfort 18
2.5 Summary of literature survey 21
Chapter 3: Development of classification models **23**
3.1 Introduction 23
3.2 Data format and classification sensor 23
3.3 Multiple-class classifier 24
3.3.1 Convolutional layer 24
3.3.2 Activation function 26
3.3.3 Pooling layer 28
3.3.4 Flattening 29
3.3.5 Dropout 29
3.3.6 Dense layer 29
3.3.7 Optimisation function 30
3.3.8 Loss function 30
3.3.9 Fitting 31
3.3.10 Backpropagation 31
3.3.11 Model summary 31
3.4 Binary class classifier 33

3.4.1 Model summary 33
3.5 Pretrained convolutional neural network model 34
3.6 Conclusion 35
Chapter 4: Data capturing of off-road terrains and class creation 37
4.1 Introduction 37
4.2 Experimental setup 37
4.3 Off-road terrain classes 39
4.3.1 Sand 40
4.3.2 Gravel 41
4.3.3 Boulder/rock 41
4.3.4 Grass 42
4.3.5 Belgian paving 42
4.3.6 Parallel and angled corrugations 43
4.3.7 Fatigue track 45
4.3.8 Ride and handling track 46
4.3.9 Rough track 47
4.4 Off-road terrain classes based on ISO 8608 international standard 48
4.5 Conclusion 49
Chapter 5: Off-road terrain classification 50
5.1 Introduction 50
5.2 Data processing 50
5.3 Classification setup 51
5.4 Multiple-class classification results 52
5.4.1 Forward-facing results using multiple-class classifier 52
5.4.2 Downward-facing results using a multiple-class classifier 55
5.5 Pretrained model classification results 57
5.5.1 Forward-facing results using the pretrained model 57
5.5.2 Downward-facing results using a pretrained model 60
5.6 Classification results omitting certain off-road terrains 61
5.6.1 Classification without parallel corrugated terrain from the class D road profile 62
5.6.2 Classification without fatigue track from the class D road profile 63
5.6.3 Classification without rough track from the class H road profile 64
5.7 Binary classification results between corrugated terrains 67
5.7.1 Binary classification model 67
5.7.2 Pretrained binary classification model 72
5.8 Summary 75
Chapter 6: Conclusion and recommendations 77
6.1 Conclusion 77
6.2 Recommendations and future work 78
6.2.1 Different road profiles 78
6.2.2 Types of learning 78
6.2.3 Stereo vision 78

6.2.4 Multiple sensors....79
6.3 Limitation....79
References....80
Appendix A: Accuracy and loss value plots for untrained off-road terrains....85

List of Figures

Figure 2.1: A) AGV experimental platform B) Camera and laser setup (Lu, et al., 2009) 7
Figure 2.2: Fourteen different terrain classes used by the AGV (Weiss et al., 2008) 7
Figure 2.3 Stereo camera and single-axis lidar setup (Manduchi, et al., 2005) 10
Figure 2.4: Road classification example (Stolee & Wang, 2018) 12
Figure 2.5: Image noise present when capturing images from inside the vehicle with a camera mounted in front of the rear-view mirror (Omer & Fu, 2010) 13
Figure 2.6: Images displaying the rapid adaption of Stanley's computer vision routines (Thrun, et al., 2006) 14
Figure 2.7: Two stages for classifying images (Roychowdhury, et al., 2019) 15
Figure 3.1: RGB image showing the three dimensions of a colour image (Upadhyay, 2017) 24
Figure 3.2: Kernel sliding over the image convolutional 2D (Verma, 2019) 25
Figure 3.3: Convolution kernel (Baskin, et al., 2017) 26
Figure 3.4: ReLU activation function (Brownlee, 2021) 27
Figure 3.5: Softmax activation graph (Brownlee, 2021) 28
Figure 3.6: Max pooling vs average pooling (Keras Help Documentation, 2020) 28
Figure 3.7: How flattening works in a CNN (Keras Help Documentation, 2020) 29
Figure 3.8: Multiple-class classifier model hierarchy 32
Figure 3.9: Sigmoid activation function (Brownlee, 2021) 33
Figure 3.10: Example of the application of feature extraction (Chollet, 2018) 34
Figure 4.1 Forward-facing camera setup 37
Figure 4.2: Downward-facing camera setup 38
Figure 4.3: Forward-facing camera setup on test vehicle 38
Figure 4.4: Downward-facing camera setup on test vehicle 39
Figure 4.5: Experimental setup showing both forward- and downward-facing camera setups 39
Figure 4.6: Can-Can road profiling machine (Becker, 2008) 40
Figure 4.7: Sand road profile example of downward- and forward-facing data 41
Figure 4.8: Gravel road profile example of downward- and forward-facing data 41
Figure 4.9: Boulder/rock road profile example of downward- and forward-facing data 42
Figure 4.10: Grass road profile example of downward- and forward-facing data 42
Figure 4.11: Belgian road profile example of downward- and forward-facing data 43
Figure 4.12: Can-Can profile of Belgian paving (Becker, 2008) 43
Figure 4.13: Straight corrugated road profile example of downward- and forward-facing data 44
Figure 4.14: Can-Can profile of straight corrugated road (Becker, 2008) 44
Figure 4.15: Angled corrugation road profile of downward- and forward-facing camera data 45
Figure 4.16: Can-Can profile of angled corrugated road with two location points in millimetre (Becker, 2008) 45
Figure 4.17: Fatigue track profile of downward- and forward-facing camera data 46
Figure 4.18: Can-Can profile of the fatigue handling track (Becker, 2008) 46
Figure 4.19: Dynamic handling road profile of downward-facing and forward-facing camera data 47
Figure 4.20: Rough track road profile of downward-facing and forward-facing camera data. 47

Figure 4.21: PSD results obtained from Can-Can data captured at Gerotek (Becker, 2008) 48
Figure 5.1: Example of a cropped image to omit unwanted noise 51
Figure 5.2: Training and validation accuracy results for the forward-facing dataset using a multiple class classifier 53
Figure 5.3: Training and validation loss results for the forward-facing dataset using a multiple class classifier ... 53
Figure 5.4: Training and validation loss results for the forward-facing dataset using a multiple class classifier and removing the outlier at 431 epochs 54
Figure 5.5: Confusion matrix for multiple class classifier prediction on unseen forward-facing road profiles 54
Figure 5.6: Training and validation accuracy results for the downward-facing dataset using a multiple class classifier 55
Figure 5.7: Training and validation loss results for the downward-facing dataset using a multiple class classifier 56
Figure 5.8: Confusion matrix for multiple class classifier prediction on unseen downward-facing road profiles .. 56
Figure 5.9: Training and validation accuracy results for the forward-facing dataset using a pretrained model 58
Figure 5.10: Training and validation loss results for the forward-facing dataset using a pretrained model 58
Figure 5.11: Training and validation loss results for the forward-facing dataset using a pretrained model without the outlier 59
Figure 5.12: Confusion matrix for pretrained model predictions on unseen forward-facing road profiles 59
Figure 5.13: Training and validation accuracy results for the downward-facing dataset using a pretrained model 60
Figure 5.14: Training and validation loss results for the downward-facing dataset using a pretrained model 60
Figure 5.15: Training and validation loss results for the downward-facing dataset using a pretrained model without major outliers 61
Figure 5.16: Confusion matrix for the pretrained model predictions on unseen downward-facing road profiles 61
Figure 5.17: Confusion matrix plot for the forward-facing multiple-class classification results omitting the rough track 65
Figure 5.18: Confusion matrix plot for the forward-facing pretrained model results omitting the rough track ... 65
Figure 5.19: Confusion matrix plot for the downward-facing multiple-class classification results omitting the rough track 66
Figure 5.20: Confusion matrix plot for the downward-facing pretrained model results omitting the rough track 66
Figure 5.21: Forward-facing binary training and validation accuracy results 68
Figure 5.22: Forward-facing binary training and validation loss value results 69
Figure 5.23: Confusion matrix plot for the forward-facing binary classification between the angled and parallel corrugated terrains 69
Figure 5.24: Downward-facing binary training and validation accuracy results 70
Figure 5.25: Downward-facing binary training and validation loss value results 70
Figure 5.26 Confusion matrix plot for the downward-facing binary classification between the angled and parallel corrugated terrains 71
Figure 5.27: Forward-facing binary training and validation accuracy results for the pretrained model 72
Figure 5.28: Forward-facing binary training and validation loss value results for the pretrained model 73
Figure 5.29: Confusion matrix plot for the forward-facing binary classification between the angled and parallel corrugated terrains for the pretrained model 73
Figure 5.30: Downward-facing binary training and validation accuracy results for the pretrained model 74

Figure 5.31: Downward-facing binary training and validation loss value results for pretrained model 74
Figure 5.32: Confusion matrix plot for the downward-facing binary classification between the angled and parallel corrugated terrains for the pretrained model 75
Figure A.1: Training and validation accuracy for the forward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier 85
Figure A.2: Training and validation loss value for the forward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier 85
Figure A.3: Training and validation accuracy for the forward-facing camera setup omitting the parallel corrugated terrain pretrained model 86
Figure A.4: Training and validation loss value for the forward-facing camera setup omitting the parallel corrugated terrain pretrained model 86
Figure A.5: Training and validation accuracy for the downward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier 87
Figure A.6: Training and validation loss value for the downward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier 87
Figure A.7: Training and validation accuracy for the downward-facing camera setup omitting the parallel corrugated terrain pretrained model 88
Figure A.8: Training and validation loss value for the downward-facing camera setup omitting the parallel corrugated terrain pretrained model 88
Figure A.9: Training and validation accuracy for the forward-facing camera setup omitting the fatigue track multiple-class classifier 89
Figure A.10: Training and validation loss value for the forward-facing camera setup omitting the fatigue track multiple-class classifier 89
Figure A.11: Training and validation accuracy for the forward-facing camera setup omitting the fatigue track pretrained model 90
Figure A.12: Training and validation loss value for the forward-facing camera setup omitting the fatigue track pretrained model 90
Figure A.13: Training and validation accuracy for the downward-facing camera setup omitting the fatigue track multiple-class classifier 91
Figure A.14: Training and validation loss value for the downward-facing camera setup omitting the fatigue track multiple-class classifier 91
Figure A.15: Training and validation accuracy for the downward-facing camera setup omitting the fatigue track pretrained model 92
Figure A.16: Training and validation loss value for the downward-facing camera setup omitting the fatigue track pretrained model 92
Figure A.17: Training and validation accuracy for the forward-facing camera setup omitting the rough track multiple-class classifier 93
Figure A.18: Training and validation loss value for the forward-facing camera setup omitting the rough track multiple-class classifier 93
Figure A.19: Training and validation accuracy for the forward-facing camera setup omitting the rough track pretrained model 94
Figure A.20: Training and validation loss value for the forward-facing camera setup omitting the rough track pretrained model 94

Figure A.21: Training and validation accuracy for the downward-facing camera setup omitting the rough track multiple-class classifier 95
Figure A.22: Training and validation loss value for the downward-facing camera setup omitting the rough track multiple-class classifier 95
Figure A.23: Training and validation accuracy for the downward-facing camera setup omitting the rough track pretrained model 96
Figure A.24: Training and validation loss value for the downward-facing camera setup omitting the rough track pretrained model 96

List of Tables

Table 2.1: Number of samples per class for the study performed by Weiss, et al. (2007) 8
Table 2.2: Summary of classification approaches found in the literature 16
Table 3.1: Different pretrained models (Keras Help Documentation, 2020) 35
Table 5.1: Number of images per class and total images vs reduced total images used during training 51
Table 5.2: Multiple class classifier model summary 52
Table 5.3: Pretrained model summary 57
Table 5.4: Accuracy and loss value training results for classification without the parallel corrugated terrain 62
Table 5.5: Prediction accuracy including the omitted terrain during training vs the omitted terrain during training for the parallel corrugated off-road terrain 62
Table 5.6: Prediction accuracy on the untrained parallel corrugated off-road terrain only 63
Table 5.7: Accuracy and loss value training results for classification without the fatigue track 63
Table 5.8: Prediction accuracy including the omitted terrain during training vs omitted terrain during training for the fatigue track 64
Table 5.9: Prediction accuracy on the untrained fatigue track only 64
Table 5.10: Accuracy and loss value training results for classification without rough track 64
Table 5.11: Prediction accuracy including the omitted terrain during training vs omitted terrain during training for the rough track 64
Table 5.12: Prediction accuracy on the untrained rough track only 67
Table 5.13: Model summary for the binary classification between the angled and corrugated terrain 67
Table 5.14: Pretrained model summary for the binary classification 72
Table 5.15: Summary of the training, validation and prediction results for the forward- and downward-facing datasets using both the multiple-class classifier and pretrained classifier 75

Lists of Symbols and Abbreviations

Lowercase Roman symbols

Symbol	Description	Unit
e	Mathematical constant	[dimensionless]
i	Vector number	[dimensionless]
k	Vector number counter	[dimensionless]
m	Number of columns	[dimensionless]
n	Number of rows	[dimensionless]
n	Wavelength	[cycles/metre]
p	Probability	[dimensionless]
$p(w_i\|\boldsymbol{x})$	Probability of w_i given x	[dimensionless]
$p(w_j\|\boldsymbol{x})$	Probability of w_j given x	[dimensionless]
$sm(x)$	Softmax activation function output values	[RGB Value]
t	Time	[s]
t_{flight}	Time of flight	[s]
v	Wave number	[cycles/metre]
v_0	Cut-off wave number	[cycles/metre]
w	PSD dimensionless parameter – 2	[dimensionless]
w_i	Neural net class	[dimensionless]
$w_j\|$	Neural net class	[dimensionless]
x	Input vector	[dimensionless]
y	Coordinate	[dimensionless]
z	Coordinate	[dimensionless]

Uppercase Roman symbols

Symbol	Description	Unit
C	General roughness parameter	[dimensionless]
D_{object}	Distance to object	[m]
G_0	Roughness magnitude parameter	[dimensionless]
$G_d(n)$	Displacement power spectral density amplitude	[metre^2/cycle/metre]
$G_Z(v)$	Power spectral density amplitude	[metre^2/cycle/metre]
$R(z)$	Activation function output value	[dimensionless]
$S(x)$	Sigmoid activation function output value	[dimensionless]
$\mathbb{T}$	Nearest neighbour test pattern	[dimensionless]
V_{light}	Speed of light	[m/s]

Greek symbols

Symbol	Description
π	Pi, mathematical constant ≈ 3.14159

Abbreviations

Abbreviation	Description
2D	Two-dimensional
3D	Three-dimensional
AGV	Autonomous Ground Vehicle
CCD	Charged Coupling Device
CNN	Convolutional Neural Network
DSD	Displacement Spectral Density
FNN	Feed-forward Neural Network
GPS	Global Positioning System
GPU	Graphics Processing Unit
HSI	Hue, Saturation, Intensity
HSV	Hue, Saturation, Value
IMU	Inertial Measurement Unit
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
KNN	K-nearest Neighbour
Lidar	Light Detection and Ranging
PSD	Power Spectral Density
PNN	Probabilistic Neural Network
Radar	Radio Detection and Ranging
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
RMS	Root Mean Square
SVM	Support Vector Machine

Units of measure

Abbreviation	Description
cycle/m	Cycle per Metre
fps	Frames per Second
GB	Gigabyte
GHz	Gigahertz
Hz	Hertz
km	Kilometre
km/h	Kilometre/Hour

Abbreviation	Description
m	Metre
mm	Millimetre
mm/s	Millimetre/Second
MP	Megapixel

Chapter 1: Introduction

1.1 Introduction

Information about the road terrain a vehicle drives over is some of the important information a vehicle's control system should acquire (Wang, 2019). Distinct types of off-road terrain have various different physical characteristics. Off-road terrain classification using on-board sensors can provide important information regarding safety, fuel efficiency and passenger comfort. The vast majority of terrain classification work has focused on on-road terrain classification and less so on off-road terrain classification. A wide range of applications including defence, agriculture, conservation, and search and rescue could benefit from off-road terrain classification (Shaban, et al., 2021). Understanding the terrain surrounding the vehicle is crucial to traverse the road successfully.

Different types of off-road terrain can have a significant impact on vehicle handling, ride quality, and stability (Coyle, 2010). The vehicle's tyre traction properties that affect the lateral and longitudinal wheel slip are determined by the physical parameters of the off-road terrain, including the friction coefficient, soil cohesion, and internal friction angles. The normal force acting on a vehicle tyre is modulated by the terrain roughness, which affects the handling characteristics of a vehicle. Driving a vehicle on mud, rocks and snow could each require a different driving manoeuvre (Wang, 2019).

When a vehicle drives on muddy terrain, the wheels tend to get stuck. To prevent the vehicle from digging into the mud and spinning, the traction must be controlled carefully. The driver should drive slowly and avoid high acceleration. If on sand, good operation is to steer smoothly with gear changes at a high revolution per minute. Soft sand should be avoided at the base of dunes and gullies. If a driver drives on dunes and gullies, he/she should make turns as wide as possible and not brake rapidly. Slow deceleration is better than sudden breaking. When a vehicle moves on rocky terrain, slip tends to occur between the tyres and the slick rock surface. Sharp turns as well as quick acceleration/deceleration that may cause the vehicle to skid or flip over should be avoided (Wang, 2019).

When a vehicle travels on icy terrain or snow, the force of the tyres to grip the road surface reduces significantly. The reduction in grip makes the motions of speeding up, slowing down, and changing direction potentially dangerous. A moderate driving manner is the best way to drive in such conditions. To avoid locking the wheels when braking, the driver is suggested to shift into a lower gear allowing engine braking. This reduces vehicle speed, which should be followed by gentle operation of the brakes. If skidding occurs, the driver should stop stepping on the acceleration pedal and turn into the skid direction (Wang, 2019).

The estimation of the off-road terrain is critical to the drivers of vehicles. Terrain classification is categorised as vision-based, reaction-based or a combination of reaction- and vision-based methods (Coyle, 2010). The purpose of terrain classification is to determine what information to provide to the driver or what changes to make to the vehicle. Reaction-based classification can only be done once the vehicle has driven over the specific part of the terrain, whereas vision-based classification classifies the terrain before the vehicle drives over that part of the terrain (Coyle, 2010). This could make vision-based classification more ideal than reaction-based classification, which has also become crucial for autonomous vehicles as vision-based classification provides the vehicle with information of what is to come.

The benefits of off-road terrain classification become clear when considering the above. This dissertation studied off-road terrain classification to determine whether off-road terrain classification is possible before a vehicle drives over specific terrain. This information will assist the vehicle's control system to make changes to the vehicle if needed before driving over the specific terrain. The captured data could potentially

increase the quality of level 1 and level 2 autonomy (Harner, 2020). The goal is to provide the vehicle with quality data of the road ahead.

The next chapter describes the in-depth research that was conducted to establish how terrain classification is performed. Various solutions, results and methods used were studied. Research was conducted on off-road terrain classification to determine what has been done and how these results could be used to aid in this research study.

Chapter 2: Literature survey

This chapter describes the in-depth research that was done to determine the existing terrain classification models. The purpose of the literature survey was to summarise and better understand how the different models work and to propose a plan on how to solve off-road terrain classification.

The findings of each paper were investigated further to gain a better understanding of each method. The results were used to choose the best possible solution to attempt off-road terrain classification and to determine whether it would be possible to classify between different terrains before a vehicle drives over specific terrain.

2.1 Types of terrain classification

Both reaction and vision-based approaches are analogous to a human driver's recognition of a terrain based on what is seen visually and what is felt through the vehicle's reactions while traversing the terrain (Coyle, 2010). Vision-based terrain classification is usually performed using lidar, cameras and radar. Radar is not a vision-based system, but is used for vision-based classification. Although some visual techniques provide detail about the environment, they do not identify the traversed terrain. These techniques include three-dimensional (3D) maps to show navigability, trees, bushes and vegetation, and stereo imagery to better detect unexplored terrain. Image processing is applied to detect surface characteristics such as slope and roughness. Research has further shown that visual detection from aerial vehicles is an effective means of recognising topology, which can be used for detection and planning of road conditions (Coyle, 2010).

Reaction-based classification relies on proprioceptive sensor measurements such as wheel slip, wheel sinkage and vehicle vibrations. This is measured while the vehicle is in operation and is also known as terrain classification using proprioceptive sensors. Vehicle slip and wheel sinkage can be difficult to measure accurately when doing off-road terrain classification. Reaction-based terrain classification is most often performed using vehicle vibrations as inertial sensors and accelerometers can measure vibrations easily (Coyle, 2010). Reaction-based classification relies on the feedback given by the proprioceptive sensor after the vehicle has driven over certain terrain (Coyle, 2010). As stated, the goal of the dissertation is to establish whether off-road terrain classification is possible *before* a vehicle drives over certain terrain. The focus forward will be on vision-based classification and the sensors and techniques used to achieve this goal.

2.2 Different types of sensor used for vision-based terrain classification

A large amount of research has been done on off-road terrain classification and autonomous vehicles. These papers studied several different sensors that are the current state-of-the-art solutions. The vision-based classification studies included sensors such as cameras, and lidar and radar systems (Coyle, 2010). A brief introduction into each of the different vision-based sensors is provided in this section.

2.2.1 Cameras

Some autonomous vehicles are equipped with cameras at every angle, enabling them to have a 360° view around the vehicle. This image data of the surroundings is used to perform several different classification tasks depending on the required application (Khvoynitskaya, 2020). Image data can provide the driver with detail about the surroundings. Furthermore, the images from the cameras can potentially be used to aid off-road terrain classification. Cameras have become the most essential component of advanced driver-assistance systems (Autocrypt, 2021). There are two types of camera sensor, namely a charged coupling device (CCD) and a complementary metal oxide semiconductor (Rudolph & Voelzke, 2017).

Image data is widely used for classification tasks and quite often with convolutional neural networks (CNNs) (Tripathi, 2021). However, image data requires additional sensors/information to calculate the distance to an object (Rosebrock, 2015). It is difficult for cameras to detect objects when there is low visibility caused by fog, rain or night-time (Khvoynitskaya, 2020). Cameras are far from perfect and are greatly affected by poor weather conditions. In some instances, the images captured are just not good enough, thereby making classification impossible (Khvoynitskaya, 2020).

Advantages of cameras (Autocrypt, 2021, Review Autopilot, 2017):

- Less expensive than lidar systems.
- Recognise two-dimensional (2D) information (capable of detecting 2D shapes and colours, making it crucial for reading lanes and pavement markings).
- See the world the same way as humans do.
- Can easily be incorporated into the design of a car and hidden within its structure.

Disadvantages of cameras (Review Autopilot, 2017)

- Subject to same visual issues faced by humans.
- Affected by rain, fog and snow (adverse weather conditions).
- Distance is not measured directly and requires postprocessing to obtain the distance to an object.

2.2.2 Radar

Radar sends out radio waves that detect objects and captures the speed and distance of these objects in relation to the vehicle's position and speed in real time. Short-range radar sensors (24 GHz) enable blind-spot monitoring and are ideal for lane-keeping assistance and parking aids. Long-range radar sensors (77 GHz) assist in providing radar data for automatic distance control and brake assistance. Radar sensors do not have the same vision problem as cameras when it comes to rain and fog (Khvoynitskaya, 2020). Because radar detects objects using radio waves, radar can supplement camera vision in times of low visibility, which allows for better detection of objects in non-ideal conditions.

Radar works by transmitting radio waves in pulses. These waves hit an object, and the wave signal is sent back to the radar. The radar sensor provides data in the form of the distance and speed of the object that reflected it. Radar typically surrounds the car and accurately measures the distance and speed, but it cannot identify the type of object/terrain or what the exact object is. This means postprocessing of the data is required, which is combined with supervised learning (or other types of learning model) to label the data captured (Burke, 2019).

Advantages of radar (LiDAR and RADAR information, 2017):

- Can penetrate clouds, fog, mist and snow.
- Can penetrate insulators such as rubber and plastic.
- Can provide the exact position of an object.
- Can determine the velocity of, and distance to an object.
- Relatively cheap compared with lidar.

Disadvantages of radar (LiDAR and RADAR information, 2017):

- Takes longer to lock a target.
- Large objects that are close to the transmitter can saturate the receiver.
- Several objects and mediums in the air can interfere with radar.
- Cannot differentiate the colour of an object.

2.2.3 Lidar

Lidar operates similarly to radar sensors, but lidar uses light waves/beams instead of radio waves. Lidar creates 3D images of detected objects and maps the surroundings. Lidar can be configured to create a full 360° map around the vehicle rather than relying on a narrow field of view (Khvoynitskaya, 2020).

Lidar is a remote sensing method that uses light in the form of a pulse sent from a laser to measure the distance to objects. Lidar consists of two principal components, namely a laser and scanner. Lidar was initially used by aeroplanes and helicopters, but its use has grown in the field of autonomous driving to assist the vehicle in being able to decide, detect and classify objects (National Ocean Service, 2016).

Lidar measures the exact distance (distance between the sensor and the target object) to an object. These light pulses create accurate 3D information about the object and the earth's surface surrounding the object. Besides the scanner, laser and global positioning system (GPS), which is not in all types of lidar, the photodetector and optics of the lidar also play a vital role in the capturing of data (Sharma, 2019).

Lidar sends out laser signals and then calculates the time the signals take to return. Measurements are done at the speed of light, namely 299 792 458 m/s. The following formula explains how the distance to the object is calculated (Sharma, 2019):

$$D_{object} = \frac{(V_{light} \times t_{flight})}{2} \quad 2.1$$

Where D_{object} is the distance to the object, V_{light} is the speed of light (299 792 458 m/s), and t_{flight} is the time of flight.

Lidar data is usually in a point cloud form in the format of x-, y-, and z-coordinates, which make it easy to produce a 3D representation of the detected objects. However, the data still requires postprocessing to determine which data points to cluster.

Advantages of lidar (LiDAR and RADAR information, 2017):

- Fast data capturing with high accuracy.
- Surface data has higher sample density compared to radar data.
- Capable of collecting elevation data in a dense forest.
- Can be used day and night.
- No geometric distortions.
- Can map inaccessible and featureless areas.

Disadvantages of lidar (LiDAR and RADAR information, 2017):

- High operational costs.
- Ineffective during heavy rain or low hanging clouds.
- Degraded at high sun angles and reflections.
- Elevation errors due to the inability to penetrate very dense forests.
- May affect the human eye in cases where the beam is powerful (near-infrared lasers with wavelengths up to 1440 nano meters).
- Requires complex data analysis techniques with high computational costs.

The above concludes basic information about some of the most widely used vision-based classification sensors. Each of the three has its respective advantages and disadvantages. Cameras have vision-like sensory enabling them to distinguish easily between shapes and colours. Cameras can quickly identify the

object based on this information. Camera data is compared with the human eye, which leads to the main disadvantage. Cameras have poor vision during extreme weather conditions. This is where radar and lidar become a better option because their vision is not affected by weather conditions. Although radar and lidar are better suited for bad weather conditions, they also have their disadvantages. Radar has low-definition modelling, which means that it is accurate at detecting objects but is usually outperformed by cameras (depending on the quality camera and radar used). Lidar, on the other hand, has high relative cost and is highly sophisticated. Further information is required to establish which sensors most state-of-the-art solutions used in previous studies and which classifier specific sensors mostly used to obtain optimal results.

2.3 State-of-the-art solutions

The previous section discussed the main sensors used for vision-based classification. This section further discusses how these sensors are combined with different off-road terrain classification techniques to achieve off-road terrain classification. This section reviews the state-of-the-art solutions available and considers the sensor and classification technique combinations, dataset, preprocessing, results, shortcomings and recommendations of each study. This information was used to decide which sensor and classification technique would be used going forward.

Lu, et al. (2009) studied terrain surface classification for autonomous ground vehicles (AGVs) using a 2D laser stripe-based structured light sensor. The paper considered vision-based terrain surface classification for the path directly in front of the AGV (< 1 m ahead). The research paper employed a laser stripe-based structured light sensor that used a laser in conjunction with a camera, which could therefore be used at night. The benefit of this sensor was that it did not rely on colour changes caused by illumination and weather and only relied on spatial relationships. The study classified between the following terrains (Lu, et al., 2009):

- Asphalt
- Grass
- Gravel
- Sand
- White sand
- Red sand

The size of each terrain patch captured by the camera was about 45 cm × 35 cm. The images were greyscale with a resolution of 640 × 480 pixels collected at a rate of 15 frames per second (fps). The study by Lu, et al. (2009) mentioned the frame rate at which the images were captured, but unfortunately did not mention the size of the dataset captured. However, it could be calculated based on the time the robot was allowed to drive over each road profile. The experimental procedure mentioned that the robot captured data for 10 seconds for each road profile. The robot captured images at 15 fps and captured images for six different road profiles. This led to a dataset consisting of 900 images (150 images per class). The vehicle had a maximum translational speed of 1 400 mm/s. The frame-per-second rate and speed of the vehicle refer to a robot, which does not mean that the same setup would be suitable for vehicles. Vehicles usually travel at much higher speeds even if they are travelling off-road.

Lu, et al. (2009) used a probabilistic neural network (PNN) to classify the feature vector as a particular terrain due to the PNN's simplicity, robustness to noise, and fast training speed. The PNN was based on a Bayesian classifier and used a supervised learning model. The laser was pointed 25° downwards with respect to the horizontal plane and the camera was pointed 48° with respect to the horizontal plane. Figure 2.1 shows the AGV experimental setup and the camera and laser setup (Lu, et al., 2009).

Figure 2.1: A) AGV experimental platform B) Camera and laser setup (Lu, et al., 2009)

Lu, et al. (2009) achieved a classification accuracy greater than 86.0%. Nonetheless, grass and gravel were the two terrain classes that were confused the most often (Lu, et al., 2009).

Weiss, et al. (2008) presented a paper on terrain classification that fused terrain predictions based on image data with predictions made by a vibration-based method. The research study utilised colour images to classify the terrain in front of it using a support vector machine (SVM) classifier. This was followed by the robot driving over the classified terrain and using a vibration-based classification technique to verify the image classification results. Fourteen different terrain classes were captured as shown in Figure 2.2 (Weiss et al., 2008).

The camera sensor did not face the AGV wheel, but faced the area in front of the robot. The robot classified the area in front of it based on the textures of the images using integral invariant features. Invariant features are image characteristics that remain unchanged under the action of a transformation group (Schulz-Mirbach, 1995). The image classification consisted of an offline and online classification phase. Although the robot was equipped with a stereo camera system, only the left camera was used. Colour images of 600 × 600 pixels were captured 0.1 m to 2 m in front of the robot during the offline training phase. The study captured 180 images per terrain, but ended up only using 44 images per class. Figure 2.2 shows the different terrain classes. An average terrain classification of 71.26% was achieved for the vision-based classification and 87.33% for a combination between vision-based and vibration-based classification (Weiss et al., 2008). One of the limitations of the project was the sample size.

Figure 2.2: Fourteen different terrain classes used by the AGV (Weiss et al., 2008)

Another study done by Weiss, et al. (2007) focused on comparing the different approaches to vibration-based terrain classification. The vibration-based study used vibrations induced in the robot while being pulled over rough terrain. The acceleration in the z-direction was extracted from this data and the results were compared with the vision-based approaches followed by Weiss, et al. (2008) The study investigated the use of different classification techniques. Data was collected by their RWI ATRV-Jr outdoor robot. The most common data used for terrain classification is data collected by cameras and laser. Lidar-based methods often focus on ground segmentation from vegetation or all kinds of obstacles instead of estimating the type of ground surface, whereas vision-based methods use texture or colour information. Four groups of classification method were considered, including: kernel methods, neural networks, likelihood, and decision trees. More specifically, the following classifiers were considered (Weiss, et al., 2007):

- SVM classifier
- PNN
- K-nearest neighbour (KNN)
- Decision trees
- Brooks's method
- Naïve Bayes

The study considered the following classes (Weiss, et al., 2007):

- Indoor floor
- Asphalt
- Gravel
- Grass
- Paving
- Clay

The robot traversed over six different types of terrain and captured data points at three different speeds including 0.2 m/s, 0.4 m/s and 0.6 m/s. This led to a total dataset of 10 225 images (2 510 for 0.2 m/s, 3 724 for 0.4 m/s, and 3 991 for 0.6 m/s). It took the robot 2h50 min to capture all the terrain images. Table 2.1 shows the number of samples captured per class.

Table 2.1: Number of samples per class for the study performed by Weiss, et al. (2007)

Class	0.2 m/s	0.4 m/s	0.6 m/s	Total
Indoor floor	282	549	581	1 412
Asphalt	499	513	600	1 612
Gravel	311	323	392	1 026
Grass	482	572	631	1 685
Paving	314	573	567	1 454
Clay	423	579	605	1 607
No motion	199	615	615	1 429
Total	**2 510**	**3 724**	**3 991**	**10 225**

The results concluded that an SVM gave the best results followed by a PNN (Weiss, et al., 2007).

Byl and Filitchkin (2012) presented a terrain classification approach that used a single compact camera that maintained high classification rates robust to changes in illumination. The terrains were classified using a bag of visual words created from speed-up robust features using an SVM classifier. The goal of the research was to provide a terrain classification framework that was not only more robust than colour-based classification, but also suitable for real-time applications. The study experimented using different image

sizes for the data ranging between 192 to 640 square pixels. The given feature extraction methods delivered inaccurate results for images smaller than 192 × 192 pixels. The data further indicated that the colour classifier ineffective in underexposed lighting conditions. Images of 320 × 320 pixels that extracted between 200 and 250 features per image gave the best results. The study used six different terrain classes (Byl & Filitchkin, 2012)

- Asphalt
- Grass
- Gravel
- Mud
- Soil
- Woodchips

The study did not give an indication of the number of images that was used overall. The classifier achieved a 100% verification accuracy on both 640 × 480 and 320 × 240 image sets (Byl & Filitchkin, 2012).

Howard and Seraji (2001) presented a technique for real-time terrain characterisation and assessment of terrain traversability using a vision-based system and artificial neural networks. The key terrain traversabilities established were terrain roughness, discontinuity, hardness, and slope. These characteristics were extracted from image data obtained from cameras. The paper attempted to incorporate all four mentioned terrain characteristics to perform terrain classification. A three-layer feed-forwards neural network (FNN) was used consisting of an input layer, hidden layer and output layer. No real information was given on the image size, dataset size, and classification accuracy results. The paper noted that illumination was a limitation to the research. Illumination can lead to misclassification of terrain profiles due to shadowing (Seraji & Howard, 2001)

Manduchi, et al. (2005) did a study on obstacle detection and terrain classification for autonomous off-road navigation. The paper presented a new sensor-processing algorithm suitable for cross-country autonomous navigation. The study used two sensor systems working together, namely a colour stereo camera and a single-axis lidar. The study proposed an obstacle detection technique based on stereo range measurements. A colour-based classification system was used to label the detected obstacles according to a set of terrain classes. An algorithm was used to analyse the lidar data, which allowed the vehicle to distinguish between obstacles (including tree trunks and rocks) and grass (Manduchi, et al., 2005).

Manduchi, et al. (2005) used two approaches: the first approach was based on stereo and colour analysis from colour cameras, and the second approach was based on processing the range data from the lidar. The two approaches were complementary in that they discriminated between terrain cover classes: local range statistics in the first and surface reflectivity in the second. Colour classification was used to classify between a few distinctive classes such as grass, foliage, dry vegetation, bark, rocks, and soil. The colour stereo camera setup was the most effective for characterising and detecting isolated obstacles and capturing elevation profiles of the scene. The single-axis lidar was placed on the lower portion of the front of the vehicle and was used for safe navigation in tall grass where obstacles were partially hidden by vegetation, which could therefore go undetected by stereo analysis (Manduchi, et al., 2005).

Figure 2.3 shows the test setup used by Manduchi, et al. (2005). For the obstacle detection task, the stereo image size was 320 × 240 pixels. The results showed that the stereo setup could only compute disparities where the left and right images in the pair overlapped, thereby leaving the left-most and right-most columns of the image without range measurements. The algorithm was able to detect 'negative obstacles' (a ditch, for example) (Manduchi, et al., 2005).

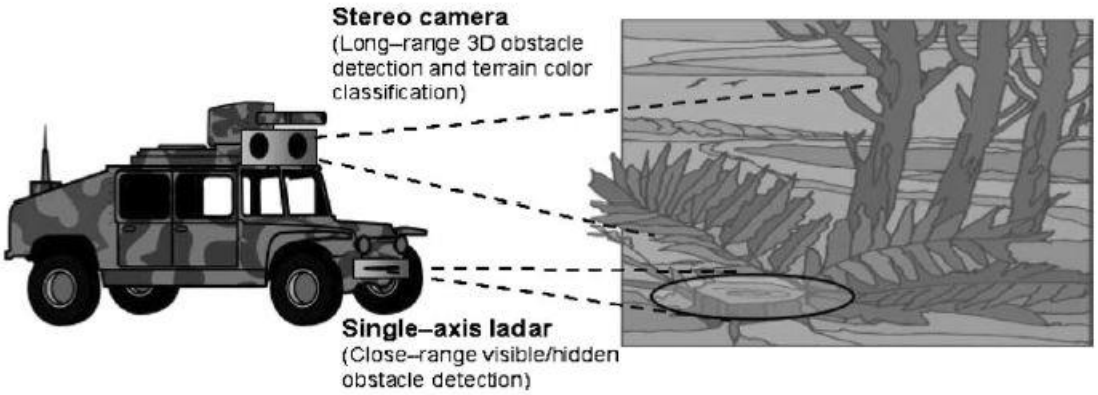

Figure 2.3 Stereo camera and single-axis lidar setup (Manduchi, et al., 2005)

The study listed some advantages and disadvantages between cameras and lidar. Colour cameras are passive and have several advantages: they are relatively cheap compared with other sensors of the same quality, are small and light, produce full-frame data at an acceptable rate, and can be used to compute range by stereopsis (Manduchi, et al., 2005). However, regular cameras cannot be used at night and the quality of stereo data is not always satisfactory, especially in the case of grass patches and dense vegetation. Lidar can be used day and night and produces accurate and dense range data. However, the interpretation of range data might not be as straightforward as with camera data (Manduchi, et al., 2005).

The following classes were considered for colour-based classification using cameras (Manduchi, et al., 2005):

- Soil/rock
- Green (photosynthetic) vegetation
- Dry (non-photosynthetic) vegetation (which includes tree bark)
- 'None of the above'

The following classes were considered for the range analysis from the lidar (Manduchi, et al., 2005):

- Obstacles characterised by relatively smooth surfaces (such as rocks or tree trunks)
- Grass/foliage

The results showed that the camera data for classification gave very good results, but required a substantial amount of training data that was labelled (supervised learning), which made it unpractical. The lidar classification attempted to distinguish grass patches from non-traversable obstacles such as rocks and tree trunks and managed to do so with a high success rate. The experiments ran on eight scenes between 15 and 90 seconds. The scenes included environments with sparse and dense grass, and with and without obstacles. From the 1 313 scans produced by these scenes, four false alarms were given (Manduchi, et al., 2005). Although the study did not mention the number of samples used to train the model, it did mention that the algorithm ran through an average of 512 samples per experiment during live application.

Selvathai, et al. (2017) published a paper that investigated road and off-road terrain classification for AGVs. They used colour images as the data source and a neural network-based classification technique. While Manduchi, et al. (2005) classified between several classes, Selvathai, et al. (2017) only classified between two classes: on-road and off-road. Images of 50 × 50 pixels were used. A feature set was extracted for each class and a neural network was used to classify between high-dimensional feature vectors. The feature extraction process utilised both colour and texture distributions and a multilayer FNN. The algorithm was trained on video data obtained from a front-looking camera mounted on a vehicle. A supervised learning model was used to train the neural network (Selvathai, et al., 2017).

Selvathai, et al. (2017) showed that high dynamic range and constantly varying lighting conditions were the major challenges while classifying the terrain. The input vector for training the back propagation neural network (the batch size) was a feature set of a 100 manually labelled images (supervised classification model) that consisted of 6 250 images per class. Training was terminated once the mean square error was satisfactorily low. The neural network classifier delivered a successful classification of 93%. The future goal of the project is to be able to classify between more classes (Selvathai, et al., 2017).

A study done by Sung, et al. (2010) performed terrain classification based on the colour and texture features of images. The study classified between different classes including soil, gravel, foliage, and sky. Discrete wavelet transform coefficients were used to extract features from images. The spatial coordinates of where the specific terrain was situated in an image were also located. A neural network was used to train off-road terrain images. The study found that the feature vectors extracted using the Daub2 wavelet that transformed in the hue, saturation, and intensity (HSI) colour space had the best classification performance. Using the wavelet features and spatial coordinates improved the terrain cover classification performance (Sung, et al., 2010).

The wavelet transform extracted the distinct differences in the images, which assisted in classifying between different terrains. To ensure feature extraction robustness under various light conditions, the brightness channel was separated from each colour channel and intensity normalisation was performed (Sung, et al., 2010). During training, the weighting parameters were computed through offline training by means of supervised learning using error backpropagation. The weights were adjusted through backpropagation, which minimised the mean square error between the actual and desired output. After training was completed, the weighted parameters were fed into the neural network.

A multilayer perception is a representative static neural network that performs classification using supervised learning (Sung, et al., 2010). The multilayer perception consisted of two hidden layers and used the sigmoid activation function. The input layer consisted of a 24- and 26-dimensional feature vector. The output layer had six nodes, which could change depending on the number of classes present. Sung, et al. (2010) found that using two hidden layers with between 12 and 18 nodes delivered a good performance. Thereafter, the classified results were computed using feature vectors (including colour features and spatial features) extracted from the input image. The images were captured with an NTSC 50 × 38 degree of field view Sony XC-555 colour camera. The images were captured at 720 × 480 pixels at 30 fps. A 100 random images were used for extracting training chips, and ten images were selected from the random images. The study obtained an average prediction accuracy of 81.8% (Sung, et al., 2010).

The training set only used 200 images per class and a total of 1 200 images were used during training. This was a small dataset that may not represent the best results. A bigger dataset will give a better idea of what the actual results would be. During training, the images were reduced to 16 × 16 pixels (Sung, et al., 2010). Although this reduced the training time, reducing the size of the images could potentially remove distinctive class information present in the image.

Stolee and Wang (2018) conducted a survey studying machine learning techniques for road detection to determine how different machine learning techniques could be used to predict the driveable road path. Road detection could be done using several sensors including cameras, radar, lidar, GPS and vehicle inertial measurement unit (IMU) to implement practical road detection (Stolee & Wang, 2018).

To evaluate the performance of these methods, the study used the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset, which consisted of a labelled training dataset of photos taken from car-mounted cameras. In addition, the authors provided photos taken from a right-mounted camera, as well as Velodyne laser point data, GPS data and IMU data. The study took data points from the front and both sides of the vehicle. Although photos were taken in ideal weather conditions, the sample photos still

had variabilities in terms of shadows created by cars, trees and buildings. The photo size of images was 375 × 1 242 pixels, and with this number of pixels, training could be expensive. Therefore, the pixels were transformed into superpixels (Stolee & Wang, 2018).

Figure 2.4 shows an example image on which the classifier was performed. The left side shows the ground truth and the right-hand side shows the classification results.

Figure 2.4: Road classification example (Stolee & Wang, 2018)

Stolee and Wang (2018) experimented with different numbers of superpixels, feature sets and classifiers. The study only used 289 images – 60% of the images for training, 10% for validation, and 30% for testing (Stolee & Wang, 2018). The validation set was very small, which could have limited the statistical significance of the cross-validation of the results. Stolee and Wang (2018) found that increasing the number of superpixels increased the performance of the classifier. When using the hue, saturation and value (HSV) colour domain, they found that varying the hue and saturation components greatly affected the results. Varying the red, green and blue (RGB) colour space and the size of the superpixel did not affect the performance of the classifier as much. However, using the RGB colour space did reduce the accuracy sensitivity when performing classification (Stolee & Wang, 2018).

Stolee and Wang (2018) studied the following classifiers:

- Logistic regression
- KNN
- Decision tree
- Mixture of Gaussians
- Neural net
- Ensemble methods
- Extra-trees and random forest
- Bagging
- AdaBoost

The paper concluded that using a neural net with two layers delivered the best results. The first layer consisted of 50 hidden units and the second layer of 25 hidden units. It is important to note that the neural net performed better in this study than the other methods (Stolee & Wang, 2018). The exact angle at which the camera was placed was not mentioned, but Figure 2.4 gives a relatively good idea of camera positioning. It would have been ideal to determine the classification performance of each classifier. Several studies up to this point used images as the data source and a neural net to perform the classification.

Omer and Fu (2010) investigated the feasibility of classifying winter road surface conditions using images from low-cost cameras mounted on regular vehicles. An SVM classifier was trained by means of feature extraction, whereafter the SVM classified the images into the respective categories. Different training schemes were used to study the effect they had on classifying the images into the respective categories. Images were captured from a camera mounted inside the car in front of the rear-view mirror. The images contained significant noise such as obstacles outside the boundaries of the road. The camera did not only capture the road, but also a portion of the vehicle and objects on the side of the road (Omer & Fu, 2010).

Figure 2.5: Image noise present when capturing images from inside the vehicle with a camera mounted in front of the rear-view mirror (Omer & Fu, 2010)

Classification was done between bare, covered and track snow roads as shown in Figure 2.5; each becoming more difficult to drive over. The classification accuracy results using an SVM were 84% for bare, 86% for covered, and 85% for track roads. Appropriate training of the SVM was essential for achieving good results, but there were still problems with changes in light intensity and shadows (Omer & Fu, 2010).

Giese, et al. (2017) did research regarding road course estimation using a deep-learning approach on radar data to estimate the course of the ego lane based on occupancy grids generated by radar sensors. 'Ego lane' was the name given to the lane where the vehicle was positioned. Classification is usually done via cameras and lidar, but in this research, classification was attempted using radar. Radar is mostly used to track traffic participants due to its ability to measure the speed of an object. Furthermore, radar is less influenced by adverse weather conditions than cameras and lidar (Giese, et al., 2017).

Giese, et al. (2017) used deep learning, specifically a CNN, to train the model. Although CNNs have been applied quite successfully to classify camera images, they have not been used to classify radar data yet. The study used image data with the resulting images 120 pixels wide and 440 pixels high, which relates to 60 m × 220 m. No information was given about the terrains considered and the size of the dataset. The study concluded that using a CNN was beneficial, especially on images. A CNN could be used for radar data, but it was more challenging and required further research (Giese, et al., 2017).

Thrun, et al. (2006) described the robot Stanley that won the 2005 DARPA Grand Challenge. The goal of the DARPA challenge was to develop an autonomous vehicle (robot) capable of driving an unrehearsed off-road terrain route. Stanley was developed for high-speed desert driving without manual intervention using the state-of-the-art artificial intelligence technologies available at that time. Stanley further applied machine learning and probabilistic reasoning (Thrun, et al., 2006).

The main interest of Thrun, et al. (2006) was studying how Stanley classified the route it drove. Stanley had five lasers, a camera interface, and a radar interface. The information gathered by these sensors was sent to the perception layer. Three different mapping modules built 2D environmental maps based on the lasers, camera, and radar system (Thrun, et al., 2006).

Five SICK laser range finders were used for Stanley's short- and medium-range obstacle detection. These lasers were mounted on the roof and tilted downwards to scan the road. The lasers could effectively measure up to 22 m, which was acceptable for speeds up to 40 km/h. To extend the range at which obstacles could be detected, a colour camera was used to extend the classification range up to 70 m (Thrun, et al., 2006).

Figure 2.6 shows two rows of three images each. Red shows the areas where Stanley classified the terrain it would be able to drive on. In the first row, Stanley used lasers to perform obstacle detection to classify drivable areas without the use of cameras. The first row shows that the grass was classified as not drivable when only using lasers. In the second row, a camera was added to the setup to classify drivable areas. This extended Stanley's classification range, and grass was classified as drivable (Thrun, et al., 2006).

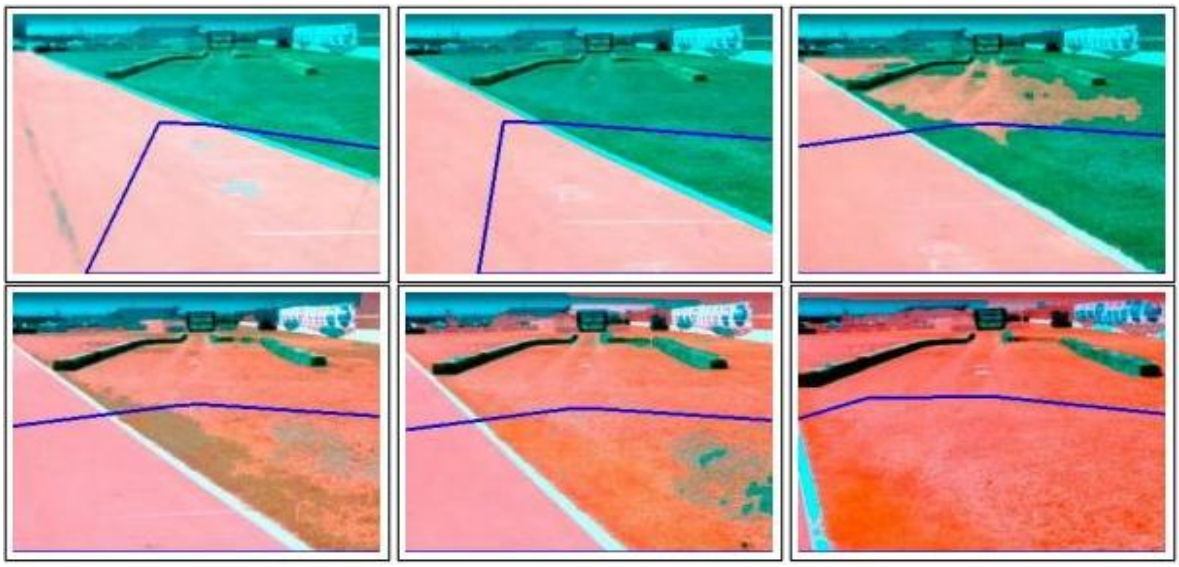

Figure 2.6: Images displaying the rapid adaption of Stanley's computer vision routines (Thrun, et al., 2006)

Stanley was unable to navigate in traffic where the surroundings were not static. For autonomous vehicles to succeed, vehicles should be able to function in traffic. Although Stanley's software was able to classify most objects it faced, it was not able to distinguish between tall grass and rocks (Thrun, et al., 2006).

Roychowdhury, et al. (2019) used several classification machine learning models for road surface and friction estimation using front-camera images. The study was done to increase safety using real-time road friction estimates. The increase in safety was due to the ability to adapt driving styles depending on the road conditions. The work consisted of two stages: the first stage used a CNN to train region-specific features for road surface condition classification. The second stage used a rule-based model that relied on domain-specific guidelines and that was implemented to segment the ego lane surface into 5 × 3 patches. Stage 1 achieved a 97% accuracy and stage 2 an 89% accuracy (Roychowdhury, et al., 2019).

Figure 2.7 shows the difference between the two stages. During stage 1, the images were classified into one of four classes: dry, wet/water, slush, or snow/ice. The classes and data were collected from 48 different YouTube videos. Each video was sampled to isolate frames greater than one second apart. Each image captured was treated as an independent sample and had a range between 640 × 360 pixels and 1 280 × 720 pixels. The images were annotated manually to assign whether the road was driveable.

Thereafter, 40–100 frames were extracted per video, resulting in a total of 3 750 training images and 1 550 test images. During stage 2, the road was broken up into segments, and each segment was labelled as one of the following three options: low driveability, medium driveability, or high drivability. The model used the applied structural similarly index metric and coefficient of variation to quantify between the different driveability classes. A CNN was chosen because CNNs are well known for feature extraction from images (Roychowdhury, et al., 2019).

The results showed that stage 1, in which the images were placed into one of four categories, achieved an accuracy between 94% and 99%. Stage 2 achieved an accuracy of up to 89.9%. This showed that CNNs give outstanding results when using cameras in combination with deep learning (Roychowdhury, et al., 2019).

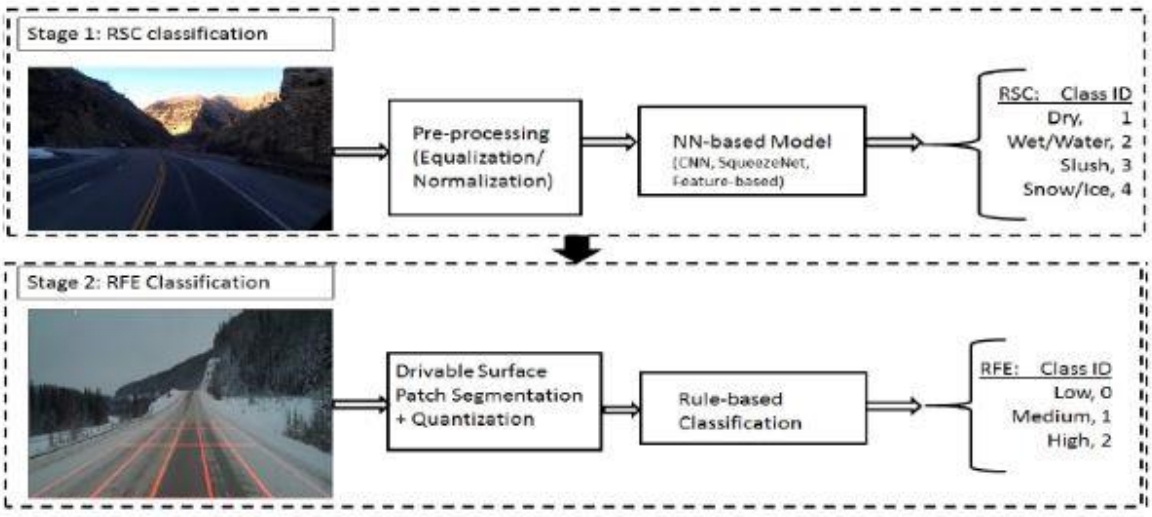

Figure 2.7: Two stages for classifying images (Roychowdhury, et al., 2019)

A summary of the different state-of-the-art solutions can be found in Table 2.2.

Table 2.2: Summary of classification approaches found in the literature

Reference	Classifier details	Sensors	Dataset specifics	Results
Selvathai et al. (2017)	Neural network with supervised learning. Binary classifier (on-road and off-road)	Camera	6250 images, 50x50p	93%
Sung et al. (2010)	Neural network with supervised learning. Multiple classes (soil, gravel, foliage and sky)	Camera	200 images per class, 16x16p	81.8%
Lu et al. (2009)	Probabilistic neural network with supervised learning. Multiple classes (asphalt, grass, gravel, sand, white sand, red sand)	Camera image of line laser projected on terrain	900 images, 640x480p	>86%
Seraji & Howard (2001)	Extraction of features (Slope, roughness, discontinuity and harness) using different methods to classify traversability using fuzzy logic	Camera	No information provided	No information provided
Roychowdhury et al. (2019)	Convolutional neural network with supervised learning. Multiple classes (dry, wet/water, slush and snow/ice)	Camera	3750 training images and 1550 testing images	97%
Weiss et al. (2007)	Comparison of support vector machines, neural network, Bayes classifier, k-Nearest Neighbour Multiple classes (indoor floor, asphalt ,gravel, grass, paving clay)	Accelerometer	10225 samples of 1s (100Hz) sampled vertical acceleration	87.33%
Omer and Fu (2010)	Support vector machine applied to extracted image features supervised learning. Classified between winter profiles that included bare, snow covered and track roads.	Camera	200 training images and 90 test images 160x400p	84 to 86%

Reference	Classifier details	Sensors	Dataset specifics	Results
Byl & Filitchkin (2012)	Support vector machine with supervised learning. Multiple classes (asphalt, grass, gravel, mud, soil and woodchips)	Camera	54 images total 320x320p	100%
Manduchi et al. (2005)	Terrain classification based on colour of pixels using maximum likelihood Gaussian mixture model. Terrain classification using Lidar and range measurements	Camera and Lidar	1313 Lidar scans, no information on camera	99%
Stolee and Wang (2018)	Different classification methods used (logistic regression, K-NN, decision trees, Mixture of Gaussian, NN) applied to 1) average RGB value, positional difference of pixels in superpixel and 2) average hue saturation value (HSV) varaiance in RGB, average grayscale entropy and frequency of edge pixels within superpixel to classify pixels as tarred road or not.	Camera	289 images	90.56%
Giese et al. (2017)	Course estimation of ego lanes based on occupancy grids generated by radar sensors.	Radar (operates at 76 GHz, 200 m range)	3.2 million grids with edge lengths of 0.5 m.	No information provided.
Vulip et al (2022)	LSTM, SVM and CNN applied to spectrogram different sensors. Multiple classes: Concrete, Dirt, ploughed and unploughed	Tri-axial accelerometer, gyroscope, left and right wheel seeds and motor current (10 channels in total)	Not specified, contains over 156 (5sec) recordings of which multiple samples are drawn	CNN 91.5%
Goto, T. and Ishigami, G., 2021	CNN applied to RGB and Infra Red (IR) images of different soils with different moisture content Multiple classes: Perlite, Pumice, Leaf Mold and Dark Soil	Camera	600 images of each class 227x227p	RGB 100% IR 99.8%

The state-of-the-art solutions provided great insight into the different sensors used for terrain classification. It further gave good insight into the different classification techniques available. Most studies used camera sensors to capture image data. A camera sensor might have been used for most studies, but the image

formats varied significantly. Several studies used the images in the RGB colour format, whereas other studies used the images in the greyscale (Lu, et al. (2009)) or HSV format (Stolee and Wang (2018)). More extreme cases included separating the brightness level from the image to create robustness under light conditions (Byl & Filitchkin, 2012).One study extracted integral invariant features from the images (Weiss, et al., 2008). The RGB colour spectrum was used most often. As stated earlier, CNNs are often used with image data because of the good results CNNs give. This could easily be seen during the state-of-the-art studies performed. Most studies used CNNs or another form of neural network. Another classification model used with good results was the SVM model. Both CNNs and SVMs gave good results.

All the studies used a supervised machine learning model, but the dataset sizes for each study were different. It is expected that data capturing was limited by the number of classes available in the area and the amount of space available to capture as many data points as possible. Each study consisted of arbitrarily selected classes. It was quite a shock to see that not a single study attempted to use some sort of standard to determine their classes.

2.4 Ride comfort

As stated at the end of the previous section, none of the state-of-the-art solutions used some sort of standard to determine their classes. This section describes how terrain classification ties into vehicle dynamics and which standards to follow.

Vehicles travel at high speeds and, therefore, experience a broad spectrum of different vibrations. These vibrations are transformed to the driver by tactile, visual or aural paths. These spectra of vibration are divided into two categories: vibrations between 0 Hz and 25 Hz are classified as ride and vibrations higher than 25 Hz (25–20 000 Hz) are classified as noise. The vibrational environment is one of the more important criteria by which the design and quality of a vehicle are judged (Gillespie, 1992).

The lower-frequency ride vibrations are the result of the dynamic behaviour common to rubber-tyred vehicles. These modes are an important area of vehicle dynamics. The vehicle is a dynamic system, but only exhibits vibration in response to an excitation input. The vehicle's response properties determine the magnitude and direction of vibrations imposed on the passenger compartment. These vibrations ultimately determine the passenger's perception of the vehicle. It is, therefore, crucial to know which excitation inputs will result in a certain response perceived by the passenger. Understanding ride comfort involves the study of three main topics (Gillespie, 1992):

- Ride excitation sources
- Basics mechanics of vehicle vibration response
- Human perception and tolerance of vibrations

This dissertation focuses on off-road terrain classification and thus only the ride excitation sources are addressed. The ride excitation sources can further be broken down into four categories (Gillespie, 1992):

- Road roughness
- Tyre/wheel
- Driveline
- Engine

Again, the focus is on terrain classification; therefore, only road roughness was considered. Road roughness includes everything from potholes to random deviations reflecting the practical limits of precision to which a road surface can be constructed. Roughness is described by the elevation profile along the wheel tracks

over which a vehicle passes. A very popular way of describing the roughness of a road is using the terrain's displacement spectral density (DSD) function. The road elevation profile measured can be decomposed by the Fourier transform into a series of sine waves varying in their amplitude and phase relationships. The power spectral density (PSD) is the plot of the amplitude versus spatial frequency (expressed as a wave number with units cycles/metre and the inverse of the wavelength). The PSD for an average road is represented by the following equation (Gillespie, 1992):

$$G_z(v) = G_0[1 + (v_0/v)^2]/(2\pi v)^2 \quad 2.2$$

Where (Gillespie, 1992):

- $G_Z(v)$ = PSD amplitude (metre^2/cycle/metre)
- v = Wave number (cycles/metre)
- G_0 = Roughness magnitude parameter (roughness level)
  = 1.25 × 10^5 for rough roads
  = 1.25 × 10^6 for smooth roads
- v_0 = Cut-off wave number

In actual applications, the ISO 8608 (2016) standard is often used to classify between different terrains. A terrain is classified from a class A to a class H road. The PSD of the terrain is calculated and plotted against the class A to class H road to determine the type of road. The standard incorporates the vibrations experienced by the driver and performs classification based on the severity of the road roughness using the ISO 8608 (2016) standard. The ISO 8608 (2016) approximation makes use of the following formula (Andrén, 2006):

$$G_d(n) = Cn^{-2} \quad 2.3$$

Where:

- $G_d(n) =$ Displacement PSD amplitude (metre^3)
- $C =$ General roughness parameter
- $n =$ Wave number (cycles/m)

Andrén (2006) presented a paper that discussed four different methods to approximate the PSD of a road terrain. The fit of the different approximations was done on the entire Swedish road network and was evaluated using the residual from a least square minimisation. The following methods were discussed (Andrén, 2006):

- ISO 8608
- BSI 1972
- Two split
- Sayers method

The first method is the ISO 8608 approximation that makes use of the formula discussed in equation 2.3. The BSI 1972 approach found that the single straight-line approximation was not suitable for many roads, and made use of a more general form (Andrén, 2006):

$$G_d(n) = \begin{cases} Cn^{-w_1} \; for \; n \leq n_0 \\ Cn^{-w_2} \; for \; n \geq n_0 \end{cases} \quad 2.4$$

Where:

- $n_0 = \frac{1}{2\pi}$
  = Wave number (cycles/metre)
- $w_1 = 3$ Dimensionless parameter
- $w_2 = 2$ Dimensionless parameter

The ISO 8608 makes use of a straight-line approximation, and the BSI 1972 splits the straight-line approximation into two. The two split method further splits the BSI 1972, and makes use of the following approximation (Andrén, 2006):

$$G_d(n) = \begin{cases} C_1 n^{-w_1} \; for \; n \leq n_1 \\ C_2 n^{-w_2} \; for \; n_1 \leq n \leq n_2 \\ C_3 n^{-w_3} \; for \; n_2 \leq n \end{cases} \quad 2.5$$

Where n_1 is a lower break frequency of 0.21 cycles/m and n_2 a higher break frequency of 1.22 cycles/m. These values produced the lowest least square error for the Swedish road network (Andrén, 2006).

The Sayers (1986) proposed a step-wise method used to fit the approximation to measured data. The following formula is used (Andrén, 2006):

$$G_d(n) = \frac{C_1}{n^4} + \frac{C_2}{n^2} + C_3 \quad 2.6$$

The C_2 coefficient is calculated as the mean value for the slope PSD covering wavelengths from 0.08 to 0.5 cycles/m. C_1 is determined from the acceleration PSD over the 0.003 to 0.05 cycles/m range and C_3 is calculated from the elevation PSD over the 0.7 to 3 cycles/m range (Andrén, 2006).

The paper concluded that using the ISO 8608 method is the simplest method that requires the least number of parameters but gave the highest least square fit residual values for the Swedish road network. The two split method gave the lowest least square residual followed by the Sayers and BSI 1972 method (Andrén, 2006).

Reina, et al. (2020) presented a paper that specifically made use of the ISO 8608 standard to do off-road terrain classification. Reina, et al. (2020) presented a method that automatically estimates the roughness of a terrain making use of a stereocamera (Point Grey XB3). The rover also measured the wheel mechanical torque, that consisted of encoders that measured the wheel angular velocity, and an inertial measurement unit (XSENS MTi-300). Images were captured at 16 fps with the rover moving up to 5 m/s. The system was tested in a rural environment where three main surfaces were present including (Reina, et al., 2020):

- Ploughed terrain: vineyard terrain broken and turned over
- Compact terrain: unbroken agricultural land
- Gravel: unconsolidated mixture of white/grey rock fragments or pebbles.

The PSD-based results suggested using the overall energy and waviness, collectively referred to as roughness parameter. The system performance was evaluated in terms of sensitivity and the influence to vehicle tilting was proved to be limited with errors always less than 8% (Reina, et al., 2020).

Reina, et al. (2020) noted that measurements obtained from various surfaces or different robots can only be compared when they refer to the same inspection waveband (Reina, et al., 2020).

Instead of arbitrarily selecting the classes, this study will use the ISO 8608 standard for classification. The model will attempt to classify the terrain classes based on the terrain roughness, which can then be used in the suspension systems to improve decision-making. The off-road terrain classifier will be used to establish whether it is possible to distinguish between these different road types to improve the ride quality of the vehicle.

2.5 Summary of literature survey

Based on the literature survey conducted, the following could be summarised and concluded. Most state-of-the-art solutions made use of image data from a single camera, stereo vision, or a camera in combination with a different sensor. Some papers also used lidar and radar. Image data can be used in various ways, including:

- Colour extraction
- Texture extraction
- Object detection
- Greyscale classification that is independent of the colour of the image data
- Spatial relationship
- Different colour spectra

It was interesting to see how different studies utilised image data in various different ways. The capability to use image data in multiple different ways was an advantage. Based on the research done it is expected that using image data for classification would be easier than making use of a different vision-based sensor. Giese, et al. (2017) was the only paper that attempted classification that did not use image data and concluded that although it was possible, it was more challenging to work with the radar data. The potential pitfall of using image data is the effect that light has on the data. Several studies noted that light affected the accuracy results of the model. This further means that classification is not possible using normal camera sensors at night if image data is used without the input of other sensors. A camera was chosen based on the advantages it presents and the number of studies that used cameras.

Past results based on image data either used an SVM or a neural network to perform the actual classification. Different types of neural network were used including PNNs, FNNs and CNNs. The studies that made use of image data all included a supervised learning model.

There were quite some inconsistencies when it came to the number of classes, dataset size and image sizes used during training. A bigger dataset does represent a truer representation of the classification capabilities of the method used. Increasing the image size increases the accuracy as noted in some of the papers, but the computational power of the computer used could be a limitation (Byl & Filitchkin, 2012).

All the state-of-the-art solutions had arbitrarily selected classes that were not selected based on some sort of standard. It seemed that most studies tried to capture the image data of as many classes possible. It was expected that some sort of international standard would be used to decide accurately between classes. The different terrain classes will be selected using the ISO 8608 (2016) standard, which none of the other studies did.

Based on the findings and the discussion it was decided to use the following methods going forward. The study will use a camera sensor that uses a neural network, more specifically a CNN applying a supervised learning model. The research question can be refined to:

Is off-road terrain classification possible by means of image data using a CNN classifier that uses the ISO 8608 (2016) standard to select the off-road terrain classes?

Chapter 3: Development of classification models

3.1 Introduction

This chapter outlines the creation of the different CNN models that were used to classify the off-road terrains. The previous chapter concluded that a CNN classifier would be used with image data. This chapter goes into further detail on how the CNNs were created by discussing the process, steps and setup. The chapter describes further research that was done into the sensor that would be used and how to work with the given image data.

The chapter discusses how the two main CNN models were developed for training: the first CNN model was created from scratch, and the second CNN model used a pretrained CNN (a feature extracting pretrained model). A multiple-class classifier as well as a binary model were created. The Python programming language was used to create the models. The Keras application programming interface was used, which functions on top of TensorFlow. Keras is a neural network library while TensorFlow is an open-source library. A live classifier was created to perform live classification.

The chapter gives a detailed outline of the architecture of the CNN models and explains the sensor and format of the data.

3.2 Data format and classification sensor

A Basler Dart 1 MP colour camera was used to capture the required image data. The camera captured video footage at 10 fps and images in an RGB format (Basler Information, 2021). Chapter 2 mentioned that many past research papers used image data in different formats, including HSV, greyscale, hue saturation lightness and other formats (Park, 2017).

Agarwal (2020) did a study to compare the different colour spaces as inputs to CNNs to determine which colour space is the best. The study considered the RGB, HSV, YCbCr, LAB ('L' is for lightness, 'A' is for green/ magenta, and 'B' is for blue/yellow), LUV and XYZ colour spaces. The study concluded that using the RGB colour spectrum delivered the best results, although the other colour spaces also did well except for the HSV colour space. Agarwal (2020) did note that the pretrained model was trained on RGB colour space images, which could have induced the bias. The conclusion was that any colour space could yield good results and the model should be trained according to the colour space (Agarwal, 2020).

RGB images have three layers with each layer consisting of $n \times m$ pixels. Each pixel value ranges from a value between 0 and 255 (but it could also range between 0 and 1 if the image uses a floating-point value), but instead of the intensity being a different greyscale, it is a different colour intensity. The combination of these matrices creates a coloured image (Upadhyay, 2017). Since it is possible to achieve good results with any colour space and because the sensor captured the images in the RGB colour space, this colour space was used in this dissertation. Figure 3.1 shows an example of a colour image and how the three dimensions consist of the RGB values ranging from 0 to 255.

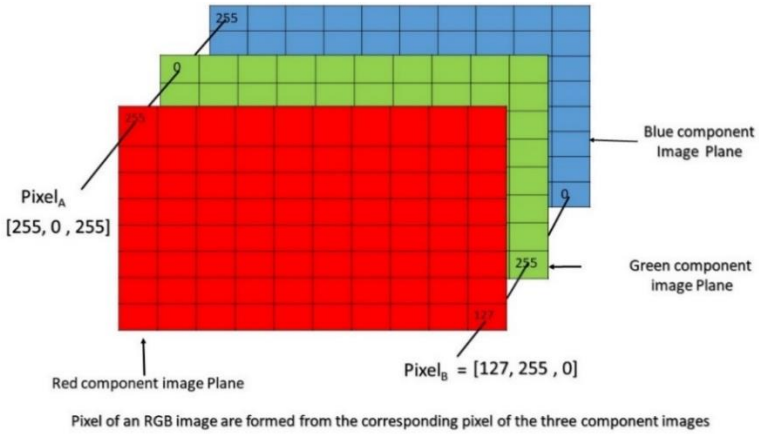

Figure 3.1: RGB image showing the three dimensions of a colour image (Upadhyay, 2017)

The RGB colour format was further chosen above the others because the camera image data was already in that format, and the study aimed to determine whether the same accuracy results could be achieved using a different colour format than past research papers.

3.3 Multiple-class classifier

This section describes the design of the multiple-class classifier. The model was designed in such a way that it automatically detects the number of off-road terrains (assuming more than two) and uses this information to update the CNN architecture. The CNN consists of a training phase and a prediction phase. The training phase requires the following components to successfully create a CNN classifier (Versloot, 2018):

- Convolutional layers
- Pooling layers
- Activation functions
- Flattening
- Dropout
- Dense layer
- Optimisation function
- Loss function
- Fitting

An in-depth study was done of these required components to create an off-terrain classifier.

3.3.1 Convolutional layer

The convolutional layer contains a set of filters whose parameters require learning using backpropagation. The number of parameters in each layer is the count of 'learnable' elements for a filter (parameters for the filter for that specific layer). The filter size is always smaller than the input array. These filters extract features about the input data with the goal of maintaining the features of the data while reducing the size of the output matrix (Xiang & Seeling, 2020).

The CNN filters can be for one-dimensional, 2D, and 3D CNNs. A one-dimensional CNN is usually used for time series data, a 2D CNN is used for image data, and a 3D CNN is mainly used with 3D image data such as magnetic resonance imaging data and computerised tomography scans (a series of X-ray images taken from different angles) (Verma, 2019).

A 2D CNN was used on the image data. It is called a 2D CNN because a 2D kernel slides along the height and width to extract features. Figure 3.2 shows the kernel and how it was dragged over the image in the two directions.

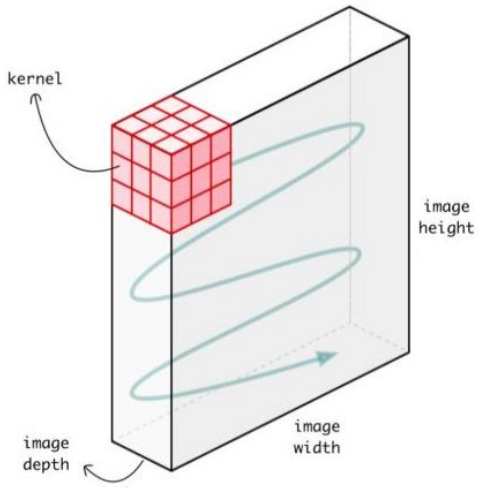

Figure 3.2: Kernel sliding over the image convolutional 2D (Verma, 2019)

Past research studies used a maximum of three convolutional layers, whereas the multiple-class classifier model created for off-road terrain classification also used three convolutional layers. Past results showed that using three convolutional layers gave optimal results. This is seen by the study done by Stolee and Wang (2018) as well as Sung, et al. (2010). A higher number of convolutional layers can also be used to determine whether the accuracy of the results will increase. The following information was required for each convolutional layer (Keras Help Documentation, 2020):

- Number of filters
- Kernel size
- Input image array size (required by the first convolutional input layer)

The proposed multiple-class classifier was designed in such a way that it would automatically detect whether an image was in the RGB spectrum or was just a greyscale image. The image input size was reduced to 150 × 150 pixels. Based on past research papers, it was difficult to determine the perfect input size of the image. The image input sizes ranged from as little as 15 × 15 pixels to 720 × 480 pixels. The pixel values were changed based on the results achieved. If the classification accuracy was low, the image size was increased. Increasing the size of the image subsequently increased the classification time required and the memory required by the computer. The same approach was followed by this dissertation where the base value of 150 × 150 pixels was used and if the results were inaccurate, an attempt was made to increase it.

The kernel is an $n \times n$ matrix made up of different values, which is also referred to as a filter. Generally, different filters have different arrangement values. The kernel is dragged along the input data of the image and corresponding values from the input matrix are multiplied by the corresponding kernel value. The sum of these values are added to create a new matrix. The goal is to reduce the size of the output data matrix while still retaining important feature information about the image. Figure 3.3 demonstrates how a kernel works.

The number of parameters increases quadratically with kernel size. This makes big convolution kernels less cost-efficient. It is often advised to use a larger kernel for the first convolutional layer as it will extract more features from a higher level, but is not set in stone (Chazareix, 2020). Popular CNNs used to have large kernels (such as the 11 × 11 size for AlexNet in 2012), but reduced over time without reducing the

performance of the models. The kernels were reduced all the way to 3 × 3 (Chazareix, 2020). A 3 × 3 kernel was therefore chosen over other kernel sizes.

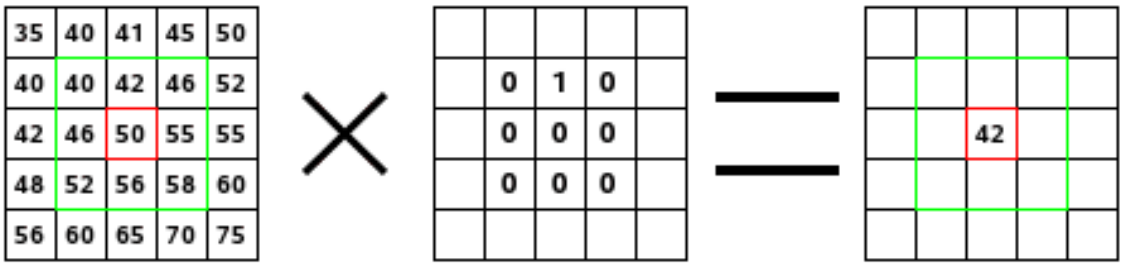

Figure 3.3: Convolution kernel (Baskin, et al., 2017)

The number of positions the kernel is moved every time is known as the stride. If the kernel is displaced by one pixel each time, the stride is equal to 1, and so on. A stride value of 1 was used for all three convolutional layers, which is also referred to as non-strided. A stride of one was selected because it reduces information loss. A popular approach is to use a stride value of 2 or smaller (Sahoo, 2018).

The fist convolution layer is responsible for low-level feature capturing such as edges, colour, and gradient orientation. Therefore, fewer filters are used. The first convolutional layer uses 32 filters, the second 64 filters, and the third 128 filters. The images are either valid padded or same padded. With valid padding, the size of the image remains the same, whereas with same padding, the size of the image is increased to allow the kernel to extract features at the edges of the image (Verma, 2019). The proposed model in this study used same padding to ensure that information about the off-road terrain would be extracted at the edges and that no important information would be lost.

3.3.2 Activation function

Given a particular input x, the activation function multiplies that input with a function, which is known as the activation function, containing a certain weight value and structure (formula). The output y is fed to the next convolutional layer and provides a certain probability between 0 and 1 as an output to predict an answer (Kizrak, 2019).

Activation functions are divided into two classes, namely linear and non-linear. The derivative of a linear function is just a constant. If the activation function is not applied, the output signal becomes a simple linear function. Linear activation functions are only single-grade polynomials. A non-activated neural network acts as a linear regression with limited learning power. The world we live in is referred to as a stochastic world where nothing is constant or linear. Road classification requires a non-linear activation function (Kizrak, 2019). Popular non-linear activation functions include:

- Sigmoid
- Rectified linear unit (ReLU)
- Tanh
- Softmax
- Exponential linear unit
- Scaled exponential linear unit
- Softplus
- Softsign
- Linear

Each convolutional layer and dense layer generally consist of an activation function. Non-linear activation are usually used to add non-linearity to the system. The proposed model uses two activation functions: the first activation function is used throughout each convolutional layer (input and hidden layers) and the first dense layer. The second activation function is used in the final dense layer (output layer). The most popular and effective activation function for image classification is the ReLU activation function. The ReLU activation function uses the following formula (Brownlee, 2021):

$$R(z) = \max(0, z) \tag{3.1}$$

Where $R(z)$ is the activation function output value and z is a coordinate.

The first input layer is a 2D array containing values between 0 and 255. The hidden layers only contain positive values. The ReLU function returns 0 if it receives any negative input and for any positive value x, it returns x multiplied by the gradient. The ReLU activation function is applied to image data to increase non-linearity in the image. Images are naturally non-linear due to the transitions between pixels, borders, colours, etc. The purpose of the ReLU activation function is to further break up linearities created during the convolution operation and is the advised activation function to use for input and hidden layers (Brownlee, 2021). Figure 3.4 shows a graph of the ReLU activation function.

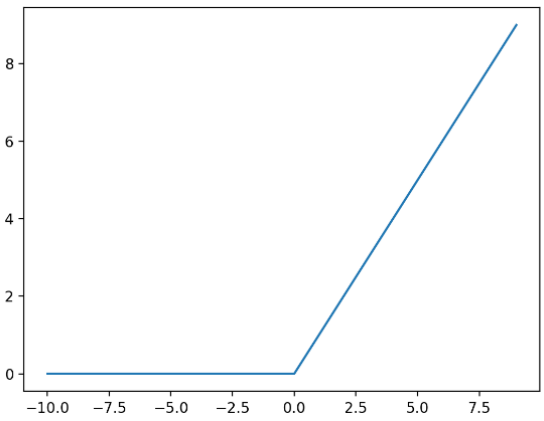

Figure 3.4: ReLU activation function (Brownlee, 2021)

The softmax activation function is used for the final dense layer. Softmax converts a real vector to a vector of categorical probabilities. The output vector is in the range from 0 to 1 and sums to 1. Softmax activation is used in the final dense layer because the result could be interpreted as a probability distribution. Softmax is used for multiple class classification problems where class membership is required on more than two class labels (Kizrak, 2019).

The softmax function uses the following formula (Brownlee, 2021):

$$sm(x) = \frac{e^{x_i}}{\sum_k e^{x_k}} \tag{3.2}$$

Where $sm(x)$ is the softmax activation function output values, e is a mathematical constant, x is the input vector, i is the vector number, and k starts at 1.

Figure 3.5 shows how the input data provided to a softmax function is transformed to a probability value.

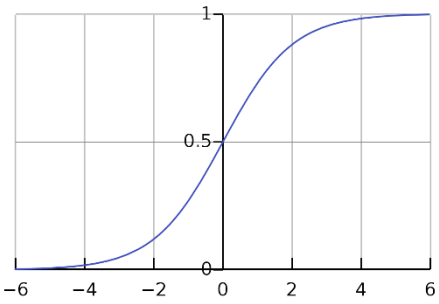

Figure 3.5: Softmax activation graph (Brownlee, 2021)

3.3.3 Pooling layer

Pooling is similar to the convolution layer using kernels. The purpose of the pooling layer is to further reduce the size of the matrix to decrease the computational power required. Pooling is useful for extracting dominant features that are rotational and positional invariant to maintain the practical training of the model. There are two types of pooling, namely max pooling and average of all values (Keras Help Documentation, 2020).

With max pooling, the maximum value within the kernel across that specific portion of the matrix is extracted. The kernel is moved across all the pixels with a specific stride value. The maximum value is taken from each, whereafter a new matrix of a smaller size is constructed while the essential features are extracted (Keras Help Documentation, 2020).

Max pooling performs as a noise suppressant. Since max pooling extracts the maximum values instead of the average values, it further creates non-linearity in the data. Max pooling layers give sharper images while retaining the most prominent features of the feature map (Sharma, n.d). Max pooling is chosen over average pooling. A non-strided 2 × 2 kernel is used. A smaller kernel size is selected to reduce the number of parameters and a non-strided approach is used to avoid missing important feature information. The pooling layer is added after each convolutional layer. Max pooling is preferred over average pooling because max pooling better extracts low-level features that include edges and points. Such features form an important part of distinguishing between different terrains (Basavarajaiah, 2019). Figure 3.6 shows the difference between max pooling and average pooling.

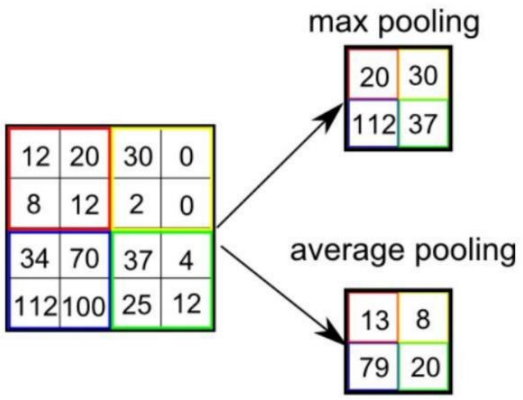

Figure 3.6: Max pooling vs average pooling (Keras Help Documentation, 2020)

3.3.4 Flattening

After convolutional layers and pooling layers have been applied to the image, the next step is to flatten the image. Figure 3.7 shows how flattening works. The flattening layer of the CNN takes the pooled feature matrix and converts it to a vector (SuperDataScience Team, 2018).

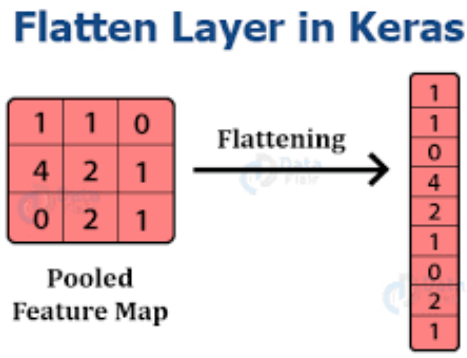

Figure 3.7: How flattening works in a CNN (Keras Help Documentation, 2020)

The flattening layer reduces the image dimensionality without losing important features or patterns.

3.3.5 Dropout

The term 'dropout' refers to dropping out both hidden and visible units in a neural network. During dropout, certain units are ignored. These ignored units are not considered during a particular forward and backward pass. At each training stage, individual nodes are either dropped out of the net with a probability $1 - p$ or kept with a probability p. This reduces the network size (Budhiraja, 2016).

Dropout is performed to prevent overfitting. A fully connected layer occupies most of the parameters, which implies that the neurons develop co-dependency among one another during training. This reduces the power of the individual strength of each neuron, leading to overfitting of training data (Budhiraja, 2016).

The dropout layer in the model randomly sets input units to 0 with a probability of 0.5. The rate is a floating value between 0 and 1 and is the fraction of input units that are dropped. Inputs not set to 0 are scaled up by $\frac{1}{1-rate} = \frac{1}{1-0.5} = 2$ such that the sum over all inputs is changed. The dropout layer in the model is only applied when training is set to 'true'. This ensures that no values are dropped during inference (Budhiraja, 2016).

3.3.6 Dense layer

A dense layer is the implementation of the equation:

$$output = activation(dot(input, kernel) + bias) \quad 3.3$$

Equation 3.3 is the dot product between the input image and the kernel with an added bias. The most important input given to the dense layer is the unit, which is the size of the output vector provided by the dense layer. For example, if one wants to classify between five images, the last dense layer has a unit value equal to 5. This gives an output of the five possible values an image can be classified as (Ferdinand, 2020).

The model consists of two dense layers. The first dense layer calculates a dimensionality output space vector of size 512. The second dense layer has an output space vector equal to the number of classes. The model automatically calculates the number of classes present during preprocessing and uses this value to update the dimensionality output space vector of the second dense layer.

A ReLU activation function is used in the first dense layer and a softmax activation function is used in the second dense layer. For both dense layers, no bias initialiser is used. No kernel, bias or activity regularises are used. No kernel and bias constraints are applied.

3.3.7 Optimisation function

The goal of an optimiser algorithm is to reduce the computation time to find the weights for the neural network. An optimiser algorithm helps to minimise the objective function. The objective function is simply a mathematical function dependent on the model's internal learnable parameters that are used to compute the target values. In this case, it is the classification of off-road profiles and the condition of each off-road profile during a multilabel classification (Walia, 2017).

Popular optimisation functions include (Walia, 2017):

- Stochastic gradient descent with momentum
- AdaGrad
- Adadelta
- RMSprop
- Adam
- AdaMax
- Nadam
- AMSGrad

The Adam optimisation function is used. Adam is a replacement optimisation algorithm for stochastic gradient descent for training deep-learning models such as image data. Adam combines the best properties of the RMSprop and AdaGrad algorithms to provide an optimisation algorithm capable of handling sparse gradients on noisy problems as found with image data (Walia, 2017).

3.3.8 Loss function

The model uses a cross-entropy loss function on the one-hot encoded output. One-hot encoding represents the categorical data as binary values. The categorical values are mapped to integer values. Each integer value is represented as a binary vector that consists of 0 values except for the index of the integer, which is marked with a 1. For a single image, the cross-entropy loss looks as follows (Brownlee, 2019):

$$-\sum_{c=1}^{M}(y_c \cdot log\hat{y}_c) \qquad 3.4$$

Where M is the number of different road terrains, and $\hat{y}_c$ is the model's class prediction for the terrain. The class labels are one-hot encoded and $M \times 1$ is a vector of ones and zeros. y_c is either 0 or 1. In Keras, the loss function is called 'categorical_crossentropy', which is the loss function for performing multiple-class classification.

3.3.9 Fitting

The final part of the CNN model is the actual training. Important inputs during the training phase are the batch size, epochs used, and validation split. The literature survey results did not present any information about what these parameters are. The batch size is a hyperparameter that defines the number of samples to work through before the model updates the internal parameters. It is difficult to determine the ideal batch size values, but good options are 32, 64, 128, 256, or higher (Shen, 2018). The multiple classification model used a batch size of 128.

The number of epochs of the model is a hyperparameter that defines the number of times the learning algorithm works through the entire dataset (Sharma, 2017a). The larger the epoch value, the better the training results are; however, if the epochs use it too much, overfitting occurs. The training results will be excellent, but the validation accuracy will not. During testing and training, the number of epoch values will be increased until overfitting occurs. The loss and accuracy results will be monitored to determine the point at which the loss value increases and the accuracy decreases.

Training data is split into training data and validation data. A validation split is as follows: around 80/20% and 90/10% for very large datasets, and around 70/30% and 60/40% for smaller datasets (Kumar, 2020). A validation split of 70/30% was used in this study. A larger dataset than other studies was captured but remained small.

3.3.10 Backpropagation

Backpropagation is a vital part of any neural network. Backpropagation updates the weights of a neural network based on the error rate obtained in the previous epoch. Effective tuning of the weights allows the CNN to reduce error rates and makes the model reliable by increasing its generalisation. Backpropagation stands for backward propagation of errors. Backpropagation is a standard method for training artificial neural networks, which helps to calculate the gradient of a loss function with respect to all the weights calculated during the feed-forward step in the network (Jaumier, 2019).

Jaumier (2019) did an in-depth study that explains the mathematics behind backpropagation. Although it is important to understand the function of backpropagation, most CNN libraries perform backpropagation without the user having to understand how it works (Jaumier, 2019).

3.3.11 Model summary

The model further captures how long the model trains on different epoch sizes, activation functions, and optimisations functions. The time can be used to compare the effectiveness of the training sequence of the model. Figure 3.8 shows the model hierarchy of the multiple-class classifier model.

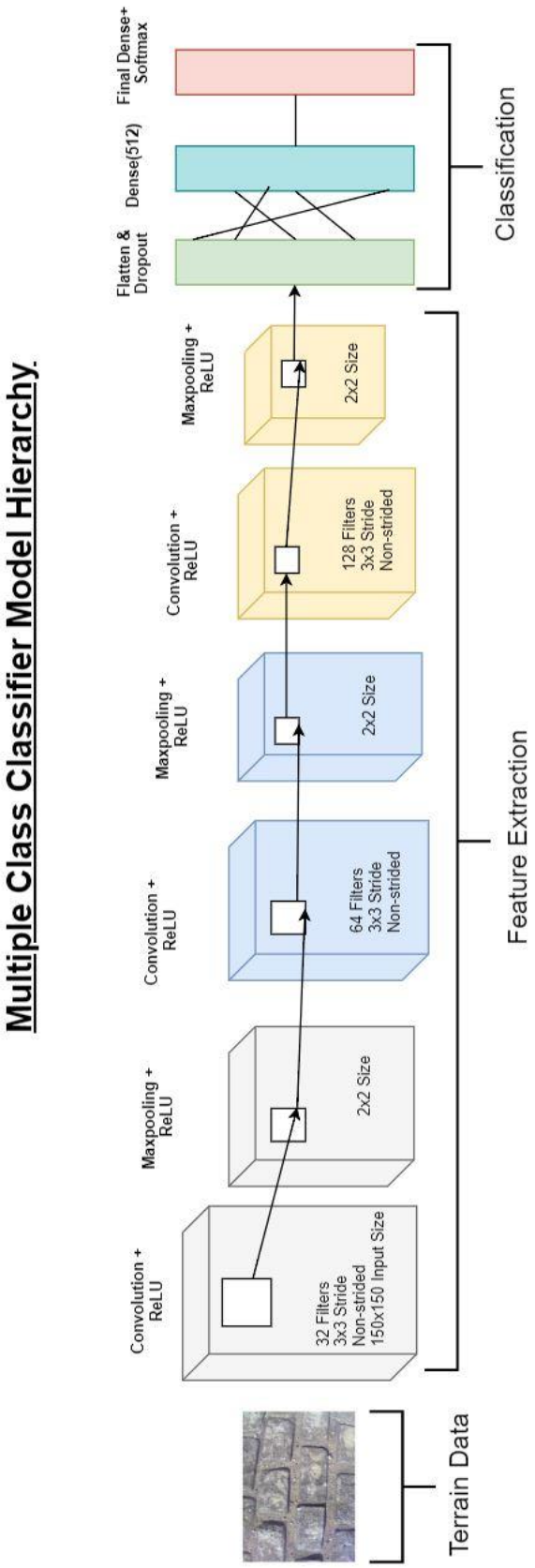

Figure 3.8: Multiple-class classifier model hierarchy

3.4 Binary class classifier

A binary classifier classifies between two different off-road terrain classes, whereas a multiple-class classifier classifies between three or more classes. A binary classifier was used to isolate two classes when a multiple-class classifier was unable to classify between two classes effectively. The CNN architecture can be changed to determine whether a different CNN setup would improve the classification results between the two classes.

The same CNN architecture was used as with the multiple-class classifier with two changes: a different loss function was used, and a different activation function was used in the final dense layer. The binary cross-entropy loss function was based on the cross-entropy loss function discussed in Section 3.3.8. The function is used when there are only two classes. The sigmoid or logistic activation function uses the following formula (Godoy, 2018):

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.5}$$

Where $S(x)$ is the sigmoid activation function output value, e is the mathematical constant, and x is the input vector (Sharma, 2017b).

Figure 3.9 shows the sigmoid activation function. The upper limits of the y-axis are between 0 and 1. The input data is either classified as a 1 or a 0, hence it is binary. If the output value is between 0.5 and 1, the output result is considered to be 1. If the output value is between 0 and 0.5, the output value is considered to be 0. A threshold is applied to the output results to decide how close the answer should be to 0 or 1 to be classified as 1 or 0.

The rest of the model is the same. Three convolutional layers are used with ReLU activation and max pooling. The model has a dropout layer, flattening layer, and two dense layers. The final dense layer consists of two outputs, namely 0 and 1.

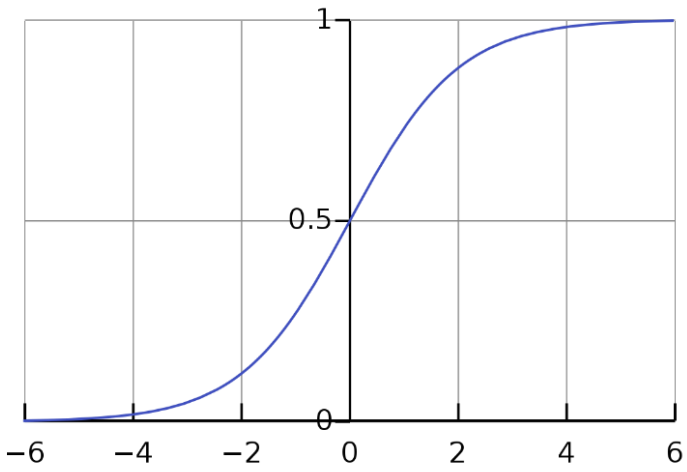

Figure 3.9: Sigmoid activation function (Brownlee, 2021)

3.4.1 Model summary

Figure 3.8 show the model summary of the binary class classifier. The only difference is that the final dense layer uses a sigmoid activation function instead of a softmax activation function.

3.5 Pretrained convolutional neural network model

Pretrained models have the benefit of being trained on very large database sets and learning to identify more unique features. These features, however, might not have applied directly to the problem at hand, which was classifying off-road terrain images. Pretrained models are simple to use and can achieve good or even better accuracy results than a smaller model that is trained from scratch (Shao, 2019).

Pretrained models can be used in two ways – either as feature extraction or fine-tuning. Fine-tuning involves unfreezing a few of the top layers of a frozen model base used for feature extraction, and jointly training both the newly added part of the model and the unfrozen top layers. It is called fine-tuning because it slightly adjusts the more abstract representations of the model being reused to make it more relevant to the specific problem at hand (Chollet, 2018).

Feature extraction uses the representations learned by a previous network to extract features from new samples. These features are run through a new classifier, which is trained from scratch. Convolutional networks for image classification comprise two parts: starting with a series of convolution and pooling layers and subsequently ending with a densely connected classifier. The first part is referred to as the convolutional base of the model. Feature extraction involves taking the convolutional base of the previously trained network and running the new data through it to train a new classifier on top of the output (Chollet, 2018).

Figure 3.10 shows a general approach to the application of feature extraction. The trained convolutional base is first trained on a large dataset (for example the ImageNet dataset consisting of 1.6 million images and 1 000 classes). Thereafter, the model is saved and applied to a new problem, such as terrain classification. The convolutional base is kept the same, but the dense layer at the end is changed.

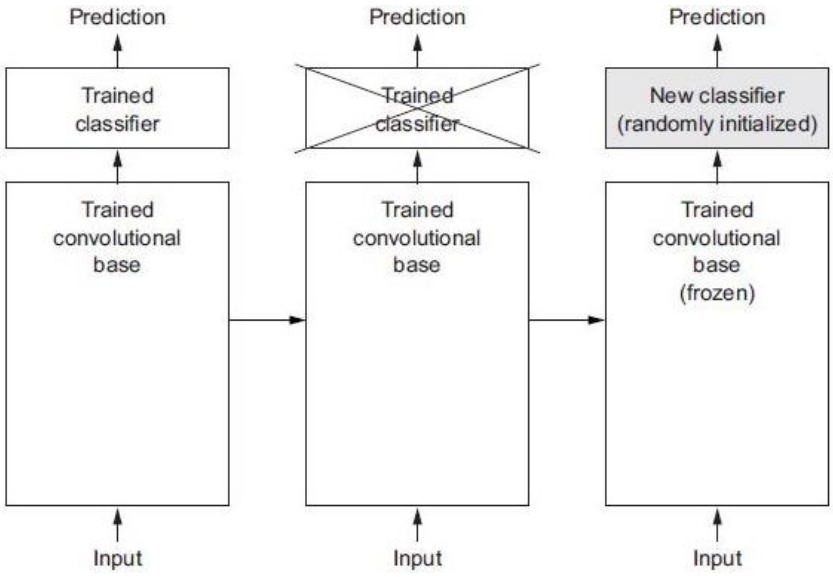

Figure 3.10: Example of the application of feature extraction (Chollet, 2018)

The convolutional base is likely to be more generic and, therefore, more reusable. The feature maps of a CNN are presence maps for generic concepts over a picture, which are expected to be useful regardless of the computer vision problem at hand. The representations learned by the classifier are specific to the set of classes on which the model was trained. It only contains information about the present probability of the given class in the entire picture. Representations found in densely connected layers no longer provide any information about where objects are located in the input image. The dense layer gets rid of the notion of space, whereas convolutional feature maps still describe the object's location. Layers that come earlier in the model extract local, highly generic feature maps, whereas layers that are higher up extract more abstract concepts. If the dataset differs substantially from the dataset used to train the original model, it is better to only freeze the first few layers instead of the entire convolutional base (Chollet, 2018).

A feature extracting pretrained model was used. Keras consists of several pretrained models built into the Keras library. Table 3.1 shows a list of some available pretrained models. The entire architecture of each pretrained model remains the same. The top performing model was chosen, namely the Xception model. Only the Xception model was used and not any of the others.

The following layers were added to the end of the pretrained model to be able to do off-road classification: A flattened layer was added followed by a 50% dropout layer to help prevent overfitting. The dropout layer was followed by two dense layers. The first dense layer had an output value of 128 and the final dense layer had an output value equal to the number of off-road terrain classes.

Table 3.1: Different pretrained models (Keras Help Documentation, 2020)

Model	Size (MB)	Top 1 accuracy	Top 5 accuracy	Parameters	Depth
Xception	88	0.790	0.945	22 910 480	126
VGG16	538	0.713	0.901	138 357 544	23
VGG19	549	0.713	0.900	143 667 240	26
ResNet50	98	0.749	0.921	25 636 713	–
ResNet101	171	0.764	0.928	44 701 176	–
ResNet50V2	98	0.760	0.930	25 613 800	–
ResNet101V2	171	0.772	0.938	44 675 560	–
InceptionV3	92	0.779	0.937	23 851 784	159
MobileNet	16	0.704	0.895	4 253 865	88
MobileNetV2	14	0.713	0.901	3 538 984	88

The top 1 and top 5 accuracies refer to the model's performance on the ImageNet validation dataset. Depth refers to the topological depth of the network. Depth includes activation layers and batch normalisation layers (Keras Help Documentation, 2020).

3.6 Conclusion

The different required models were created:

- Multiple-class classifier
- Binary classifier
- Pretrained classifier

The models applied research gathered from past papers that used neural networks as reference points. The goal of these models were to improve and expand on past results. Full details were provided of how the

models were built and the entire architecture of the models was discussed. Instead of using two convolutional layers, three layers were used. Pretrained models were also considered. Research was done to ensure that the most suitable activation and optimisation functions were chosen based on what most studies used. The models can omit the effect of unwanted noise, which improves the legitimacy of the classification results obtained compared with data classification that consists of noise.

The next chapter discusses how data of the different off-road terrains was captured and how the different terrain classes were identified.

Chapter 4: Data capturing of off-road terrains and class creation

4.1 Introduction

This chapter discusses the method used to capture the profiles of the different off-road terrains. The experimental setup and the method used to capture the data are described. The chapter explains how the data was broken up into several different classes based on the material of the road and the road roughness. Data capturing was conducted at the Gerotek Test Facilities in South Africa. After the different off-terrains were established, the ISO 8608 (2016) standard was used to distinguish accurately between the different off-road classes that would be used during the classification process.

4.2 Experimental setup

Two Basler Dart 1 MP colour cameras were used to capture data. The cameras were mounted on a Land Rover Defender. There were two different camera setups with one camera capturing images at a relatively small angle from the horizon and the second camera facing downwards at a close to 90° angle with the road surface. The first camera was tilted between 5° and 15° from the horizon. These angles varied during testing and were referred to as forward facing (5° and 15° initial setup) and downward facing (90° initial setup). Figure 4.1 shows the forward-facing camera setup with the tilted angle A.

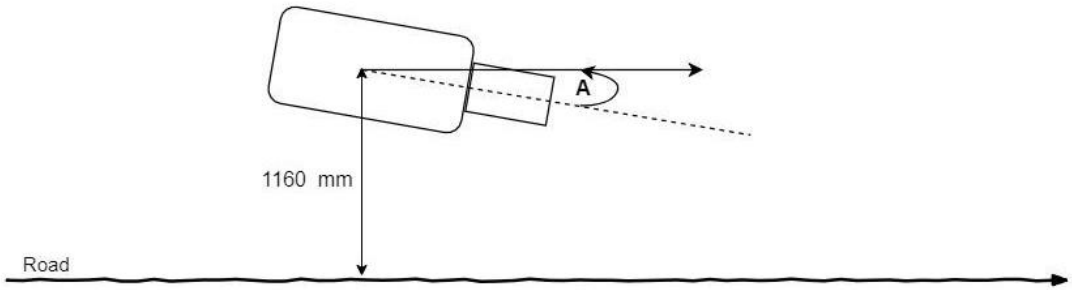

Figure 4.1 Forward-facing camera setup

The camera data was not the only data captured during the data capturing process. The vertical acceleration, GPS speed and other IMU measurements such as pitch rate, roll rate and yaw rate were also captured together with the lidar and radar data. However, it is important to note that these data points were not used in this project as the project only made use of the camera data.

Figure 4.2 shows how the downward-facing camera was set up with an angle, A, close to 90° with the horizon (horizontal solid line). The camera was placed around 630 mm from the ground. Figure 4.3 shows how the forward-facing camera was placed on the vehicle. The forward-facing camera was about 1 160 mm from the ground attached at bonnet height. The main reason for having two camera setups was to be able to compare the results of each. The goal was to establish whether a particular setup yields better results

and to analyse the advantages and disadvantages of each setup. Figure 4.4 shows how the downward-facing camera was placed on the test vehicle.

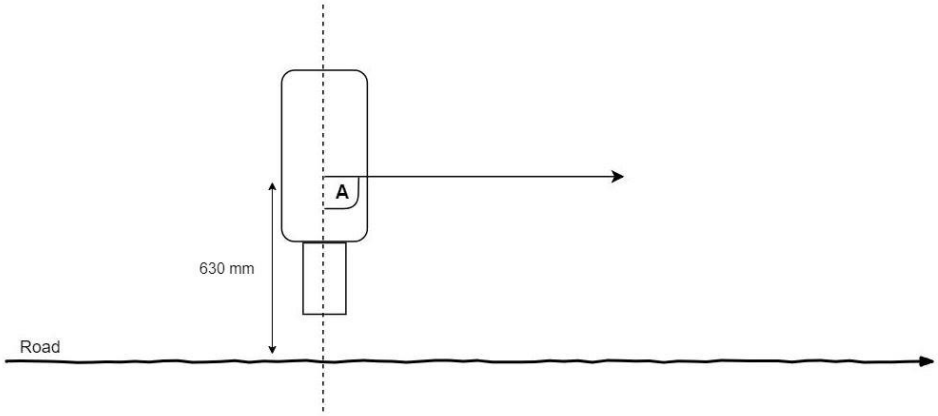

Figure 4.2: Downward-facing camera setup

Figure 4.3: Forward-facing camera setup on test vehicle

Figure 4.4: Downward-facing camera setup on test vehicle

Figure 4.5 shows both camera setups. The forward-facing camera was placed higher and captured data further away from the vehicle. The camera captured a larger part of the terrain than the downward-facing camera. As stated, the downward-facing camera was placed closer to the ground at a 90° angle. The downward-facing camera captured a smaller section of the off-road terrain.

Figure 4.5: Experimental setup showing both forward- and downward-facing camera setups

4.3 Off-road terrain classes

The next step was to capture data of the different off-road terrain classes, which was divided in two ways. The first method distinguished between the different terrain materials (tar, gravel, concrete etc.) and the second distinguished between the different types of terrain roughness value. Becker (2008) performed profiling of rough terrains at the University of Pretoria. The study used the Can-Can machine developed by the Vehicle Dynamics Group to capture the vertical displacement of different terrains at the Gerotek Test Facilities (Becker, 2008).

The Can-Can machine is a road profiler that has 90 arms spaced 40 mm apart. The height of the rear beam is adjustable between 160 mm and 300 mm. The track width can be set at 1.5 m, 3 m, or 4.5 m. The Can-Can machine measures the specific road profiles used during vehicle simulations accurately or, in this case,

determines the different road profiles from which the road roughness value can be calculated using the ISO 8608 (2016) standard. The Can-Can machine moves at 1 km/h with the rear beam arms designed with a maximum of 3 mm deflection. The vertical displacement at the tip of the arms are calculated measuring the angle at the pivot point and is done making use of high precision single turn potentiometers. The resolution of the potentiometers are equal to 0.12° which is equal to a vertical displacement of 0.98 mm. Figure 4.6 shows a photo of the Can-Can machine (Becker, 2008).

The following five different road materials were captured during the data capturing phase:

- Sand
- Gravel
- Boulder/rock
- Grass
- Concrete

Figure 4.6: Can-Can road profiling machine (Becker, 2008)

4.3.1 Sand

Sand is a granular material consisting of finely divided rock and mineral particles. Although sand comprises various compositions, it is defined by its grain size. The grain sizes of sand are smaller than gravel, but coarser than silt. The composition of sand varies, but it is mostly referred to as silica, silicon dioxide, or SiO_2. Sand grain sizes vary between 0.05 mm and 2.00 mm (Yun, et al., 2007).

Figure 4.7 shows an example of the sand road profile. The image on the left was captured using the downward-facing camera and the image on the right was captured using the forward-facing camera. The Can-Can machine could not be used on the sand track because the sand track constantly changes. This track is not used for road roughness calculations and was not considered as a one of the off-road terrains.

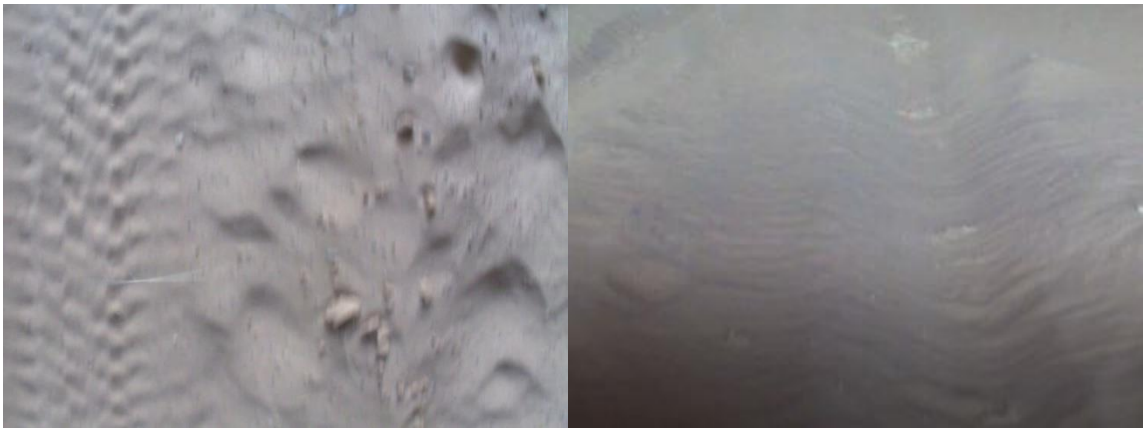

Figure 4.7: Sand road profile example of downward- and forward-facing data

4.3.2 Gravel

Sand and gravel differ in the size of their particles. Sand particles typically vary from 0.05–2 mm, whereas gravel particles range from 4.75–75 mm. Gravel particles are larger than sand, but smaller than boulders. Both sand and gravel have a brown colour (Yun, et al., 2007).

Figure 4.8 shows an example of a gravel road profile. The image on the left was captured using the downward-facing camera and the image on the right was captured using the forward-facing camera.

Figure 4.8: Gravel road profile example of downward- and forward-facing data

4.3.3 Boulder/rock

If the size of a rock fragment exceeds 256 mm, it is considered a boulder. Pieces with smaller sizes are considered rocks. The next terrain consisted of large rock fragments ranging around 250 mm. This road profile consisted of boulders and rocks (Yun, et al., 2007).

Figure 4.9 shows an example of a boulder/rock road profile. The image on the left was captured using the downward-facing camera and the image on the right was captured using the forward-facing camera.

Figure 4.9: Boulder/rock road profile example of downward- and forward-facing data

4.3.4 Grass

Compared with sand, gravel and boulder/rocks, grass is usually green and has a different shape and size than the previously mentioned classes. Figure 4.10 shows an example of a grass road profile. The image on the left was captured using the downward-facing camera and the image on the right was captured using the forward-facing camera.

Figure 4.10: Grass road profile example of downward- and forward-facing data

4.3.5 Belgian paving

The next road profiles were all made of concrete, but they had different road profile properties. To differentiate between these profiles, the Can-Can machine developed by the Vehicle Dynamics Group was used. The study performed by Becker (2008) profiled rough terrains. The study applied different methods and using the Can-Can machine provided good and consistent results.

Figure 4.11 shows an example of a Belgian paving road profile. The image on the left was captured using the downward-facing camera and the image on the right was captured using the forward-facing camera.

Figure 4.11: Belgian road profile example of downward- and forward-facing data

Figure 4.12 shows the profiling of Belgian paving. Belgian paving has a vertical displacement value between 80 and −100 mm and is classified as a class D road (Becker, 2008).

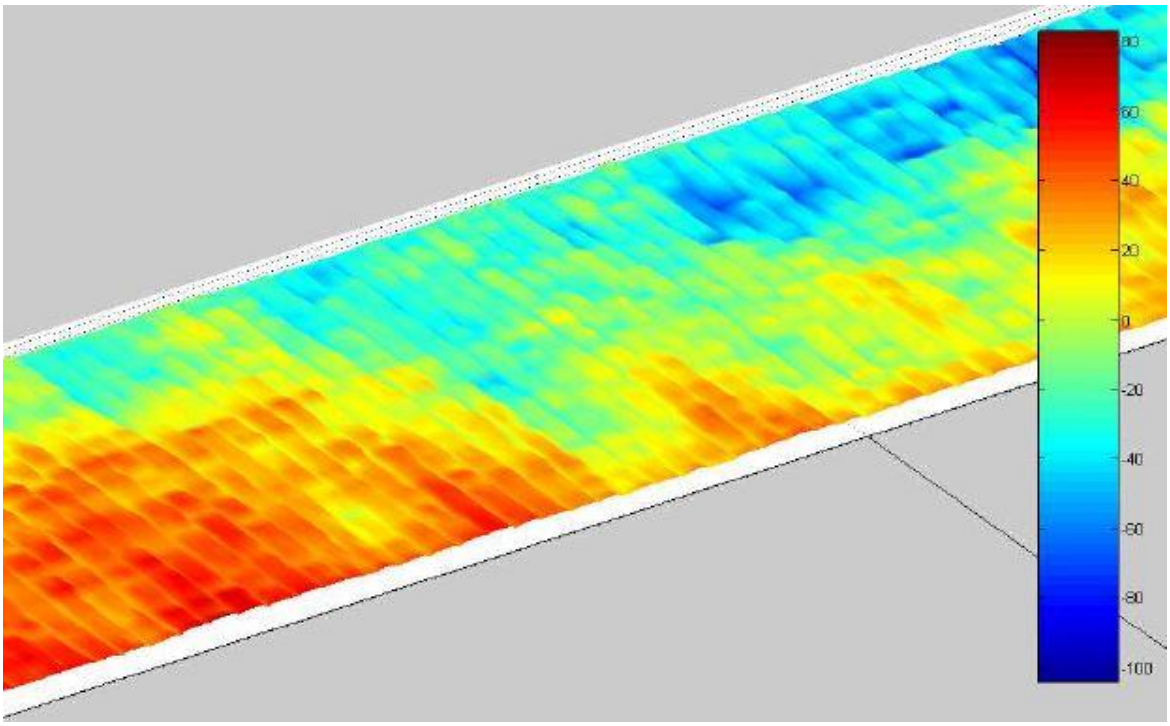

Figure 4.12: Can-Can profile of Belgian paving (Becker, 2008)

4.3.6 Parallel and angled corrugations

Parallel and angled corrugation roads were used to simulate a corrugated gravel road. The road profile consisted of 25 mm bumps and was made from concrete to ensure repeatability of tests. The corrugated roads were 4 m wide and 100 m long (Becker, 2008). Figure 4.13 shows an example of a corrugated road profile. The image on the left was captured using the downward-facing camera and the image on the right was captured using the forward-facing camera.

Figure 4.13: Straight corrugated road profile example of downward- and forward-facing data

The Can-Can machine profiling results ranged between 20 mm and 26 mm vertical RMS. Figure 4.14 shows the profiling of the straight corrugated road profile (Becker, 2008).

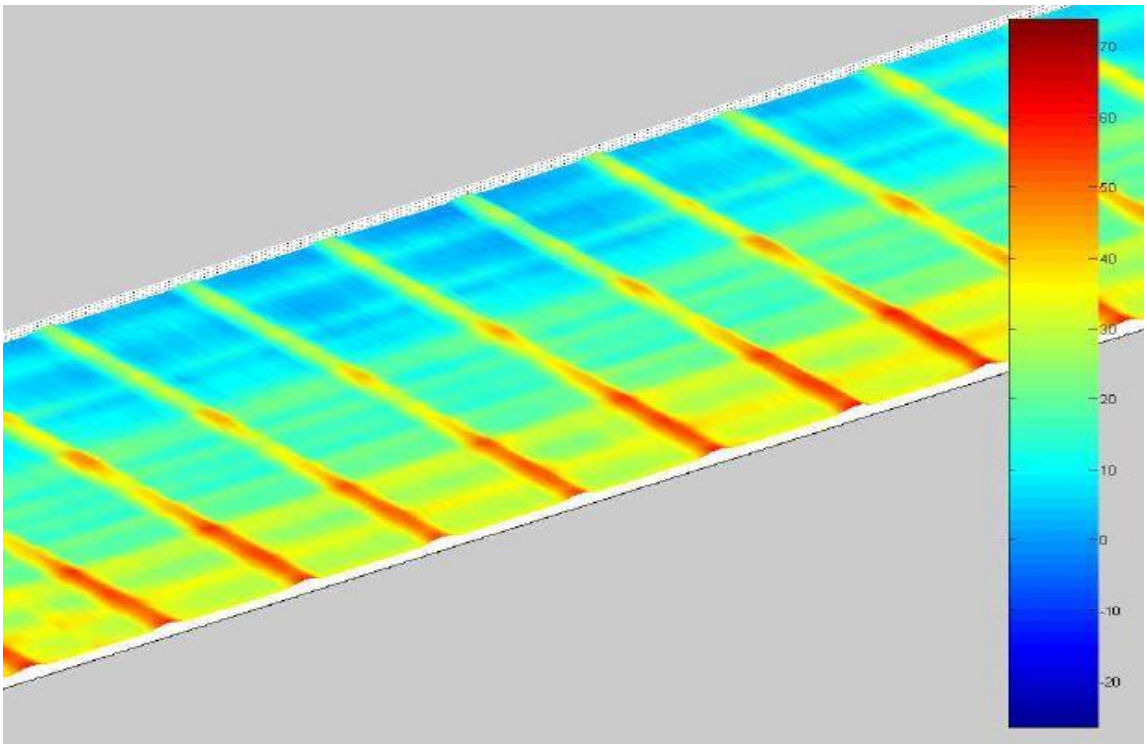

Figure 4.14: Can-Can profile of straight corrugated road (Becker, 2008)

The straight corrugated road was compared with a class A, class D, and class H road (Becker, 2008). The angled corrugated road had the same vertical RMS value as the straight corrugated road. When a vehicle drives over a straight corrugated road, the left and right wheels move up and down in sync. No lateral movement is induced by the vehicle – assuming the vehicle drives straight over the road and not at an angle. With an angled corrugated road, the left and right wheels of the vehicle do not move vertically in sync, and lateral movement of the vehicle's centre of gravity is present. Figure 4.15 shows an example of the downward- and forward-facing data of the angled corrugated road profile.

Figure 4.15: Angled corrugation road profile of downward- and forward-facing camera data

Figure 4.16 shows the Can-Can profile data of the angled corrugated road profile. The units of the values in the figure are in millimetre.

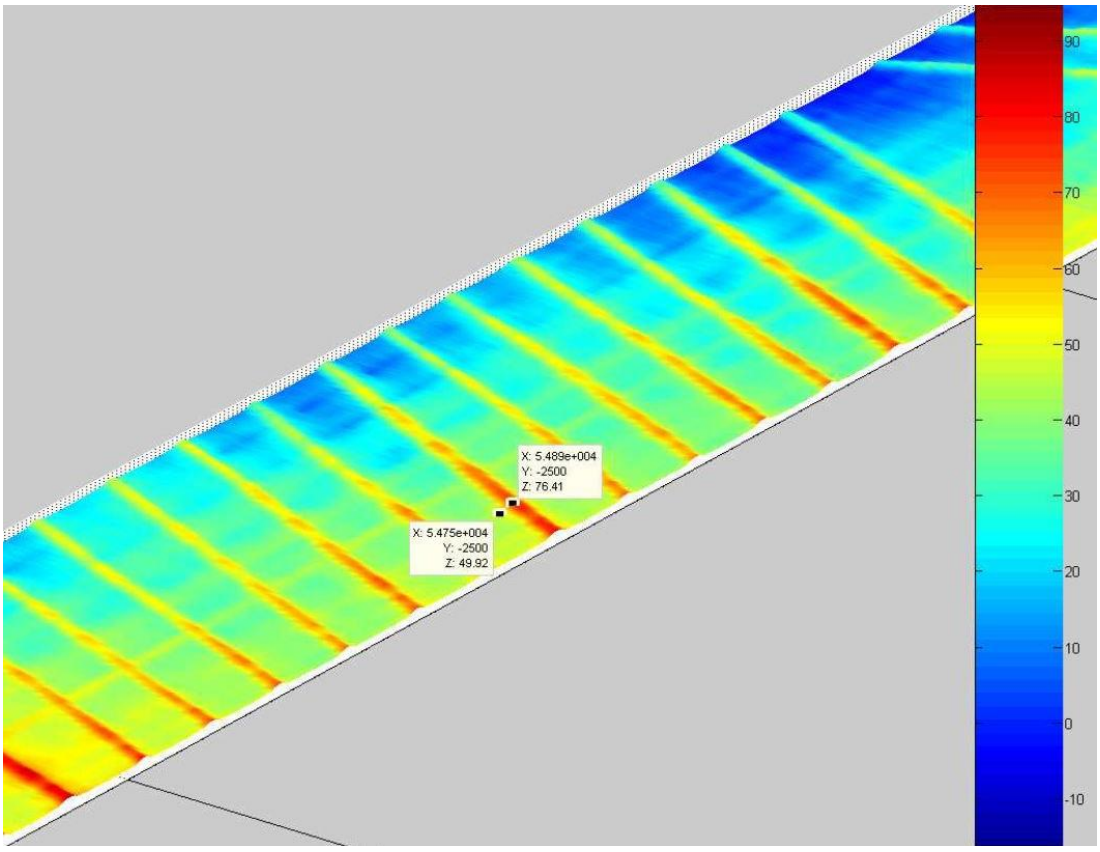

Figure 4.16: Can-Can profile of angled corrugated road with two location points in millimetre (Becker, 2008)

4.3.7 Fatigue track

The next road profile captured was the fatigue track. The goal of the track is to accelerate the fatigue life of a vehicle's suspension and chassis to test the durability thereof. The induced vertical RMS values of the fatigue track were higher than those of the Belgian paving road. Figure 4.17 shows an example of the downward- and forward-facing data of the fatigue road.

Figure 4.17: Fatigue track profile of downward- and forward-facing camera data

Becker (2008) concluded that if the spatial frequency of the profile is between 0.5 cycles/m and 10 cycles/m, the DSD data exhibits a class D road. Below 0.5 cycles/m and above 10 cycles/m, the road profile's DSD data is lower, suggesting a smoother track. The vertical RMS obtained values up to 60 mm at certain points along the track. Figure 4.18 shows how the Can-Can machine profiled the road of the fatigue handling track (Becker, 2008).

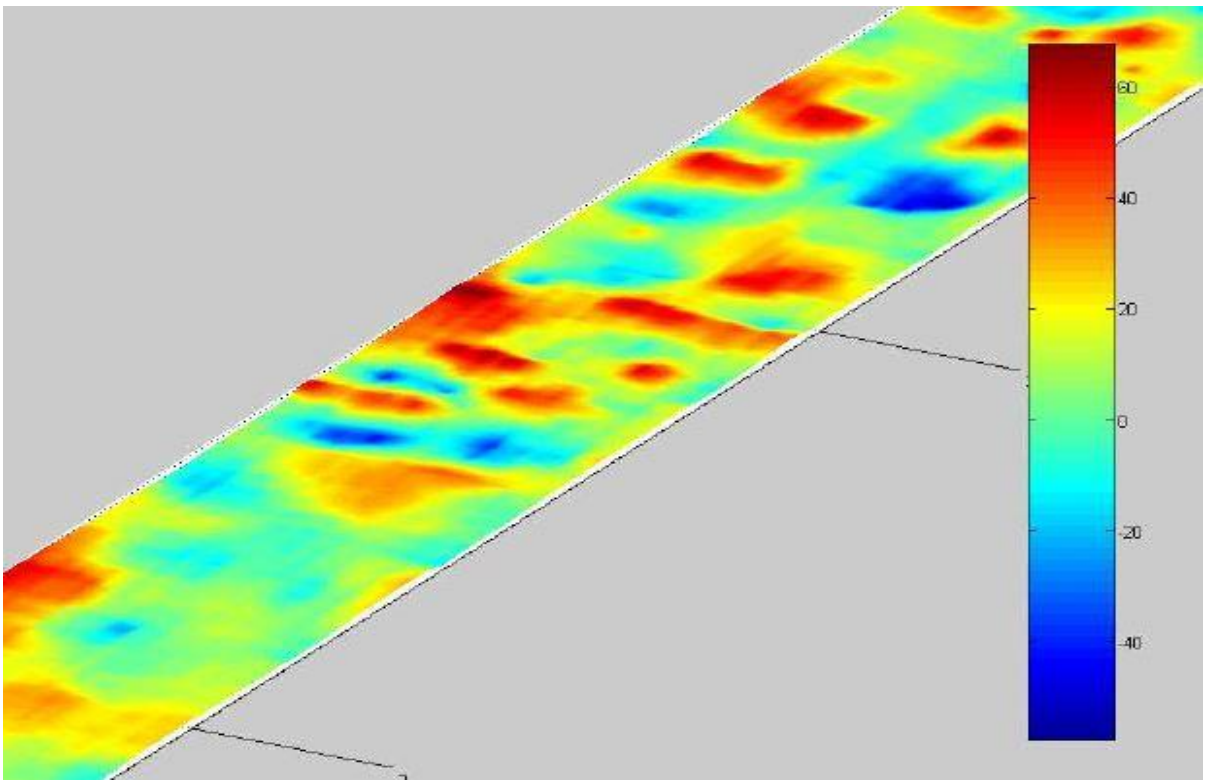

Figure 4.18: Can-Can profile of the fatigue handling track (Becker, 2008)

4.3.8 Ride and handling track

The purpose of the ride and handling track is to test the ride comfort, driveline endurance, and handling characteristics of a vehicle. Some of the properties of the track include:

- Up and down hills
- Constant radius turns

- Decreasing radius turns
- Positive and negative turns
- High- and low-speed corners
- Sections for low- and high-mobility vehicles

Figure 4.19 shows that it is more difficult to see the grooves in the forward-facing data than the downward-facing data.

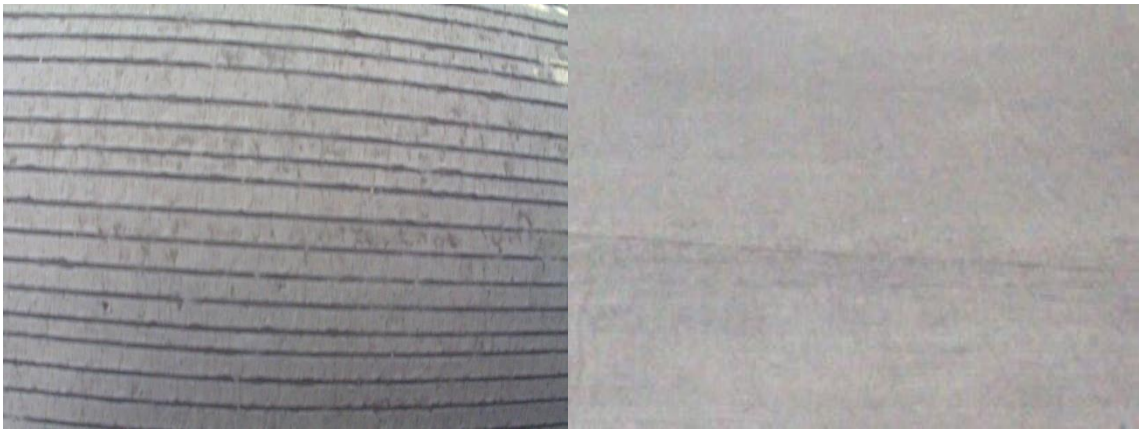

Figure 4.19: Dynamic handling road profile of downward-facing and forward-facing camera data

The road is 4.2 km long, and due to the large overall climb of the road, the small instantaneous vertical changes within the road profile are not clear. The DSD results classified the road as a class D road for spatial frequencies of 1 cycle/m and higher. The road was classified between class A and D for lower spatial frequencies (Becker, 2008).

4.3.9 Rough track

The rough track is used to evaluate the rough terrain mobility as well as the structural endurance of all-terrain vehicles. The tracks test vehicle durability under accelerated conditions and relative movement between the vehicle's body and cabin. The chassis and wheels are also evaluated together with ride comfort and interior noise levels. Vehicle speeds above 20 km/h are seldom achieved. The downward- and forward-facing image data can be seen in Figure 4.20.

Figure 4.20: Rough track road profile of downward-facing and forward-facing camera data.

The DSD data classified the road as a class H road (Becker, 2008).

4.4 Off-road terrain classes based on ISO 8608 international standard

None of the state-of-the-art results used any standard to determine their terrain classes, which was a shortcoming of those studies. This dissertation determined the terrain classes to classify between classes based on the ISO 8608 (2016) international standard. Testing concluded on the following off-road terrains:

- Sand track
- Gerotek rally track (gravel track)
- Rough track
- Belgian paving
- Parallel and angled corrugations
- Fatigue track
- Ride and handling track

These off-road terrains were the terrains available at Gerotek testing facilities. The ISO 8608 (2016) standard was used to determine the roughness by plotting the PSD plot of each of these tracks. The terrains were classified between a class A, class D and class H road. The results from the study done by Becker (2008) are shown in Figure 4.21. It is difficult to determine which class each road profile was as they differed with a change in spatial frequency. A reference spatial frequency, namely 0.01 cycle/m, was used to determine the roughness values of the different road profiles. Note that 0.1 cycle/m is the reference frequency used by the ISO 8608 (2016) standard. Table C.3 in the ISO 8608 (2016) shows the wavelength intervals in cycles/m that is used to classify terrains between class A and H (ISO 8608:2016, 2016).

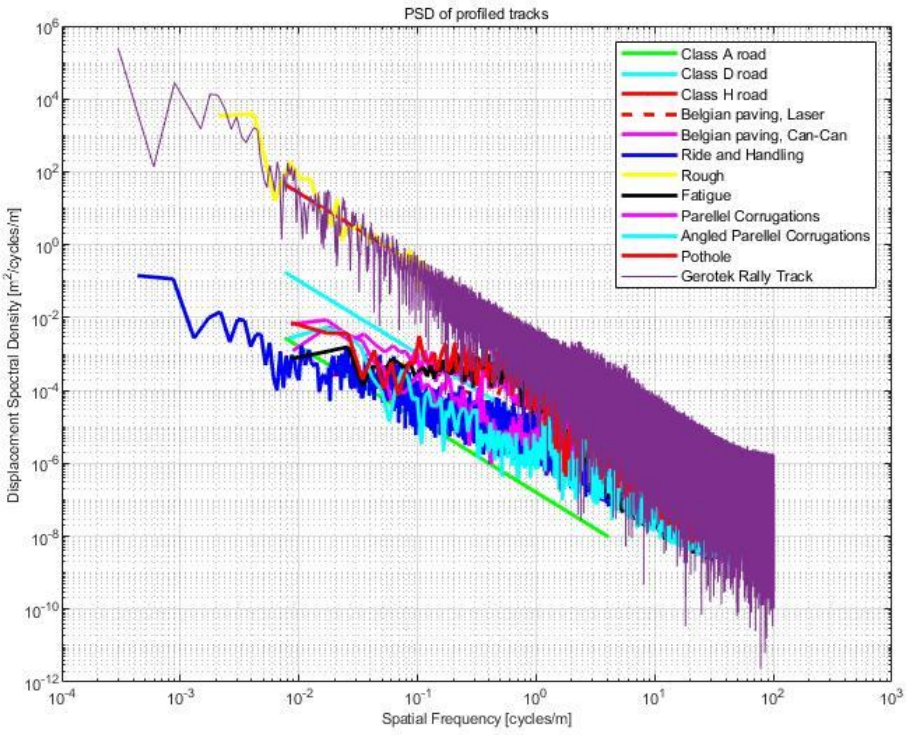

Figure 4.21: PSD results obtained from Can-Can data captured at Gerotek (Becker, 2008)

The terrains were classified as follows:

- Class B:

    1. Ride and handling track
- Class D:
    1. Parallel and angled corrugations
    2. Belgian paving
    3. Fatigue track
- Class H:
    1. Rough track
    2. Gerotek rally track (gravel track)

The sand track data could not be used due to the continuous change in the road profile when driving over the road. Data was captured for grass, gravel and mud terrains. The captured data also consisted of boulders that were extracted from the rough track. In this analysis, the grass, gravel and mud terrain all formed part of the Gerotek rally track (gravel track), which is a class H road. The boulders formed part of the rough track. This reduced the number of road profile images available. Some of the smaller datasets, such as the parallel and angled corrugations, consisted of fewer images than the rally track and rough track. The classifier created the classes and used the same number of randomly selected images during training. This meant that if one class consisted of more data points than another, the classifier would only use the same number of data points as the class with the fewest number of data points.

The total dataset for the downward- and forward-facing categories contained 15 308 images. Weis, et al. (2008) captured 2 520 images and performed classification on 616 images, whereas Lu, et al. (2009) used 900 images. Another study done by Weiss, et al. (2007) consisted of 10 225 images over 14 different classes. Selvathai, et al. (2017) considered 6 250 images, while Sung, et al. (2010) considered 1 200 images. Stolee and Wang's (2018) dataset only consisted of 289 images, whereas Roychowdhury, et al.'s (2019) dataset consisted of 5 300 images. By comparing these dataset sizes to this project's dataset, it can be concluded that this study's dataset uses a larger number of images.

4.5 Conclusion

In this chapter, the data captured during testing was split up into different classes, road materials and characteristics. The ISO 8608 (2016) standard was used to distinguish between the different road profiles. The next chapter describes how the classification task was performed between the following classes as calculated:

- Class B:
    1. Ride and handling track
- Class D:
    1. Parallel and angled corrugations
    2. Belgian paving
    3. Fatigue track
- Class H:
    1. Rough track
    2. Gerotek rally track (gravel track)

Chapter 5: Off-road terrain classification

5.1 Introduction

This chapter discusses how the off-road terrain classification was done based on the models created in Chapter 3 and the data captured in Chapter 4. If the models could classify between the different off-road road profiles with good enough accuracy, the study would be considered successful. The models should further be able to classify the terrain before a vehicle drives over that part of the road.

5.2 Data processing

Before the classifier could classify between the different off-road terrains, the data required preprocessing first. Preprocessing varied between the downward- and forward-facing datasets.

The data was captured in video format, and the image data had to be extracted from these videos. The OpenCV library in Python was used to extract images from the video footage. The algorithm automatically detected the frame rate of the video (10 fps) and extracted the images at that specific rate. The algorithm extracted sequential frames. During the preprocessing phase, the class labels of each image were assigned to the corresponding image because the classification model used a supervised learning model. Each image was given its class name based on the classes created in Chapter 4. The videos of each off-road terrain were captured separately during the data capturing phase. This allowed the user to manually label each type of off-road terrain. Based on the classes determined in Chapter 4, three folders were created and labelled according to the class it belonged to (class B, class D, or class H). Thereafter, the off-road terrain videos belonging to each of those classes were copied manually into their respective folders. Afterwards, a Python program ran through each folder and extracted the images of all the videos in that folder. It assigned a class label to each image depending on the folder/class it belonged to. The program created and labelled a separate folder in which the extracted images of each class were placed. This was done purposefully to make it easier for the classification model to run through the different classes to create, train, test and validate the datasets.

Noise could be present in the image when the image data was captured. Noise is defined as obstacles that do not form part of the terrain profile. The CNN trained and extracted features of whatever input data was provided to the model. Capturing images that included the surroundings gives inaccurate results because the model uses the surroundings to extract features from it instead of just using the terrain profile. It was important to ensure that the data provided to the CNN was correct. To prevent this error, the forward-facing dataset images first had to be cropped to only capture the features of the road by omitting the surroundings. Figure 5.1 shows how an image with noise was cropped to capture relevant data about the terrain profile.

This process took some time as each video was analysed manually to determine the correct bounding box for cropping the video. This is a process that can be improved in future. The bounding box ensured that no noise was present throughout the video. The bounding box of each terrain was analysed, and the biggest possible bounding box was selected to ensure that no noise was present if any of the off-road terrain videos were played. The bounding box for extracting images was determined by finding the extreme pixel positions of the different videos where noise was present. The $x1$, $x2$, $y1$ and $y2$ coordinates were provided to the model and the original image was cropped to the new size. The actual cropping of the images was automated in Python. Because the camera was fixed relatively to the vehicle when the data was captured, the camera moved up and down when the vehicle traversed the different terrains.

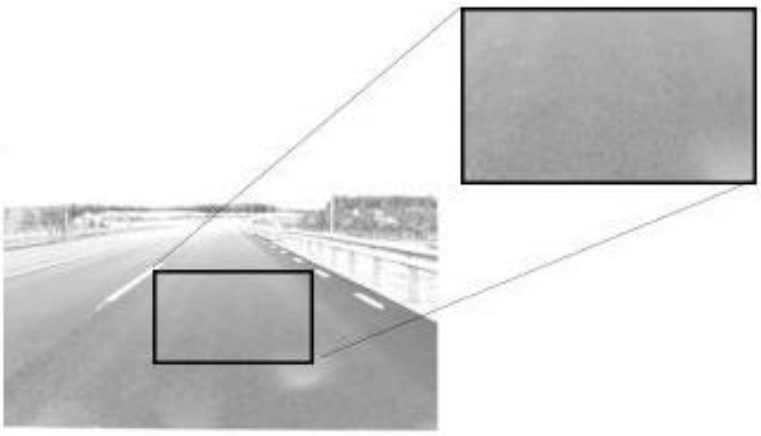

Figure 5.1: Example of a cropped image to omit unwanted noise

This step was only required for the forward-facing camera setup. The forward-facing camera was placed in such a way and at a higher height that allowed it to capture a significant amount of noise. The downward-facing camera was placed much closer to the ground and did not capture any noise. The forward-facing camera was not placed in the centre of the vehicle allowing more noise to be present at the side closer to the edge of the vehicle. To reduce the amount of noise captured in the future, the forward-facing camera can be placed differently and a different lens can be used to reduce the horizon and vertical field of view.

5.3 Classification setup

Two datasets each consisting of three classes were captured at the Gerotek Test Facilities. A total number of 15 308 images was captured during the data capturing phase: 7 783 images for the forward-facing dataset and 7 525 images for the downward-facing dataset. A test train split of 70/30% was used. Some classes had more images than others, thus the class with the smallest dataset was chosen to be the limit of the number of images allowed per class, as shown in Table 5.1. Table 5.1 lists the total number of images captured per class as well as the reduced total (12 645 images). The reduced total is the total number of images in the class with the fewest number of images for both the forward- and downward-facing dataset multiplied by 3 (number of classes). The reduced total was split into a train and a test set and was used as input into the classification models. This meant that 4 095 training images and 1 755 testing images were used respectively for the downward-facing setup, whereas 4 756 training images and 2 039 testing images were used for the forward-facing setup. If a class consisted of more images than allowed, the images were selected randomly using the random function in NumPy.

Table 5.1: Number of images per class and total images vs reduced total images used during training

	Downward facing	**Forward facing**
Class B	1 950	2 265
Class D	2 713	2 751
Class H	2 862	2 767
Total	**7 525**	**7 783**
Reduced total	**5 850**	**6 795**

The original captured images were 1 280 × 960 pixels. After removing the noise in the forward-facing dataset, the images reduced to 900 × 900 pixels. The image size of the downward-facing dataset was not reduced during the preprocessing phase. Both the forward-facing and downward-facing dataset image sizes were resized automatically to 150 × 150 pixels by TensorFlow and given as input for the first convolutional layer. TensorFlow resized the images using the smart_resize function in Keras that was applied

automatically during the first convolutional layer. The full image size could not be used as the computer did not have the memory capabilities to run the simulation. The image was resized to the desired size, which induced a blur in the image that created information loss. The batch size, number of feature maps, image height, and width determine the graphics processing unit (GPU) memory required. Increasing these input variables increases the amount of memory required. A higher batch size is ideal because it reduces sudden drops in training and validation accuracy and loss results. Increasing the image size increases the image resolution, which increases the feature extracting capabilities. Increasing the number of feature maps increases the number of features extracted from the images (Al-Shweiki, 2021). Therefore, a good balance had to be established between these values by considering other solutions and what is suggested by research, and by trial and error comparing the results of the different configurations.

Both the multiple-class classifier and binary class classifier were used. The binary class classifier was used to compare specific classes. These results are shown in Section 5.7.

5.4 Multiple-class classification results

The CNN architecture is shown in Table 5.2. The model had a total of 1 995 466 trainable parameters. The model was trained on 1 000 epochs with a 128 batch size. A 1 000 epochs were used as it showed the point at which overfitting started to become clear. There was an attempt to use a bigger batch size, but the computer ran out of memory. Table 5.2 summarises the model summary of the multiple class classifier.

Table 5.2: Multiple class classifier model summary

Layer (type)	Output shape	Number of parameters
Conv2D	(None, 148, 148, 32)	896
MaxPooling2D	(None, 74, 74, 32)	0
Conv2D	(None, 72, 72, 64)	18 496
MaxPooling2D	(None, 36, 36, 64)	0
Conv2D	(None, 34, 34, 128)	73 856
MaxPooling2D	(None, 17, 17, 128)	0
Conv2D	(None, 15, 15, 256)	295 168
MaxPooling2D	(None, 7, 7, 256)	0
Flatten	(None, 12 544)	0
Dropout	(None, 12 544)	0
Dense	(None, 128)	1 605 760
Dense	(None, 3)	387

5.4.1 Forward-facing results using multiple-class classifier

Figure 5.2 shows both the training accuracy against the number of epochs and the validation accuracy against the number of epochs. A training accuracy of 100% was achieved after 1 000 epochs. The validation accuracy ranged between 99.3% and 100%. At the end of the 1 000 epoch values, the validation accuracy started to decrease slightly, which was an indication of the model overfitting.

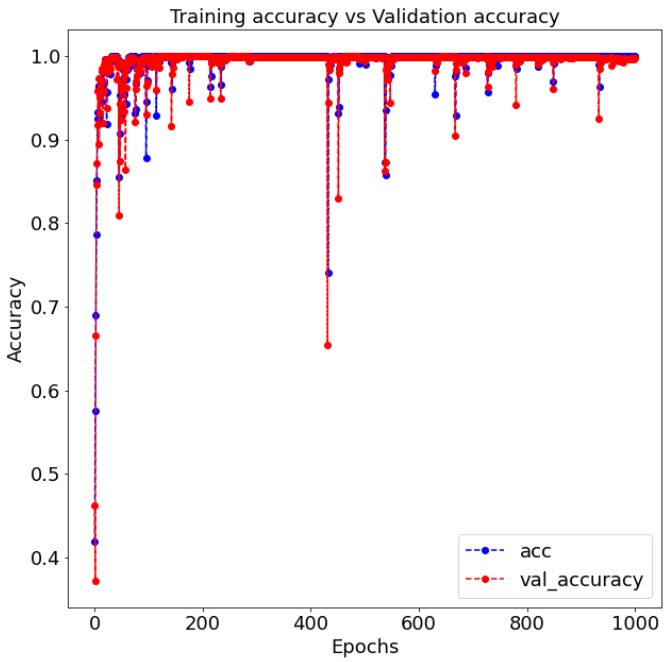

Figure 5.2: Training and validation accuracy results for the forward-facing dataset using a multiple class classifier

The training and validation loss values are shown Figure 5.3, which is a representation of whether enough epoch values were used during the training process. As the training and validation loss results approached 0, it indicated that the model had trained for enough epoch values. The results showed that the validation loss was slightly larger than the training loss at the end. This meant that the model had started to overfit. The outlier in the data at 431 epochs, as seen in Figure 5.3, made it difficult to spot when overfitting occurred. When the outlier was removed, as seen in Figure 5.4, it became clear that overfitting started to occur after 600 epochs. This is because the validation loss and training loss values no longer decrease to zero but starts to increase. The validation loss value remained constant between 300 and 600 epoch values.

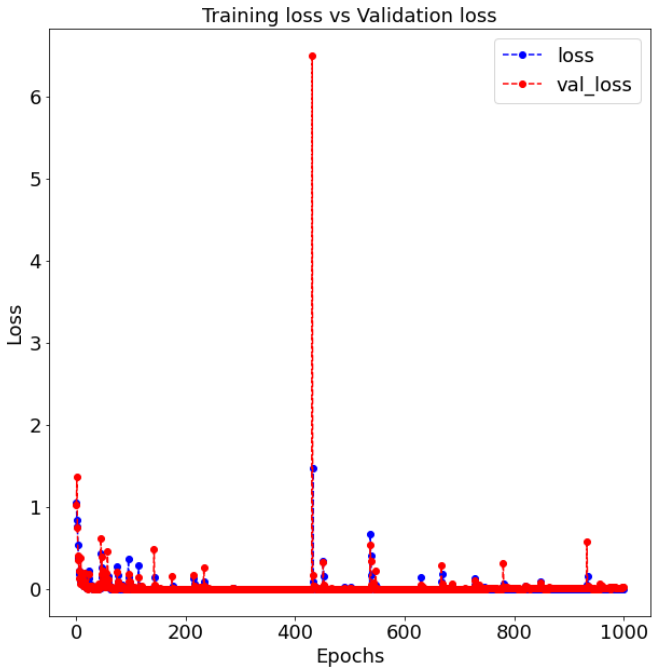

Figure 5.3: Training and validation loss results for the forward-facing dataset using a multiple class classifier

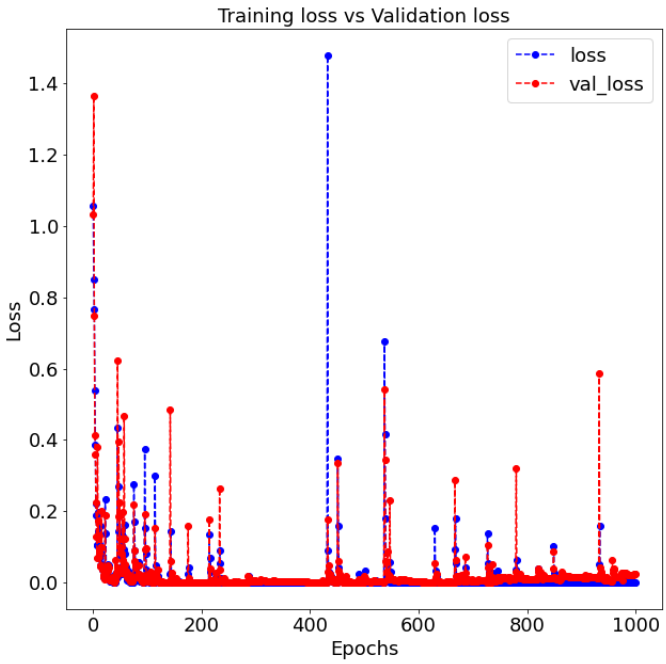

Figure 5.4: Training and validation loss results for the forward-facing dataset using a multiple class classifier and removing the outlier at 431 epochs

Figure 5.5 shows the confusion matrix results when the classifier tried to predict the road type of unseen road profiles. The model achieved a prediction accuracy of 100% for both the class B and class D road types. A 99% prediction accuracy was achieved for the class H road with the other 1% being predicted as a class D road. An overall prediction accuracy of 99.71% was achieved.

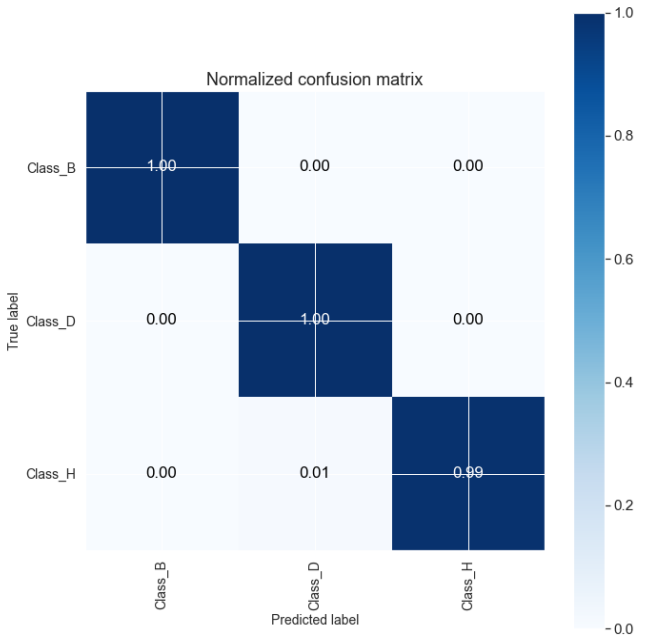

Figure 5.5: Confusion matrix for multiple class classifier prediction on unseen forward-facing road profiles

5.4.2 Downward-facing results using a multiple-class classifier

Figure 5.6 shows both the training accuracy against the number of epochs and the validation accuracy against the number of epochs. A training accuracy of 100% was achieved after 1 000 epochs. The validation accuracy ranged between 99% and 100%. At the end of the 1 000 epoch values, the validation accuracy also started to decrease, indicating that the model was overfitting. The goal was to train the model for as few epochs as possible while achieving accuracy as high as possible. Therefore, many epochs were used to determine at which point overfitting might occur. This point could then be the number of epochs to use for the model, whereafter the model can be retrained for only that number of epoch values.

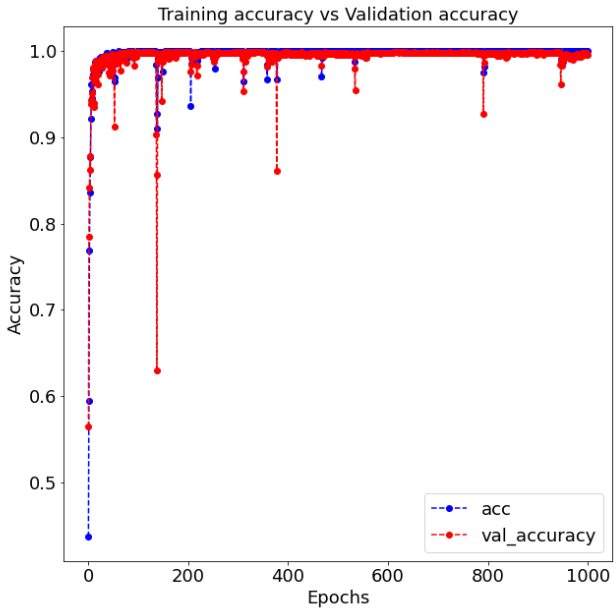

Figure 5.6: Training and validation accuracy results for the downward-facing dataset using a multiple class classifier

Figure 5.7 shows the training and validation loss values. The training and validation loss values approached 0, indicating the model had trained for a large enough number of epoch values. As the epoch values approached 1 000, the results again started to indicate overfitting. The accuracy and loss plots for the downward-facing setup had fewer spikes in the results, but shared the same overall shape.

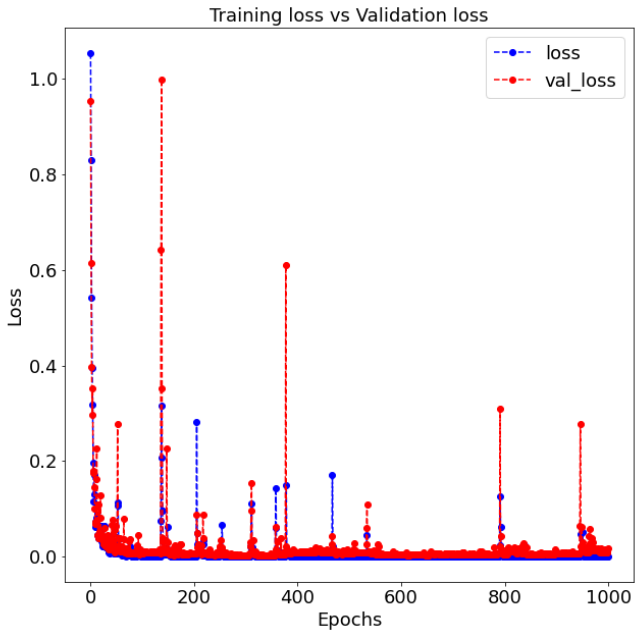

Figure 5.7: Training and validation loss results for the downward-facing dataset using a multiple class classifier

Figure 5.8 shows the confusion matrix results when the classifier predicted the road type of unseen road profiles for the downward-facing dataset. The model achieved a prediction accuracy of 100% for both the class B and class D road types. A 99% prediction accuracy was achieved for the class H road with the other 1% being predicted as a class D road. An overall prediction accuracy of 99.65% was achieved. The prediction results were 0.06% better than the forward-facing results; however, this cannot be used to state that using the downward-facing results are better because of the small improvement. Changing the epoch values or other hyperparameters of the CNN could alter the results slightly. It is safer to assume that both the forward- and downward-facing setups gave satisfactory results when using the multiple-class classifier.

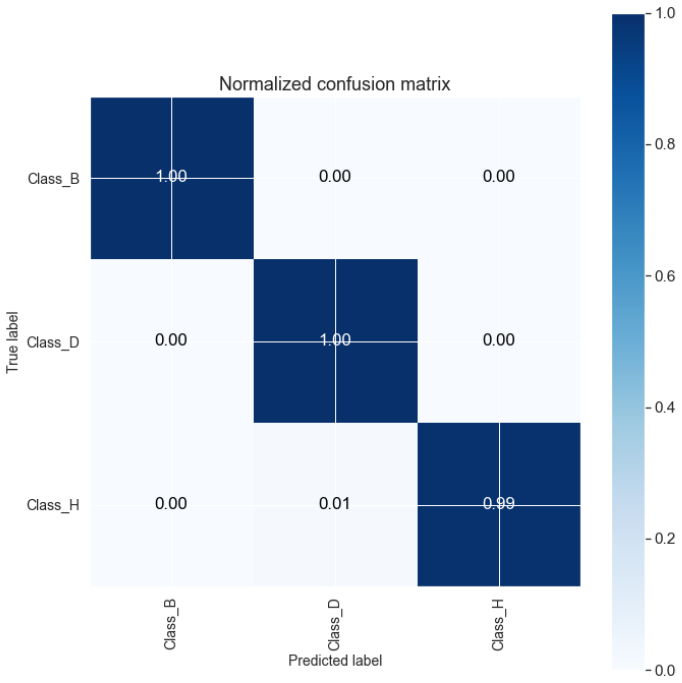

Figure 5.8: Confusion matrix for multiple class classifier prediction on unseen downward-facing road profiles

There were sudden drops in accuracy and loss percentages for both the forward- and downward-facing results. Based on online research, this could have been due to two reasons. First, the model is built on TensorFlow, which can cause a sudden drop in accuracy performance due to momentary memory loss. However, this did not affect the overall accuracy results and prediction accuracy on the prediction set. Second, the accuracy could drop due to the learning rate being too high, which was unlikely. If the learning rate is too high, the model usually does not recover and yield good accuracy results again. The accuracy results will drop suddenly and remain that low (Stack Exhange Information, 2019).

5.5 Pretrained model classification results

Table 5.3 displays the model summary of the Xception model. The full Xception model is not shown – only the layers added.

Table 5.3: Pretrained model summary

Layer (type)	Output shape	Number of parameters
Xception (model)	(None, 5, 5, 2048)	20 861 480
Flatten	(None, 51 200)	0
Dropout	(None, 51 200)	0
Dense	(None, 128)	6 553 728
Dense	(None, 3)	1 290

The model consisted of a total of 27 416 498 parameters: 27 361 970 trainable parameters and 54 528 non-trainable parameters. The non-trainable parameters were present because the input layers of the Xception model were imported with the weight values. These were frozen and remained unchanged during the training process.

5.5.1 Forward-facing results using the pretrained model

A training accuracy of 100% was achieved after 1 000 epochs as shown in Figure 5.9. The validation accuracy results ranged between 99% and 100%. Compared with the multiple class classifier, the pretrained model did not experience the gradual decrease in validation accuracy, but did have more spikes in the data. These spikes happened from time to time and caused a sudden decrease in validation and training accuracy. It was expected that the sudden drops in the validation and training accuracy were caused by the batch size. A batch size of 16 images was used compared with the batch size of 128 for the multiple class classifier. The smaller batch size was used because of the memory limitations of the computer being exceeded during training (16 GB of RAM was available and 4 GB of GPU memory). The convergence graph would be smooth if the model trained on the entire dataset, instead of with batches.

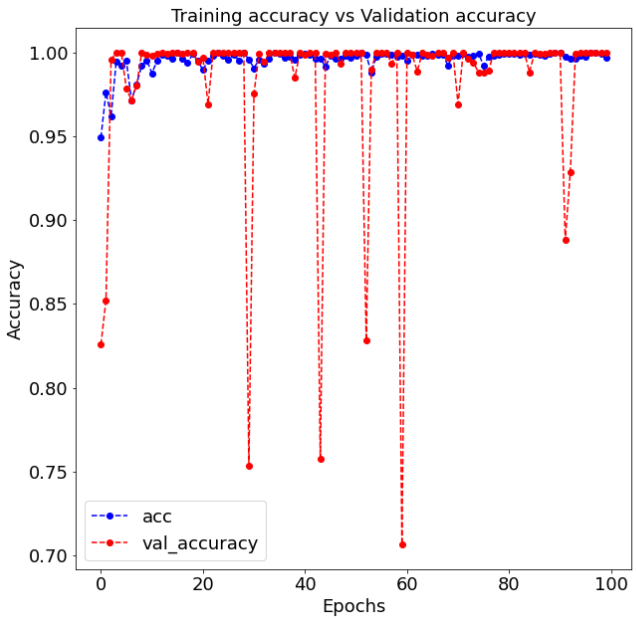

Figure 5.9: Training and validation accuracy results for the forward-facing dataset using a pretrained model

The training and validation loss values are shown in Figure 5.10. The model experienced one big spike in validation loss during the early stages of training. Although several other spikes also occurred in the data, they were not as clear in Figure 5.10 due to the outlier. The outlier was removed in Figure 5.11 and the validation loss spikes can be seen easily. As mentioned at the start of Section 5.5, these spikes occurred because of the small batch size.

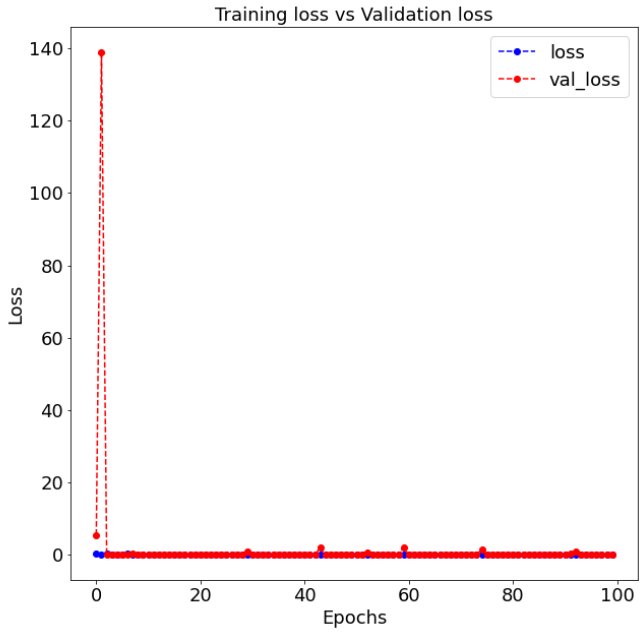

Figure 5.10: Training and validation loss results for the forward-facing dataset using a pretrained model

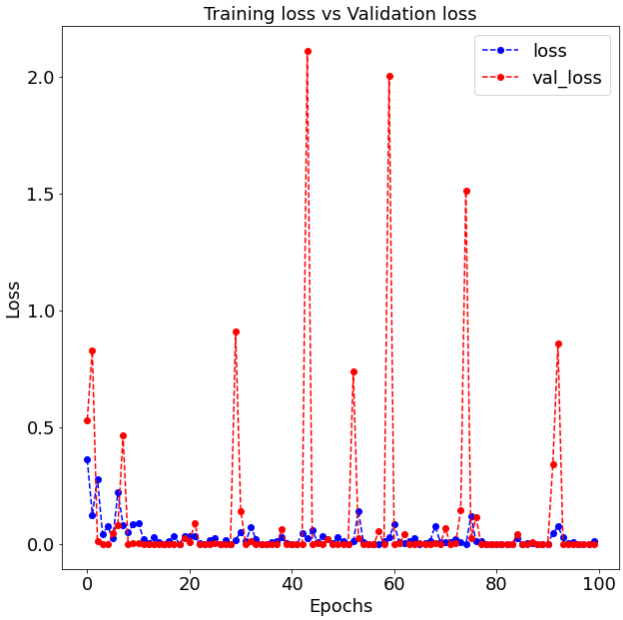

Figure 5.11: Training and validation loss results for the forward-facing dataset using a pretrained model without the outlier

Figure 5.12 shows the confusion matrix results. The pretrained model had a 100% accuracy when performing a prediction task on unseen road profile images.

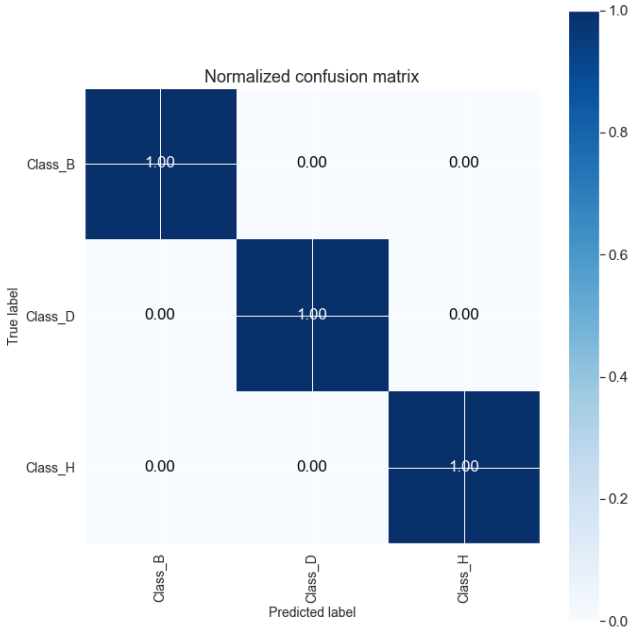

Figure 5.12: Confusion matrix for pretrained model predictions on unseen forward-facing road profiles

5.5.2 Downward-facing results using a pretrained model

A validation and training accuracy of 100% was achieved after 1 000 epochs. The validation and training accuracy again showed several spikes during the training process. The training and validation accuracies are shown in Figure 5.13.

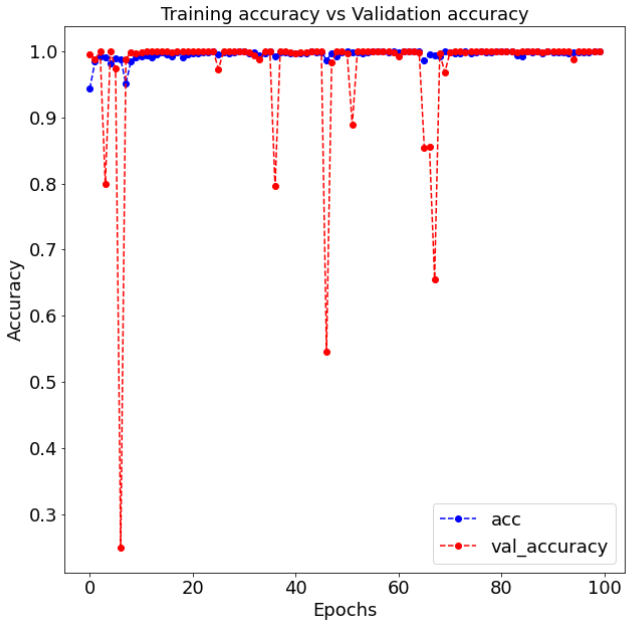

Figure 5.13: Training and validation accuracy results for the downward-facing dataset using a pretrained model

Figure 5.14 shows the validation and accuracy loss results. There was a sudden spike in loss results again. Three outliers were removed at epochs 6, 42 and 67 to get Figure 5.15, which shows a clearer picture. Fewer loss spikes are seen than for the forward-facing pretrained model.

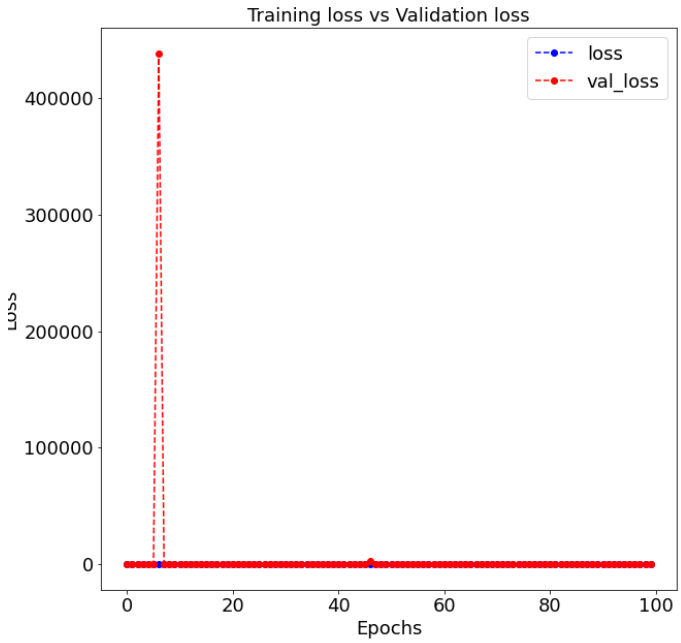

Figure 5.14: Training and validation loss results for the downward-facing dataset using a pretrained model

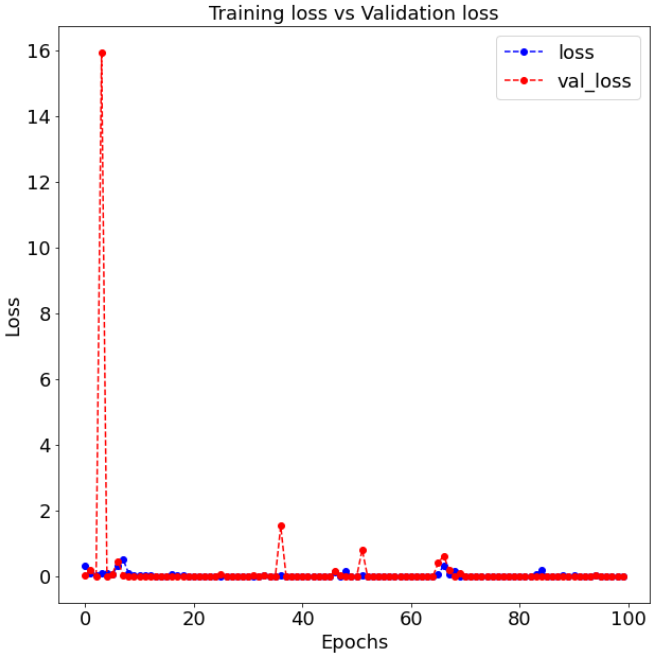

Figure 5.15: Training and validation loss results for the downward-facing dataset using a pretrained model without major outliers

Figure 5.16 shows a prediction accuracy of 100% for all the classes. The prediction accuracy is higher than the multiple class classifier results for the downward-facing dataset.

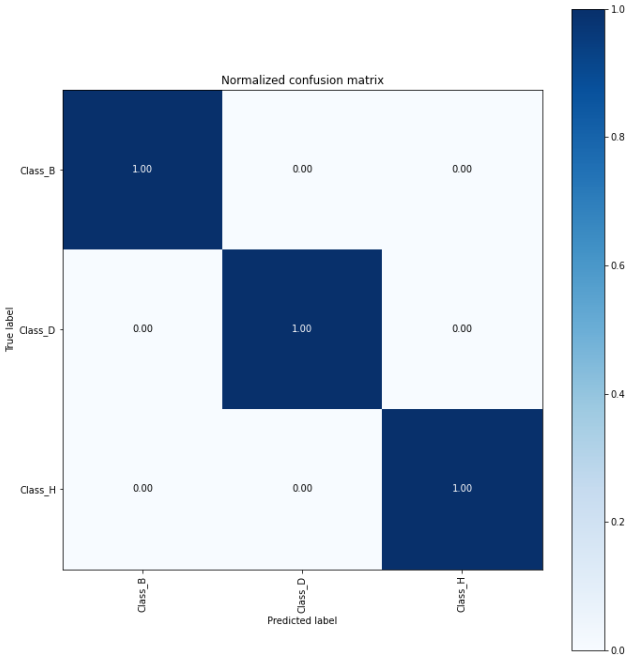

Figure 5.16: Confusion matrix for the pretrained model predictions on unseen downward-facing road profiles

5.6 Classification results omitting certain off-road terrains

Section 5.4 and Section 5.5 discussed how terrain classification was carried out while training the classifier on all the terrains it would perform the prediction on using supervised learning. The results showed that classification of off-road terrains can be achieved using a CNN and image data if the model had been trained on those off-road terrains. For this section, the classification was done again, but the off-road terrain was

purposefully omitted during training. The goal was to determine what would happen if one of the off-road profiles from class B, class D or class H was omitted and training was done on the rest of the off-road terrains. Would the classifier be able to predict the class of the untrained terrain correctly? The same classification models were used as with Section 5.4 and Section 5.5. The following off-road terrains were omitted from each class:

- Test 1: Parallel corrugations from class D
- Test 2: Fatigue track from class D
- Test 3: Rough track from class H

It was not possible to omit a terrain from class B as it only consisted of one terrain. The classification was performed on both the forward- and downward-facing datasets. Classification was done for both the multiple-class classifier as well as the pretrained model. After the models had been trained, the prediction model were run using two approaches. For the first approach, the validation dataset consisted of the off-road terrains used during training. The untrained terrain was added back to the corresponding class it belonged to. The prediction results were captured for this approach. The second approach performed the prediction task only on the untrained class to determine how many of the terrain images were classified as the correct class.

5.6.1 Classification without parallel corrugated terrain from the class D road profile

The training and validation accuracy and loss plots are attached in Appendix A. Table 5.4 summarises the training accuracy and loss results from both the forward- and downward-facing datasets for both the multiple-class classifier and pretrained model approach. The same model setups were used as for Section 5.4 and Section 5.5. Table 5.4 shows that the training for all the models delivered good results for all the tests.

Table 5.4: Accuracy and loss value training results for classification without the parallel corrugated terrain

		Training accuracy [%]	Training loss	Validation accuracy [%]	Validation loss
Forward-facing classification	Multiple-class classifier	99.88–100	0.00169	99.48–100	0.0157
	Pretrained model	99.92–100	0.00199	100	4.517e-05
Downward-facing classification	Multiple-class classifier	100	2.427e-06	99.13–100	0.0776
	Pretrained model	100	0.000615	100	0.00021

Table 5.5 summarises the prediction accuracy results for the first approach where the untrained terrain, the parallel corrugated terrain, was added back to the rest of the off-road terrains. As stated earlier, the models did not train on the untrained terrain but attempted to predict the class of the terrain. The results showed a slight decrease in prediction accuracy. The pretrained model still got a 100% prediction accuracy.

Table 5.5: Prediction accuracy including the omitted terrain during training vs the omitted terrain during training for the parallel corrugated off-road terrain

		Prediction accuracy including untrained terrain during training [%]	Prediction accuracy excluding untrained terrain during training [%]
Forward-facing classification	Multiple-class classifier	99.71	98.53
	Pretrained model	100	100
Downward-facing classification	Multiple-class classifier	99.65	99.32
	Pretrained model	100	100

Table 5.6 summarises the second approach during which the trained model only attempted to predict the class of the off-road terrain that was omitted, namely the parallel corrugated off-road terrain, during the training phase.

Table 5.6: Prediction accuracy on the untrained parallel corrugated off-road terrain only

		Prediction accuracy only on parallel corrugated off-road terrain [%]
Forward-facing classification	Multiple-class classifier	87.97
	Pretrained model	100
Downward-facing classification	Multiple-class classifier	99.46
	Pretrained model	100

The prediction results were still satisfactory even though the model never trained on the parallel corrugated terrain. This could be due to the similarities between the angled and parallel corrugated terrains. These terrains share similarities, which make it easier for the model to predict the class of the untrained profile. The pretrained model results were better than the multiple-class classifier results. The prediction accuracy was 87.97% for the forward-facing multiple-class classifier results with 8.24% of the parallel corrugated terrain being classified as a class H road and 3.78% as a class B road. The prediction was done on 740 images. The prediction accuracy results in Table 5.6 are lower than in Table 5.5 because Table 5.5 contained terrain images that the model did train on for the other terrains that were part of the class D road. These terrain results increased the prediction results whereas Table 5.7 only contained the untrained profile. This is also seen in Section 5.6.2 (Table 5.8 and Table 5.9) and Section 5.6.3 (Table 5.11 and Table 5.12).

5.6.2 Classification without fatigue track from the class D road profile

Table 5.7 summarises the training accuracy and loss results from both the forward- and downward-facing datasets for both the multiple-class classifier and pretrained model approach. In Section 5.6.1, the parallel corrugated terrain was omitted. The fatigue track was now omitted because it was expected to be less correlated to the other tracks in the class D class.

Table 5.7: Accuracy and loss value training results for classification without the fatigue track

		Training accuracy [%]	Training loss	Validation accuracy [%]	Validation loss
Forward-facing classification	Multiple-class classifier	99.98–100	0.001021	99.71–100	0.01203
	Pretrained model	99.83–100	0.00883	99.13–100	0.02687
Downward-facing classification	Multiple-class classifier	100	3.4119e-05	99.40–100	0.01592
	Pretrained model	98.59–100	0.10366	99.66–100	0.00855

Table 5.8 summarises the prediction accuracy results for the first approach where the untrained terrain, namely the fatigue track, was added to the rest of the off-road terrains. As stated earlier, the models did not train on the untrained terrain, but attempted to predict the class of the terrain. Again, it was shown that the prediction accuracy results were very high even with an untrained track that shared fewer similarities. The prediction results did decrease, but by less than 1%.

Table 5.8: Prediction accuracy including the omitted terrain during training vs omitted terrain during training for the fatigue track

		Prediction accuracy including untrained terrain during training [%]	Prediction accuracy excluding untrained terrain during training [%]
Forward-facing classification	Multiple-class classifier	99.71	99.56
	Pretrained model	100	99.41
Downward-facing classification	Multiple-class classifier	99.65	99.49
	Pretrained model	100	99.66

Table 5.9 summarises the second approach in which the trained model only attempted to predict the class of the off-road terrain that was omitted, namely the fatigue track, during the training phase. All the results were close to 100%. The prediction was again done on 740 fatigue track images. The forward-facing results for the multiple-class classifier delivered better results than the parallel corrugated results.

Table 5.9: Prediction accuracy on the untrained fatigue track only

		Prediction accuracy only on fatigue track [%]
Forward-facing classification	Multiple-class classifier	98.71
	Pretrained model	100
Downward-facing classification	Multiple-class classifier	99.03
	Pretrained model	98.71

5.6.3 Classification without rough track from the class H road profile

Table 5.10 summarises the training accuracy and loss results from both the forward- and downward-facing datasets for both the multiple-class classifier and pretrained model approach.

Table 5.10: Accuracy and loss value training results for classification without rough track

		Training accuracy [%]	Training loss	Validation accuracy [%]	Validation loss
Forward-facing classification	Multiple-class classifier	99.37–100	0.02137	99.39–100	0.03389
	Pretrained model	99.89–100	0.0032	100	0.00018
Downward-facing classification	Multiple-class classifier	100	6.576e-05	99.87–100	0.002359
	Pretrained model	100	0.000255	100	6.92998e-05

Table 5.11 summarises the prediction accuracy results for the first approach in which the untrained terrain, namely the rough track, was added to the rest of the off-road terrains. The results were all above 90% with the lowest being the multiple-class classifier prediction results for the forward-facing dataset. The results were lower than the previous sections.

Table 5.11: Prediction accuracy including the omitted terrain during training vs omitted terrain during training for the rough track

		Prediction accuracy including untrained terrain during training [%]	Prediction accuracy excluding untrained terrain during training [%]
Forward-facing classification	Multiple-class classifier	99.71	93.66
	Pretrained model	100	97.20
Downward-facing classification	Multiple-class classifier	99.65	95.73
	Pretrained model	100	99.32

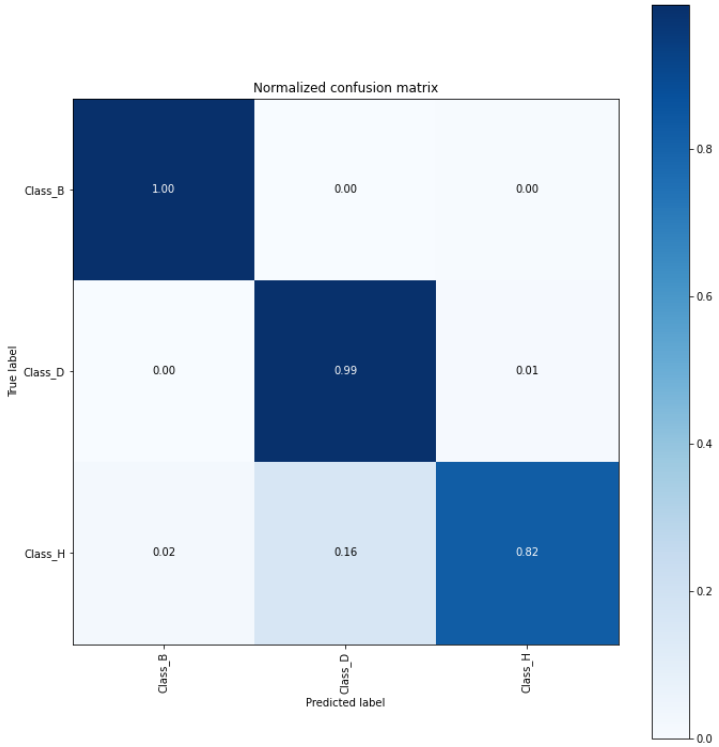

Figure 5.17: Confusion matrix plot for the forward-facing multiple-class classification results omitting the rough track

From Figure 5.17 it became clear that a terrain had been omitted from the training phase. The similarities between the omitted terrain and the rest of the terrain in class H was not as closely correlated. The same is seen in Figure 5.18 and Figure 5.19. The results in Figure 5.20 were still high.

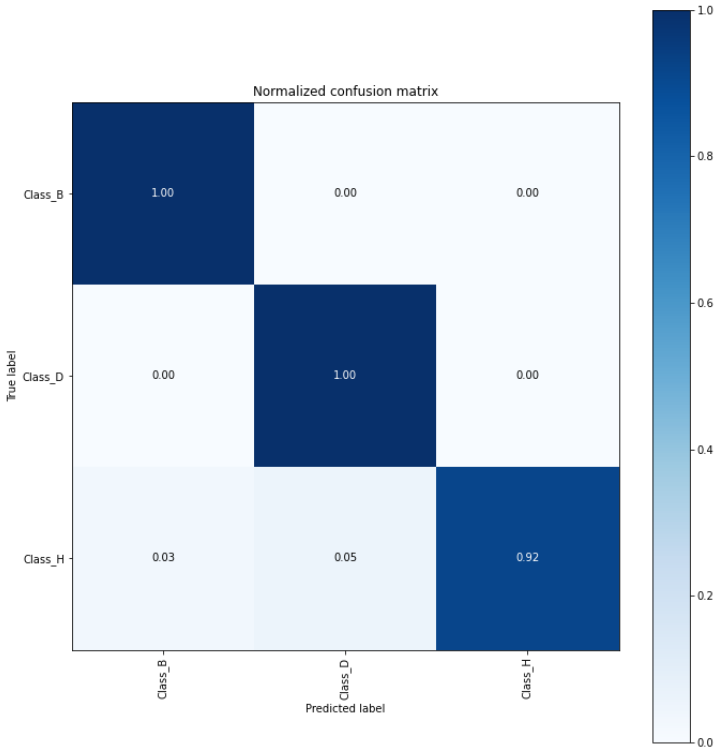

Figure 5.18: Confusion matrix plot for the forward-facing pretrained model results omitting the rough track

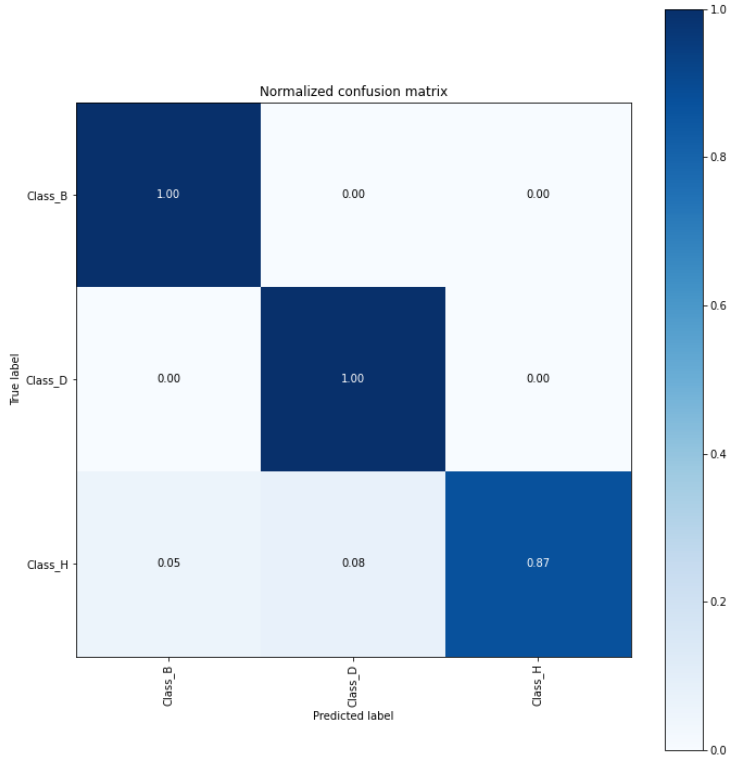

Figure 5.19: Confusion matrix plot for the downward-facing multiple-class classification results omitting the rough track

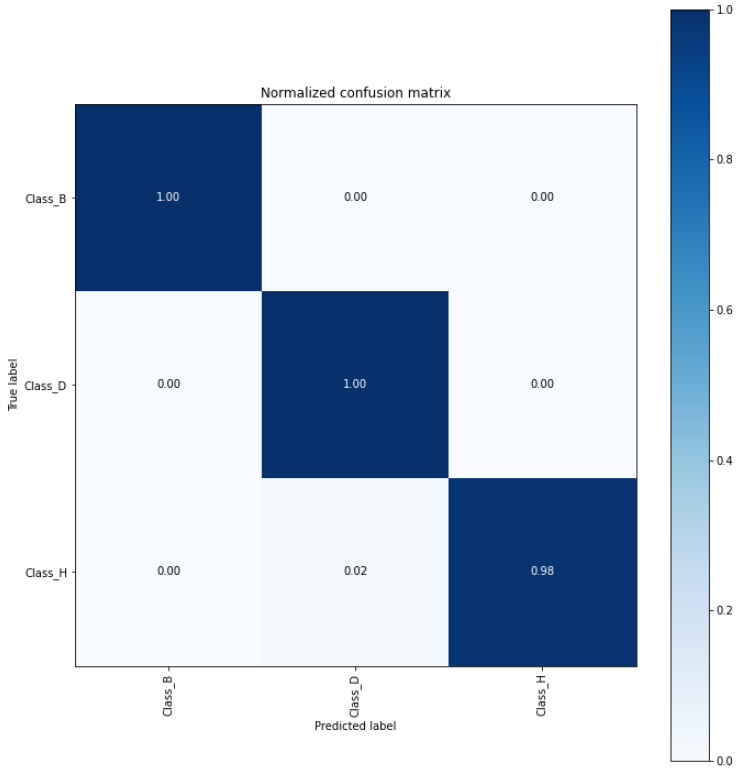

Figure 5.20: Confusion matrix plot for the downward-facing pretrained model results omitting the rough track

Table 5.12 summarises the second approach in which the trained model only attempted to predict the class of the off-road terrain that was omitted, namely the fatigue track, during the training phase. It was obvious that the prediction results were not good. When the model only attempted to predict the untrained profile,

the prediction results were much lower than the previous sections. The pretrained models gave the best results, but only the downward-facing pretrained model had good results.

Table 5.12: Prediction accuracy on the untrained rough track only

		Prediction accuracy only on rough track [%]
Forward-facing classification	Multiple-class classifier	28.91
	Pretrained model	53.59
Downward-facing classification	Multiple-class classifier	37.16
	Pretrained model	83.58

Section 4.4 concluded that the Gerotek rally track and the rough track form part of the class H road. The omitted section from the rough track is made from concrete. The rally track consists of grass, mud, rocks and gravel. The appearance of these terrains is completely different than the concrete rough track. If the untrained profile is completely different than the other terrains in its respective classes, the prediction accuracy decreases. To predict a class of an untrained profile accurately using a supervised learning model, there must be some similarities between the untrained profile and the other profiles of that class. This was the case for the parallel corrugated terrain and the fatigue track. These results indicated that colour plays an important role. The Gerotek rally track and the rough track are different colours. It appears that the classification model made strong use of colour to distinguish between terrains and potentially less use of the actual roughness. There could be a correlation between colour and roughness, whereas the roughness of the terrain was paired with a certain colour scheme, but this will not always be the case. This issue could potentially be solved by adding a second sensor (an accelerometer, for example).

If colour plays an important role, it is important to capture data points for different weather conditions. The weather conditions will affect the colour of the terrain. If it is raining, the terrain will appear completely different. Different times of the day will also make the terrains appear differently due to the position of the sun (for example, shadows and when the sun rises and sets). The dataset used was for a sunny day with no rain and the data capturing occurred when the sun was directly above the vehicle.

5.7 Binary classification results between corrugated terrains

Parallel and angled corrugations are similar in nature as explained in Section 4.3.6. Furthermore, Section 5.6 showed that they were closely correlated. A binary classification was performed to establish whether it would be possible to distinguish between these profiles. The model is described in Section 3.4 and Section 3.5. For the pretrained model, the final dense layer used the sigmoid activation function and the binary cross-entropy loss function.

5.7.1 Binary classification model

Table 5.13 summarises the model summary for the binary classification model.

Table 5.13: Model summary for the binary classification between the angled and corrugated terrain

Layer (type)	Output shape	Number of parameters
Conv2D	(None, 148, 148, 32)	896
MaxPooling2D	(None, 74, 74, 32)	0
Conv2D	(None, 72, 72, 64)	18 496
MaxPooling2D	(None, 36, 36, 64)	0
Conv2D	(None, 34, 34, 128)	73 856

MaxPooling2D	(None, 17, 17, 128)	0
Conv2D	(None, 15, 15, 256)	295 168
MaxPooling2D	(None, 7, 7, 256)	0
Flatten	(None, 12 544)	0
Dropout	(None, 12 544)	0
Dense	(None, 128)	1 605 760
Dense	(None, 1)	129

5.7.1.1 Forward-facing results

Figure 5.21 shows that a training and validation accuracy of 100% was achieved.

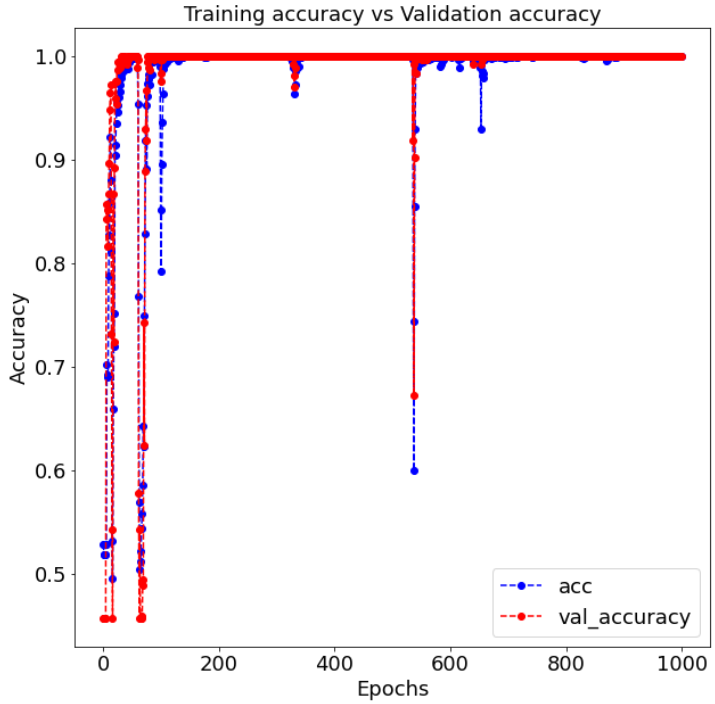

Figure 5.21: Forward-facing binary training and validation accuracy results

Figure 5.22 shows that a training and validation loss value of close to 0 was achieved. There was no real indication of overfitting, but further training was not required because a prediction accuracy of 100% was achieved as shown in Figure 5.23.

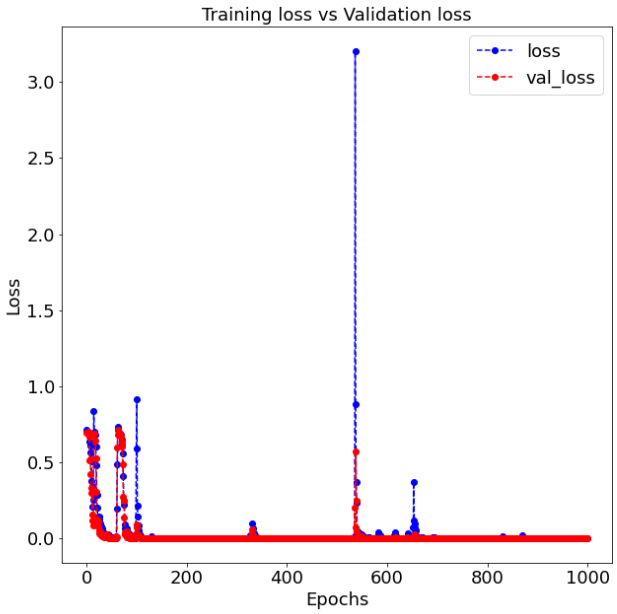

Figure 5.22: Forward-facing binary training and validation loss value results

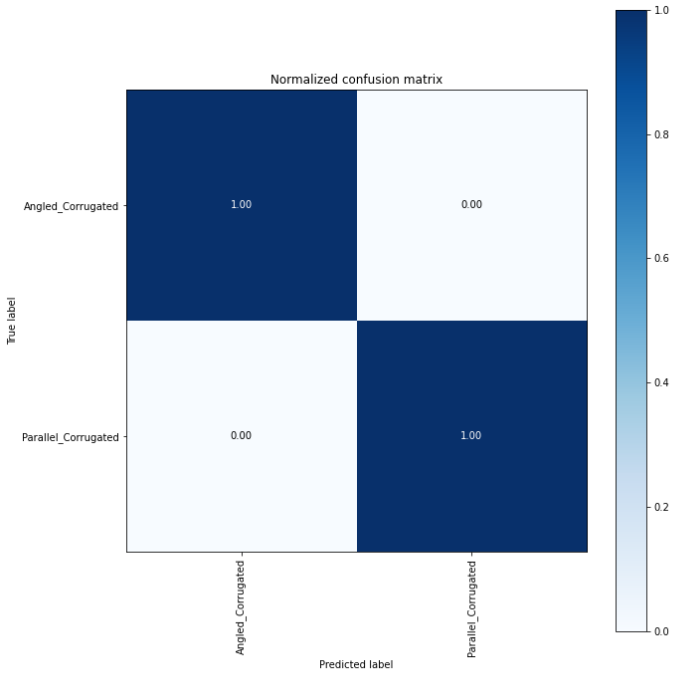

Figure 5.23: Confusion matrix plot for the forward-facing binary classification between the angled and parallel corrugated terrains

5.7.1.2 Downward-facing results

Figure 5.25 shows the training and validation accuracy results. At around 200 epoch values, the training accuracy results reached about 100% and remained there. The validation accuracy results reached a maximum of 89% and were not able to improve. This indicated that the model was not able to get a validation accuracy higher than 89%. To improve, either more data would be required or the model had to be improved.

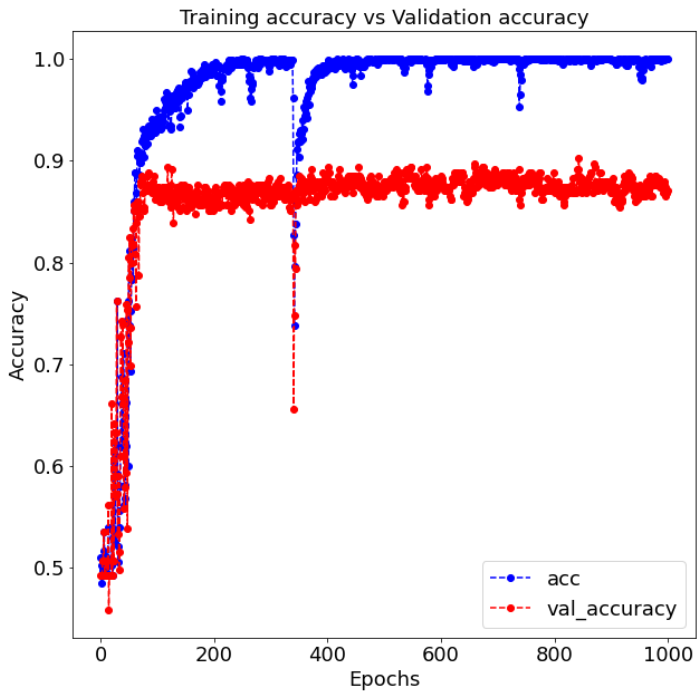

Figure 5.24: Downward-facing binary training and validation accuracy results

Figure 5.25 shows that overfitting occurred quite early in the training process. The validation loss value started to increase after around 50 epochs.

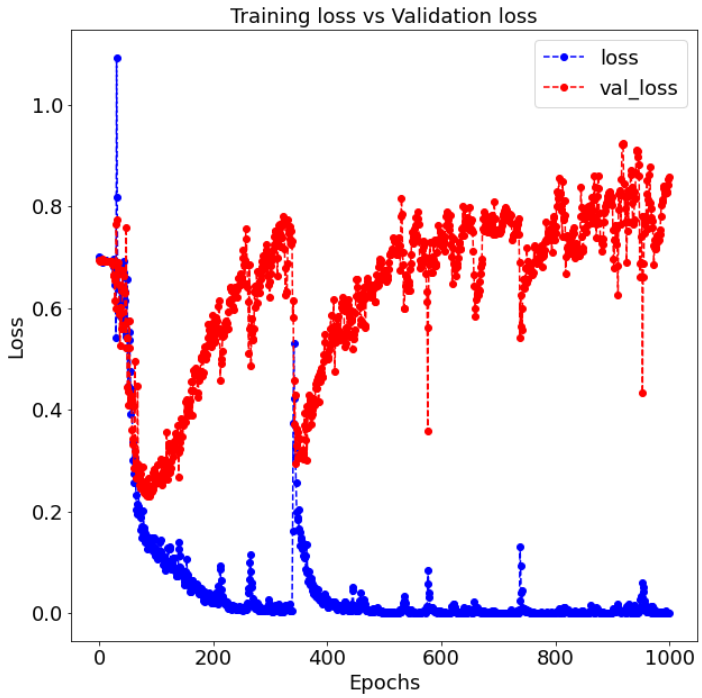

Figure 5.25: Downward-facing binary training and validation loss value results

The confusion matrix in Figure 5.26 shows that on average the model was able to classify 87.5% of the unseen data correctly.

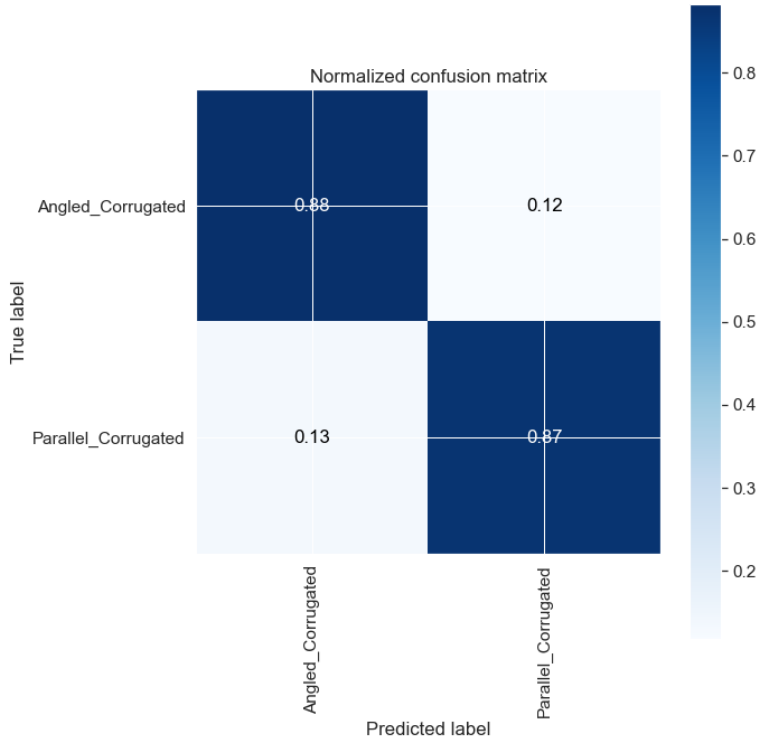

Figure 5.26 Confusion matrix plot for the downward-facing binary classification between the angled and parallel corrugated terrains

5.7.2 Pretrained binary classification model

Table 5.14 shows the model summary of the binary pretrained model.

Table 5.14: Pretrained model summary for the binary classification

Layer (type)	Output shape	Number of parameters
Xception (model)	(None, 5, 5, 2048)	20 861 480
Flatten	(None, 51 200)	0
Dropout	(None, 51 200)	0
Dense	(None, 128)	6 553 728
Dense	(None,1)	129

The model contained 27 415 337 parameters: 27 360 809 trainable parameters and 54 528 non-trainable parameters.

5.7.2.1 Forward-facing results

Figure 5.27 shows the training and validation accuracy for the pretrained model training on the forward-facing binary classification data between the angled and straight corrugated off-road terrain. A training and validation accuracy of 100% was achieved after 100 epochs. However, there were some sudden drops in the training accuracy results, which were expected due to the small batch size.

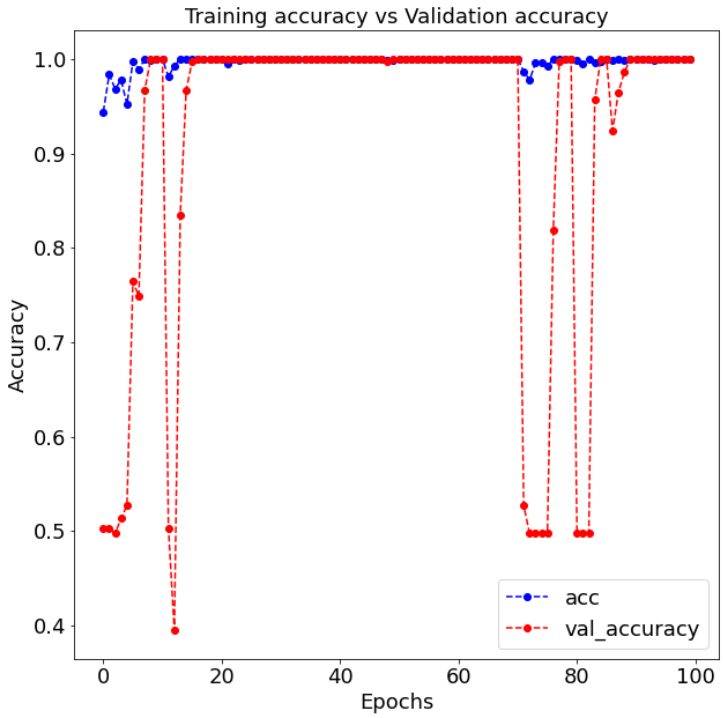

Figure 5.27: Forward-facing binary training and validation accuracy results for the pretrained model

Figure 5.28 shows that the training and validation loss results were equal to 0 and the model did not overfit. The prediction accuracy was 100% as shown in Figure 5.29.

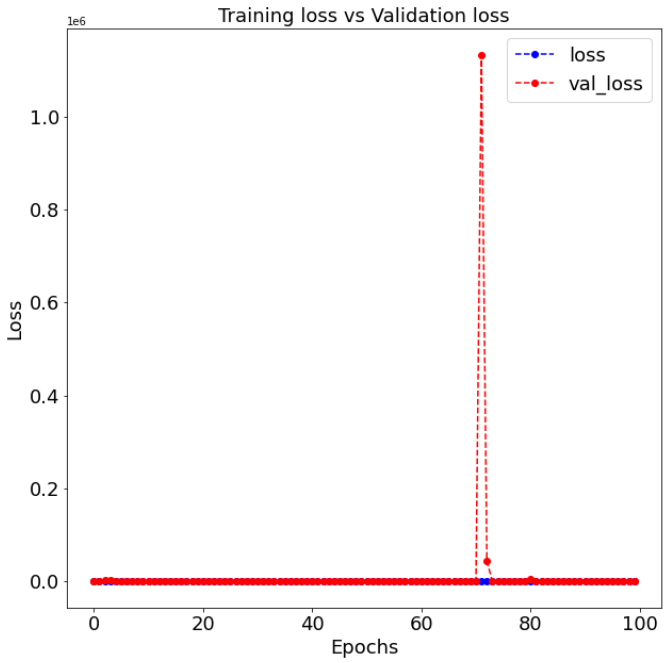

Figure 5.28: Forward-facing binary training and validation loss value results for the pretrained model

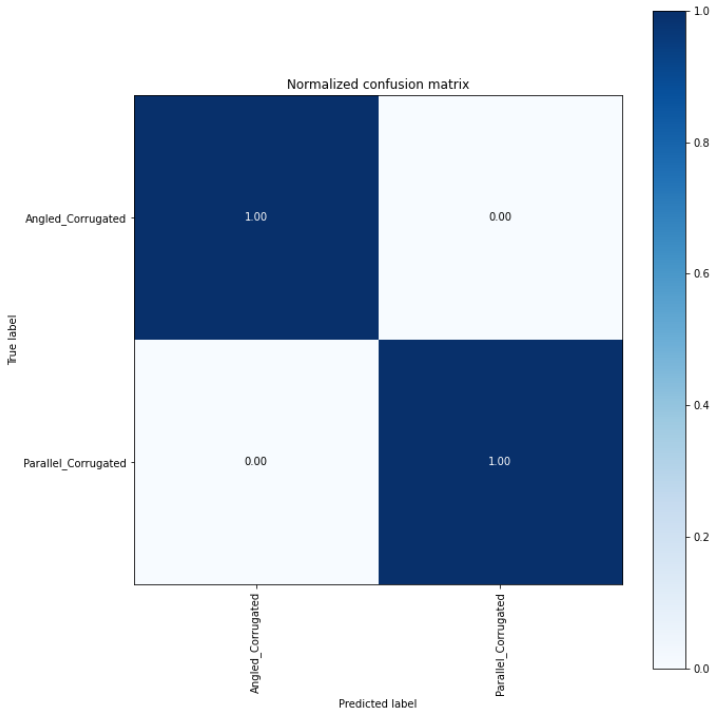

Figure 5.29: Confusion matrix plot for the forward-facing binary classification between the angled and parallel corrugated terrains for the pretrained model

5.7.2.2 Downward-facing results

Figure 5.30 shows the training and validation accuracy results for the pretrained downward-facing results. The validation results were not as good again, which agreed with the validation results for the binary classifier. The sudden drops were due to the small batch size. Although a validation accuracy of 88% was satisfactory, the graph showed that the results were not as clean as the others.

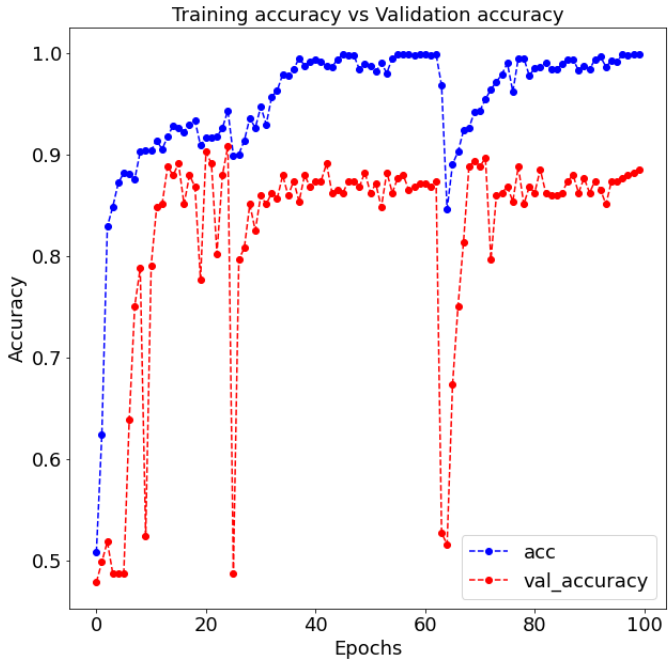

Figure 5.30: Downward-facing binary training and validation accuracy results for the pretrained model

Figure 5.31 shows the training and validation loss values for the pretrained binary classification task. From the graph, it appeared that the results went down to 0. The outlier in the data made the results appear better than they were. Upon closer inspection it was determined that overfitting occurred as the epoch values increased.

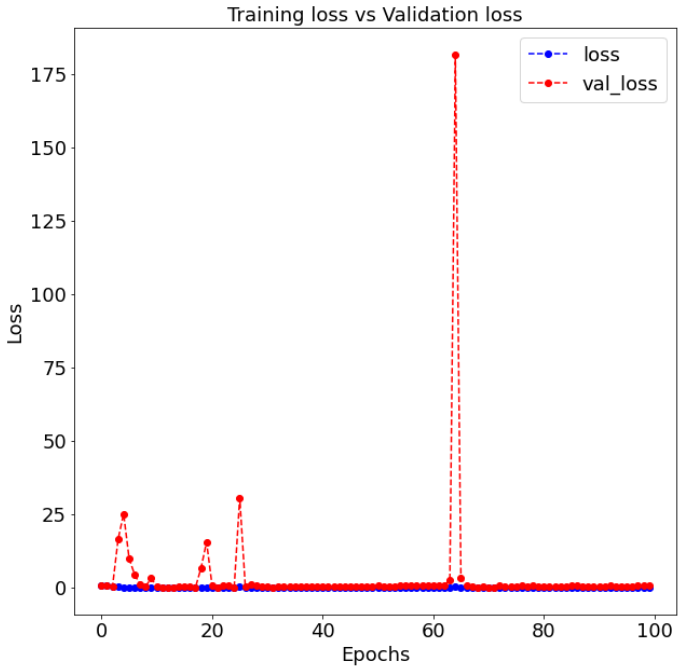

Figure 5.31: Downward-facing binary training and validation loss value results for pretrained model

An overall prediction accuracy on unseen data of 87.5% was achieved with 84% of the angled corrugated road being classified correctly and 91% of the parallel corrugated terrain being classified correctly as shown in Figure 5.32.

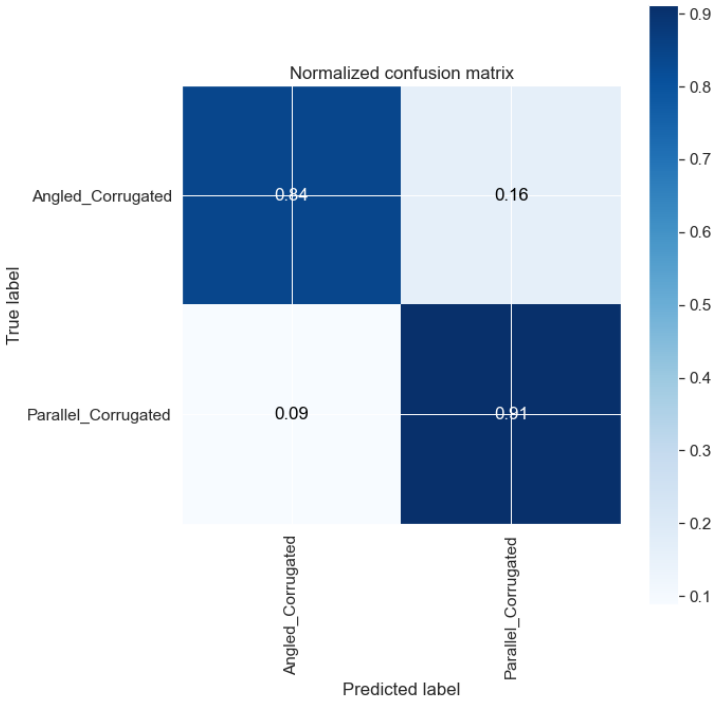

Figure 5.32: Confusion matrix plot for the downward-facing binary classification between the angled and parallel corrugated terrains for the pretrained model

The classification prediction results were not as good as Section 5.4 and Section 5.5 where multiple classes were used that focused on the downward-facing dataset. This indicated that there were strong correlations between the angled and parallel corrugated terrains, which made it more challenging to distinguish between them. The forward-facing results all achieved an accuracy of 100%. The downward-facing results were 87.5%. The downward-facing data was captured significantly closer to the actual road than the forward-facing data. The downward-facing data did not capture the spatial frequency differences between the angled and parallel corrugated road profiles the same way the forward-facing data did. This made it more difficult to classify the downward-facing data than the forward-facing data.

5.8 Summary

Chapter 5 discussed the results from the classification models and the two dataset setups. This first step was to ensure that the preprocessing was done correctly and that both datasets did not contain unwanted noise. Noise was only present in the forward-facing dataset.

Table 5.15 summarises the training, validation and prediction results for the forward- and downward-facing datasets using both the multiple-class classifier and pretrained classifier.

Table 5.15: Summary of the training, validation and prediction results for the forward- and downward-facing datasets using both the multiple-class classifier and pretrained classifier

Model	Training results [%]	Validation results [%]	Prediction results [%]
Multiple-class classifier using forward-facing dataset	100	99.3	99.71
Multiple-class classifier using downward-facing dataset	100	99–100	99.65
Pretrained model using forward-facing dataset	100	99–100	100
Pretrained model using downward-facing dataset	100	99–100	100

The training and validation results were all close to 100%. The first difference between the multiple-class classifier and pretrained classifier was that overfitting occurred after a certain number of epochs for the multiple-class classifier. Overfitting did not occur for the pretrained classifier. The number of epochs used for the multiple-class classifier was higher, which could be the reason why overfitting occurred in only the one setup. The number of epochs was not increased for the pretrained classifier because of memory limitations and because the validation accuracy and loss values were satisfactory.

The second difference between the two models was the sudden spikes that occurred more frequently with the pretrained classifier due to a lower batch size being used because of memory limitations. Even with these two differences, both models achieved prediction accuracy results close to 100%. However, the pretrained model did achieve slightly better results, which could be due to the pretrained model being trained on far larger datasets than the dataset used in this project. This could have allowed the model to train and learn useful features, even though it trained on completely different datasets, which assisted the classifier to make better predictions.

It was suspected that the models would only be able to classify off-road terrains it had trained on and would not be able to classify the class of an unknown terrain. To test this theory, a second approach was followed whereby certain off-road terrains were omitted from the training phase. These terrains were reintroduced in the prediction phase and the results captured. The results showed that the models were able to predict the omitted classes of off-road terrains, but only if they shared similarities with other terrains in that class. Once a terrain has been omitted that looked completely different than the other terrain in its class, the prediction results were unsatisfactory. This was the case when the rough terrain was removed from class H. It appeared that the models relied on the colour of the off-road terrain to make predictions instead of using the terrain's roughness value. These results were summarised in Table 5.6, Table 5.9 and Table 5.12.

A final study was done to establish whether a binary classifier could distinguish between two very similar terrains. The binary classifier tried to predict between the parallel corrugated and angled corrugated terrains. The results were very interesting. For the forward-facing dataset, both models achieved a 100% prediction accuracy. For the downward-facing dataset, the multiple-class classifier got a prediction accuracy of 87.5% and the pretrained model got a prediction accuracy of 84%. This is because the forward-facing dataset captured the data from a further distance revealing the spatial frequency difference and orientation differences between the two terrains. These spatial frequency differences were not as clear for the downward-facing dataset.

Chapter 6: Conclusion and recommendations

6.1 Conclusion

The study set out to develop a model that could perform off-road terrain classification before a vehicle drives over that specific part of the road. Existing research showed that other vision-based classification studies used cameras, lidar and radar, with most studies using cameras. Several classification techniques were used in these studies, with most using neural networks or an SVM classifier. More recent studies used CNN models to classify camera data. All the studies had relatively small datasets and limited off-road terrain classes. The way the classes were picked was also not based on any concrete standard.

Based on past results and after conducting a literature study, it was decided to use a camera sensor to capture the data. A CNN model was chosen to perform the classification due to its capability to perform classification predictions accurately on image data. Two main CNN models were created, namely a multiple-class classifier built from scratch and a pretrained CNN model. The data was captured at the Gerotek Test Facilities and three different classes were selected using the ISO 8608 (2016) standard:

- Class B:
    1. Ride and handling track
- Class D:
    1. Parallel and angled corrugations
    2. Belgian paving
    3. Fatigue track
- Class H:
    1. Rough track
    2. Gerotek rally track (gravel track)

Two camera setups were used: a downward-facing camera pointing directly at the ground and a forward-facing camera setup pointed at an angle between 5° and 15° relative to the horizon. A total number of 15 308 images was captured. Due to some classes consisting of fewer images than others, 12 645 images were used during the actual training, testing and validation phase. A train test split of 70/30% was used. For the downward-facing setup, 4 095 training images and 1 755 testing images were used, whereas for the forward-facing setup, 4 756 training images and 2 039 testing images were used.

The forward-facing data required preprocessing to ensure that only relevant information about the off-road terrain was present in each data set. After preprocessing, the images were trained and validated using the CNN model. The trained CNN model was used to make predictions regarding unseen off-road terrains. The following prediction results were achieved:

- Multiple-class classifier on downward-facing data – 99.65%
- Multiple-class classifier on forward-facing data – 99.71
- Pretrained model on downward-facing data – 100%
- Pretrained model on forward-facing data – 100%

A second study was done to determine whether one of the off-road profiles from class B, class D or class H was omitted. If the classifier was able to predict the class of the untrained terrain correctly, training was done on the rest of the off-road terrains. The same classification models were used as with Section 5.4 and Section 5.5. The results showed that the models could predict the class of untrained terrain accurately if it shared similarities with the rest of the off-road terrain profiles that formed part of that class. When the

terrain significantly differed from the rest of the off-road terrains in its class, the prediction accuracy results decreased. This was expected because a supervised learning model was used that could only make predictions based on the features it was trained on. It was further expected that the models would classify the terrains based on their colour instead of the actual terrain roughness. This was seen when the rough track was omitted from class H. The rough track had a different colour than the rest of the off-road terrains in class H, resulting in unsatisfactory results. When a terrain was omitted that shared similar colour features with terrains in the same class, the results were satisfactory.

Although terrain classification using image data and a CNN was possible and could be achieved with great accuracy, it did have limitations. The model was only able to predict the classes of the terrains accurately if it had been trained on that class or if the untrained terrain shared close similarities with the other terrain profiles. A supervised learning model had the limitation in that it required the label of the class during the training phase. This meant the class type of the road was required as an input into the model.

6.2 Recommendations and future work

Although there were limitations to this study, the limitations of the current results do create opportunity for improvement. The following recommendations are made based on the results and limitations discovered during this dissertation.

6.2.1 Different road profiles

The project can be expanded by classifying new and different road profiles. The road profiles used in this study were captured at a testing facility, namely Gerotek Test Facilities. To use this classifier in real life, the model would have to be trained on all the different types of road it would be required to drive over. This makes the accuracy of the classifier dependent on the location where it will be used. Since a supervised learning model is used, the model must be trained on each of the road profiles it will be driving on.

Data of the off-road terrains should further be captured during different weather conditions. The dataset is very limited in the sense that it only includes data captured at a specific time of the day under ideal sunny conditions. The models would potentially give unsatisfactory results if weather conditions changed and the colour of the off-road terrains changed subsequently. Therefore, it is necessary to capture a more robust dataset for the same terrains.

6.2.2 Types of learning

Supervised learning is limited by its capabilities in the sense that all the data captured requires labelling. If a new type of road is given to the model, it will not be able to classify the type of road because the model had not been trained on that type of road. To improve this, semi-supervised, unsupervised or reinforcement learning can be used. Although using these types of learning model is more difficult, it improves the quality and capabilities of the model. With a more advance learning model, the current model can attempt to make a class prediction of an unseen terrain. A second sensor can be used to measure the type of road the vehicle drives over and either confirm the prediction given or update the model.

6.2.3 Stereo vision

Stereo vision is an extraction of 3D information from digital images. Stereo vision can be used to capture the exact depth and position of objects inside an image. The current solution can classify the type of road and use that information to make a prediction regarding the roughness of that road. Stereo vision can be used to capture the distance to an object in an image.

6.2.4 Multiple sensors

Multiple sensors can further improve the quality of the classification model. A camera can be used to classify the type of road. A second sensor, for example, an IMU sensor, can then be used to measure the acceleration and change in the direction of the vehicle's chassis and suspension. These results can be plugged into a cost function, and the supervised learning model can be expanded further to a semi-supervised model or an unsupervised learning model.

Much can be done to improve the future capabilities of the project and continue the research that can be solved.

6.3 Limitation

In this study, lighting was a limitation. If too much light shines directly into the camera, it can prevent the camera from classifying the type of road. The success of the model is highly dependent on the quality of the data captured. CNN models perform best when high-quality data is provided to the model. Therefore, it is crucial to ensure that the data is captured correctly. The project was also limited to the tracks at Gerotek.

References

Agarwal, V., 2020. Different colorspaces as inputs to CNNs. [Online] Available at: https://towardsdatascience.com/different-colorspaces-as-inputs-to-cnns-406ae62d1bd6 [Accessed 18 03 2022].

Al-Shweiki, J., 2021. Which image resolution should I use for training for deep neural network?. [Online] Available at: https://www.researchgate.net/post/Which_Image_resolution_should_I_use_for_training_for_deep_neural_network [Accessed 13 11 2021].

Armstrong, J., 2018. How do driveless cars work?. [Online] Available at: https://www.telegraph.co.uk/cars/features/how-do-driverless-cars-work/ [Accessed 12 09 2019].

Autocrypt, 2021. Camera, radar and LiDAR: A comparison of the three types of sensors and their limitations. [Online] Available at: https://autocrypt.io/camera-radar-lidar-comparison-three-types-of-sensors/ [Accessed 10 03 2022].

Basavarajaiah, M., 2019. Maxpooling vs minpooling vs average pooling. [Online] Available at: https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9 [Accessed 23 12 2021].

Baskin, C., Liss, N., Mendelson, A. & Zheltonozhskii, E., 2017. Streaming architecture for large-scale quantized neural networks on an FPGA-based dataflow platform. [Online] Available at: https://www.researchgate.net/figure/Image-convolution-with-an-input-image-of-size-7-7-and-a-filter-kernel-of-size-3-3_fig1_318849314 [Accessed 29 09 2020].

Basler Information, 2021. *Basler dart – Small cameras with big performance*. [Online] Available at: https://www.baslerweb.com/en/embedded-vision/embedded-vision-portfolio/embedded-vision-cameras/#frame_rate=20;resolution=y12mpx [Accessed 15 12 2021].

Becker, C. M., 2008. *Profiling of Rough Terrain,* Pretoria: University of Pretoria.

Brownlee, J., 2019. How to one hot encode sequence data in Python. [Online] Available at: https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/ [Accessed 07 03 2021].

Brownlee, J., 2021. How to choose an activation function for deep learning. [Online] Available at: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/ [Accessed 23 12 2021].

Budhiraja, A., 2016. Dropout in (deep) machine learning. [Online] Available at: https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5 [Accessed 4 10 2019].

Burke, K., 2019. How does a self-driving car see?. [Online] Available at: https://blogs.nvidia.com/blog/2019/04/15/how-does-a-self-driving-car-see/ [Accessed 3 10 2020].

Byl, B. and Filitchkin, P., 2012. Feature-based terrain classification for LittleDog. *IEEE/RSJ International Conference on Intelligent Robots and Systems,* p. 6.

Chazareix, A., 2020. About convolutional layer and convolution kernel. [Online] Available at: https://www.sicara.ai/blog/2019-10-31-convolutional-layer-convolution-kernel [Accessed 23 12 2021].

Chollet, F., 2018. Deep learning with Python. In: *Deep Learning With Python.* s.l.: Manning Publications, pp. 143–180.

Coyle, E., 2010. *Fundamentals and Methods of Terrain Classification Using Proprioceptive Sensors,* Tallahassee, FL: Florida State University Libraries.

Ferdinand, N., 2020. A simple guide to convolutional neural networks. [Online] Available at: https://towardsdatascience.com/a-simple-guide-to-convolutional-neural-networks-751789e7bd88 [Accessed 29 09 2020].

Francis, S., 2017. Honda aiming for level 4 automated driving capability by 2025. [Online] Available at: http://roboticsandautomationnews.com/2017/06/08/honda-aiming-for-level-4-automated-driving-capability-by-2025/12691/ [Accessed 25 02 2020].

Giese, T., Klappstein, J., Dickmann, J. & Wohler, C., 2017. Road course estimation using deep learning on radar data. [Online] Available at: https://ieeexplore.ieee.org/abstract/document/8008125/authors#authors [Accessed 5 10 2019].

Gillespie, T. D., 1992. *Fundamentals of Vehicle Dynamics,* Warrendale, PA: Society of Automotive Engineers, Inc.

Godoy, D., 2018. Understanding binary cross-entropy/log loss: A visual explanation. [Online] Available at: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a [Accessed 21 11 2022].

Harner, I., 2020. The 5 autonomous driving levels explained. [Online] Available at: https://www.iotforall.com/5-autonomous-driving-levels-explained/ [Accessed 25 02 2020].

Hendrickson, J., 2019. What are the different self-driving car 'levels' of autonomy?. [Online] Available at: https://www.howtogeek.com/401759/what-are-the-different-self-driving-car-levels-of-autonomy/ [Accessed 18 09 2019].

Howard, A. & Seraji, H., 2001. *Vision-based Terrain Characterization and Traversability Assessment,* Pasadena, CA: California Institute of Technology.

Jaumier, P., 2019. Backpropagation in a convolutional layer. [Online] Available at: https://towardsdatascience.com/backpropagation-in-a-convolutional-layer-24c8d64d8509 [Accessed 29 06 2020].

Jeong, J., 2019. The most intuitive and easiest guide for convolutional neural network. [Online] Available at: https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480#:~:text=Rectangular%20or%20cubic%20shapes%20can,a%20single%20long%20feature%20vector [Accessed 23 12 2021].

Keras Help Documentation, 2020. *Keras Applications.* [Online] Available at: https://keras.io/api/applications/ [Accessed 08 06 2020].

Khvoynitskaya, S., 2020. 3 types of autonomous vehicle sensors in self-driving cars. [Online] Available at: https://www.itransition.com/blog/autonomous-vehicle-sensors [Accessed 25 02 2020].

Kizrak, A., 2019. Comparison of activation functions for deep neural networks: Step, linear, sigmoid, hyperbolic tangent, softmax, ReLU, leaky ReLU, and swish functions are explained with hands-on!. [Online] Available at: https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a [Accessed 11 06 2020].

Kumar, S., 2020. Data splitting technique to fit any machine learning model. [Online] Available at: https://towardsdatascience.com/data-splitting-technique-to-fit-any-machine-learning-model-c0d7f3f1c790 [Accessed 23 12 2021].

Lu, L., Ordonez, C., Collins, E. G. & DuPont, E. M, 2009. Terrain surface classification for autonomous ground vehicles using a 2D laser stripe-based structured light sensor. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2174–2181.

LiDAR and RADAR Information, 2017. *Advantages and disadvantages of RADAR systems.* [Online] Available at: https://lidarradar.com/info/advantages-and-disadvantages-of-radar-systems [Accessed 18 07 2019].

Manduchi, R., Castano, A., Matthies, L. & Talukder, A., 2005. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robots*, vol. 18, no. 1, pp. 81-102.

National Ocean Service, 2016. *What is lidar?.* [Online] Available at: https://oceanservice.noaa.gov/facts/lidar.html [Accessed 19 07 2019].

Omer, R. & Fu, L., 2010. An automatic image recognition system for winter road surface condition classification. [Online] Available at: https://ieeexplore.ieee.org/abstract/document/5625290/authors#authors [Accessed 26 09 2019].

Park, O., 2017. *A brief explanation of colour formats.* [Online] Available at: https://castfromclay.co.uk/commentary/a-brief-explanation-of-colour-formats/ [Accessed 15 12 2021].

Review Autopilot, 2017. LiDAR vs. Cameras for self driving cars – what's best?. [Online] Available at: https://www.autopilotreview.com/lidar-vs-cameras-self-driving-cars/ [Accessed 6 10 2020].

Rosebrock, A., 2015. *Find distance from camera to object/marker using Python and OpenCV.* [Online] Available at: https://pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/ [Accessed 10 03 2022].

Roychowdhury, S., Zhoa, M., Wallim, A. & Ohlsson, N., 2019. Machine learning models for road surface and friction estimation using front-camera images. *International Joint Conference on Neural Networks (IJCNN)* [Online] Available at: https://www.researchgate.net/publication/328402375_Machine_Learning_Models_for_Road_Surface_and_Friction_Estimation_using_Front-Camera_Images [Accessed 10 09 2019].

Rudolph, G. & Voelzke, U., 2017. *Three sensor types autonomous vehicles.* [Online] Available at: https://www.fierceelectronics.com/components/three-sensor-types-drive-autonomous-vehicles [Accessed 02 08 2019].

Sahoo, S., 2018. Deciding optimal kernel size for CNN. [Online] Available at: https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363 [Accessed 23 12 2021].

Schulz-Mirbach, H., 1995. Invariant features for gray scale images. [Online] Available at: https://link.springer.com/chapter/10.1007/978-3-642-79980-8_1 [Accessed 16 03 2022].

Selvathai, T., Varadhan, J. & Ramesh, S., 2017. Road and off road terrain classification for autonomous ground vehicle. *International Conference on Information, Communication & Embedded Systems (ICICES 2017),* p. 3.

Shaban, A., Meng, X., Lee, J., Boots, B. & Foz, D., 2021. *Semantic Terrain Classification for Off-Road Autonomous Driving,* s.l.: University of Washington.

Shao, C., 2019. Approach pre-trained deep learning models with caution. [Online] Available at: https://medium.com/comet-ml/approach-pre-trained-deep-learning-models-with-caution-9f0ff739010c [Accessed 18 03 2022].

Sharma, B., 2019. What is LiDAR technology and how does it work?. [Online] Available at: https://www.geospatialworld.net/blogs/what-is-lidar-technology-and-how-does-it-work/ [Accessed 20 08 2019].

Sharma, P., n.d. MaxPool vs AvgPool. [Online] Available at: https://iq.opengenus.org/maxpool-vs-avgpool/#:~:text=As%20you%20may%20observe%20above,the%20features%20in%20the%20image [Accessed 18 03 2022].

Sharma, S., 2017a. Activation functions in neural networks. [Online] Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 [Accessed 06 09 2022].

Sharma, S., 2017b. Epoch vs batch size vs iterations. [Online] Available at: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9 [Accessed 23 12 2021].

Shen, K., 2018. Effect of batch size on training dynamics. [Online] Available at: https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e [Accessed 23 12 2021].

Stack Exchange Information, 2019. What is the cause of the sudden drop in error rate that often sees when training a CNN?. [Online] Available at: https://stats.stackexchange.com/questions/312766/what-is-the-cause-of-the-sudden-drop-in-error-rate-that-one-often-sees-when-trai [Accessed 26 12 2021].

Stolee, J. & Wang, Y., 2018. *A Survey of Machine Learning Techniques for Road Detection,* Toronto, ON: Toronto University.

Sung, G. Y., Kwak, D.-M. & Lyou, J., 2010. Neural network based terrain classification using wavelet features. *Journal of Intelligent & Robotic Systems,* vol. 59, pp. 269–281.

SuperDataScience Team, 2018. Convolutional neural networks (CNN): Step 3 – Flattening. [Online] Available at: https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening [Accessed 18 03 2020].

Thrun, S., Montemerlo, M., Dahlkamp, H. & Stavens, D., 2006. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, vol. 23, pp. 661–692.

Tripathi, M., 2021. Image processing using CNN: A beginners guide. [Online] Available at: https://www.analyticsvidhya.com/blog/2021/06/image-processing-using-cnn-a-beginners-guide/#:~:text=CNN%20is%20a%20powerful%20algorithm,contain%20data%20of%20RGB%20combination [Accessed 10 03 2022].

Upadhyay, D., 2017. How a computer looks at pictures: Image classification. [Online] Available at: https://medium.com/datadriveninvestor/how-a-computer-looks-at-pictures-image-classification-a4992a83f46b

Verma, S., 2019. Understanding 1D and 3D convolution neural network | Keras. [Online] Available at: https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610 [Accessed 24 02 2020].

Versloot, C., 2018. *Convolutional neural networks and their components for computer vision.* [Online] Available at: https://www.machinecurve.com/index.php/2018/12/07/convolutional-neural-networks-and-their-components-for-computer-vision/ [Accessed 7 12 2021].

Walia, A. S., 2017. Types of optimization algorithm used in neural networks and ways to optimize gradient descent. [Online] Available at: https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f [Accessed 20 04 2020].

Wang, S., 2019. Unmanned system technologies. In: *Road Terrain Classification Technology for Autonomous Vehicle.* Changchun: Springer Nature Singapore, pp. 16–18.

Webster, S., 2017. Self-driving cars explained: How do self-driving cars work – and what do they mean for the future?. [Online] Available at: https://www.ucsusa.org/resources/self-driving-cars-101 [Accessed 25 02 2020].

Weiss, C., Fechner, N., Stark, M. & Zell, A., 2007. Comparison of different approaches to vibration-based terrain classification. *Proceedings of the 3rd European Conference on Mobile Robots,* EMCR 2007, September 19–21, 2007, Freiburg, Germany, p. 6.

Weiss, C., Tamimi, H. & Zell, A., 2008. A combination of vision- and vibration-based terrain classification. *IEEE/RSJ International Conference on Intelligent Robots and Systems,* p. 6.

Xiang, Z. & Seeling, P., 2020. Training convolutional nets to detect calcified plaque in IVUS sequences. [Online] Available at: https://www.sciencedirect.com/topics/engineering/convolutional-layer [Accessed 03 03 2021].

Yun, T. S., Santamarine, J. C. & Ruppel, C., 2007. Mechanical properties of sand, silt, and clay containing tetrahydrofuran hydrate. *Journal of Geophysical Research*, vol. 12, no. B4. [Online] Available at: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2006JB004484 [Accessed 09 03 2021].

Appendix A: Accuracy and loss value plots for untrained off-road terrains

A.1. Classification without angled corrugated off-road terrain for class D

Figure A.1 to Figure A.8 show the training and validation accuracy results and the training and validation loss values for the untrained parallel corrugated road terrain from class D.

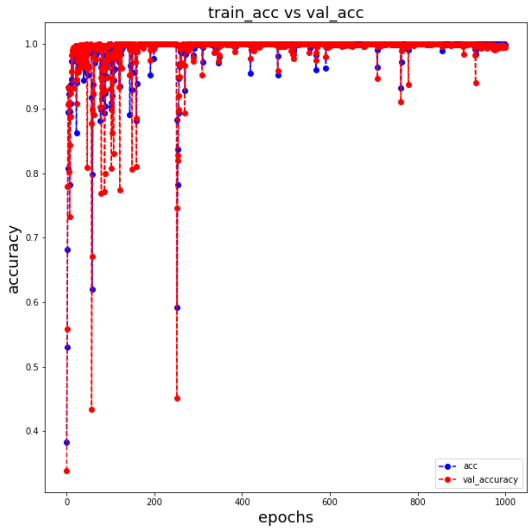

Figure A.1: Training and validation accuracy for the forward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier

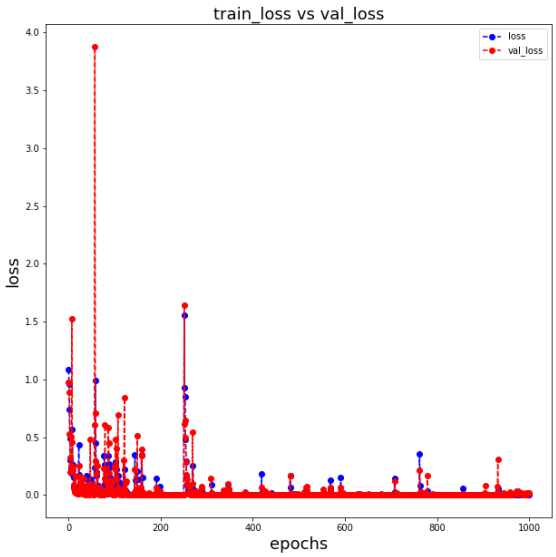

Figure A.2: Training and validation loss value for the forward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier

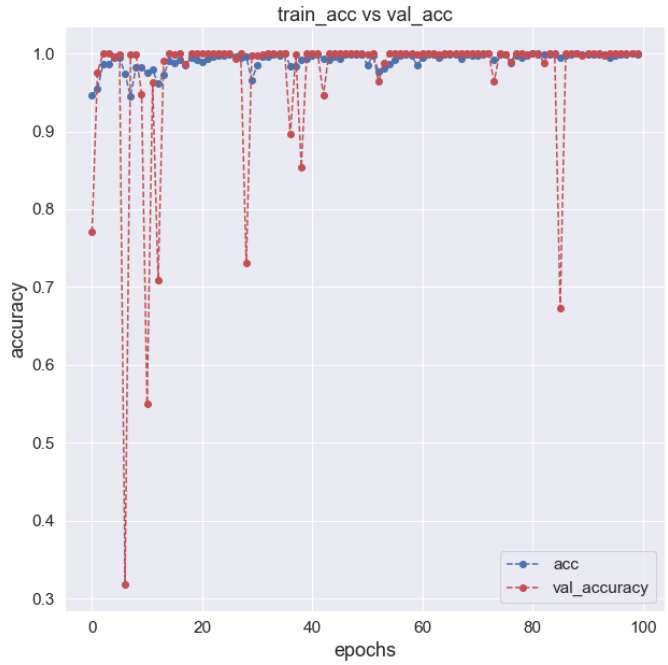

Figure A.3: Training and validation accuracy for the forward-facing camera setup omitting the parallel corrugated terrain pretrained model

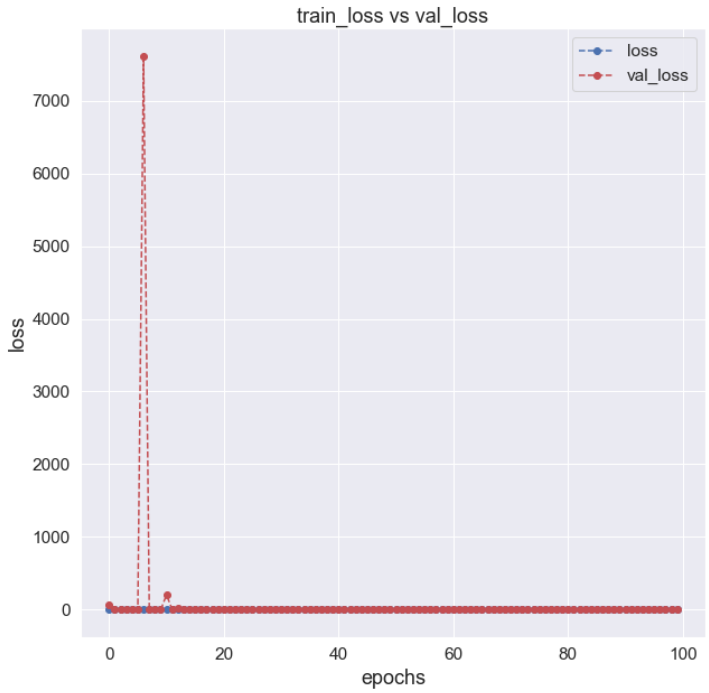

Figure A.4: Training and validation loss value for the forward-facing camera setup omitting the parallel corrugated terrain pretrained model

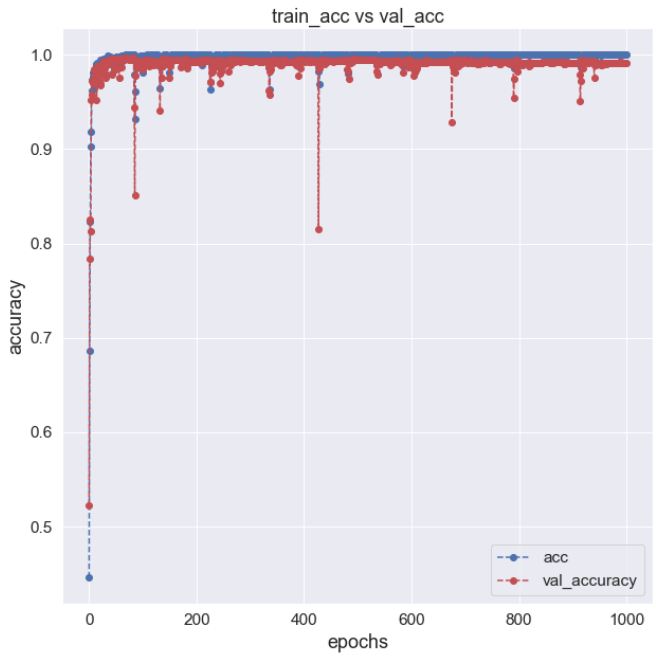

Figure A.5: Training and validation accuracy for the downward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier

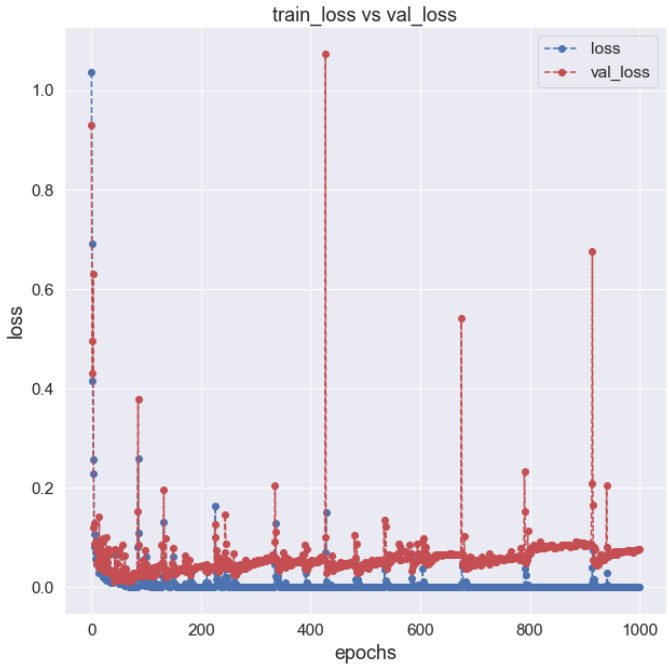

Figure A.6: Training and validation loss value for the downward-facing camera setup omitting the parallel corrugated terrain multiple-class classifier

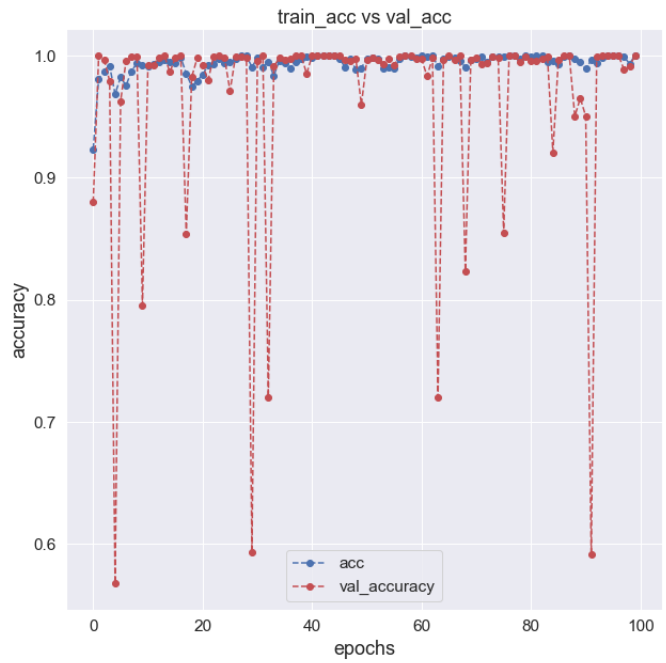

Figure A.7: Training and validation accuracy for the downward-facing camera setup omitting the parallel corrugated terrain pretrained model

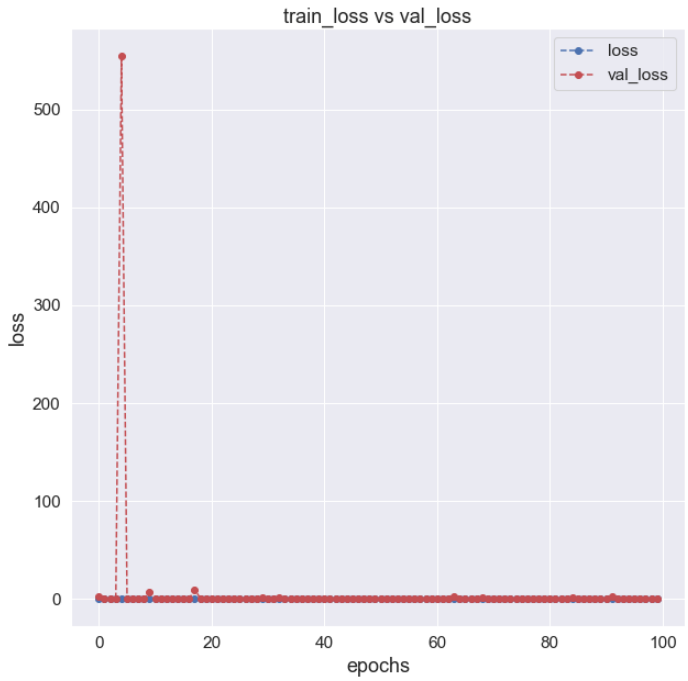

Figure A.8: Training and validation loss value for the downward-facing camera setup omitting the parallel corrugated terrain pretrained model

A.2. Classification without fatigue track for class D

Figure A.9 to Figure A.16 show the training and validation accuracy results and the training and validation loss values for the untrained fatigue track from class D.

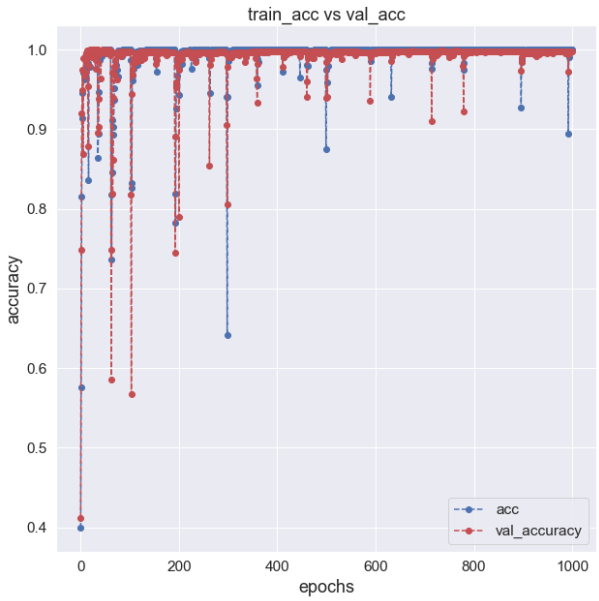

Figure A.9: Training and validation accuracy for the forward-facing camera setup omitting the fatigue track multiple-class classifier

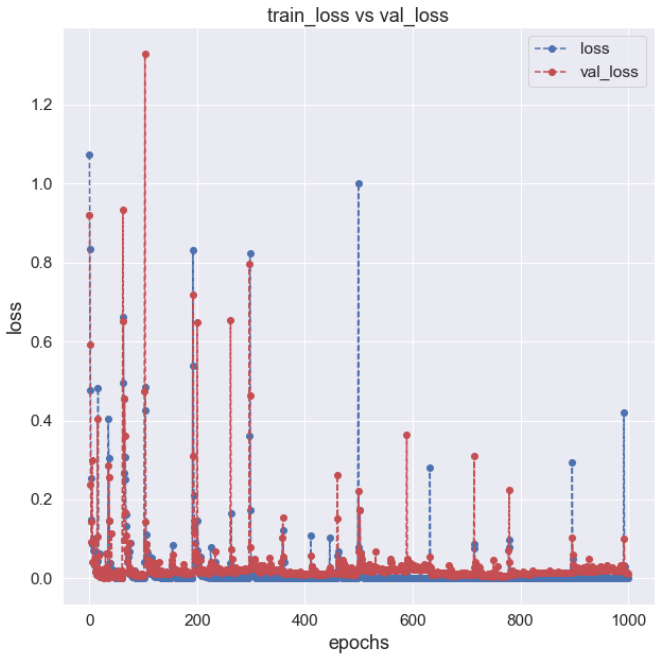

Figure A.10: Training and validation loss value for the forward-facing camera setup omitting the fatigue track multiple-class classifier

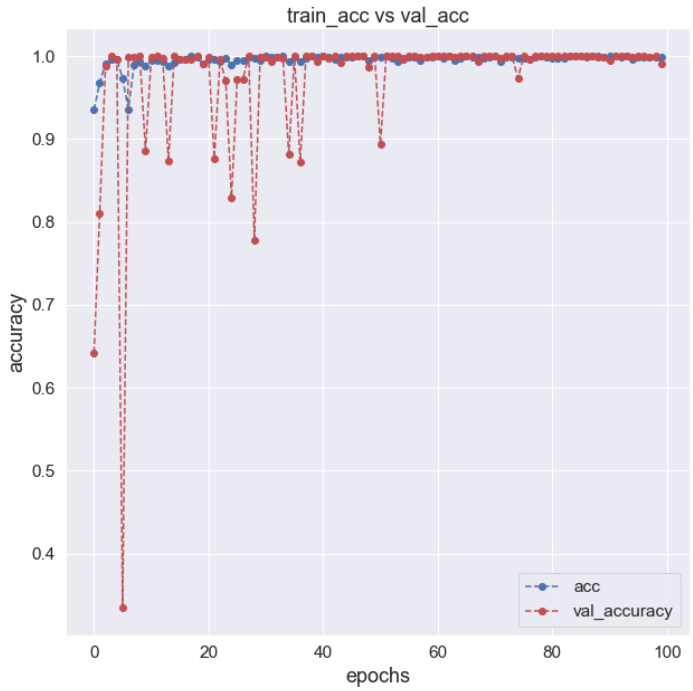

Figure A.11: Training and validation accuracy for the forward-facing camera setup omitting the fatigue track pretrained model

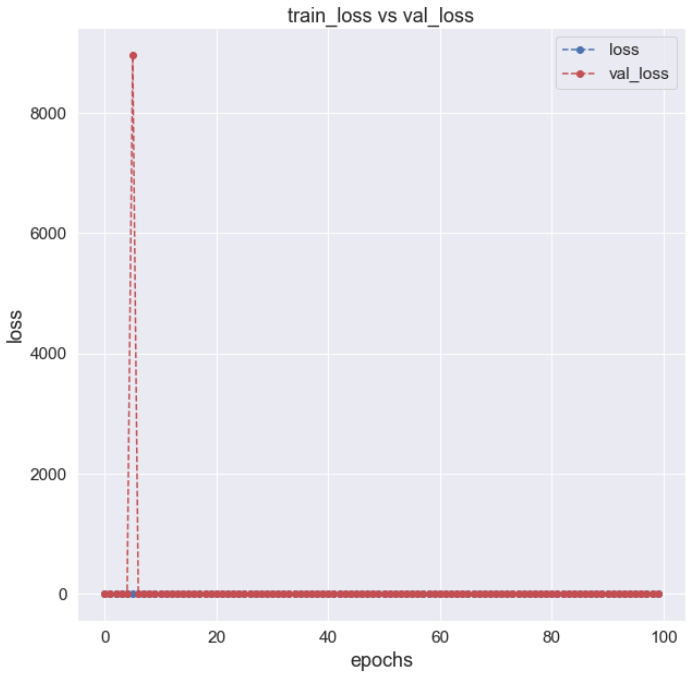

Figure A.12: Training and validation loss value for the forward-facing camera setup omitting the fatigue track pretrained model

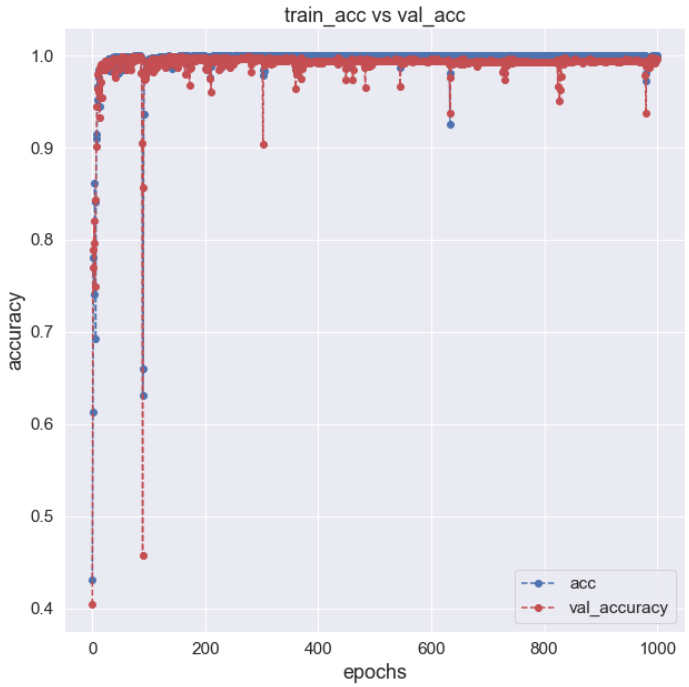

Figure A.13: Training and validation accuracy for the downward-facing camera setup omitting the fatigue track multiple-class classifier

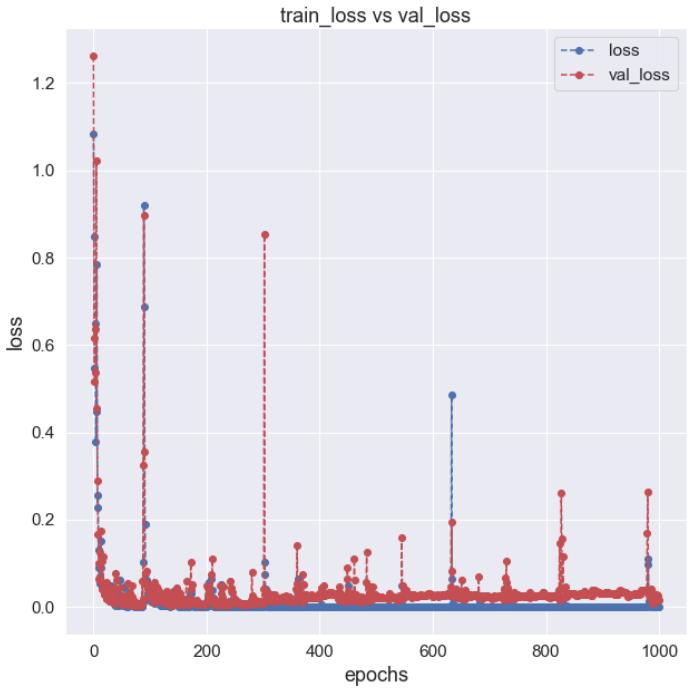

Figure A.14: Training and validation loss value for the downward-facing camera setup omitting the fatigue track multiple-class classifier

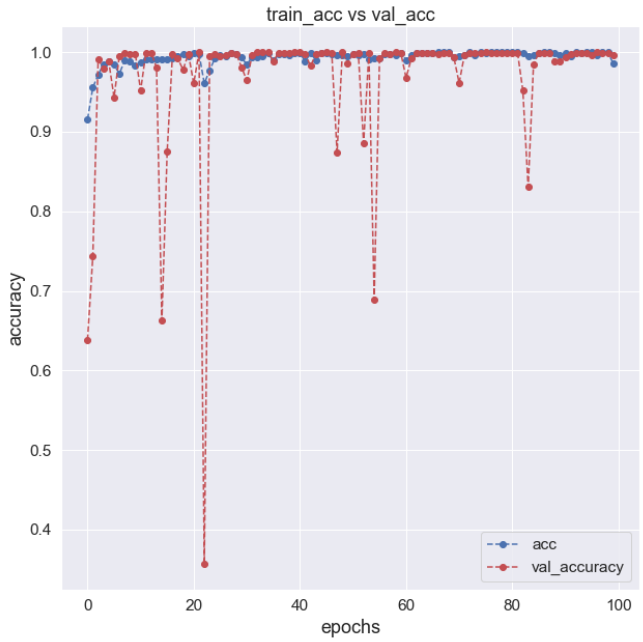

Figure A.15: Training and validation accuracy for the downward-facing camera setup omitting the fatigue track pretrained model

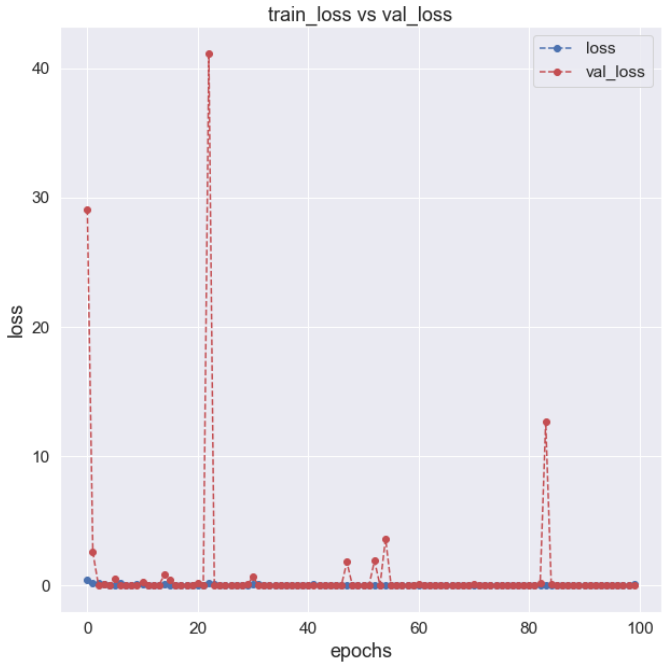

Figure A.16: Training and validation loss value for the downward-facing camera setup omitting the fatigue track pretrained model

A.3. Classification without rough track for class H

Figure A.17 to Figure A.24 show the training and validation accuracy results and the training and validation loss values for the untrained rough track from class H.

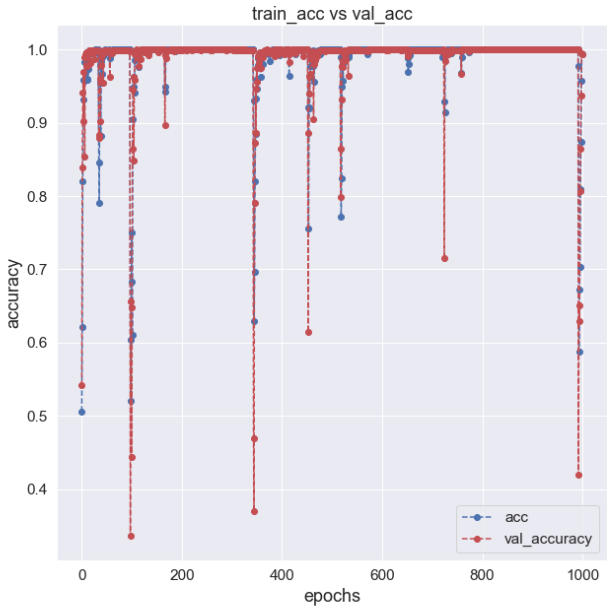

Figure A.17: Training and validation accuracy for the forward-facing camera setup omitting the rough track multiple-class classifier

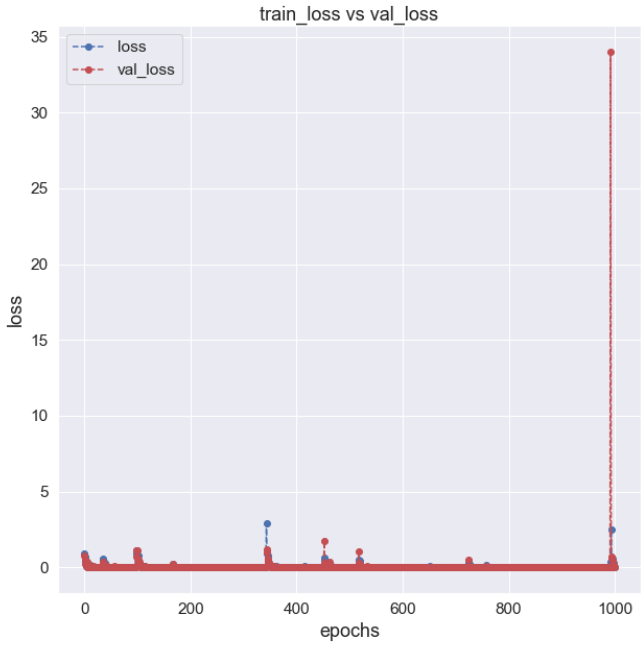

Figure A.18: Training and validation loss value for the forward-facing camera setup omitting the rough track multiple-class classifier

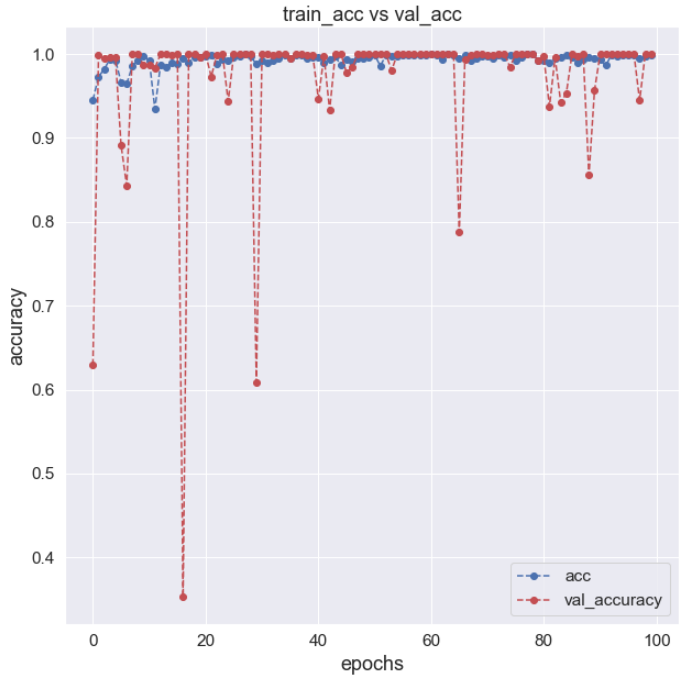

Figure A.19: Training and validation accuracy for the forward-facing camera setup omitting the rough track pretrained model

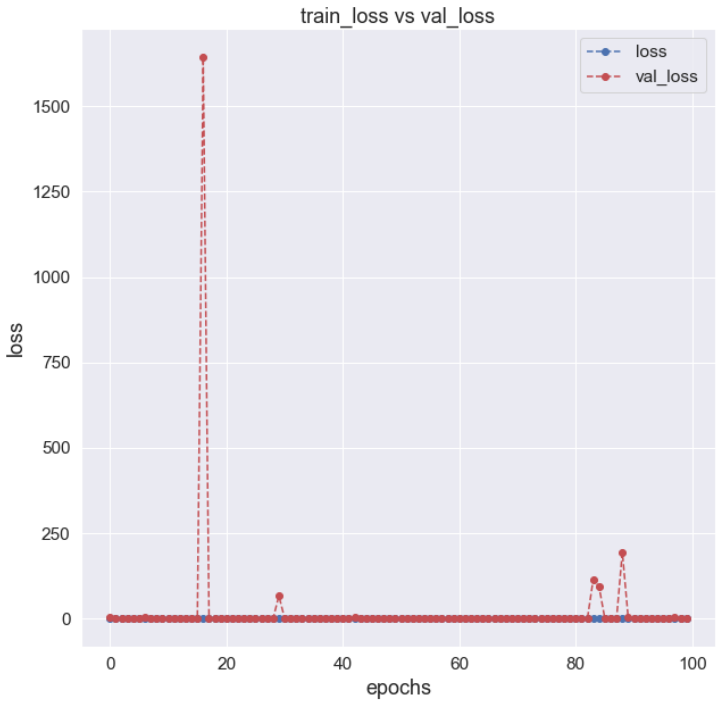

Figure A.20: Training and validation loss value for the forward-facing camera setup omitting the rough track pretrained model

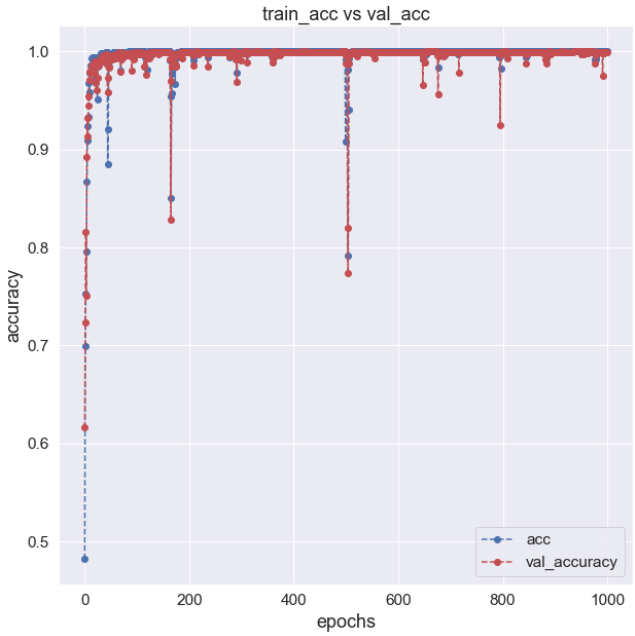

Figure A.21: Training and validation accuracy for the downward-facing camera setup omitting the rough track multiple-class classifier

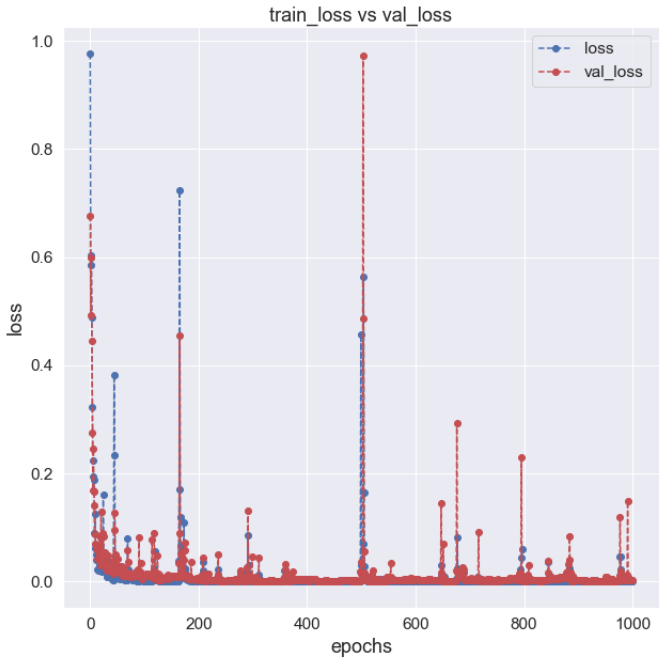

Figure A.22: Training and validation loss value for the downward-facing camera setup omitting the rough track multiple-class classifier

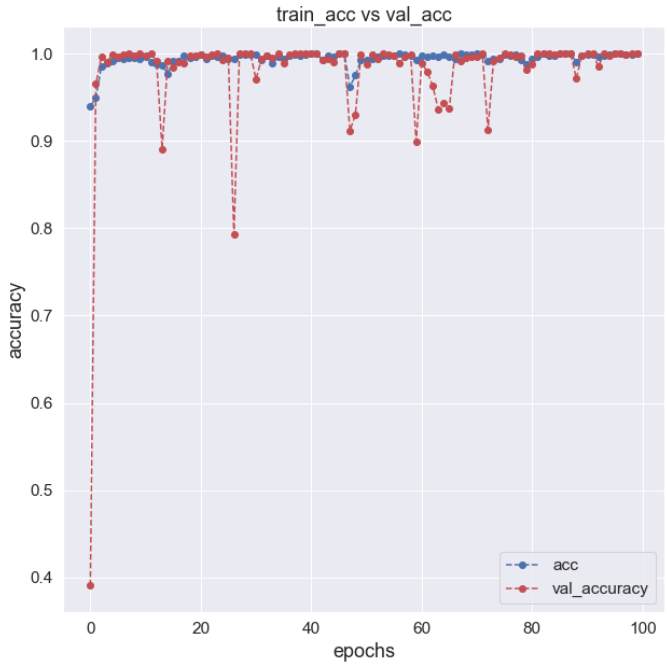

Figure A.23: Training and validation accuracy for the downward-facing camera setup omitting the rough track pretrained model

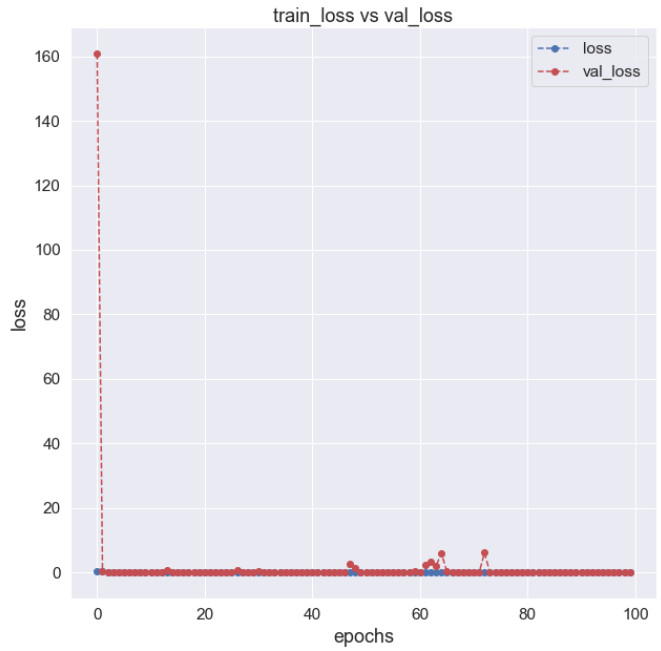

Figure A.24: Training and validation loss value for the downward-facing camera setup omitting the rough track pretrained model