Correcting inter-sectional accuracy differences in drowsiness detection systems using generative adversarial networks (GANs)

Submitted in partial fulfillment of the requirements of
Doctor of Philosophy Degree (Computer Eng.)

By

Mkhuseli Ngxande
Student No. 216077070

Under the supervision of:

Prof. Jules-Raymond Tapamo
&
Dr Michael Burke

UNIVERSITY OF
KWAZULU-NATAL

University of Kwa-Zulu Natal

February 2020

UNIVERSITY OF KWAZULU-NATAL, COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE DECLARATION

The research described in this thesis was performed at the University of KwaZulu-Natal under the supervision of Prof. Jules-Raymond Tapamo and Dr Michael Burke. I hereby declare that all materials incorporated in this thesis are my own original work except where the acknowledgement is made by name or in the form of reference. The work contained herein has not been submitted in part or whole for a degree at any other university.

Signed: MNgxande
Name: Mkhuseli Ngxande
Date: February 2020

As the candidate's supervisor, I have approved this thesis for submission.

Signed:..
Name: Prof. Jules-Raymond Tapamo
Date: February 2020

As the candidate's co-supervisor, I have approved this thesis for submission.

Signed:...MBurke..........................
Name: Dr Michael Burke
Date: February 2020

UNIVERSITY OF KWAZULU-NATAL, COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE DECLARATION 2 - PUBLICATIONS

I, Mkhuseli Ngxande, declare that the following publications from part of this dissertation.

1. M. Ngxande, J.-R. Tapamo, and M. Burke, “Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques,” in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, 2017, pp.156–161 [Online]. Available: `https://ieeexplore.ieee.org/document/8261140/`

2. M. Ngxande, J.-R. Tapamo, and M. Burke, “DepthwiseGANs: Fast Training Generative Adversarial Networks for Realistic Image Synthesis ,” in *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, 2019, pp.111–116 [Online]. Available: `https://ieeexplore.ieee.org/document/8704766`

3. M. Ngxande, J.-R. Tapamo, and M. Burke, “Bias Correction in population-based Driver Drowsiness Dataset through Generative Adversarial Network,”*IEEE Transactions on Intelligent Transportation Systems*, 2019 [**Under Review**]

4. M. Ngxande, J.-R. Tapamo, and M. Burke, “Bias Remediation in Driver Drowsiness Detection systems using Generative Adversarial Networks,”*IEEE Access*, 2019 [**Under Review**]

Signed:...

Dedication

This thesis is dedicated to my Grandmother.

Acknowledgements

This work was supported and funded by the Council for Scientific and Industrial Research (CSIR) through a studentship programme. I am grateful to have this opportunity to fulfill my dreams.

To my supervisor, Prof. Tapamo, thank you for giving me this opportunity to work with you and share ideas. I have learned a lot in this three year journey of my Ph.D. studies. To my co-supervisor, Dr Michael Burke, I am deeply grateful for your support, chats, and quick responses. Furthermore, thank you for your undivided time and dedication through this period of my research and from believing in me.

I am grateful to the image processing group where we share ideas and our difficulties. They have introduced me to a lot of industrial ethics which will help to expand my career. The weekly group meetings have strengthened the importance of teamwork and provided support in how to address problems.

To my family, this thesis is for you because you believed in me and my dreams. It was difficult to leave my comfort zone and step out on own but your support and guidance has made it not only possible, but worthwhile.

Abstract

Road accidents contribute to many injuries and deaths among the human population. There is substantial evidence that proves drowsiness is one of the most prominent causes of road accidents all over the world. This results in fatalities and severe injuries for drivers, passengers, and pedestrians. These alarming facts are raising the interest in equipping vehicles with robust driver drowsiness detection systems to minimise accident rates. One of the primary concerns of motor industries is the safety of passengers and as a consequence they have invested significantly in research and development to equip vehicles with systems that can help minimise to road accidents. A number research endeavours have attempted to use Artificial intelligence, and particularly Deep Neural Networks (DNN), to build intelligent systems that can detect drowsiness automatically. However, datasets are crucial when training a DNN. When datasets are unrepresentative, trained models are prone to bias because they are unable to generalise. This is particularly problematic for models trained in specific cultural contexts, which may not represent a wide range of races, and thus fail to generalise. This is a specific challenge for driver drowsiness detection task, where most publicly available datasets are unrepresentative as they cover only certain ethnicity groups. This thesis investigates the problem of an unrepresentative dataset in the training phase of Convolutional Neural Networks (CNNs) models. Firstly, CNNs are compared with several machine learning techniques to establish their superior suitability for the driver drowsiness detection task. An investigation into the implementation of CNNs was performed and highlighted that publicly available datasets such as NTHU, DROZY and CEW do not represent a wide spectrum of ethnicity groups and lead to biased systems. A population bias visualisation technique was proposed to help identify the regions, or individuals where a model is failing to generalise on a picture grid. Furthermore, the use of Generative Adversarial Networks (GANs) with lightweight convolutions called Depthwise Separable Convolutions (DSC) for image translation to multi-domain outputs was investigated in an attempt to generate synthetic datasets. This thesis further showed that GANs can be used to generate more realistic images with varied facial attributes for predicting drowsiness across multiple ethnicity groups. Lastly, a novel framework was developed to detect bias and correct it using synthetic generated images which are produced by GANs. Training models using this framework results in a substantial performance boost.

Contents

Declaration of Authorship i

List of Publications ii

Acknowledgements iv

Abstract v

Contents vi

List of Figures viii

List of Tables x

List of Algorithms xi

List of Acronyms xii

1 Introduction 1
1.1 Motivation 1
1.2 Problem Statement 4
1.3 Aims and Objectives 4
1.4 Contributions to knowledge 5
1.5 Organisation 6

2 Background and Context 9
2.1 Fundamental Concepts of Drowsiness 10
2.1.1 What is drowsiness ? 10
2.1.2 Causes of drowsiness and its signs 11
2.1.2.1 Signs of drowsiness 12
2.1.3 Countermeasures against drowsiness 12
2.1.4 Measurement Methods for Drowsiness Detection 13
2.1.4.1 Physiological Methods 13
2.1.4.2 Subjective Methods 15
2.1.4.3 Vehicle-Based Methods 17
2.1.4.4 Behavioural Methods 18
2.1.4.5 Meta-Analysis 23
2.1.4.6 Hybrid Methods 25
2.2 Commercially Available Systems 26
2.3 Conclusion 27

3 Fundamentals of Deep Learning 30
3.1 Artificial Neural Networks 30
3.2 Convolutional Neural Networks 31
3.2.1 Convolution Layer 32
3.2.2 Activation Functions and Non-linearity 33

3.2.3 Pooling Layers . . . 35
3.2.4 Fully-Connected Layers . . . 36
3.2.5 Batch normalisation . . . 36
3.3 Neural Network Training . . . 37
3.3.1 Loss Function . . . 37
3.4 Optimisation and Learning Rates . . . 38
3.5 Regularisation . . . 41
3.6 CNNs Visualisation . . . 42
3.7 Transfer Learning . . . 44
3.8 Conclusion . . . 45

4 Intersectional accuracy differences in drowsiness detection systems 47
4.1 Related Work . . . 47
4.2 Bias in Machine Learning . . . 49
4.3 Population Bias Visualisation . . . 50
4.4 Datasets . . . 53
4.5 Experiments . . . 55
4.5.1 Architectures . . . 55
4.6 Results . . . 58
4.7 Conclusion . . . 63

5 Generative Adversarial Network (GAN)-based data augmentation 66
5.1 Generative Adversarial Networks (GANs) . . . 66
5.2 Conditional Generative Adversarial Networks (Conditional Generative Adversarial Network (CGAN)) . . . 69
5.3 Comparing Performance of GANs . . . 71
5.4 Depthwise Separable Convolutions . . . 72
5.5 Experiments . . . 73
5.5.1 Architectures . . . 73
5.5.2 Datasets . . . 74
5.5.3 Training process . . . 75
5.6 Results . . . 75
5.6.1 Results on CelebA . . . 77
5.6.2 Results on RaFD and Stirling . . . 80
5.7 Conclusions . . . 81

6 GAN-based bias correction 84
6.1 Boosting . . . 85
6.2 Data Augmentation . . . 86
6.3 Proposed Framework . . . 87
6.3.1 GAN Architecture . . . 88
6.3.2 CNN Architecture . . . 89
6.3.3 Population Bias Visualisation . . . 90
6.3.4 Sampler . . . 90
6.4 Experiments and Results . . . 91
6.4.1 Training . . . 91
6.4.2 Results . . . 92
6.4.3 GAN Augmentation Results . . . 92
6.4.4 Learning rate Results . . . 93
6.4.5 Population Bias Visualisation Results . . . 95
6.5 Conclusion . . . 97

7 Conclusions and future work 99
7.1 Conclusion . . . 99
7.2 Limitations . . . 100
7.3 Future work . . . 100

Bibliography 102

List of Figures

1.1 Road traffic fatality rates . . . 2
1.2 Numbers of road crashes in South Africa . . . 2
1.3 Percentages of road accidents by provinces . . . 3

2.1 Readings of Electroencephalogram (EEG) signals on a person . . . 14
2.2 Driver drowsiness detection process . . . 20
2.3 Meta-Analysis Report . . . 24
2.4 Bayerische Motoren Werke (BMW) active driving assistant . . . 26
2.5 Mercedes-Benz attention assist . . . 27

3.1 Convolution inputs . . . 33
3.2 Sigmoid graph . . . 33
3.3 Hyperbolic Tangent (Tanh) graph . . . 34
3.4 Layer-Activation . . . 42
3.5 Conv Filter . . . 43
3.6 Activation maps . . . 43
3.7 Saliency maps . . . 44
3.8 t-SNE . . . 44

4.1 2-dimensional grid . . . 52
4.2 Organised Principal Component Analysis (PCA) image . . . 53
4.3 NTHU dataset . . . 54
4.4 ULg Multimodality Drowsiness Database (DROZY) . . . 54
4.5 Closed Eyes In The Wild (CEW)dataset . . . 54
4.6 South African dataset . . . 55
4.7 ResNet . . . 56
4.8 Visual Geometry Group (VGG)-19 . . . 56
4.9 Visual Geometry Group-FaceNet (VGG-Face) . . . 56
4.10 Architecture . . . 57
4.11 Training results . . . 57
4.12 Testing results . . . 58
4.13 Saliency results . . . 60
4.14 Testing results on Resnet model . . . 62
4.15 Testing results on VGG model . . . 62
4.16 Testing results on VGG-Face model . . . 63

5.1 Simple GANs architecture . . . 67
5.2 Deep Convolutional Generative Adversarial Network (DCGAN) architecture . . . 68

5.3 DSC vs Standard Convolutions . . . 73
5.4 DepthwiseGAN architecture . . . 74
5.5 Training phase of DepthwiseDG . . . 76
5.6 Training phase of DeeperDepthwiseG . . . 78
5.7 CelebA results . . . 79
5.8 Fréchet Inception Distance (FID) scores of CelebA . . . 79
5.9 FID scores for RaFD dataset . . . 80
5.10 FID score for Stirling dataset . . . 81
5.11 Synthetic data for RaFD and Stirling dataset . . . 81

6.1 Proposed Framework . . . 88
6.2 GAN Architecture . . . 89
6.3 GAN generated image with different facial attributes . . . 94
6.4 Visualisation from Standard datasets . . . 96
6.5 Visualisation from GAN generated images . . . 97

List of Tables

2.1 The Karolinska Sleepiness Scale (KSS) Sleepiness Scale 16
2.2 The Stanford Sleepiness Scale (SSS) ratings 16
2.3 Multiple Sleep Latency Test 17
2.4 Convolutional Neural Network (CNN) based detection systems 21
2.5 Support Vector Machines (SVMs) based detection systems 22
2.6 Hidden Markov Models (HMM) based work related 23

4.1 Accuracies in experimentation 58

5.1 GANsArchitecture information 74

6.1 Generator network architecture information 92
6.2 Discriminator network architecture information 92
6.3 Learning rate - accuracy tradeoff 93
6.4 Model classification accuracy 96

LIST OF ALGORITHMS

1 **Image overlay generation** 52
2 **GAN original algorithm.** 68
3 **Model Training** 77

List of Acronyms

AdaBoost	Adaptive Boosting
Adam	Adaptive Moment Estimation
ADAGRAD	Adaptive Gradient
ADADELTA	Adaptive Learning Rate Method
ANN	Artificial Neural Network
ArtGAN	Artwork Synthesis with Conditional Categorical GAN
AUC	Area under the ROC Curve
ACCV	Asian Conference on Computer Vision
BN	Batch Normalization
BMW	Bayerische Motoren Werke
BC-GAN	Bayesian Conditional Generative Adversarial Network
BAC	Blood Alcohol Concentration
CIFAR-10	Canadian Institute For Advanced Research-10
CEW	Closed Eyes In The Wild
COCO 2	Common Objects in Context
CGAN	Conditional Generative Adversarial Network
CNNs	Convolutional Neural Networks
CNN	Convolutional Neural Network

DAGAN	DataAugmentation Generative Adversarial Networks
DCGAN	Deep Convolutional Generative Adversarial Network
DDD	Deep Drowsiness Detection
Deep-LSTM	Deep Long-Short-Term Memory Network
DNN	Deep Neural Networks
DSC	Depthwise Separable Convolutions
DGAN	DrowsyGAN
DROZY	ULg Multimodality Drowsiness Database
ECG	Electrocardiogram
EEG	Electroencephalogram
EMG	Electromyogram
EOG	Electrooculograms
ESS	Epworth Sleepiness Scale
Eye-Chimera	Eye-Chimera Database
EAR	Eye Aspect Ratio
FFT	Fast Fourier Transform
FI-DDD	Forward Instant Driver Drowsiness Detection
FID	Fréchet Inception Distance
FCNN	Fully-Convolutional Neural Network
GANs	Generative Adversarial Networks
GAN	Generative Adversarial Network
GBoost	Gradient Tree Boosting
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradient

ILSVRC	ImageNet Large Scale Visual Recognition Competition
ICA	Independent Components Analysis
IsoMap	Isometric Mapping
IR	Infra-red
IS	Inception Score
KSS	Karolinska Sleepiness Scale
LSUN	Large-scale Scene Understanding
Leaky-ReLU	Leaky Rectified Linear Units
AlignGAN	Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks
LSGAN	Least Squares Generative Adversarial Networks
LBP	Local Binary Patterns
LLE	Locally Linear Embedding
MRI	Magnetic Resonance Imaging
MSLT	Multiple Sleep Latency Test
NTHU	National Tsing Hua University Dataset
PReLU	Parametric Rectified Linear Units
PERCLOS	Percentage of the Eye Closure
PBV	Population Bias Visualisation
PCA	Principal Component Analysis
PVTs	Psychomotor Vigilance Tests
RFID	Radio Frequency Identification
ReLU	Rectified Linear Units
R-CNN	Regions with Convolutional Neural Networks

RTMC	Road Traffic Management Corporation
RSU	Roadside Unit
RMSprop	Root Mean Square Propagation
SVD	Singular Value Decomposition
SWAI	Sleep-Wake Activity Inventory
SSS	Stanford Sleepiness Scale
StarGan	Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation
SGD	Stochastic Gradient Descent
SVHN	Street View House Numbers
SVMs	Support Vector Machines
t-SNE	t-Distributed Stochastic Neighbor Embedding
Tanh	Hyperbolic Tangent
V-I	vehicle-infrastructure
V-V	Vehicle-Vehicle
VGG	Visual Geometry Group
VGG-Face	Visual Geometry Group-FaceNet
WGAN	Wasserstein Generative Adversarial Networks
WGAN-GP	Wasserstein Generative Adversarial Networks with Gradient Penalty
WHO	World Health Organization
XGBoost	A Scalable Tree Boosting System
YawnDD	Yawn Detection Dataset
ZJU	Zhejiang University Eyeblink Database

1 | Introduction

1.1 Motivation

The significant cost of traffic accidents around the world is estimated to be billions of rands, as reported by the American National Highway Traffic Safety Administration [118]. The World Health Organization (WHO) reveals that South Africa is among the African countries with the highest road traffic accident fatality ratio, of about 26.6 per 100 000 of the population [186]. Arrive Alive reported that the first road collision in South Africa occurred in Maitland, Cape Town in October 1903 [9]. Since then, there has been a drastic increase in road accident occurrence.

In 2006, the Road Traffic Management Corporation (RTMC) reported an increase in road accidents to approximately 12 454 from 8 802 fatal crashes in 2001 [139]. Figure 1.1 shows the fatality ratio analysis that was carried out by the WHO, highlighting the high rate of car accidents in Africa.

Figure 1.2 shows numbers of road accidents over a ten year period from 2007 to 2016 in South Africa. The contribution of each province to the percentage of accidents in South Africa in the year 2016 is illustrated in Figure 1.3 [157]. This shows that Gauteng and KwaZulu-Natal provinces had the highest accidents rates in 2016. Moreover, 1 700 people died on South African roads during festive season of 2016 alone, which is a 5% increase from the previous season [115]. Statistics South Africa has shown that one of the major causes of accidents is driver drowsiness [182]. In addition, statistics have shown the top three causes of accidents on South African roads are distracted drivers (for example, a driver using a phone), speeding, or driving under the influence of alcohol [96].

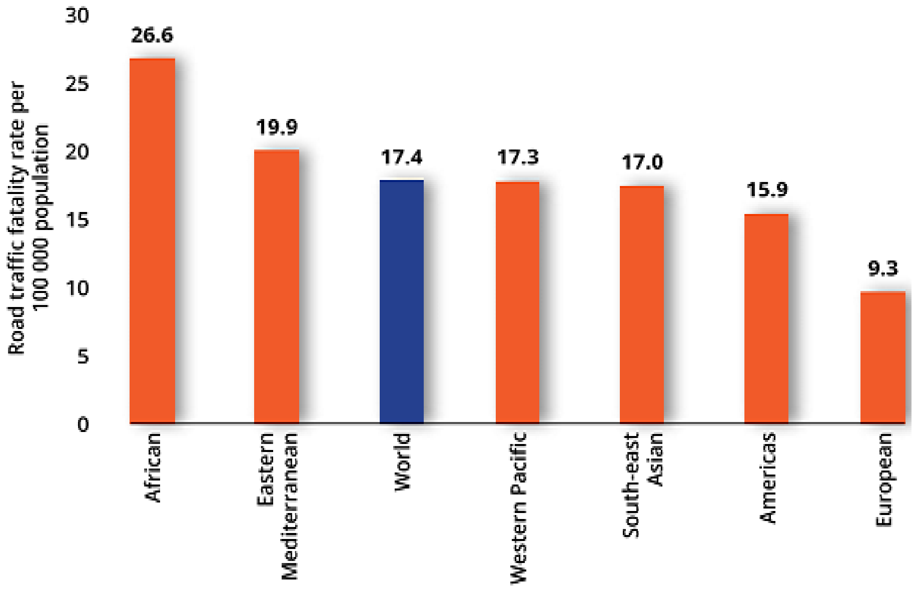


FIGURE 1.1: Road traffic fatality rates per 100 000 people per region [139].

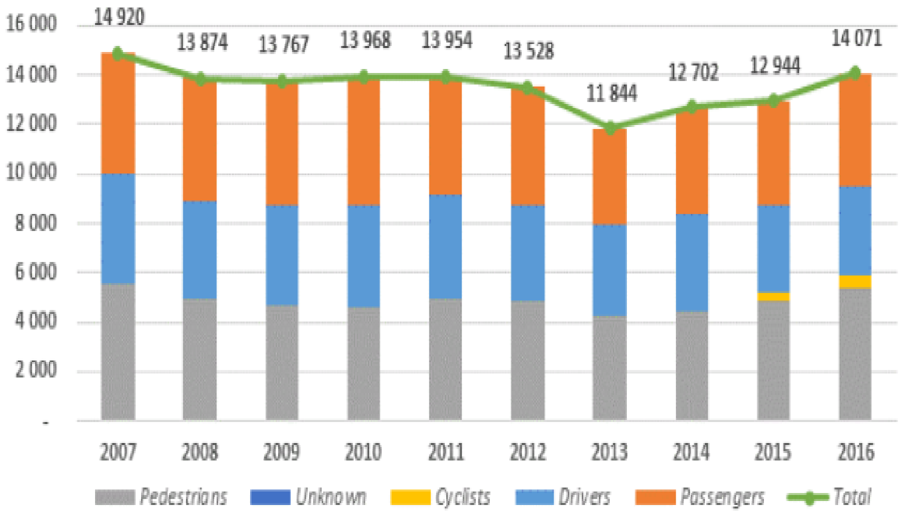


FIGURE 1.2: Numbers of road crashes in South Africa from 2007 to 2016 [157].

As machine learning increasingly becomes a technology that is used across many tasks in people's lives, the automobile industry has started to develop systems aimed at reducing the high rate of road accidents. Deep learning is a sub-field in machine learning that is actively being investigated, along with techniques that can communicate with car electronics to alert a driver of any anomalies. Examples of these systems include collision detection, drowsiness detection and driver skill monitoring [64].

Despite the great success of driver aids in the motor industry, there is evidence to show that these techniques can fail, which may then lead to accidents. A recent case involved an accident

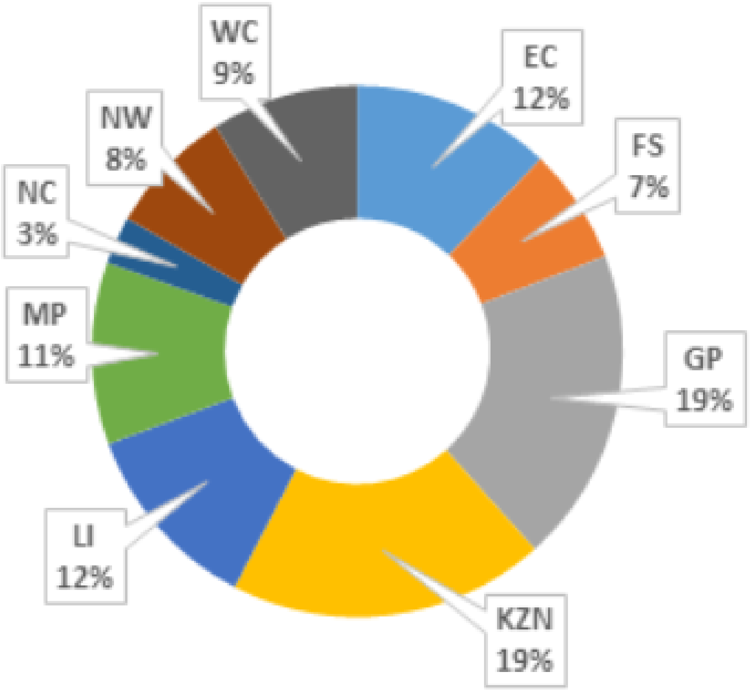


Figure 1.3: Percentages of road accidents per province in the 2016 financial year [157].

caused by a Tesla car in the United States of America, where the driver died while the car was in autopilot mode and crashed into a car-lane divider [13]. Moreover, an Uber self-driving car collided and killed a pedestrian while in autonomous mode in Tempe, Arizona [50].

As deep learning starts to gain traction in the car industry, accidents arising from poor decision making by deep learning systems have become a concern. In particular, systems like driver drowsiness detection could yield incorrect results based on the training data used. This thesis investigates intersectional accuracy differences in drowsiness detection systems and the bias implications caused by using unrepresentative training datasets. It further develops a novel framework that remedies this effect by using synthetic data and a re-sampling technique to generate fairer training datasets, allowing for more accurate models.

Computer vision is a field of research where there are multiple methods to approach numerous visual problems such as face recognition using machine learning techniques. However, there are concerns surrounding the use of people's faces on training machine learning techniques. In addition, there are privacy issues and questions surrounding faces such as who will own the collected image data, how will this data be shared and stored to prevent the misuse. Companies such as Clearview, have collected data without people's concerns and that goes further to datamine more images on their social networks [78]. In addition, some researchers use machine learning to train on faces of people to violate their rights by deciding whether a person is gay or not [79]. This is a clear indication that there is a need for privacy regulations when dealing with human data and European United has provided some ethics in artificial intelligence [168]. In this thesis, images are collected from publicly available channels and blurred to prevent the violation of using people's faces.

1.2 Problem Statement

Currently, driver drowsiness detection systems are only available in high-end vehicle models, which are expensive. However, there is an increased demand for accurate and efficient driver drowsiness detection systems on all vehicle models to reduce the rate of road accidents. This has led to the development of deep learning techniques to support this demand. However, deep learning based drowsiness detection systems can be flawed, due to bias that can be influenced by many factors including training data, parameters and the choice of the algorithm used. This work investigates the effects of using publicly available unrepresentative training datasets and shows how this impacts the performance of drowsiness detection systems when tested in an African context. An analysis of current drowsiness detection systems and their training datasets shows that these systems tend to work better on certain ethnicities (light skinned). This poses a problem in the South African context because imported cars with these systems can fail if trained on datasets captured in different demographics. This is due to the fact that South Africa has a diversity of races with different skin complexions. Most individuals in South Africa have dark skin and there is no drowsiness detection dataset that covers dark skinned drivers. The aim of this research is to provide an approach that can close this gap in training data for driver drowsiness detection systems. A novel framework is introduced to remedy the bias in these training datasets.

1.3 Aims and Objectives

The main aim of this research is to develop a framework that corrects a CNN trained for prediction using GANs for targeted data augmentation based on a population bias visualisation strategy that groups faces with similar facial attributes and highlights where the model is failing. The primary objectives are:

1. To investigate which machine learning technique should be used for the driver drowsiness detection task and the availability of training datasets.

2. To develop a CNN model for the detection of driver drowsiness detection.

3. To develop a data augmentation technique for balancing the training dataset where the CNN model fails to generalise.

4. To develop a framework that remedies generalisation failures in under represented population groups in the training dataset, and which boosts the performance of drowsiness detection across all population groups.

1.4 Contributions to knowledge

The contributions in this thesis can be summarised as follows:

- A meta-analysis study reviewing machine learning techniques in the detection of drowsiness was completed. This study showed that CNNs produce more accurate results when compared to the other machine learning techniques. This study further showed that there is a lack of benchmark datasets for driver drowsiness detection.

- A DNN was trained on publicly available datasets for the driver drowsiness detection task and was used to show that models trained on publicly available datasets exhibit traits of bias when tested in South African contexts.

- A novel population bias visualisation technique is proposed. The visualisation technique highlights individuals and population groups where the model fails to generalise.

- The training speed of GANs is improved by incorporating DSC, and findings on the importance of capacity are provided.

- Synthetic GAN images are used for data augmentation, relying on the translation of drowsiness facial features.

- A framework is proposed to remediate bias in the training datasets used for driver drowsiness detection tasks. This framework can also be used in other tasks that have unrepresentative datasets.

1.5 Organisation

The structure of this thesis is as follows. It should be noted that each chapter has its own literature review, that is based on the chapter's objective, before presenting implementation details and results. In Chapter 2, fundamental concepts and a review of driver drowsiness detection systems and techniques used are discussed. For driver drowsiness detection systems and the techniques used, special attention is paid to those methods reported as yielding the most accurate results. This is accomplished by conducting a meta-analysis on recent papers. Behavioural methods for driver drowsiness detection systems are of primary interest in this thesis as they are vulnerable to intersectional bias.

Chapter 3 introduces the fundamentals of machine learning and also the components that build up the CNN network. Training techniques are also discussed in an attempt to improve the accuracy of the network. Transfer learning techniques are also introduced to help where there is limited training data. Chapter 4 discusses the implementation of CNNs in-depth and introduces publicly available datasets for the driver drowsiness detection task. A new population bias visualisation technique that uses Principal Component Analysis (PCA) as a dimensional reduction technique, to identify bias in deep learning models trained for drowsiness detection, is presented. Here, trained model accuracy is overlayed on a grid of faces organised by similarity. This allows for the identification of population groups where the model does not generalise well.

In Chapter 5, a method that can be used to generate multi-domain realistic images from an input image is introduced. GANs are a generative model that consist of two networks competing with one another to produce realistic outputs. The generator network was modified by using the DSC technique to reduce the training time while retaining the quality of the produced synthetic images. The fundamentals of GANs are discussed and related work is introduced.

In addition, Chapter 6 proposes a novel framework that remedies generalisation failures under represented population groups in the training dataset, which boosts the performance of drowsiness detection across all population groups. The framework is composed of a GAN architecture for generating synthetic data, a CNN model for detecting drowsiness, a population bias visualisation technique to highlight where the model is failing and a sampler to target these population groups and search for similar images in the synthetic data.

Finally, Chapter 7 concludes the thesis by reviewing the results achieved. This is followed by a discussion on future work and expansion of the research field of road safety, with a particular focus on human behaviour.

2 | Background and Context

We all use transportation to reach our final destinations, whether going to school, a workplace or on vacation. Driving for long hours without taking breaks can be dangerous and leads to accidents. Most accidents are caused by not only drowsiness, but also distraction, weather conditions, high speeds, and alcohol usage. To date, the number of road accidents has increased each year. High-end car models are often manufactured with technologies that can detect whether the driver is falling asleep and can warn the driver. In addition, the motor industry is rapidly growing and improving safety features because driver and passenger safety is among their primary concerns.

This thesis focuses on behavioural methods to measure the level of drowsiness in drivers. Behavioural methods extract facial information which can be useful when used with deep learning techniques. Deep learning techniques are increasingly used to predict the driving state efficiently using the output from behavioural methods. Other methods are also discussed as they can be used in combination with behavioural methods to yield more accurate and robust drowsiness detection systems.

This chapter starts by introducing the fundamental concepts behind drowsiness as these behavioural cues are of importance to this research. This is followed by a discussion of signs and countermeasures that minimise the chances of falling asleep while driving. A number of commercially available driver drowsiness detection systems are discussed.

This chapter also provides a systematic literature review that was conducted on machine learning techniques for driver drowsiness detection, along with a meta-analysis on the performance of these techniques. Importantly, the results from this meta-analysis showed that CNNs yield more accurate results than alternative machine learning techniques, and are growing in popularity.

2.1 Fundamental Concepts of Drowsiness

This section briefly explains the fundamentals of driver drowsiness, which play an important role in identifying the methods investigated in this research.

2.1.1 What is drowsiness ?

Drowsiness or sleepiness are interchangeable terms that define a biological state where the body is in the transition from a state of wakefulness to a sleeping state [161]. It is a biological process that has a drastic effect on the body and mind of a person. In this state, the driver can lose focus on the road because of a lack of concentration. Drowsiness is often circumstantial, with many aspects of the driver's state and the environment being potential causes. For example, driving for long hours without taking breaks or on a highway at night can induce drowsiness. The body has a biological clock that alerts the person when there is a change in the body such as being hungry or a change in body temperature [44]. The process of being awake and being asleep is controlled both by a circadian pacemaker and by homeostatic factors. Homeostatic factors are those that control circadian rhythms to regulate the timing of states of sleep and wakefulness.

Being awake for long hours builds pressure that triggers the sleeping state, which is difficult to resist [39]. The circadian pacemaker is the internal clock that completes a cycle every 24-hours. This allows a person to prepare for changes in the physical environment that are associated with day or night activities [179]. For people between the ages of 18 and 65 years, normal sleeping duration ranges between 7 and 9 hours [62]. Sleep deprivation leads to sleep debt, which accumulates and can be observed through the emotions and mental state of a person [105]. A driver that drives for 17 hours without taking breaks has abilities that are equivalent to that of a driver with a Blood Alcohol Concentration (BAC) of 0,05%. After driving for more than 21 hours, the driver's BAC equivalence is about 0,10% which is above South Africa's legal level of 0.08% [82].

There are many internal and external factors that can interrupt a sleeping schedule. External factors that contribute to changing sleeping routines include work shifts, the effects of light, and driving times [73, 120]. On the other hand, internal factors include lifestyle (such as the intake of alcohol), effects of medication, and poor sleep quality (including insomnia).

2.1.2 Causes of drowsiness and its signs

There are many factors that influence the drowsy state of drivers. These effects can be caused by drivers themselves or by environmental conditions.

Work Shift Schedules

Shift-schedules are often divided into day or night shifts. Shifts vary between different companies and depend on the type of work being done. Night shifts can interrupt the normal sleep-wake cycle that is controlled by a circadian pacemaker. This results in a change of sleeping routines according to the shifts worked. The length and quality of sleep on a night shift is affected by physiological factors which include body temperature.

Light Effect

Light is an external factor that contributes to the sleep routine changes by influencing the need for sleep or by making it difficult to sleep [20]. Light influences the circadian cycle through cells in the retina of the eyes. These cells work together with rods and cones which are responsible for telling the brain when it is night or day, thereby setting the sleeping pattern. Levels of drowsiness in most drivers increase at night because there is no light effect and also due to the temperature inside the car. When driving at night, retina cells send signals to the brain that it is night time and the body will start to adjust and prepare for a sleeping state.

Lifestyle and Medication

The intake of medication such as over the counter prescriptions can have an impact on cognitive and driving abilities, which can lead to road accidents [101]. People often do not take serious note that many medications have side effects. Moreover, every person reacts differently to medication, for example in some cases medication will cause the body to react, resulting in high blood pressure or blurred vision [101]. Alcohol is the most commonly used form of drug consumed by drivers and its side effects include loss of concentration, blurred vision, poor judgement, and drowsiness [47]. Alcohol also affects sleeping cycles and physiological processes that occur during the sleeping process. The effects of alcohol in the human body has been extensively studied and early experiments were performed by Kleitman in 1939 [140].

2.1.2.1 Signs of drowsiness

There are common signs that can indicate a person is drowsy, and these typically involve facial expressions. Some of these features are obvious and are not difficult to notice, such as the following:

- Frequently yawning;
- Being unable to keep eyes open and reduced eye blinking rate;
- Swaying the head forward;
- Eyes start to itch and become red;
- Paying less attention than usual to road markings and signs;
- Crossing of road lanes and a tendency to accelerate; and
- A tendency to stop too close to cars up ahead.

2.1.3 Countermeasures against drowsiness

There are countermeasures that a person can take to prevent drowsiness. If a person needs to drive for a long distance, or perhaps at night, it is recommended to have sufficient sleep before the journey.

Consumption of Caffeine

Most people use caffeine products such as coffee, energy drinks, soft drinks, and tablets or capsules to avoid falling asleep [34, 28]. Although people think this is an effective solution, caffeine has a short-term effect, which last for only one to two hours and thereafter actually helps to induce sleep. Furthermore, it takes about thirty minutes to enter the bloodstream and for those who consume caffeine regularly, its effect can be minimal [65].

Taking Naps

The South African Traffic Department recommends that drivers should take naps during stops when on a long journey [138]. A nap of about 15- to 20 minutes in duration is enough to energise the body, and also has an additional benefit in helping to overcome drowsiness after stretching your legs. Driving with a passenger on a long drive is also helpful because this person can keep the driver awake or alert the driver if they notice symptoms of drowsiness. An alarm could be also used to help alert the driver.

2.1.4 Measurement Methods for Drowsiness Detection

A number of methods used to detect and measure driver drowsiness state are currently being researched and implemented. These methods are grouped into four categories: physiological, behavioural, vehicle-based, and hybrid methods. Behavioural methods are of primary interest to this research and are currently being developed for detection of drowsiness using publicly available datasets for training and testing. The section below provides a brief survey of drowsiness detection methods in each of the above-mentioned categories.

2.1.4.1 Physiological Methods

Physiological methods are those that obtain direct information about the driver's state by assessing their physical conditions. This can be done by using an electronic device that is connected to different parts of the body to collect information about a person's state, such devices include Electrooculograms (EOG), Electroencephalogram (EEG), Electrocardiogram (ECG), and Electromyogram (EMG) [55, 89, 155]. Using these devices can yield highly accurate results for driver drowsiness identification, but these methods are not widely accepted because of practical implications [36]. Moreover, electro-physical devices use electrodes that are attached to the driver's head, chest, or the face. The implementation of a real-time application using such methods is not acceptable because the electrodes can cause irritations and discomfort to the driver [171]. A brief discussion on electro-physical devices is discussed below:

EEG − is a device that reads electrical activity generated within the brain by the firing of neurons [41]. Readings are collected by placing electrodes on the scalp of the driver to find brain signals at a certain frequency. Figure 2.1 shows a person wearing an electrode headset for reading EEG signals. Drowsiness levels appear on the EEG spectrum when there is an increase of activity in the frequency bands that are generated by the parietal and central regions of the brain. Results are obtained by placing the device on the person and measuring signals. The collected signals are then converted into a waveform that can be visualised [80]. In much research, EEG is

Figure 2.1: Readings of EEG signals being captured on a person [80].

used as a reference indicator because it is capable of producing efficient results when detecting traces of drowsiness in drivers. Belakhdar et al. investigated the use of an Artificial Neural Network (ANN) on a single EEG channel, to detect drowsiness [14]. Ten people were tested and nine features were captured from one EEG channel using the Fast Fourier Transform (FFT) features. These features were used to train the ANN classifier. An accuracy rate of 86.1% and 84.3% was achieved when classifying users as drowsy or alert, respectively. [14].

ECG – is a graphical presentation that depicts electrical activity generated by the heart pulse [97]. ECG makes use of electrodes that are placed on the skin to detect an electrical change on the skin caused by heart muscles [122]. It is a portable device which uses a non-invasive method where waves from the heart are measured with a variation of blood volume in tissues where a light source is used together with a detector.

EMG – is a device used to record the activity of the muscles when they are at rest and when there is movement [32]. The primary purpose of EMG is for diagnosis based on information from muscle and motor neurons [59]. EMG reads and translates signals from the motor neurons into graphs, numerical values or sounds that can be interrupted by a computer to make a diagnosis. A needle electrode is inserted into the muscles to measure muscle signals.

EOG – is a technique used for measuring different levels of eye movement through the resting potential of the retina [151]. The signals produced by EOG record the difference between the electrical voltage of the front and back of the eye [111]. To obtain recordings, an electrode is placed on the skin near both eyes. These electrodes are placed in different positions, horizontally to reflect the horizontal eye movement, with a vertical electrode to reflect the vertical movement. Signals can be acquired by sampling at a frequency of 256Hz and resolution of 16 bit [23]. Zhu et al. presented EOG-based drowsiness detection using CNNs. An unsupervised learning approach to estimate driver fatigue based on EOG was used [198]. Their results showed that ad-hoc feature extraction yielded effective results.

2.1.4.2 Subjective Methods

Subjective methods are non-invasive and consist of a series of tasks given to the driver and the completion of questionnaires. These tasks are monitored by a trained individual to assess and quantify the sleepiness levels of a driver. There are different subjective methods that can be used to evaluate the sleepiness levels of the drivers including the Epworth Sleepiness Scale (ESS), the Karolinska Sleepiness Scale (KSS), the Stanford Sleepiness Scale (SSS), the Sleep-Wake Activity Inventory (SWAI), and the Multiple Sleep Latency Test (MSLT).

ESS – is an instrument used to measure average daytime sleepiness, in order to differentiate between normal sleepiness levels and excessive daytime sleepiness. It was introduced by Dr Murray Johns at Epworth Hospital in 1991 [76] and makes use of questions designed to measure an individual's sleepiness levels. These are based around various situations an individual may find themselves in, including watching television, sitting in a car due to traffic congestion, or sitting inactive in a public place. The questionnaires consist of eight questions and are rated up to 4-points containing different weights as follows:

- No chance of dozing (0 points);
- Slight chance of dozing (1 point);
- Moderate chance of dozing (2 points); and
- High chances of dozing (3 points).

Individuals with a score from 0 to 9 are considered to be awake and those with 15 points and above show traits of severe daytime sleepiness levels.

KSS – is a tool that measures an individual's level of sleepiness at a particular time of the day [5]. This consists of a self-report process that takes about five minutes to complete. There are two versions, where the first version has labels on every other step and the second uses labels on every step on the 9-point scale. Table 2.1 defines the KSS scores.

Table 2.1: The KSS Sleepiness Scale

KSS scores	
Description	Scores
Extremely alert	1
Very alert	2
Alert	3
Rather alert	4
Neither alert nor sleepy	5
Some signs of sleepiness	6
Sleepy, but no difficulty remaining awake	7
Sleepy, some effort to keep alert	8
Extremely sleepy, fighting sleep	9

Table 2.2: The SSS ratings

SSS ratings	
Description	Scale Rating
Feeling active, vital, alert, or wide awake	1
Functioning at high levels, but not fully alert	2
Awake, but relaxed; responsive but not fully alert	3
Somewhat foggy, let down	4
Foggy; losing interest in remaining awake; slowed down	5
Sleepy, woozy, fighting sleep; prefer to lie down	6
No longer fighting sleep, sleep onset soon; having dream-like thoughts	7
Asleep	x

SSS – is a self-report questionnaire instrument that was developed by Dement Hoddes and colleagues in 1972 [63]. This measure uses an hourly test performed to evaluate the level of sleepiness. It uses a scale from 1 to 7 to determine the level of sleepiness. Table 2.2 defines the SSS level scale rating. The SSS is a simpler version of the KSS and thus takes less time to evaluate.

SWAI – is a multidimensional self-report tool used to measure the level of sleepiness in a clinical setting [142]. This measure contains 59 items which have domain scales for evaluation of sleepiness levels. The score domains include excessive daytime sleepiness, distress, social desirability, energy levels, ability to relax, and nighttime sleep. The test takes approximately 10 to 15 minutes to complete, which is longer than the KSS and SSS tests.

MSLT – is a test for evaluating excessive daytime sleepiness. It is performed by measuring

Table 2.3: Multiple Sleep Latency Test

MSLT Levels	
Minutes before falling asleep	Sleepiness Levels
0 - 5	Severe
5 - 10	Troublesome
10 - 15	Manageable
15 - 20	Excellent

how quickly an individual falls asleep in a quiet environment [21]. Those taking the MSLT are given five scheduled naps each lasting 20 minutes, two hours apart. The tests are done in different environments including a dark room, a quiet room, a comfortable environment without distractions, or rooms with environmental factors that may prevent them from sleeping. Data can be collected from brain waves, EEG tools, muscle activities, oxygen levels, and eye movements. Table 2.3 shows the MSLT scores. The sleepiness levels indicate that if an individual falls asleep within five minutes of the test, they are considered pathologically sleepy. On the other hand, individuals that take more than 10 minutes to fall asleep are considered to have normal sleepiness levels.

Subjective methods are not practical for monitoring driver drowsiness tasks, but they can be useful when used in combination with other drowsiness measuring methods to produce informed decisions and to provide ground truth labelling for behavioural detection systems.

2.1.4.3 Vehicle-Based Methods

Vehicle-based methods rely on the vehicle's control system, which includes steering wheel angle, braking system indicators, lane position information, and the speed of the vehicle [16]. Together with behavioural methods, these can be used to create robust and efficient systems.

Lane position monitoring –This technique uses road lanes to detect the position of the car on the road. The placement of the vehicle in the centre, left or right lane of the road [86, 137], can be used to assist the driver to leave a space between their own and oncoming cars and to detect if a vehicle crosses the line. Lane departure systems can warn the driver about collisions, potential distractions, and can flag drowsiness [184]. Zheng et al. proposed a new system for lane position detection based on Radio Frequency Identification (RFID) and vision [196]. The system is divided into two processes which include lateral position detection, and the detection of the absolute or relative position of other vehicles. Errors from lateral positioning GPS/DGPS were revised by using a lane judgement subsystem [196]. Furthermore, Zheng et al. proposed vehicle-infrastructure (V-I) positioning using a single Roadside Unit (RSU), and Vehicle-Vehicle (V-V) relative positioning algorithms. The system is motivated to be a useful approach preventing road accidents in dense traffic [196].

Driving at High Speed – In South Africa, the speed limit is 40 to 60km/h in urban areas, 80 to 100km/h in rural areas and 120km/h on freeways[114]. Road accidents that are caused by excessive speed have a high mortality rate. When a driver falls asleep, they become unaware of the speed they are driving at and it can be too late to regain control of the vehicle if they wake. 16.5% of accidents between 1999 and 2008 were speed-related accidents caused by drowsy driving [170]. When a driver is about to fall asleep, they tend to increase speed. The increase in the speed can lead to accidents since the driver is not paying attention to the road because they are in a sleepy state.

Steering wheel movement measurements – A number of studies have investigated the use of steering wheel measurements to detect consciousness by observing steering patterns [121]. This technique uses sensors that are mounted on the steering wheel to collect data that can be further analysed for drowsiness. When the driver is in a drowsy state, steering wheel correction is reduced. The combination of steering wheel movement measure and lane departure techniques can overcome false positive situations where small steering angles are required to follow a road safely.

2.1.4.4 Behavioural Methods

Faces contain information that can be used to interpret levels of drowsiness. There are many facial features that can be extracted to infer a driver's level of drowsiness. Behavioural methods measure levels of drowsiness through the use of mounted cameras in the car that observe facial features such as eye state, head movement, blinking rate and the presence of yawning [109].

Eye-state detection – When a driver has been driving for long hours or at night, they tend to lose focus of the road and control of the vehicle. When a driver is drowsy, their blinking rate is reduced and their eyes tend to be red. Some researchers track eye movement to detect the state of the driver [49]. It has been shown that closing the eyes can lead to head-on collisions and is a significant cause of road accidents [154]. However, drowsiness impacts the eye state and can be used as a detection measure. Methods that are used to measure the level of drowsiness include the Percentage of the Eye Closure (PERCLOS) and the Eye Aspect Ratio (EAR) [30, 173]. The EAR was introduced by Soukupora and Cech in 2016, with the EAR computed as the ratio between the height and width of the eye [173]. In contrast, PERCLOS is the percentage of eye closure over a period of time. The difference between these two methods is that EAR classifies the ratio of the eye as it decreases whereas PERCLOS classifies whether eyes are open or closed over a period of time.

Yawning – Frequent yawning is a sign that a driver needs to rest. Detection of yawning can be an early warning sign that can trigger an alarm for a driver to be made aware of their state. However, yawning alone cannot be used to determine that the driver is drowsy because it can yield false positive results. Many researchers use yawning with other features to detect and draw the conclusion that the driver is drowsy [2, 149].

Head Position Detection – The head position tends to sway forward when the driver is drowsy. This can cause a driver to lose focus on the road and potentially cross the road lanes. A number of researchers use head tilt angles to determine the state of the driver [145]. The normal angle of the driver's head is determined and an alarm is raised when the angle goes beyond a certain angle.

This thesis focuses on the use of vision-based behavioural methods to measure the level of drowsiness in drivers. This process typically relies on a mounted camera in the car that is used to monitor a driver's facial attributes, as mentioned in Section 2.1.4.4. These facial attributes can then be used by machine learning techniques such as CNNs, SVMs or HMMs for drowsiness detection. The process of identifying the driver's state starts by passing labelled data to a given machine learning technique, as part of a step termed the training phase. The series of steps that are followed in figure 2.2 suggests a common process for detection of drowsiness. These steps are as follows:

Input Video – This is the stage where video frames from a fixed camera or a smartphone are broken down into a series of images. The video frames are taken in such a manner that only the face of the driver is captured in the image.

Face Detection – The second stage typically aims to detect the face in the image frames. The Viola and Jones detector is the most commonly used algorithm to detect the driver's face from within the image [177]. However, when CNNs are used, the whole image is typically fed to a

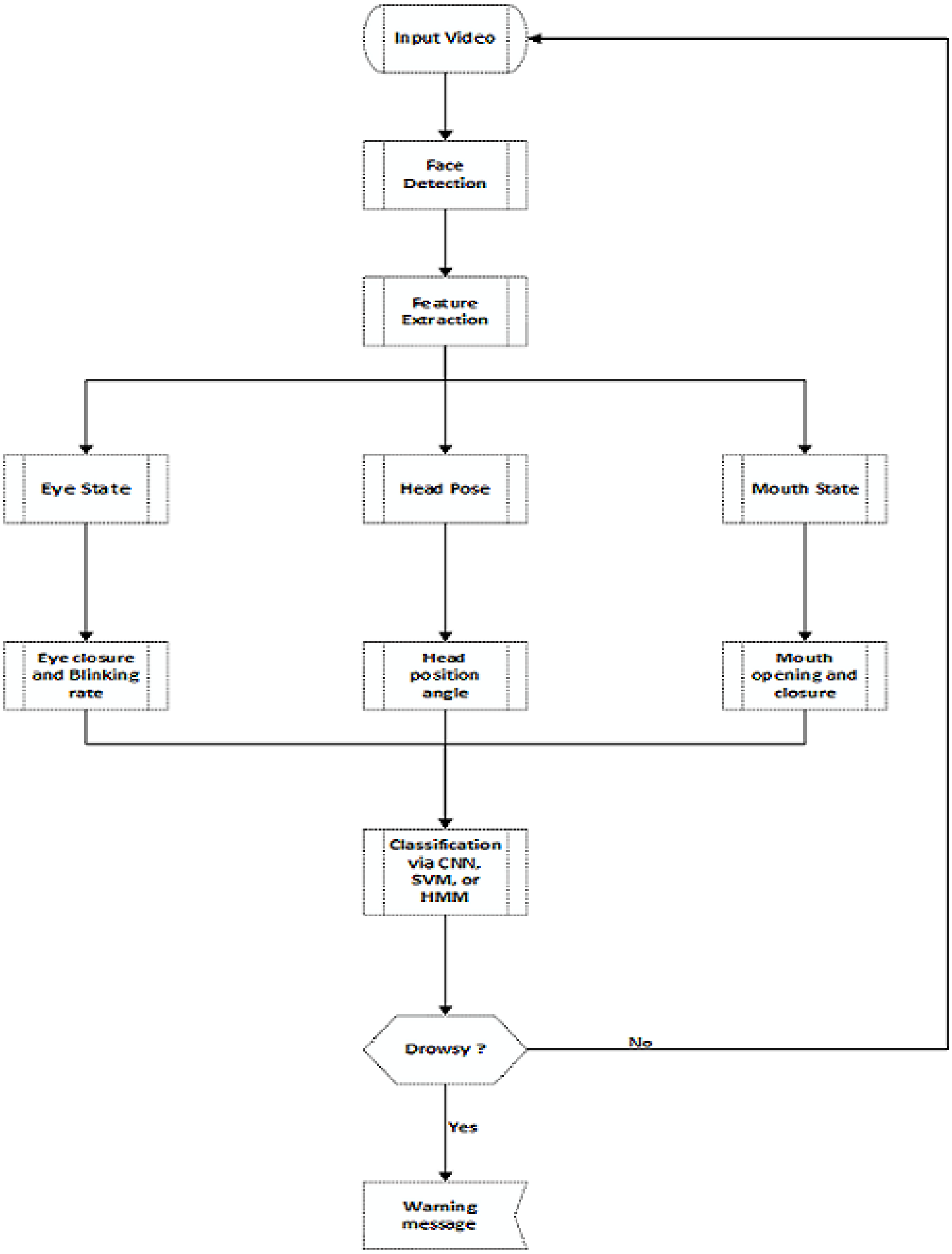


FIGURE 2.2: Driver drowsiness detection process.

network that has multiple filters and features are automatically extracted. CNNs combine the two stages of detecting the face and of feature extraction.

Feature Extraction – If face detection is applied, features are usually extracted using different methods such as landmark localisation, Histogram of Oriented Gradient (HOG), and Local Binary Patterns (LBP).

Feature Analysis – Extracted features can then be processed further, as is the case for PERCLOS or EAR for eye analysis or mouth-based methods for yawning detection.

TABLE 2.4: CNN based detection systems

Author/s	Year	Measure	Frame Rate (fps)	Accuracy
Dwivedi et al. [36]	2014	Visual features	60	78%
George and Routray [46]	2016	Eye gaze	24	98.32%
Reddy et al. [134]	2017	Eye and mouth state	72	91.6%
Zhang et al. [195]	2017	Eye state	N/A	95.18%
Jie Lyu et al. [98]	2018	Eye and mouth state	37	90.05%
Rateb Jabbar et al. [72]	2018	Eye and yawning state	30	87%
Young-Joo Han et al. [56]	2018	Eye state	22	94%
Mohammed Ghazal et al. [48]	2018	Eye state	14	95%
Wang Huan Gu et al. [52]	2018	Yawning state	50	99%

Classification – The classification stage consists of classifiers that are used for decision-making regarding a driver's level of drowsiness. If the classifier detects traits of drowsiness based on the weighted parameters, then an alarm will be activated suggesting that a driver takes a break.

Behavioural methods exhibit various limitations because their performance is affected by lighting conditions, camera movements, and the frame rate used to capture images of the driver's face. Light variation can typically be eliminated by using Infra-red (IR) cameras. Various measures are used in different studies for detecting a face and extracting features from the video feed. Unfortunately, most behavioural drowsiness detection studies use different datasets that may favour their own algorithms. This is due to the lack of standardised datasets that can be used as a benchmark. As a result, it is hard to compare approaches by simply evaluating reported accuracies. Machine learning techniques that classify different levels of drowsiness are now discussed, along with a review of measures that form a driver drowsiness detection system.

CNNs – Consist of interconnected layers of neurons, where data is passed through every layer and various computations are performed. More detailed information about CNNs is provided in Chapter 3. CNNs are the most commonly used machine learning techniques for driver drowsiness detection, as indicated in Table 2.4.

Dwivedi et al. proposed an algorithm for driver drowsiness detection using representation learning [36]. Here, the popular Viola and Jones algorithm was used to detect the faces. Images were cropped to 48 x 48 square images and fed into the first layer of the network which consisted of 20 filters. The whole network contains two layers. The output of the CNNs was passed to a softmax layer for classification. This system did not allow for a consideration of head pose changes and as a result can fail. However, Huynh et al. used a 3D DNN to obtain more accurate results [68]. Here, the face is tracked by a combination of a Kernelized Correlation filter with a

Table 2.5: SVMs based detection systems

Author/s	Year	Measure	Frame Rate (fps)	Accuracy
Sabet et al.[144]	2012	Eye state	25	98.4%
Punitha et al.[130]	2014	Eye state	15	93.5%
Pauly and Sankar[125]	2015	Eye state	5	91.6%
ALAnizy et al.[6]	2015	Eye closure	60	99.74%
Manu [102]	2017	Eye and Mouth state	15	94.58%
Zhuoni Jie et al. [75]	2018	Yawning state	10	94%
Souto et al. [158]	2018	Eye state	15	78%

Kalman filter for robust face tracking. The extracted face regions are then passed to 3D-CNNs which is followed by a gradient boosting machine for classification. This system works well even if the driver is changing head position [68].

SVMs – are a group of supervised learning methods for classification and regression problems based on decision planes to separate a set of training data into their different classes. They were first introduced by Boser et al. in 1992 in an attempt to find a hyperplane which separates training data into different classes according to their features [18]. SVMs use labelled data as inputs and for the driver drowsiness detection problem, SVMs can differentiate whether a driver's eyes are open or closed.

A great deal of work has attempted to utilize the capabilities of SVMs in the detection of drowsiness. Different measures have been used as features to determine a driver's level of drowsiness using SVMs. A comparison of these measures is presented in Table 2.5.

AL-Anizy et al. proposed a fully automatic system that is capable of detecting driver drowsiness [6]. For face detection and eye extraction, the well-known Haar feature matching algorithm was used. SVMs were then trained to classify when eyes are open or closed and to trigger an alarm. Similarly, [144] proposed a system that can detect driver drowsiness and distraction. Here, the Viola and Jones algorithm was used for face detection and colour histograms with LBP applied to track the face over frames. The system achieved an accuracy of 100% in face detection, but a potential downfall of this approach is the low frame rate achievable, which could result in missed facial expressions.

HMMs – are statistical models that use hidden states which are based on observed states defined by probabilities to make predictions. The first HMM was developed by Leonard Baum and colleagues in the late 1960s and early 1970s [12]. HMMs are now widely used in other

Table 2.6: HMM based work related

Author/s	Year	Measure	Frame Rate (fps)	Accuracy
Sun et al.[163]	2013	Eye blinks	61	90.99%
Tadesse et al.[167]	2014	Eye closure	20	97.0%
Zhang et al.[194]	2015	Eye state	N/A	95.9%
Choi et al. [24]	2016	Eye and head state	16 to 20	92%

fields including facial expression recognition, gene annotation, and computer virus classification [128, 190]. For the driver drowsiness detection task, Table 2.6 shows the range of features and approaches used by HMM-based drowsiness detectors, with the exception of Zhang et al. [194] and Choi et al. [24], who omitted information required for comparing their findings and are therefore not included in this meta-analysis. Nakamura et al. proposed a new facial feature measure by using changes in wrinkles detected by calculating the local edge intensity on the face [112]. They used an IR camera to eliminate illumination changes and allow for operation in both day and night conditions. Unfortunately, this system can yield false results when used on older people because they have deeper wrinkles. In contrast, Bagci and Ansari implemented an HMM for eye tracking based on colour and geometrical features [11]. For illumination elimination, authors used a two-level Lloyd-max quantization intended to be robust to illumination changes. Unfortunately, this system is designed for indoor conditions and it fails to detect the face if the driver is not facing forward.

2.1.4.5 Meta-Analysis

A great amount of work has been conducted in the field of road safety to minimise road accidents. Industries play a vital role in providing these safety features to detect drowsiness. However, a challenge identified in the reviewed papers is that most systems used their custom datasets. This challenge makes it difficult to compare these techniques and to have standard benchmark datasets. Most of the datasets used to test machine learning techniques are not publicly available because of issues around privacy and identity security.

In an attempt to provide a fair comparison, a systematic literature review was conducted on machine learning techniques for driver drowsiness detection, along with a meta-analysis of the performance of these techniques.

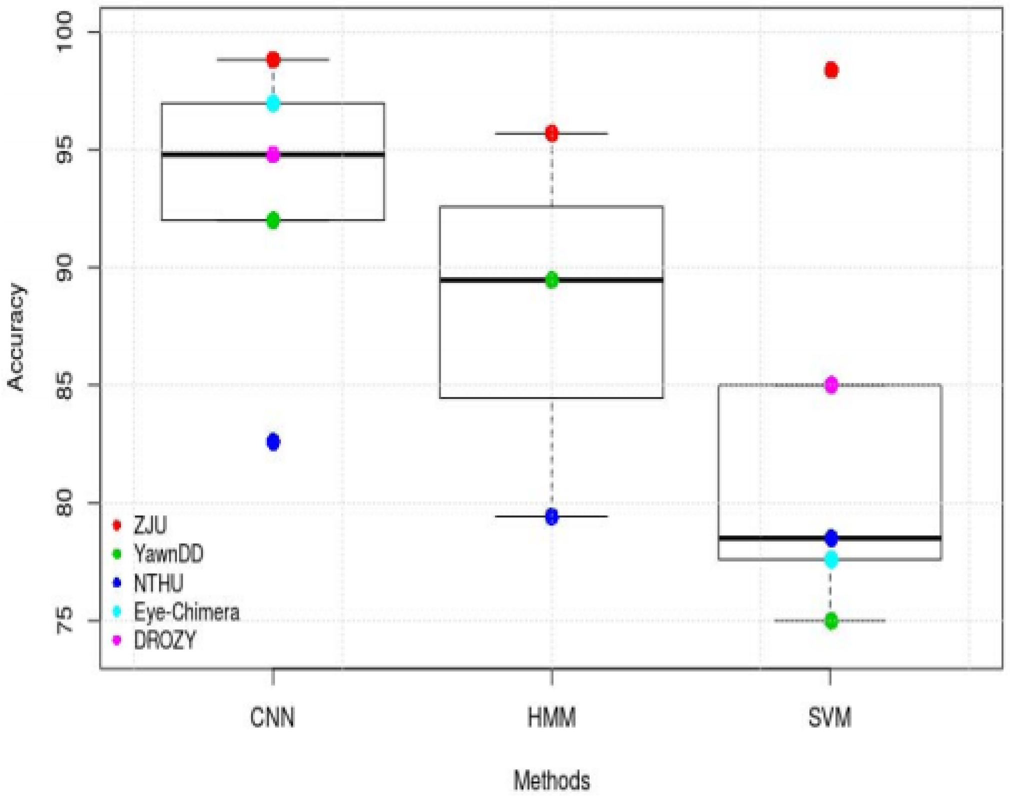


FIGURE 2.3: Meta-Analysis Report

This meta-analysis was conducted using 40 papers that were based on the three techniques described above. From the collected papers, various metrics were extracted for comparison including datasets, facial expression features, frame rate, and accuracy levels. The meta-analysis report showed that CNN methods were used most commonly and also yielded more accurate results than SVMs and HMMs. A non-parametric Skillings-Mack test was conducted and rendered a Chi-square value of 6.66, which was significant at p = 0.035. This test showed that there is a difference in performance between the compared techniques. The most common test datasets were identified from the papers to provide a fair comparison, these datasets include DROZY [104], Eye-Chimera Database (Eye-Chimera) [136], National Tsing Hua University Dataset (NTHU) [27], Yawn Detection Dataset (YawnDD) [3], and the Zhejiang University Eyeblink Database (ZJU) [199]. Figure 2.3 shows the box-plots of the accuracies obtained for each technique along with the associated dataset for comparison. The CNNs performance and number of uses show a great interest in this approach. Chapter 4 will show how CNNs are used for driver drowsiness detection. It is also important to note that the other two techniques were difficult to compare due to insufficient data.

It should be noted that the datasets that were used for meta-analysis are not representative and they cover a limited range of races. This poses challenges in the South African contexts. South Africa is a diverse country with people of a variety of races with different skin complexions. The Bantu-speaking races dominate South African demographics at 78.4%, whites make up 10.2% of the population, Coloureds (people of mixed race) contribute 8.8%, and 2.6% are Asians [183]. The population of people in South Africa is at 57 million, as reported by Statistics South Africa [187]. The value of cars imported to South Africa was about R59 billion in 2017 [38]. This represents a high number of vehicle imports and increase the potential for failure if car drowsiness detection systems are designed and tested on data that excludes a large portion of South African population groups, especially dark-skinned individuals.

2.1.4.6 Hybrid Methods

All of the previously mentioned methods have advantages and limitations. Limitations in vehicle-based methods include weather and road conditions. Vehicle-based methods work best on specific roads that have visible road markings and minimal road defects. In poor road conditions, vehicle-based methods can yield more false-positive results. On the other hand, physiological methods yield more accurate results. However, these require that a driver must wear a device that measures their state of drowsiness, so it is a challenge to implement real-world applications that incorporate this method.

Hybrid systems combine drowsiness detection methods to produce more robust systems. Combining two methods can yield better and more accurate results. For example, Veena and Subhashini used a combination of physiological and behavioural methods to detect driver drowsiness [162]. This system used wireless sensors for eye blink detection and a biomedical sensor placed on the steering wheel or on the driver's spectacles. The hybrid method proposed by Veena and Subhashini showed promising results with low-cost sensors [162]. Similarly, Agustin et al. proposed a hybrid method that combines the information from pulse rate monitoring, eye monitoring, and head movement resulting in increased the drowsiness detection accuracy [4]. This system used fuzzy logic for prediction. The pulse rate was measured using a pulse oximeter, while eye and head movements were tracked using a camera. The correctness was evaluated by using the number of correct responses over the total number of samples. Pulse rate contributed about 40% towards the final decision of the system, while eye movement accounted for 35% and head monitoring 25% [4].

2.2 Commercially Available Systems

Motor industries have been engaged in research and implementation of accurate and increasingly robust driver drowsiness detection systems. However, companies use different measuring methods for the implementation of such systems. Driver drowsiness systems are available in different vehicle model brands. The following section describes a number of systems that are currently being used.

The rest recommendation system – This is an Audi system that analyses the driver's behaviour by monitoring steering, gear lever, and pedal movements [10]. The system is automatically activated at a speed between 65 and 200 km/h. The information from all these components is monitored and if there is a change in the pattern, the driver will hear an audio notification and receive a visual prompt alerting them. The system also provides an easy to read interface that recommends if a driver needs to take a break.

Active Driving Assistant – This BMW system includes a camera-based lane departure warning and collision warning [33]. This system detects lane markings, and if the system detects that the driver has changed lane unintentionally then the steering wheel will vibrate. Figure 2.4 shows the BMW active driving assistant system. The system is activated by the car's speed, at 70km/h the system starts to collect information on unintentional lane changes and a warning is displayed on the car's cluster [17] if this is detected.

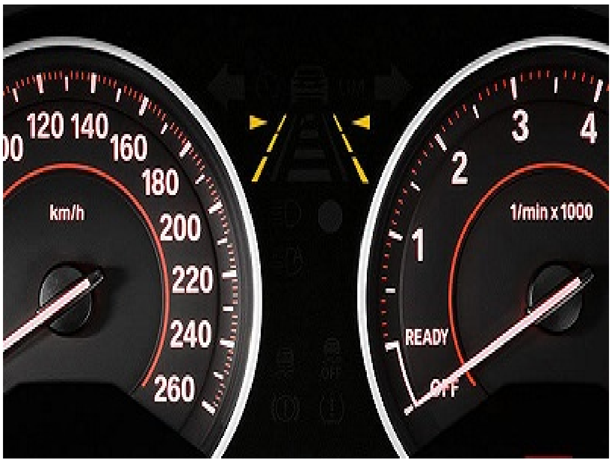
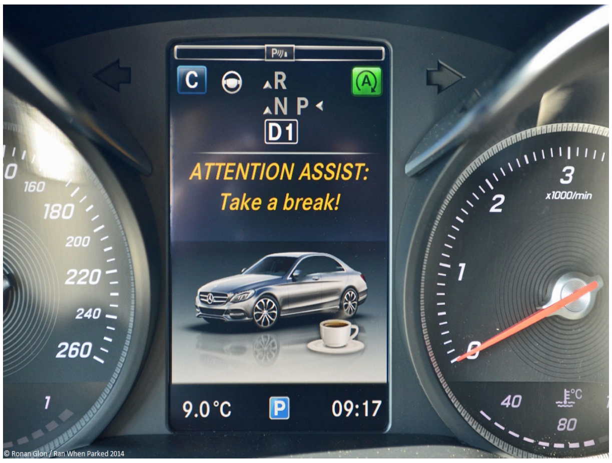


FIGURE 2.4: BMW active driving assistant [33]

Attention Assist – This is a Mercedes-Benz system that is equipped with several sensors which include EEG sensors that are placed on the steering wheel. The system is activated between speeds of 80 and 180 km/h, where it measures steering wheel patterns and combines this with other information such as duration of the trip and the time of day [107]. If the information collected shows that the driver is drowsy, a warning (a coffee cup signal) will be displayed in the car's cluster, followed by an audible tone. Figure 2.5 shows the visual alert from the system.



© Ronan Glon / Ran When Parked 2014

FIGURE 2.5: Mercedes-Benz attention assist [107].

Driver Alert – The Ford system uses small forward-facing and rear view cameras that are connected to the onboard computer [157]. The rear-view camera is trained to identify lane markings on both sides of the vehicle. When the driver is crossing road marking lanes, the system looks at the road ahead and predicts the position of where the vehicle should be relative to lane markings. Once the driver makes the mistake of crossing the lane markings, an alert will be displayed as a text message in the car's cluster. If the driver ignores the message, another alert will be triggered and the driver has to acknowledge it by pressing an "OK" button. If the driver continues to ignore the message, the last stage of this system requires that the driver stop the car and acknowledge the alert by opening the door.

2.3 Conclusion

This chapter has explored fundamental concepts of drowsiness together with measurement methods used to distinguish different levels of drowsiness. It further discussed available driver drowsiness detection systems that are currently in use on high-end car models. In particular, this chapter focused on behavioural methods, which use sensors to track facial features in order to identify drowsiness. The South African population was also discussed as a potentially problematic case for behavioural methods, as it is a particularly diverse country. A general approach to driver drowsiness detection was highlighted, which clearly shows what is needed for implementing an effective system.

A literature review was conducted on machine learning techniques for the task of driver drowsiness detection. A meta-analysis of the performance of the machine learning techniques was conducted. CNNs were shown to provide more accurate results and were extremely popular, this is due to their rapid growth since a 2012 breakthrough by Alex Krizhevsky. The following chapter discusses in more detail the fundamentals of CNNs, while details on how they are used to implement a driver drowsiness detection system are discussed in Chapter 4.

3 | Fundamentals of Deep Learning

We are entering an era where data is collected each and every day. This data can be in various forms such as video, images, and text. There is a need to interpret this data and discover valuable hidden information. One of the most commonly used techniques to interpret data is by machine learning. Machine learning consists of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to classify objects in images. In this thesis, we use deep learning, which is a subclass of machine learning that learns from large amounts of data, for the driver drowsiness detection task. The fundamentals of deep learning are described in this chapter.

3.1 Artificial Neural Networks

An artificial neuron is a mathematical model that is inspired by neurons in the human brain. This artificial model tries to simulate the structure and the functionalities of a biological neural system in a computerised form. The perceptron [127], relies on three steps, in the first step inputs x_i are accepted and multiplication is performed on each input value with individual weights w_i. The second step involves the summation of all the weighted inputs and the addition of bias to the neuron, to form the neuron's pre-activation z.

$$z = b + \sum_i w_i x_i, \tag{3.1}$$

where b is the bias value of the neuron. The final step is the activation stage, where the previously weighed inputs and bias are passed through an activation function.

3.2 Convolutional Neural Networks

CNNs are multistage mechanisms that learn a data representation in order to fulfil a specific goal. Data is passed through sequentially stacked layers to learn different features that represent the data. CNNs have dominated many computer vision tasks since the breakthrough shown by Alex Krizhevsky in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012 [87]. However, the history of CNNs goes back to the 1940s, where McCulloch and Pitts introduced mathematical models that were inspired by human neural activities in 1943 [106]. In 1958, Frank Rosenblatt showed how a probabilistic model could learn from the observations, and how the information is stored and remembered [141]. In 1980, Fukushima proposed a neural network model called neocognitron which is said to be self-organised [45]. His model involved learning without being taught and recognised stimulus patterns based on geometrical similarity. Since then, the field had slow improvement until 1989, where LeCun et al. applied backpropagation to neural networks [91]. Their work was inspired by the experiments conducted by Hubel and Wiesel in 1962, who studied the visual cortex of animals [67]. The success of Alex Krizhesky in 2012 brought much interest in the computer vision communities and this led to many progressive works. A brief summary of the highlights of some improvements made thus far is described below, with reference to the ILSVRC:

ZFNet (2013) –In 2013 the winners of the ILSVRC were Matthew Zeiler and Rob Fergus from New York University [193]. Their model achieved an 11.2% error rate, improving upon the 2012 error rate of 15.4% obtained by AlexNet. Although ZFNet is similar to AlexNet, ZFNet was modified by introducing a deconvNet and decreasing filter sizes from 11×11 pixels to 7×7 pixels.

GoogleNet (2014) – GoogleNet is a 22 layer CNN that won the 2014 ILSVRC. The novelty of this work was the introduction of the Inception module [164]. This module aims to reduce computational costs while increasing the width and depth of the network. The inception module has been extended several times with recent iterations including Inception-V3 [166] and Inception-V4 [165] models.

VGG (2014) – The VGG developed by Karen Simonyan and Andrew Zisserman consists of two versions (VGG-16 and VGG-19 models) and was awarded second place in the ILSVRC 2014 challenge [153]. The two network architectures have depths of 16 and 19 layers respectively. VGG decreased the filter sizes of ZFNet to 3 x 3 with the motivation that these smaller filter sizes are capable of gathering more information from input images.

ResNet (2015) – This network, developed by Microsoft Research Asia, won the 2015 ILSVRC with an error rate of 3.6% [57]. This model uses a residual learning framework that aims to simplify the training of deeper networks and yield higher accuracy. This network consists of 152 layers and was extended to 1001 layers on CIFAR-10, achieving an error rate of 4.62% [58].

It is clear that there is a trend of increasing the depth of the network, producing increasing performance, while reducing computational costs. However, these trends can make CNNs more vulnerable to over-fitting. Strategies for avoiding over-fitting include Batch Normalization (BN) [71] and dropout [159].

CNNs consist of multiple different layers. These layers include convolution, BN, activation, pooling, and fully connected layers. All these layers work together to achieve a specific goal. These layers are discussed in the following section.

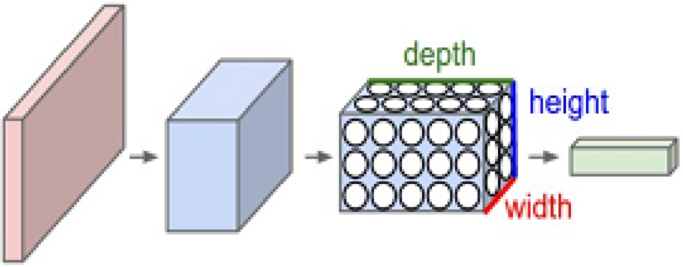
3.2.1 Convolution Layer

Convolutional layers are the most important building blocks of the CNN architecture. They are linear, shift-invariant operational layers which perform a locally weighted combination of the inputs. The benefit of this layer is that parameters are shared across the layer which results in fewer parameters. The second benefit is obtained by restricting each neuron to small localised regions of the input vector rather than the entire image. The convolution layer starts by accepting a three-dimensional image tensor, containing three channels if it is a colour image and one channel if it is black and white, sized pixels with height and the width of the image tensor as shown in Figure 3.1. These layers extract low-level features from the input tensor I by means of convolution operations using a two-dimensional kernel K,

$$\begin{aligned} c(i,j) &= (K * I)(i,j) \\ &= \sum_m \sum_n I(i-m, i-n)K(m,n) + b_{m,n} \end{aligned} \tag{3.2}$$

where $b_{m,n}$ is a bias parameter, and i, j denote the coordinates of a feature map pixel. There are also other parameters needed to be considered in this layer including stride and padding.

FIGURE 3.1: The inputs of the convolution layer.



3.2.2 Activation Functions and Non-linearity

Activation functions add non-linearity to the CNN architecture. For this research, we will focus on a number of activation functions that are most commonly used, including sigmoid, Tanh, Rectified Linear Units (ReLU), Leaky Rectified Linear Units (Leaky-ReLU), and Parametric Rectified Linear Units (PReLU).

Sigmoid function

The sigmoid is a logistic activation function which maps the intervals $(-\infty, \infty)$ onto the bounded range of $(0, 1)$ values. The sigmoid function is defined as

$$f(z) = \frac{1}{1 + e^{-z}}, \tag{3.3}$$

with a slope shown in figure 3.2. It is mostly used in models where the output is required to predict the probability of the image class. Even though it is commonly used, the sigmoid function

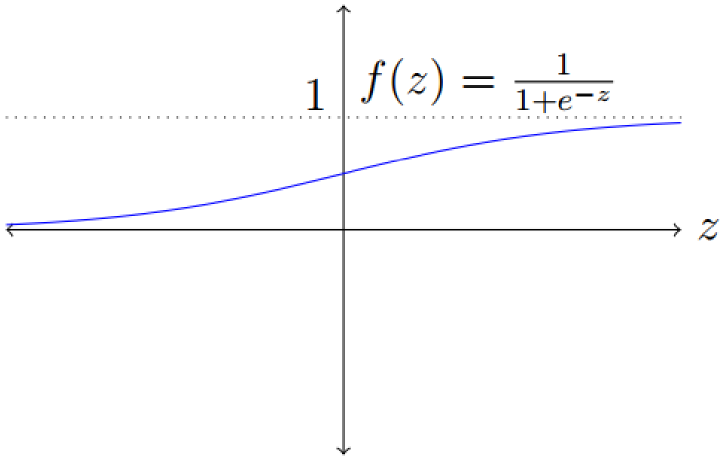


FIGURE 3.2: The sigmoid function has a characteristic S-shape curve.

treats small gradients as zeros. Furthermore, intervals need a careful selection because they can lead to saturation. This can bring instability to the network because all gradients can be negative or all can be positive during backpropagation.

Hyperbolic tangent function

In contrast to the sigmoid function, Tanh maps the intervals $(-\infty, \infty)$ onto the bounded range of $(-1, 1)$, as shown in Figure 3.3. The Tanh is a zero-centred function providing stronger gradients than the sigmoid function when z has very small gradients. The advantage of Tanh is that it results in faster training convergence for normalised data inputs [92]. The Tanh function is defined as:

$$f(z) = \text{Tanh}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{3.4}$$

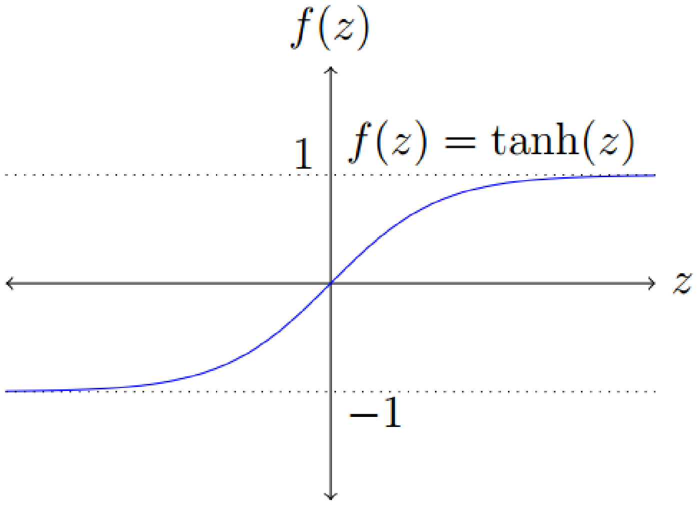


FIGURE 3.3: The tanh function

Rectified linear units

The ReLU is one of the most commonly used functions because it results in faster training time when compared to the above functions. However, this activation has the disadvantage of potentially allowing neurons to output only zero values. This is caused if neurons are never activated, as zero gradients flow through ReLU and the neurons, which causes the weights to never activate. Furthermore, having higher learning rates can also affect the output values, as these are unbounded. The ReLU function is defined as:

$$f(z) = \begin{cases} z, & \text{if } z > 0, \\ 0, & \text{if } z \leq 0. \end{cases} \tag{3.5}$$

For this research, unless otherwise specified, the ReLU activation function is selected because it is easy to backpropagate the errors and have multiple layers of neurons being activated by the ReLU function , as will be illustrated in the implementation section.

Leaky-ReLU

In an attempt to improve the ReLU function, the Leaky-ReLU uses a modification to fix the zero gradient problem. A small positive constant is introduced to multiply the pre-activation stage. It is defined as follows

$$f(z) = \begin{cases} z, & \text{if } z > 0, \\ \alpha z, & \text{if } z \leq 0. \end{cases} \tag{3.6}$$

The idea of introducing the constant assumes that there is a small non-zero gradient for negative pre-activations. The constant can be defined for the entire layer or for each neuron.

PReLU

PReLU was first introduced by He et al. in 2015. These are an extension of Leaky-ReLU that includes the constant as one of the trainable parameters. It is defined as:

$$f(\alpha, z) = \begin{cases} z, & \text{if } z > 0, \\ \alpha z, & \text{if } z \leq 0. \end{cases} \tag{3.7}$$

The α constant is a trainable parameter.

3.2.3 Pooling Layers

The pooling layer is also called the down-sampling layer and is used to reduce the dimensionality of the input. Pooling layers are mostly used between convolution layers depending on the chosen parameters for the convolution layer, to prevent overfitting. It has its own filters that contain untrainable weights, but do not contain bias parameters.

Max-pooling – takes the maximum value in a window of pixels where window size $m = (2, 2)$ is the most commonly used. For example, given input with a shape $X \times Y \times N$, max pooling could reduce the size of the input layer by 25%, to $\frac{X}{2} \times \frac{Y}{2} \times N$.

3.2.4 Fully-Connected Layers

Fully-connected layers are typically the last layers in the CNNs architecture, which contain neurons where each neuron is connected to all outputs of the previous layer. The outputs of this layer are often one-dimensional vectors, the last of which is associated with the predicted class category. For classification, outputs are reported as probability values of a certain class. The computation of the fully-connected layer is defined as follows:

$$\mathbf{Z} = \mathbf{W}\mathbf{a} + \mathbf{b} \tag{3.8}$$

where $\mathbf{W}$ is a weight matrix and $\mathbf{a}$ is the input from the previous layer.

3.2.5 Batch normalisation

This technique was first introduced by Sergey and Christian with the aim to normalise the output distribution of every node in a layer [70]. It is used to prevent internal co-variant problems and to help in speeding up the training process. The BN process consists of four steps, given values of x over a mini-batch $\mathcal{B} = \{x_1.....x_m\}$; and learned parameters γ, β. The first step is to calculate the mean of the layer input by:

$$\mu_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

The second step is to calculate the mini-batch variance:

$$\sigma_{\mathcal{B}}^2 = \frac{1}{m}\sum_{i=1}^{m} x_i(x_i - \mu_{\mathcal{B}})$$

This is followed by normalising the layer inputs:

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \varepsilon}}$$

The ε is a small constant variable for numerical stability. The final step is the scale and shift step:

$$y_i = \gamma \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$$

The γ and β are learned during training along with the original parameters of the network and the variances are fixed at the testing time.

3.3 Neural Network Training

Given all the above information about the structure of the CNN building blocks and the various layers required, we now have all the tools required to build a neural network. As mentioned before, a CNN architecture is built by sequentially stacking multiple layers on top of each other, where the output of one layer becomes the input of the next layer until the prediction layers. However, one needs to understand how to train the weights that are contained in the CNN architecture. The section below first discusses the loss function and the role it plays. This is followed by a discussion of optimisation techniques and regularisation.

3.3.1 Loss Function

The loss function plays a vital role in the training phase of the neural network model. It evaluates the performance of the network after every iteration, a lower loss value indicates that the network is performing better. There are multiple loss functions that are commonly used, but this thesis only discusses the mean squared error and cross entropy loss functions.

Mean Squared Error

The mean squared error is the most commonly used loss function for regression problems and calculates the squared error between the output values and the target variables, and then averages over the number of variables. The mean squared error is defined as

$$L = \frac{1}{M}\sum_{i=1}^{M}(y_i - o_i)^2, \tag{3.9}$$

where M represents the number of training examples passed to a neural network with an output that contains a single neuron. y_i is the target label that is associated with training examples. The network produces an output of o_i.

Cross Entropy

Cross entropy is another commonly used loss function, which is used for classification problems. This loss can be used with a logistic (binary classification) or softmax activation function on the output layer. Both of these functions produce a probability output between 0 and 1. In this research, the binary cross entropy is used because the expected predictions contain only two states, which are whether the driver is drowsy or alert. Cross entropy error is calculated as follows:

$$L = -\sum_{i=1}^{M} y_i \log(o_i). \tag{3.10}$$

Where M is the number of classes, y_i is the binary indicator, and o is the predicted probability observation i.

3.4 Optimisation and Learning Rates

Neural networks are trained by updating trainable parameters of the model using gradient descent to minimise the loss. Backpropagation is mostly used in the training process because it is an iterative and recursive technique for calculating the weight updates to improving the network efficiency. This technique was first introduced in the 1970s and gained popularity through the paper by Hinton et al. [143]. The backpropagation algorithm calculates the activation of each node and computes the error of the output to find how much each node contributes to the overall error of the network. After obtaining the error contributions, the output is used to calculate the partial derivatives with respect to the trainable parameters. These partial derivatives are then multiplied by a learning rate to update the network parameters in the direction that reduces the loss. The full derivation of the backpropagation algorithm is beyond the scope of this thesis, but readers are referred to [143] for additional details.

The learning rate value is one of the most important hyper-parameters during the training process. This has an influence on the performance of the neural network. It is a small positive constant, and poorly chosen learning rates can make the network fail to converge. There should be a number of tested learning rates on a particular chosen architecture so as to find the most suitable rate.

There are a number of optimisation techniques that can be used for gradient descent, including Root Mean Square Propagation (RMSprop), Adaptive Moment Estimation (Adam), Adaptive Gradient (ADAGRAD) and Adaptive Learning Rate Method (ADADELTA).

ADAGRAD – This is an optimisation technique that adapts the learning rates of the parameters by computing smaller updates [35]. ADAGRAD is used for speeding up the learning for slow learning parameters. It also speeds up the training time based on constant accumulation of partial derivatives. The update rule of ADAGRAD modifies the general learning rate η at each time step t for every parameter of θ which is based on past gradients. The ADAGRAD update rule is as follows:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}, \tag{3.11}$$

where G_t is the diagonal matrix with i, i diagonal elements the sum of the square of the past gradients. ϵ represents a smoothing variable. Since G_t contains the sum of the past gradients of all θ parameters, it is vectorised by using the vector product $\odot$ between G_t and g_t. The expression becomes:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t. \tag{3.12}$$

ADAGRAD automatically updates learning rates. The disadvantage of ADAGRAD is based on its accumulation of the squared gradients in the denominator which results in shrinking learning rates.

ADADELTA – This is an extension of ADAGRAD which tries to reduce the rapid decrease in its learning rate [192]. This technique restricts past accumulated gradients to a window with a fixed size instead of storing all past squared gradients. The previous and current gradients are given by:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma) g_t^2. \tag{3.13}$$

where γ is set to 0.9 which has a similar value as the momentum variable. From equation 3.13, the update vector $\triangle\theta_t$ is expressed as follows:

$$\triangle\theta_t = -\eta \cdot g_{t,i}$$

$$\theta_{t+1} = \theta_t + \triangle\theta_t$$

This yields the update rule of ADADELTA, which is given by:

$$\triangle\theta_t = -\frac{RMS[\triangle\theta]_{t-1}}{RMS[g]_t} g_t$$

$$\theta_{t+1} = \theta_t + \triangle\theta_t \tag{3.14}$$

where $RMS\triangle\theta_t$ is the root mean square of previous squared gradients up to time $t.\sqrt{E[\triangle\theta^2]_t + \epsilon}\cdot$ represents the squared gradients, where a constant ϵ is added for better results. The default learning rate is not necessary for ADADELTA and has been eliminated from the update rule.

RMSprop – this technique was first introduced by Hinton et al. [61]. The idea was to keep an exponentially weighted moving average of the magnitude of the recent partial derivatives to normalise the current partial derivatives. This is similar to the vector of ADADELTA in equation 3.13 where the value of γ is 0.9 and 0.001.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{3.15}$$

Hinton recommended that Υ to be set as 0.9 and the learning rate η to 0.001, using equation 3.13.

Adam – this is used to compute adaptive learning rates for each parameter [83]. The RMSprop stores previous decaying average squared gradients v_t, but Adam also keeps the exponential decaying averages of the past gradients m_t, which is similar to momentum. The past decaying averages of m_t and v_t can be computed as follows

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

where v_t and m_t are the estimates of the first moment (mean) and the second moment (the uncentred variance) of the gradient. There is a bias when the moments approach zero, which can be corrected by computing bias of both moment's estimates as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_2^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

These moments are then used to update the parameters which yield the Adam update rule defined as:

$$\theta_{t+1} = \theta - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t. \tag{3.16}$$

3.5 Regularisation

CNN models are vulnerable to the overfitting problem, where the network generalises to a given training dataset, but fails to generalise on testing data. Regularisation methods try to prevent this by using techniques such as BN, L_1 regularisation, L_2 regularisation, and dropout.

L_1 **regularisation** – This method tries to force regularised values close to zero by modifying the loss function. The loss function is modified by penalising the network weight magnitude:

$$L_1 = L + \lambda \sum_{k=1}^{M} \mid w_k \mid = L + \lambda \parallel w \parallel^1, \tag{3.17}$$

where L is the original loss function, w is the M-dimensional collection of all the connection weights and λ is the hyper-parameter which controls the level of regularisation [40].

L_2 **regularisation** – This is another method to prevent overfitting by introducing a square term which pushes the weights towards zero. It is expressed as:

$$L_2 = L + \lambda \sum_{k=1}^{M} w_k^2 = L + \lambda \parallel w \parallel^2_. \tag{3.18}$$

Dropout – This is the most commonly used technique to prevent overfitting and is typically applied to each layer and also after every fully-connected layer. It provides a way to combine exponentially many different architectures efficiently. This method was introduced by Krizhevsky et al. and the term “dropout ” refers to dropping some of the hidden or visible units in a network. This is done by temporarily eliminating the whole connection to a particular unit of the network and it is random which improves the generalisation of the network. Most commonly used random choices for the fractions are 0.5 and 0.7, forcing the neurons to learn robust features [87]. It is used in the training phase and does not affect testing. Unless otherwise specified, dropout in the primary regularisation method is used in this research.

Unfortunately, the limitation of publicly available training datasets can expose CNNs to overfitting, where they fail to generalise well and are considered to be biased when tested on a dataset that is not covered by the training dataset. The following section briefly discusses visualisation techniques that can be used to understand what CNNs have learned from the training data.

3.6 CNNs Visualisation

This is a technique that is commonly used to interpret what is learned by trained CNN models, to improve their performance or identify where the model fails. This section will discuss a number of visualisation methods.

Layer Activation – Visualising the activations of the network during the forward-pass operation can be used to see which image features are being used for prediction by the network. The visualisation is performed layer by layer where the early layers produce activations with around edges and texture, but as the training progresses, the activation becomes clearer. This method is illustrated in figure 3.4.

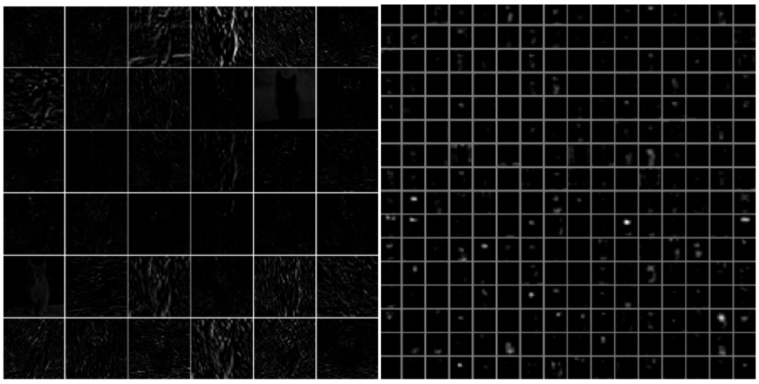


FIGURE 3.4: The left image shows the first convolution layer and the right image shows the 5th convolution layer of the AlexNet architecture from a cat image [178].

Conv Filter – This method visualises the output weights of the convolutional filters used at every layer. Filters can be inspected to see what type of image features they will activate for. 3.5.

Activation Maps – This technique is performed on the output of a particular convolution layer to produce heatmaps of class activation with respect to input images. The heatmap produced is a 2D grid of the scores associated with a particular output class. The grid represents a computation of every location from the input image and indicates the importance of each location to the output class. This technique is illustrated in Figure 3.6.

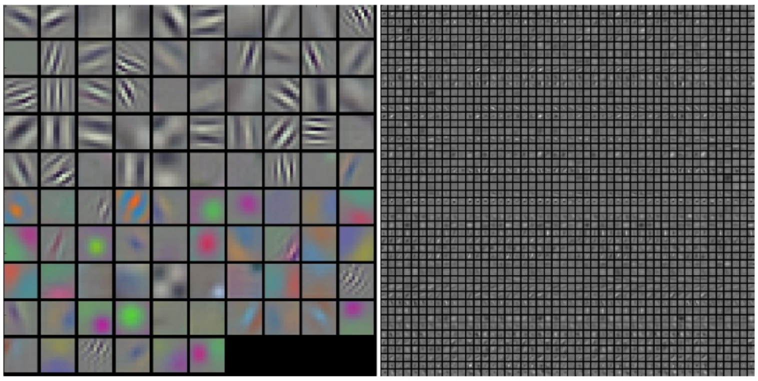


Figure 3.5: The AlexNet architecture showing the first (left) and second (right) layers. [178].

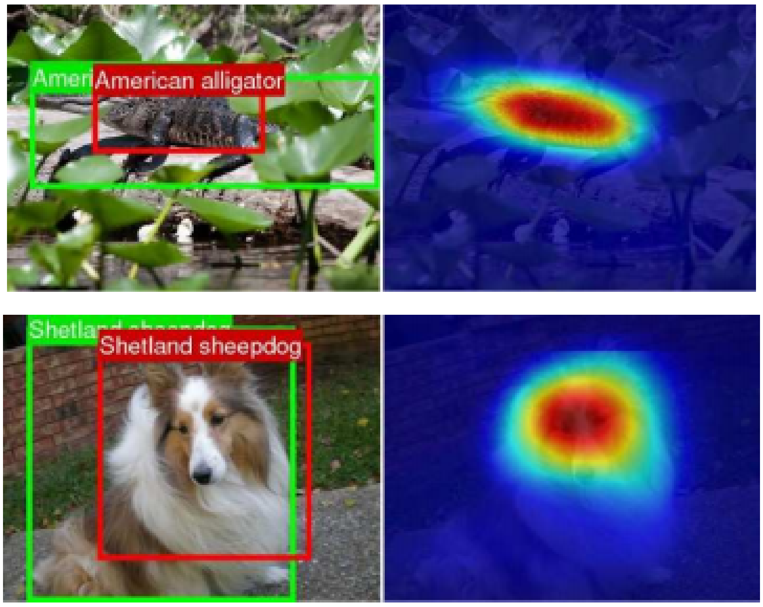


Figure 3.6: A modification of the global average pooling using the class activation mapping (CAM) localisation from GoogleNet-GAP [197].

Saliency maps are also another way of highlighting the activated regions on the input image. CNN saliency maps were first used by Simonyan et al. [152] and are derived from the concept of saliency in images. Each feature (pixels) of the image exhibits a unique context and the visualisation location in the image. The gradients of the class scores with respect to the image pixels of the final output prediction [148] determine the importance or saliency of each pixel. Figure 3.7 shows the saliency map overlays from a grid-picture, where the trained model is overlayed to highlight regions where the model is not generalising well.

Unfortunately, these visualisation techniques only allow us to investigate one image at a time. This can be employed to verify what the network was learning, but often a more overarching view is required. The t-Distributed Stochastic Neighbor Embedding (t-SNE) on the other hand minimises the divergence between two distributions by computing the probability of similarity of

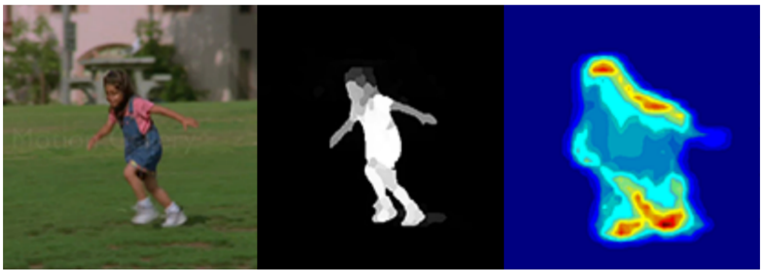


Figure 3.7: An original image from the left, the saliency map of the image in the middle and the saliency overlay on the right which represents activated regions on the image. The red colour represents activated regions where the blue represents the regions that contribute little to the classification decision of the network.

points in high-dimensional space and the probability of the similarity of points in a corresponding, learned low-dimensional space [99]. Figure 3.8 shows the t-SNE visualisation technique of the MNIST dataset.

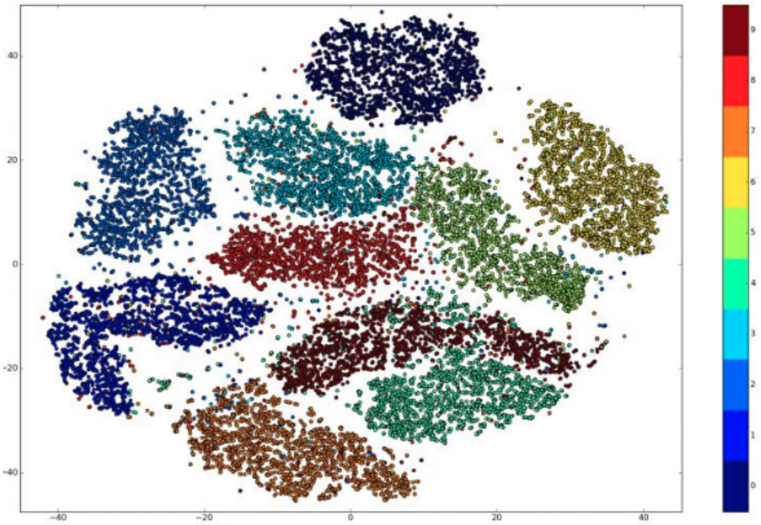


Figure 3.8: t-SNE visualisation on the MNIST dataset of handwritten digits.

3.7 Transfer Learning

When learning representational information for a classification model on a new dataset for a new target domain where the size of the training dataset is limited, there is a potential vulnerability to overfitting problems and poor generalisation performance. Collecting a sufficiently sized of dataset on a specific domain and manually labelling it is expensive and time consuming. One possible solution to this problem is the use of transfer learning. Transfer learning aims to address the generalisation problem by utilising a pre-trained network on a particular dataset. Transfer learning can be performed in two ways, including transfer learning via feature extraction or fine-tuning. In this research, transfer learning via fine-tuning is utilised. Fine-tuning is done by modifying the final layers of a pre-trained network to suit the task at hand and continuing to train the network on new data. One of the advantages of using this technique is that most of the low-level features are already learned from a large datasets such as CIFAR-10 or Imagenet, which improves the accuracy of the network.

3.8 Conclusion

This chapter has introduced the fundamentals of deep learning methods. The components that build up the CNNs have been discussed in this chapter and these components are limited to the scope of this research. Training a CNN model needs special care and careful consideration of the methods to help in the training process have also been discussed.

Visualisation techniques that help to understand what the CNN is learning were discussed in this chapter together with transfer learning methods that can be used when there is limited data available for network training. This chapter forms the basis of the implementation of CNNs in Chapter 4. In addition, a novel visualisation technique is also implemented in Chapter 4. This technique is used to gain an understanding of what the network is learning and to identify the presence of intersectional bias in the network.

4 | Intersectional accuracy differences in drowsiness detection systems

This thesis focuses on the investigation of CNNs for vision-based driver drowsiness detection. This chapter describes work conducted in an attempt to identify a dataset suited to South African contexts.

Initially, related work using CNNs for drowsiness detection is described. Chapter 2 showed that CNNs outperformed other techniques against which they were compared. However, this chapter shows that training a CNN model on a limited and unrepresentative dataset leads to a model that does not generalise well on the testing dataset. This bias arises from imbalanced training datasets captured in limited cultural contexts. With respect to the South African context, this chapter shows that standard driver drowsiness detection datasets do not cover dark-skinned races. In order to address these concerns, this chapter introduces a visualisation technique in an attempt to identify population groups where the neural network fails. Here, Population Bias Visualisation (PBV) is used to group individuals with similar appearance and an accuracy-based saliency map is used to highlight that CNNs trained on publicly available datasets fail to successfully classify the wide range of races in South Africa.

4.1 Related Work

CNNs have dominated many computer vision fields with their impressive performance. In the field of road safety, CNNs have produced outstanding results, particularly when compared with other state of the art techniques, as mentioned in the Chapter 2. Many driver drowsiness detection systems have been developed using different facial features to improve road safety by introducing early warning systems. However, despite improvements to the architecture side, there is a lack of publicly available datasets for driver drowsiness detection. This is due to the privacy and security aspects associated with publishing images of people's faces. As a result, most researchers use controlled datasets that suit their studies, and publicly available datasets are often limited.

Sanghyuk et al. [123] proposed a deep architecture called Deep Drowsiness Detection (DDD) in 2016. The architecture has three deep CNNs including AlexNet [87], VGG-Face [124], and FlowImageNet [31]. The output of these networks is concatenated and fed into a softmax classification layer for drowsiness detection. The DDD system was tested on the NTHU driver detection dataset, but the authors noted that the NTHU set lacked reliable ground truth labeling, which led them to use a substitute evaluation dataset for testing. The authors also noted that there was a lack of previously benchmarked datasets to compare with the publicly available NTHU set. Furthermore, Reddy et al. [133] proposed a compressed deep neural network model that can be deployed on an embedded board. The authors note that for their focus, the NTHU dataset had an unsuitable capture angle and inappropriate class labels. In addition, the authors noted that the DROZY dataset was also unsuitable because the images depict sensor patches attached to a subject's face, which could interfere with the results obtained. Their solution was to use a custom dataset and compare the efficacy of their approach to a number of CNNs architectures, including faster Regions with Convolutional Neural Networks (R-CNN), VGG-16, and AlexNet.

Lyu et al. [98] proposed a sequential multi-granularity deep framework for detection of driver drowsiness. The framework consists of two components, a multi-granularity CNN and a Deep Long-Short-Term Memory Network (Deep-LSTM). A contribution of this work was to utilise a group of parallel CNN extractors. The Deep-LSTM was applied on facial representations to identify long-term features of drowsiness over a sequence of frames. The model was evaluated on the NTHU set in addition to a new dataset named Forward Instant Driver Drowsiness Detection (FI-DDD). The FI-DDD dataset is a re-labeled NTHU set, as the authors note that it is difficult to locate drowsy states temporally with high precision using the NTHU labels. Following a different approach, Dwivedi et al. [37] introduced a more diverse dataset that includes persons with different skin tones, eye shapes, and eye sizes. This dataset was used to test CNN with a final softmax classification layer, but unfortunately, the dataset is not publicly available for comparison.

A recent study by Kim et al. proposed a deep CNN based on the classification of opened and closed eyes using a visible light camera sensor [81]. They used the ZJU in addition to their own dataset collected for performance analysis. Here, the ResNet-50 [57] architecture was adopted, with a modified fully connected layer. The system outperformed AlexNet [87], GoogleNet [164], VGG-Face fine-turning [124], and HOG-SVMs [126].

It is clear that many driver drowsiness detection applications rely on CNNs for robust results. The following section describes potential challenges with these techniques in more detail.

4.2 Bias in Machine Learning

Machine learning algorithms are becoming part of our everyday lives and these algorithms keep improving. However, there are growing concerns in using these algorithms because they could be flawed which can lead to be biased. In the field of road safety, it is clear that a number of modern drowsiness detection systems rely on CNNs. Bias can be present in many forms including sample bias, prejudice bias, and measurement bias. For this research, the focus is on sample bias which most likely to affect driver drowsiness detection systems.

Sample Bias – This is a problem that is encountered in the training data used for a particular task. For the driver drowsiness detection task, bias can be manifested by training a system on an unrepresentative dataset. Due to security policies regarding publishing people's faces, many datasets are not published. Publicly available datasets do not always cover a wide range of different races and ethnicities with varying facial features. In addition, publicly available datasets typically only cover cultural contexts in which they are captured [29]. For this research, the efficiency of the three most common publicly available datasets for driver drowsiness detection is evaluated, including NTHU, DROZY, CEW in South African contexts, where racial bias is likely to have a significant impact. However, the road safety community is unfortunately not the only field that is affected by the bias which is caused by using unrepresentative datasets for training.

A study conducted by Buolamwini [19] has documented extensive algorithmic bias in face detection systems, which fail on faces with darker skin-tones, while Renda et al. have highlighted bias in predictive policing [135], by showing that a system called PredoPol, which is used to dispatch police to crime hotspots tends to send police to areas where there are large numbers of dark-skinned people or Muslims. Wen et al. [181] analysed a face spoof detection algorithm, designed to recognise fake faces using image distortion analysis and reported that most current systems mis-classify individuals with dark-skinned faces as spoof attacks. Furthermore, Jiang and Nachum provided a mathematical formulation of how bias can arise and be exhibited in the training dataset [74]. The authors stated that this bias arises from the labelled dataset and this can be corrected by re-weighing the data points without changing the labels.

Unfortunately, this bias cannot be seen through training the network alone and needs additional techniques to visualise what the network is learning from the data. The following section describes a new visualisation technique, called Population Bias Visualisation (PBV), that can help in identifying the groups of people on which a model is failing to generalise.

4.3 Population Bias Visualisation

In order to address the limitations described above, a visualisation technique PBV is introduced, which makes use of PCA as a dimensional reduction method and further identifies races where trained models fail to generalise because of the lack of sufficient and representative data. This is done by using PCA to project images into a 2-dimensional grid such that images are located close to other images having similar features.

For this research, skin complexion is the primary interest, which poses a serious concern when it comes to training a network. In particular, darker skinned groups of people may fail to be analysed because of the lack of datasets captured in relevant cultural contexts.

PCA is one of the most common dimension reduction techniques that can be used to compress a very large number of features to a small number while retaining important information [156]. This is done by transforming data into an orthogonal subspace where axes (Principal components) align with the directions of maximum variance in the data. There are two methods that can be used to perform PCA, the Singular Value Decomposition (SVD) and covariance matrix method. For this research, the SVD method was used to perform PCA. Other dimension reduction techniques that are popular include Locally Linear Embedding (LLE), Isometric Mapping (IsoMap), Independent Components Analysis (ICA), and t-SNE [147, 172, 69, 99], but PCA is used because of its low computational cost and ability to retain most information on the data.

Let $\mathbf{X}$ be the matrix of images, formed by reshaping images $\mathbf{x}_i$ into row vectors (where $i = 1...N$, and N is the number of images in the dataset) and stacking these vertically to form an N x P matrix. Here, P denotes the number of pixels in each image. PCA starts by mean centering the matrix of images, which is accomplished by subtracting the mean image $\mu = \frac{1}{N} \sum_i^N \boldsymbol{X}_i$ from each 1 x P dimensional row vector, $\mathbf{X}_i$ in the matrix of images, to obtain

$$\hat{\mathbf{X}}_i = \mathbf{X}_i - \mu$$

The mean centred matrix N x P dimensional matrix of images $\hat{\mathbf{X}}$ is then decomposed using SVD, as

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$$

Here, $\mathbf{U}$ is a $P \times N$ unitary matrix, $\mathbf{V}$ is a $P \times P$ unitary matrix and $\mathbf{\Sigma}$ is a diagonal matrix comprising the singular values of $\hat{\mathbf{X}}$ in decreasing order [160]. A reduced dimensional representation of $\hat{\mathbf{X}}$ can be obtained by discarding columns of $\mathbf{U}$ and $\mathbf{V}$ as follows

$$\hat{\mathbf{X}} \approx \mathbf{U}_{0:j}\mathbf{\Sigma}_{0:j,0:j}\mathbf{V}^{\mathrm{T}}_{0:j}$$

Here, j denotes the number of columns retained. As shown above, PCA can project data into a low dimensional coordinate system, with axes provided by the columns of $\mathbf{U}_{0:j}$, and data coordinates given by $\mathbf{V}_{0:j}$.

In this research, only two columns ($j = 1$) were retained, and images projected into a two-dimensional coordinate system. Figure 4.1 shows the 2D projection (coordinates obtained from $\mathbf{V}_{0:1}$) of facial images in a test dataset compiled for this work. The grid consists of two features which are the grid position (represented by blue dots) and PCA projection (represented by red dots). The PCA projections are then overlayed over grid positions and show the scattering of the images. This projection was used to construct a grid of images, grouped by similarity. Algorithm 1 describes this process. A uniform coordinate grid was created and searched for the closest image (in the reduced dimensional coordinate system) to each point in the grid. Each image was assigned a corresponding point and ensure that no image is duplicated, by removing it from the list of available images once allocated a grid coordinate in order to produce a grid of images that groups individuals by facial similarity, as shown in figure 4.2. The process of grouping similar faces together in a grid is an unsupervised strategy which does not need a subjective classification stage. It is clear that this process successfully groups faces of similar skin tone together, with darker skinned individuals located towards the top of the image, and lighter skinned individuals towards the bottom. In addition, these faces are also sorted based on facial attributes that are important when classifying drowsiness state. These facial attributes include open or closed eyes and yawning. Even though the faces are organised by the complexion variation, the bottom right corner of figure 4.2 shows that the faces are grouped based on closed eyes and moving to the

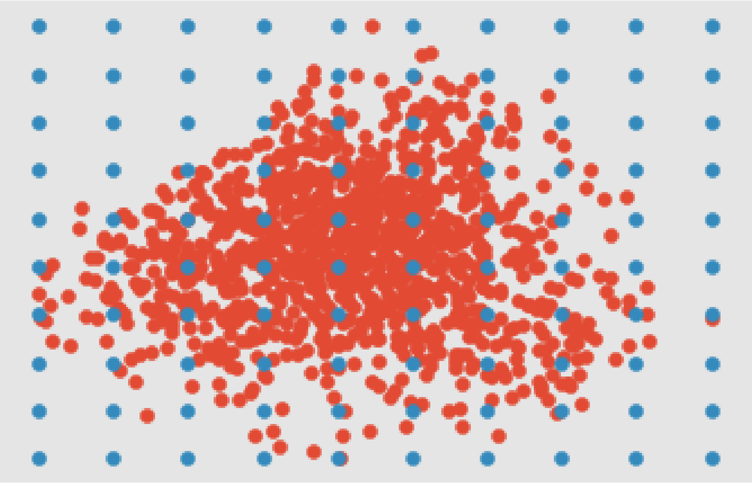


FIGURE 4.1: The figure shows 2-dimensional grid projection coordinates obtained after applying the linear PCA transformation into 2-dimensional subspace. A uniformly spaced grid is placed over the projected image coordinates, and images are assigned a grid position by finding the closest image coordinate to each grid position, ensuring each image can only be used once. The blue dots represent grid position and the red dots represent PCA projections.

bottom left corner. For each image selected, the error in drowsiness prediction was calculated, to produce a saliency map indicating model quality for the constructed grid of images.

Algorithm 1 Image overlay generation

Let $\mathbf{p}$ be the $N \times 2$ matrix $\mathbf{V}_{0:1}$

Input: $\mathbf{p}$, list of images $\mathbf{p}_i$, where $i = 1 \ldots N$, labels$_i$

```
image-grid = [ ][ ]                                   ▷ two-dimensional list of the grid images
overlay-grid = [ ][ ]                                 ▷ two-dimensional list of the model
for position-x >= x-min to maximum value x-max do
    for position-y >= y-min to maximum value y-max do
        set min-distance = 10000
        best-index = 0
        for i = 0 to maximum value of N do
            Compute the similarity distance by dist = $\sqrt{(p[i,0]-j)^2 + (p[i,1]-k)^2}$
            if distance < minimum-distance then
                minimum-distance = distance
                best-index = i
            end if
            Plot the images into a grid of images
            Apply an overlay prediction over grid of images
            Remove any duplicate images
        end for
    end for
    Output: Returns an overlayed saliency image
end for
```

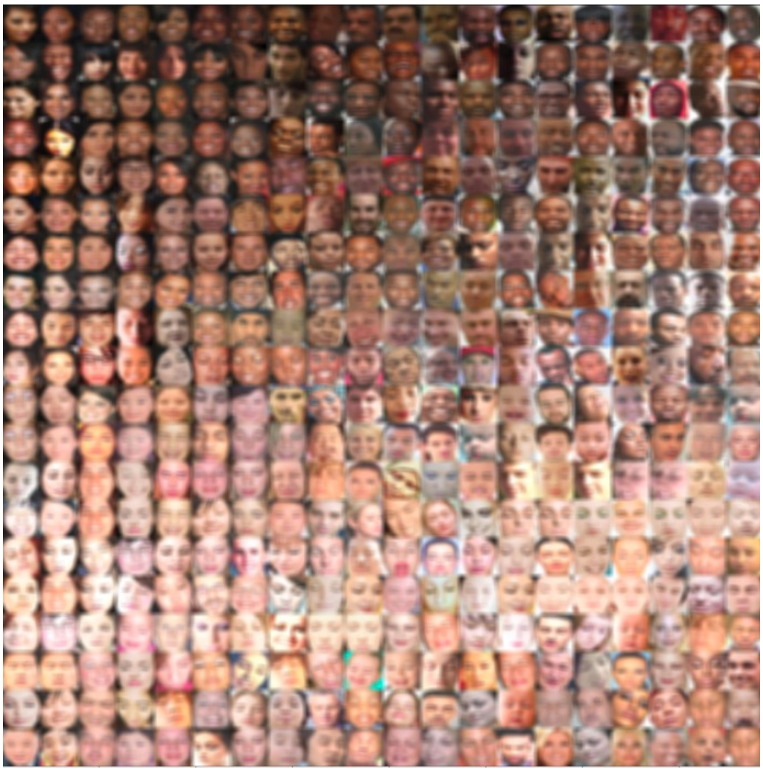


FIGURE 4.2: The proposed visualisation strategy uses PCA to sort faces by similarity without requiring meta-data.

4.4 Datasets

Neural networks rely strongly on the data used to train them, with more data typically resulting in greater performance. Despite the security constraints for publishing datasets, there has been a significant effort to build publicly available large-scale driver drowsiness datasets. This section describes the datasets that were used for training and testing the models in order to investigate the potential for racial bias to present itself in driver drowsiness detection systems.

NTHU – This dataset was introduced at the 13th Asian Conference on Computer Vision (ACCV) in 2016 [119]. The dataset is split into test and training sets. For training, there are 18 participants (10 men and 8 women) pretending to drive, with 5 scene scenarios for each participant including spectacles, no spectacles, spectacles at night, no spectacles at night, and sunglasses. For evaluation, there are images of 2 men and 2 women. Videos combining drowsy, normal and sleepy states are provided. The dataset is primarily comprised of light-skinned Asians, as illustrated in figure 4.3.

DROZY – consists of 14 participants (3 males and 11 females) [104]. Each video is approximately 10 minutes long and is accompanied by the results of Psychomotor Vigilance Tests (PVTs)

FIGURE 4.3: A sample of images from the NTHU dataset shows images with the same scenario but different behaviours like yawning, being awake, and sleeping. It also contains the same behaviours, but with different scenarios such as with and without spectacles.

FIGURE 4.4: DROZY was obtained from the Kinect v2 sensor. The images are a sample of cropped near-infrared intensity images. They only contain people from light-skinned races.

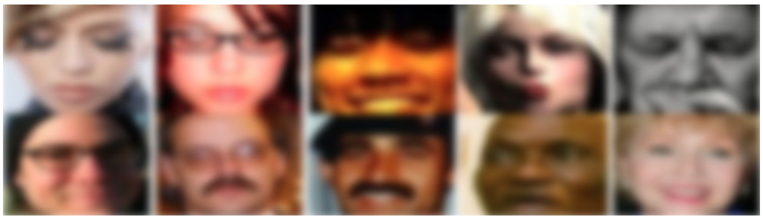


FIGURE 4.5: This dataset contains a variety of races which are collected from the internet.

regarding the drowsiness state. For each participant, the dataset (Figure 4.4) contains time-synchronised KSS scores [104].

CEW – is a collection of online images of people of different races (for example Asians and non-Asians with light-skinned faces) and contains about 2 423 participants [175]. Among the participants, 1 192 have both eyes closed and 1 231 have their eyes open. These images were selected from the labeled faces in the wild database and a selection is shown in figure 4.5.

Test Dataset – This was prepared from a collection of online videos of South African faces. It consists of 30 348 images comprising a variety of different races and ethnicities represented in South Africa. Images range from dark to light-skinned faces of both genders to provide a diverse testing dataset as shown in figure 4.6.

FIGURE 4.6: This dataset contains a variety of South African races collected over the internet.

4.5 Experiments

This section discusses experiments conducted in order to identify the bias in models trained with unrepresentative datasets, to illustrate the use of the proposed population bias visualisation system. The architectures used, together with the tuning parameters are also discussed.

4.5.1 Architectures

Three pre-trained network models are discussed in this chapter. This section will present the architectures selected, followed by a short discussion on how these architectures are constructed. These architectures were pre-trained on Imagenet, with the final layers modified to make them suitable for the driver drowsiness detection task. All experiments were done in the Keras framework using the python programming language.

ResNet50 Architecture – This architecture was introduced in 2015 and comprises 50 layers [57], which introduces the residual learning framework for easy optimisation that improves the accuracy from increased network depth. With the deeper networks, performance tends to degrade when converging, however, as the depth of the ResNet model increases, accuracy increases. This is accomplished by learning a direct mapping of the image with a residual function. The architecture is illustrated in figure 4.7.

VGG-19 Architecture – The VGG-19 architecture contains 19 layers and obtained an 8.0% error rate on a classification task. The authors also found that adding more layers could improve the efficiency [153]. The architecture contains three fully connected layers and a softmax layer as a final layer. Figure 4.8 illustrates VGG-19 architecture.

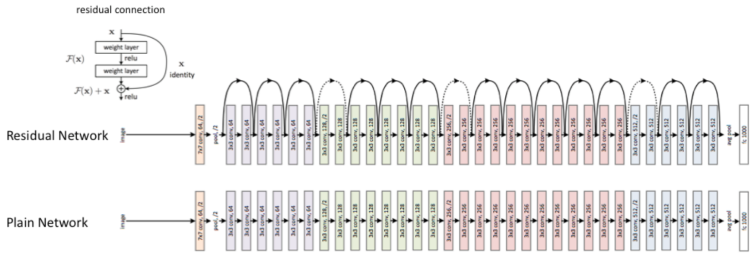


FIGURE 4.7: The RestNet architecture consists of 50 layers with 25 million trainable parameters.

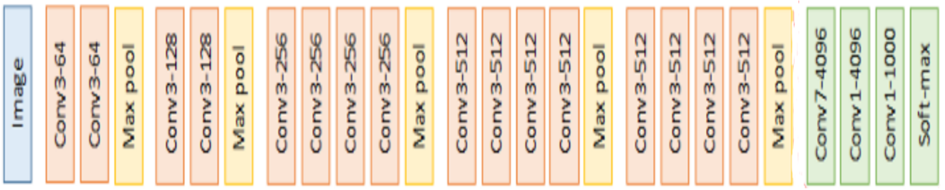


FIGURE 4.8: The VGG-19 architecture has 19 layers with 144 million trainable parameters.

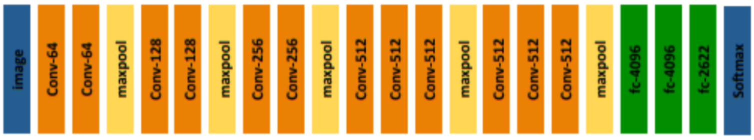


FIGURE 4.9: The VGG-Face architecture has 16 layers with 138 million trainable parameters.

VGG-Face Architecture – The VGG-Face architecture is based on the VGG-16 architecture and was pre-trained on CEW and images of faces from YouTube. The architecture was trained on 2.6 million faces. Figure 4.9 shows the VGG-Face architecture.

Proposed Architecture – The architectures in section 4.5.1 were chosen because they were pre-trained on datasets that contain faces of people which makes it more likely to generalise to drowsiness detection. These architectures were modified by replacing the final layers with BN, followed by two fully-connected layers, with two dropout layers in between and a sigmoid final layer as classification layer which is illustrated in figure 4.10.

Pre-trained models (trained for general image classification) were used as feature extractors to lower the number of parameters to be trained and to reduce training time. In addition, lower level features are already learned for the pre-trained models, which can prevent overfitting to smaller datasets. The Adam optimiser and a binary cross entropy loss was used in the training process. Data augmentation was also used in an attempt to prevent over-fitting. Here, re-scaling, shearing, zooming, and horizontal flipping was applied to extend the size of datasets used for training. Furthermore, zooming was applied to images because the face is of greatest interest in drowsiness detection. Horizontal flipping was also applied to generate different angles of the

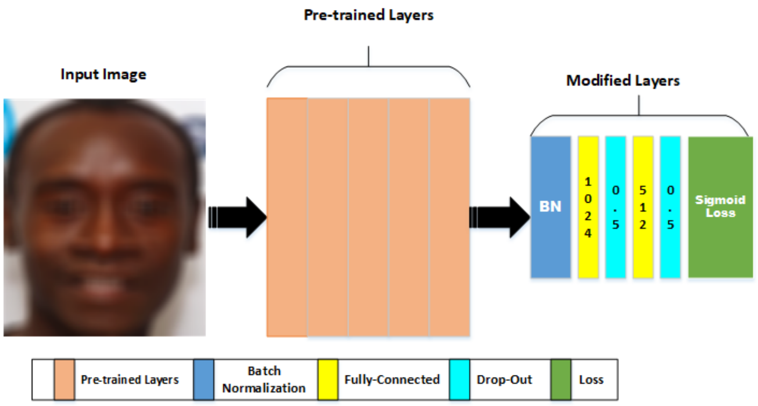


FIGURE 4.10: Proposed architecture that contains pre-trained layers which are followed by BN layer, fully-connected layers which contain dropout layers in between.

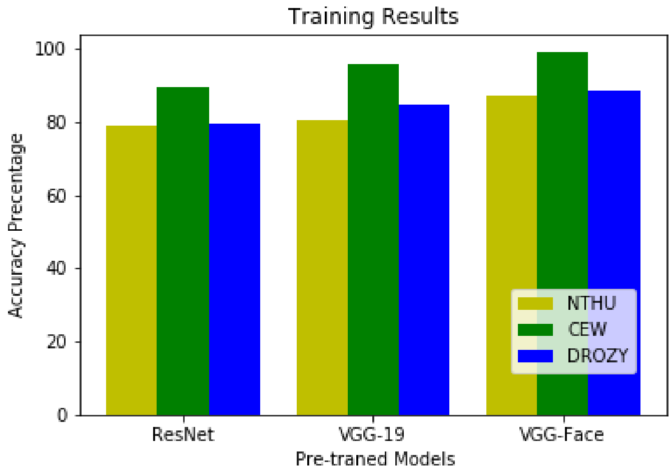


FIGURE 4.11: Accuracies obtained by testing the models on the dataset that was kept aside from the three training datasets.

drivers' faces. Dropout ($\alpha = 0.5$) was applied between fully connected layers to reduce the chances of over-fitting even further.

The drowsiness detection models were trained and evaluated on the three datasets discussed in Section 4.4, which all consist of two classes (alert and drowsy). Three models were trained on each dataset (on over 300 000 images) and evaluated on 50 000 of images that were kept aside from each of the training datasets used. Finally, the South African test dataset was used to test the three trained models.

All the images were prepared from the three datasets in the same manner for training. All images were re-sized to 150 x 150 pixels, before applying augmentation and feeding the data in batches into the model. The Adam optimiser was used to train the model and training was performed for

TABLE 4.1: Accuracies in experimentation

Experimentation Accuracy			
Pre-trained Models	Datasets	Training phase	Testing phase
ResNet	CEW	89.00%	83.77%
	NTHU	79.51%	50.08%
	DROZY	78.01%	53.23%
VGG-19	CEW	97.43%	81.73%
	NTHU	78.08%	52.39%
	DROZY	83.86%	52.96%
VGG-Face	CEW	98.98%	97.40%
	NTHU	82.43%	53.80%
	DROZY	81.89%	52.31%

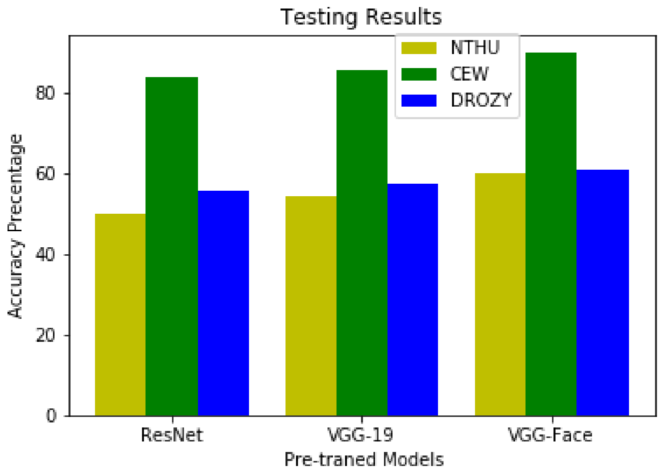


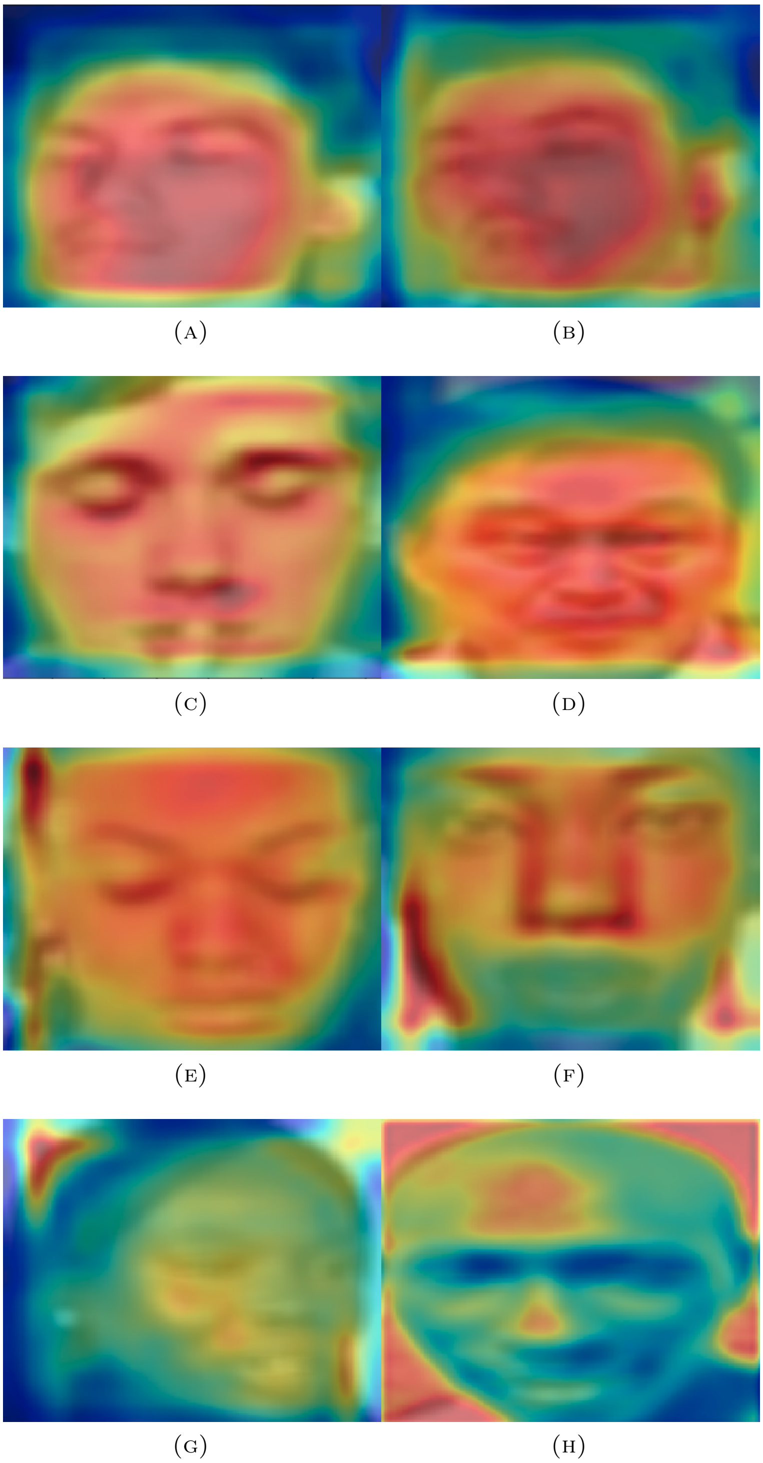
FIGURE 4.12: Accuracies obtained by testing the models on the dataset that was kept aside from the three training datasets.

30 epochs. The batch size was kept constant at 32 as it was observed that using a larger batch size degraded the model's quality.

4.6 Results

This section presents the results obtained from experiments using the described architectures. The accuracies of the modified models were tested on the testing data kept aside from the training datasets. The performance of these models is illustrated in figure 4.11. From the accuracies presented, it seems evident that all the models were generalising. The facial images are blurred for confidentiality in the presented results.

For testing the same models on the South African dataset, all images were prepared in the same way as the training dataset and used the same parameters for data augmentation. The loss and accuracy were recorded for both the training and testing phases. Pre-trained models performed well when tested on data kept aside from training sets. The data that these models were tested on first represented all the races that the models were trained on and therefore reached their highest performance. However, all models showed decreased performance when tested on South African representative dataset, as shown in figure 4.12, although the decrease in performance was marginal for the CEW. More detailed information about the accuracies of the experiments is provided in table 4.1. The South African dataset has represents all the races and more specifically targets the African ethnicities which were not represented in the training datasets. It is clear that the models trained using NTHU and DROZY completely overfit to these datasets, and failed entirely when tested on South African dataset. As a result, the NTHU and DROZY were excluded from further analysis. The dataset from NTHU and DROZY explicitly state that they only cover the cultural context in which they were captured, while the DROZY dataset was captured in an laboratory environment which does not represent real-world scenarios. As a result, training on those datasets and testing in the South African context, meant the accuracy in the training was high, as illustrated in figure 4.11, but low on the testing set. For the South African group, there is a variety of skin complexions amongst the population, training the model on more light-skinned population would lead to problems when testing the system on people with darker complexions. This is because detecting the face of the driver would be difficult. The first step is to detect the face and extract useful information and dark faces are difficult to be detected, as it is observed on the saliency map in figure 4.13. In addition to the complexion differences and the difficulty for pre-trained models to generalise to the South African context, saliency maps were extracted to understand what the models were focusing on in selected individual's faces from different ethnicities. Figure 4.13 shows a selection of saliency maps obtained when the VGG-Face models trained using the CEW dataset were tested using both light-skinned individuals (dominant in the training sets) and dark-skinned individuals (dominant in our test set). Here, red areas denote image pixels that contributed significantly to the algorithm output. Interestingly, the model seemed to focus on facial regions for the lighter skinned individuals shown here but failed to do so for darker skinned individuals, indicating a potential failure case. From the top row in figure 4.13, the model was mostly interested on the driver's face based on different scenarios (when the eyes were closed or open) which shows that the models did generalise and learned the important features to be used to decide on the state of the driver. However, as seen on the bottom row of the figure 4.13, worse results were obtained. From the bottom left image to bottom right the



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 4.13: The saliency map overlays highlight pixels in the input image that contribute to the network's final output. Areas marked in red contribute significantly, while blue regions contribute little to the final classification decision. Images (a) to (d) are from the validation set, and the saliency map highlights facial features, as would be expected for a drowsiness detector. Images (e) to (h) were sampled from our test dataset. The saliency visualisation failed to highlight facial features in the images (g) and (h).

intensity of darkness increases and the performance of the model decreases. In the bottom right image of figure 4.13, the model was mostly interested in the background of the image and the useful features from the driver's faces are not detected. The background of the image is lighter than the face of the driver which led to miss-detecting the faces and resulted in biased models for the South African context.

The proposed PBV technique was also applied and is illustrated in figures (4.14, 4.15, and 4.16). Figure 4.14 shows the PBV on the ResNet model that was trained on the CEW dataset. This model was able to generalise on the majority of races but was failing on darker faces. The CEW dataset is more diverse than the NTHU and DROZY datasets. In addition, the Resnet depth also played a role in the performance. Although the accuracy measures highlighted previously seemed to show that the models trained on CEW dataset performed well, the proposed PBV technique shows that the CEW models seemed to struggle to predict drowsiness for darker-skinned individuals at the top of the image, potentially indicating a population group for whom additional data is required to train a better model. Furthermore, figure 4.15 shows improved potential in handling the generalisation problem, but the predictions of drowsiness in the top left lie in the range from 50.06% to 67.53%, lower than that of other individuals. This shows that the model is unstable and still potentially biased. Figure 4.16 showed particularly interesting and unexpected results because the model was pre-trained on faces for the recognition task. The model's generalisation level was very low when compared to the two other two models.

The key findings of these experiments are as follows:

- All training experiments performed well when tested on data kept aside from training datasets (85.3% to 98.7%).
- All three models showed a decrease in performance when tested on our more representative test dataset, indicating some overfitting.
- Models trained using NTHU and DROZY completely failed to generalise.
- Models trained using the CEW dataset failed to perform well for certain dark-skinned individuals, indicating a need for additional training data covering these population groups.

One of the main reasons for a drastic decrease in figure 4.12 is based not on the features contributing to the level of drowsiness but is due to the failed detection of the driver's face. The main contributing factor is the complexion difference in these ethnicities, as the training set

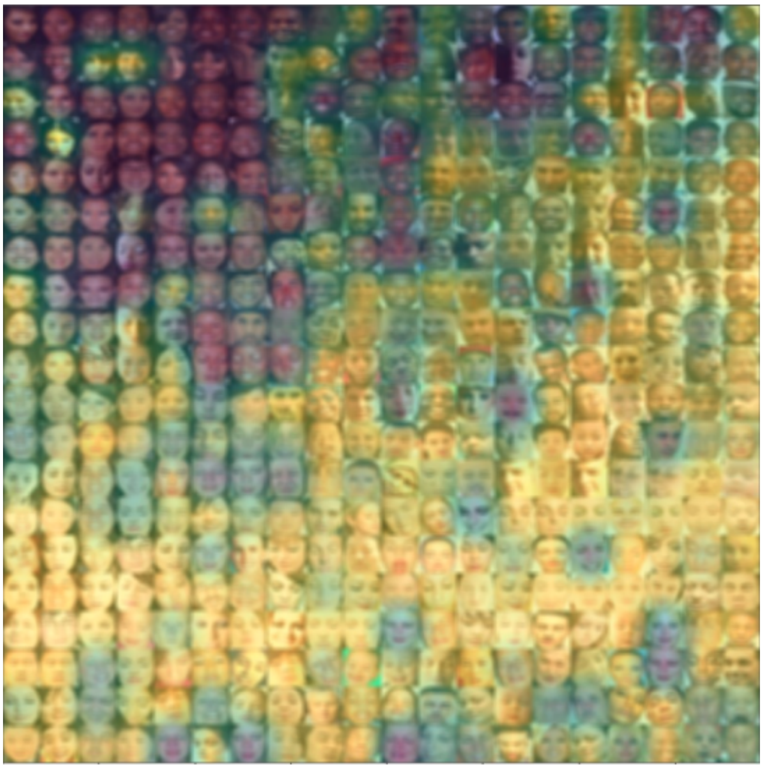


FIGURE 4.14: The figure shows the results obtained when performing the population bias visualisation technique on the Resnet model trained on CEW dataset.

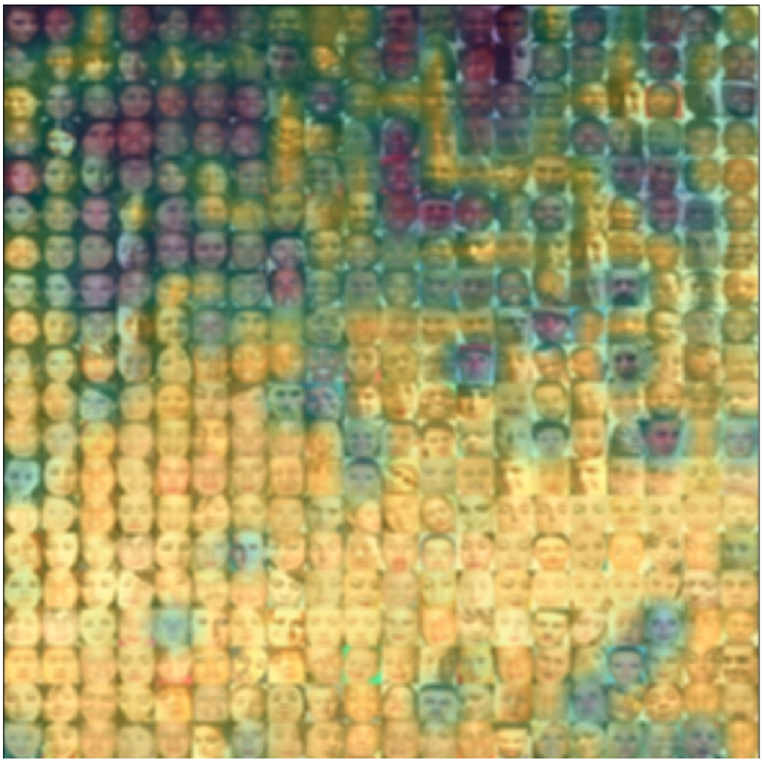


FIGURE 4.15: Visualisation of the VGG model trained on the CEW dataset.

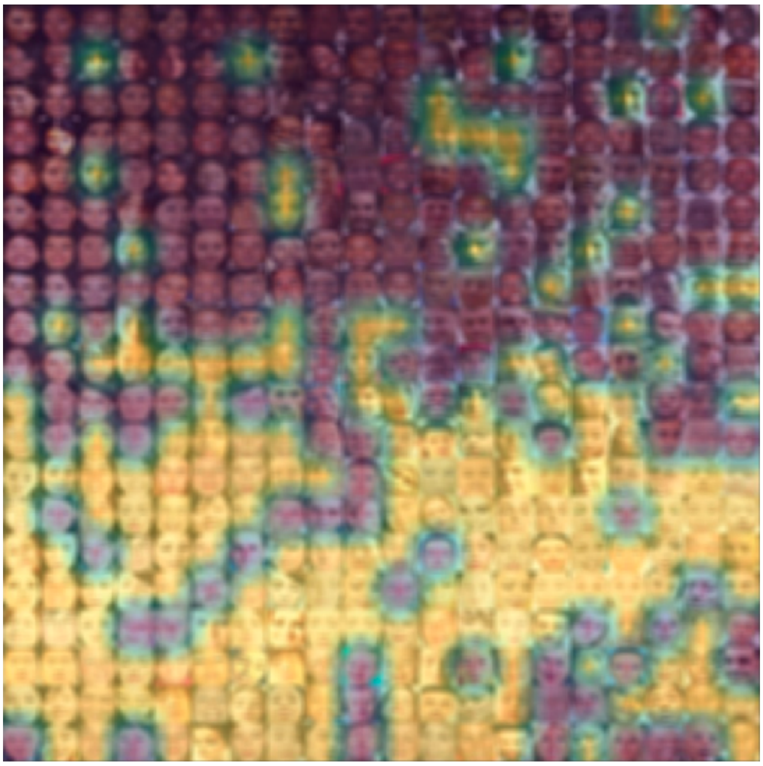


FIGURE 4.16: VGG-Face model results shows that the model is biased even though it was previously trained on the face dataset.

contains more light-skinned individuals while in the test set there is a wider range complexions among the of individuals. In addition, figures 4.13, 4.14, 4.15, and 4.16 showed that with the publicly available datasets results in biased models for the South African context.

4.7 Conclusion

This chapter implemented CNNs and has shown how these can be applied to the driver drowsiness detection task. Pre-trained model architectures were identified and the final layers modified to suit the drowsiness detection task. In addition, publicly available datasets were selected, which were used to train the pre-trained models.

The problem of bias in the machine learning field was discussed and the influence of sample bias in the detection of drowsiness task was identified. Sample bias was identified to be the most important problem in the process. Publicly available datasets do not cover a wide range of races, and dataset collection is typically limited to specific cultural contexts.

A new visualisation technique which uses PCA and trained model's predictions was developed and from the results, it showed that there is bias in models, resulting from the dataset used for training. This was observed in all tested models, showing that these models fail when tested on dark-skinned races. In addition, the VGG-face model which was trained on faces only also failed to generalise, indicating that the face dataset used for pre-training is also biased.

Furthermore, the saliency map method was applied to support the introduced visualisation technique and it also showed that dark-skinned faces were not used by the model for prediction. The following chapter provides a GAN-based data augmentation technique that can be used to generate synthetic data to populate and balance the dataset. This technique is tested on facial attributes datasets.

5 | GAN-based data augmentation

The challenge regarding the availability of driver drowsiness datasets was introduced in Chapter 4. This chapter focuses on generating synthetic data using GANs, in an attempt to motivate the use of generative models to produce more data for solving the driver drowsiness detection task. These generative models are powerful, but challenging to train since they are sensitive to parameter selection for the neural networks used to build them.

Initially, an overview of GANs for synthetic data generation is provided. The history of GANs is discussed, and this is followed by a discussion on the types of GANs together with the implementation of a new GAN architecture which extends the knowledge from related work. Furthermore, methods of comparing generative models to evaluate the efficiency of the models are provided. The introduction of DSCs and their usage as replacements for the standard convolutions is also proposed here. This chapter shows that DSCs can be used to reduce the training time and computational resources by requiring less trainable parameters when compared to standard convolutions. This technique is only applied to the generator network because this is the network that is responsible for creating synthetic images. This chapter also investigates the benefits and trade-offs of using DSC.

5.1 Generative Adversarial Networks (GANs)

GANs were introduced by Goodfellow et al. in 2014, where the goal was to generate synthetic data [51]. The architecture of GANs makes use of two sub-networks, a generator G and a discriminator D, which are trained by playing a mini-max game against one another. In the case of image translation, to learn a generator distribution p_g over data x, the generator creates a mapping function, parameterised by θ_g, from a prior latent space distribution $p_z(z)$ to data space $G(z;\theta_g)$. The discriminator $D(x;\theta_d)$, on the other hand, learns parameters θ_d to distinguish whether data is from the training data or from the generator. These two networks are trained simultaneously using a gradient-based update rule and loss for the corresponding network. The mini-max game function $V(G, D)$ is given by

$$\begin{aligned} min_G\ max_D\ V(D,G) &= \mathbb{E}_D + \mathbb{E}_G \\ \text{where } \mathbb{E}_D &= \mathbb{E}_{x \sim p_{\mathbf{data}}(x)}[\log D(x)] \\ \mathbb{E}_G &= \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \end{aligned} \tag{5.1}$$

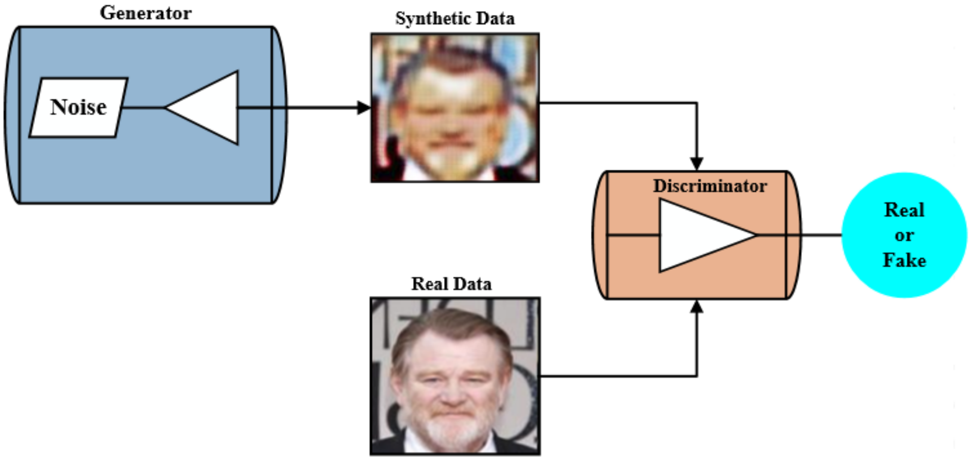


FIGURE 5.1: A simple architecture of GANs.

Figure 5.1 shows a simple graphical representation of the GAN architecture. Equation 5.1 is optimised by running optimisation on both G and D, but G is optimised once for multiple steps of D, in order to keep D in its optimal region. Algorithm 2 shows the original training algorithm that was proposed by Goodfellow et al. [51]. The efficacy of GANs are based on well researched and understood concepts such as Stochastic Gradient Descent (SGD), activation functions, and dropout layers. Although GANs are extremely popular and have produced remarkable results, they suffer from various problems including the instability of the training process, non-convergence, and diminishing gradients [174, 84, 189]. Mode collapse is a problem where the generator learns to map multiple latent distributions z input to the same output $G(z)$ which leads the generator to only create a single mode with the largest discriminator response. Numerous forms of GANs have been implemented since the original implementation.

DCGAN – This architecture was introduced by Radford et al. in 2016 with the aim to use

Algorithm 2 GAN original algorithm.
Minibatch stochastic gradient descent training of generative adversarial networks.

```
for number of training iterations do
    for $k$ steps do
        Sample minibatch of $m$ noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$.
        Sample minibatch of $m$ examples $\{x^{(1)}, ..., x^{(m)}\}$ from data generating distribution $p_{data}(z)$.
        Update the discriminator by ascending its stochastic gradient:
            $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right]$
    end for
    Sample minibatch of $m$ noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$.
    Update the generator by descending its stochastic gradient:
        $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$
end for
```

popular CNN techniques [132] for image generation. The architecture consists of a Fully-Convolutional Neural Network (FCNN), batch normalisation, and activation functions to improve the stability of the training process. The DCGAN architecture was trained on the Large-scale Scene Understanding (LSUN) bedroom dataset which contains over 3 million training images [191]. The authors experimented with the latent distribution input of the generator by using two random noise vectors which resulted in establishing a representation of the submanifold spanned by input images. The discriminator was used as a feature extractor on CIFAR10 and Street View House Numbers (SVHN) [88, 113] datasets and achieved comparative results on classification tasks when compared with other unsupervised algorithms. Figure 5.2 shows the DCGAN generator architecture.

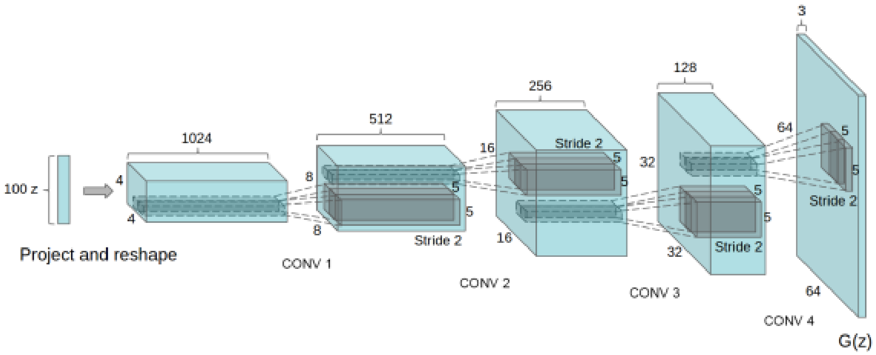


FIGURE 5.2: The DCGAN that was used for LSUN scene modeling.

Wasserstein Generative Adversarial Networks (WGAN) – introduced a new loss function for optimising the standard GANs and were developed by Arjovsky et al. in 2017 [8]. They analysed ways of measuring the distance between the model distribution and a real distribution and proposed using the Earth-Mover or Wasserstein-1 distance. The WGAN resolved the mode collapse problem encountered by the original GAN.

Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP) – improves the WGAN architecture by introducing a gradient penalty term [53]. Here, the authors stated that failures that were encountered using the WGAN were due to the way a 1-Lipschitz constraint is enforced. The weight clipping method that was presented in WGAN exhibited problems in finding the optimal functions, which lead to vanishing gradients thereby making the training unstable. This work proposed a soft penalty on the gradient norm of the discriminator weights to enforce the 1-Lipschitz constraints on WGAN. The new gradient penalty is as follows:

$$\underbrace{L = \mathbb{E}_{x\sim\mathbb{P}_g}[D(\hat{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)]}_{\text{WGAN critic loss}} + \underbrace{\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2]}_{\text{WGAN-GP gradient penalty}} \tag{5.2}$$

where λ represents the penalty coefficient, with the authors suggesting it to be set equal to 10. They claimed this works well for a multitude of experiments. The WGAN-GP architecture omitted the use of batch normalisation layer on the discriminator network because the gradient penalty is invalid with batch normalisation. Instead, they replaced the batch normalisation with layer normalisation. This architecture was trained on the LSUN bedroom dataset and experiments showed improved training stability in multiple datasets.

5.2 Conditional Generative Adversarial Networks (CGAN)

CGAN are an extension of the original GANs that introduce additional information in both the generator and discriminator networks [108]. This allows for the ability to control aspects of the output image, which results in flexible generated synthetic data. This information, y, is typically a label applied to the resulting output image, for example, this information can be skin complexion, age, gender, etc. This information can be used to modify equation 5.1 which results in the following expression:

$$
\begin{aligned}
min_G max_D V(D,G) = \mathbb{E}_{x \sim p_{\mathbf{data}}(x)}[\log D(x|y)] + \\
\mathbb{E}_{z \sim p_z}(z)[\log(1 - D(G(z|y)))]
\end{aligned}
\tag{5.3}
$$

A number of conditional GANs have been proposed. These are briefly discussed below.

Bayesian Conditional Generative Adversarial Network (BC-GAN) – introduces a framework that extends the traditional GAN by using the framework of a Bayesian semi-supervised learning problem [1]. Instead of using a random noise variable to generate data, the authors propose a random function that takes deterministic input. This architecture utilises the uncertainty in the model rather than the noise as input. The generator's generated distribution is defined as follows :

$$
\begin{aligned}
p(S'|f_D) \propto \int p(\omega) \int p(f_G|\omega) \prod_{i=1}^{n} p(f_D(f_G(y_i'), y_i')) df_G d\omega, \\
p(S'|\omega, f_G) = \int p(f_G|\omega) \prod_{i=1}^{n} p(f_D(f_G(y_-'), y_i')) df_G
\end{aligned}
\tag{5.4}
$$

where ω is the weight of the generator, and $p(\omega)$ is the prior on ω. The f_G is the discriminator function that measures the compatibility of the input x and output y. They experimented with their framework using MNIST and CIFAR-10 datasets which performed better when using mapping parameters of Monte Carlo than Langevin dynamics. This was caused by noisy gradients.

Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks (AlignGAN) – build upon the conditional GANs by introducing alignment cross-domain images and also a model that conditions on multiple information with a 2-step alternating optimisation algorithm [103]. It introduced two rules for achieving more visually appealing results than with conditional GANs. The first rule is to restrict the generator noise input layer to be unconditioned by domain vectors. The second rule is that in the discriminator, the image input layer should be conditioned by the domain vectors. They adopted the Least Squares Generative Adversarial Networks (LSGAN) architecture for training the models and kept the parameters the same. The Digits, MNIST and USPS datasets were used to evaluate the performance of AlignGAN.

Artwork Synthesis with Conditional Categorical GAN (ArtGAN) – was inspired by the backpropagation of the loss function, and allows feedback from labels given to each generated image through the loss function in D to G [169]. The use of cross entropy to back-propagate the error to the generator allows the generator to learn better. Moreover, this modification speeds up the training process. Here, the authors used the sigmoid function in the generator and also employed L2 pixel-wise reconstruction loss. This improves training stability.

5.3 Comparing Performance of GANs

It is difficult to compare different GAN models by visually inspecting the generated images. There are performance measures that are used to evaluate the quality of the synthetic images. Among those performance measures, the Inception Score (IS) and FID are discussed in this research. For experimental results, the FID was used as a performance measurement method.

Inception Score (IS) – was introduced by Salimans et al. in 2016 [146]. It evaluates the quality of images generated by the GANs by making use of distribution of the samples. The IS uses the inception-v3 pre-trained network and is computed as:

$$\text{IS}(G) = exp(\mathbb{E}_{x \backsim p_g} D_{KL}(p(y|x) \| p(y))) \tag{5.5}$$

where KL is the KL-divergence, $p(y|x)$ is the conditional probability, and $p(y)$ is the marginal probability which is computed by $\int_z p(y|x = G(z))d_z$. Having higher scores means better results, corresponding to a larger KL-divergence between two distributions. This means that the generated images with meaningful information have low entropy on the conditional label distribution $p(y|x)$, where the diverse images have high entropy.

Fréchet Inception Distance (FID) – is a method that approximates the image distributions as multivariate Gaussians over pre-trained bottleneck features. The FID is calculated by computing the Fréchet distance between multivariate Gaussians fitted to the feature representation of the Inception model [60]. The FID is computed by:

$$\text{FID}(r, g) = \| \mu_r - \mu_g \|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}), \tag{5.6}$$

where $X_r \backsim N(\mu_r, \Sigma_r)$ and $X_g \backsim N(\mu_g, \Sigma_g)$ are the 2048-dimensional activations of the inception-v3 layer for real and synthetic data [60]. In contrast to the IS, lower FID means better results, with the activation distributions closer.

5.4 Depthwise Separable Convolutions

Depthwise Separable Convolutions (DSCs) were introduced by Chollet in 2016 as a replacement for standard Convolutions [26]. They are a form of factorized convolutions that apply separate convolution operations to every input channel of a tensor. DSCs have fewer trainable parameters and have shown improved performance in image classification tasks [150]. DSCs are an extension to the Xception layer [164] and are composed of two components, namely a depthwise convolution and a pointwise convolution. The depthwise convolution performs a spatial convolution over each input channel data while the pointwise convolution layer performs a 1 × 1 convolution that merges information from the previous step across all channels. The original convolution layer dimensions can be expressed as H, W, C_{in}, C_{out} and K, where H is the height of an input tensor, W is the width, C is the depth and K is the size of the convolution kernel. A depthwise separable convolution operation can be expressed using dimensions H, W, C_{in}, K for the depthwise convolution and H, W and C_{out} for the pointwise convolution. This results in an operation reduction of $\frac{1}{C_{out}} + \frac{1}{K^2}$. Figure 5.3 illustrates the comparison between the standard convolution and depthwise separable convolution architectures.

Training a GAN, which is based on two deep neural networks takes time because of its large parameter count, requiring computationally intense processing and significant resource allocation. Trainable parameters increase as the depth of the network increases. Introduction of the DSC in the GAN results in a reduction of number of trainable parameters and speeds up the training time. Trainable parameters are very important for the accuracy of the network but this work shows that the introduction of DSCs can have limited impact for deeper networks. Reducing the number of operations required by the network reduces the training time which also eases the computation needed. As shown below, results indicate that there is a much greater improvement in the training time when DSCs are compared to standard convolutions.

Depthwise separable convolution has found success in a variety of applications [66, 77, 100, 180]. The work of Nguyen and Ray proposed an adaptive convolution block method that learns the upsampling algorithm [116]. They replace traditional convolutions with DSC in the generator to improve the performance of a weak baseline model. Moreover, Wojna et al. have also applied DSC in GANs architectures, where they introduced a bilinear additive upsampling layer, which improves performance [185]. However, this research investigates the performance of DSC on the upsampling, downsampling and bottleneck layers. This is only applied in the generator.

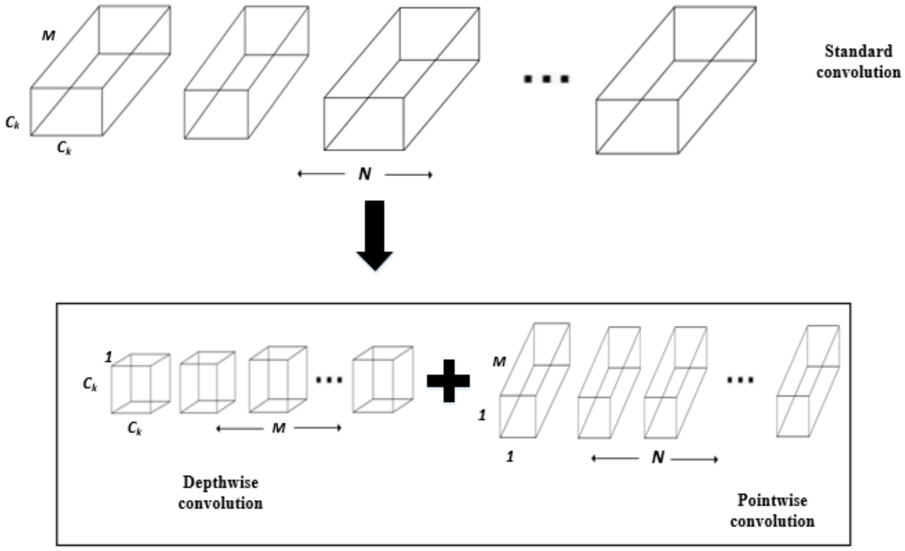


FIGURE 5.3: In contrast to traditional convolution operations performed in deep learning, DSC performs convolutions on subsets of the input tensor and aggregates the information across these subsets using a single pointwise convolution.

5.5 Experiments

This section describes the proposed modified architecture that was used for experimentation. It further discusses the datasets that were used for training and testing.

5.5.1 Architectures

For training and testing, three models which were modified from the original Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation (StarGan) architecture in order to produce lightweight networks with fewer trainable parameters on the generator network for experimental testing were prepared. The modified architecture used for testing was adapted from [25] and consists of two networks, where the generator consists of a stride size of two for downsampling and 11 DSCs (see figure 5.4). Instance normalization is an operation that removes instance-specific contrast information from the content image, which prevents mean and covariance shift and simplifies the learning process [176]. Instance normalization and ReLU activations were applied after each pointwise convolution. For the discriminator network, a Markovian discriminator (PatchGan) [93] was adopted because it is a fixed-size patch discriminator that is easily applied to 256 × 256 images.

TABLE 5.1: GANsArchitecture information

Detection Accuracy			
Parameters	DepthwiseDG	DepthwiseG	DeeperDepthwiseG
Number of layers on the Generator	6	8	11
Number of layers on the Discriminator	8	8	8
Number of trainable parameters	30.8 Million	43.9 Million	46.5 Million
Training time	9 hours	12 hours	48 hours

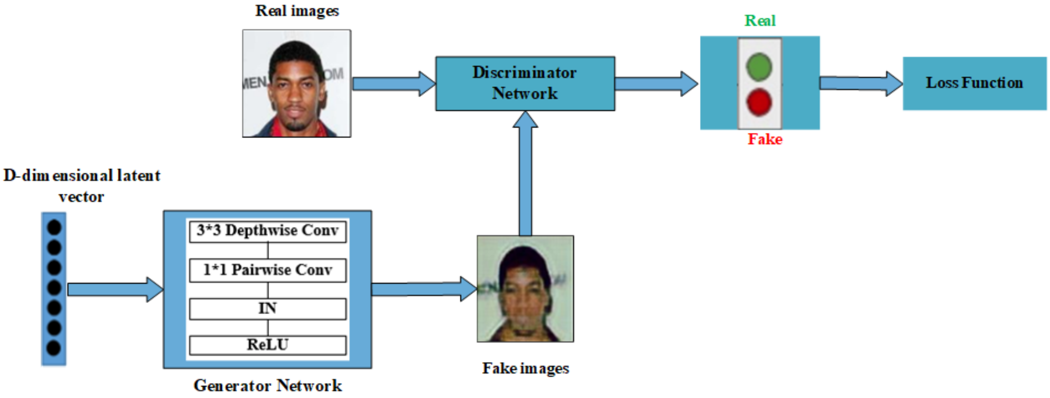


FIGURE 5.4: The DepthwiseGAN architecture is composed of two networks (Generator (G) and Discriminator (D)). DSC was used in the generator coupled with Instance Normalization (IN) and ReLu activation functions.

The three prepared models are termed DepthwiseDG, DepthwiseG, and DeeperDepthwiseG models. Architecture information is contained in table 5.1. The DepthwiseDG model replaced convolution layers with depthwise separable convolutions in both the generator and discriminator, the DepthwiseG model replaced convolutions in only the generator, while the DeeperDepthwiseG model is a deeper model with 11 depthwise separable convolution layers in the generator. All three models were compared to the original StarGan architecture.

5.5.2 Datasets

Models for conditional generation of synthetic facial images were explored using three datasets, which are discussed below.

CelebA – The CelebFace Attributes dataset contains images of about 200 000 faces of celebrities [95]. Each image is annotated with 40 attributes and faces cover a relatively large pose variation, with background clutter included. Images were cropped, with the faces centred and resized to 128 × 128 pixels. Seven facial domains were used that were investigated in [25] to train the models.

Stirling 3D Face Database – The Stirling 3D Face dataset consists of 3 339 images collected from 99 participants (45 males and 54 females) [129]. Each participant made seven facial expressions and images were captured at four different angles. The images were cropped to 256×256 pixels, with the faces centred and then resized to 128 × 128.

RaFD – The Radboud Face Database (RaFD) consists of 49 people divided into two subsets, comprising 39 Caucasian Dutch adults (19 female and 20 male) and 10 Caucasian Dutch children (6 female and 4 male) [90]. The images include eight facial expressions captured from three different angles. Images were prepared in the same way as the Stirling 3D Face Database.

5.5.3 Training process

The DepthwiseGAN training process is illustrated in algorithm 3 and uses the following adversarial loss [25]:

$$\begin{gathered}\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)]+ \\ \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x,c)))],\end{gathered} \tag{5.7}$$

where G generates an image $G(x, c)$ which is conditioned on both the input image x and the target domain label c and $D_{src}(x)$ is a probability distribution over sources given by D. The modified objective function in [25] is also adapted and used to optimize both the generator G and discriminator D as follows:

$$\begin{gathered}\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^r_{cls}, \\ \mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^f_{cls} + \lambda_{rec}\mathcal{L}_{rec},\end{gathered} \tag{5.8}$$

where λ_{cls} and λ_{rec} are hyper-parameters that trade-off domain classification and reconstruction losses. All the parameters for the training followed the StarGan training procedure and all training was conducted on a single NVIDIA Tesla K20c GPU. The Adam optimizer was used to train all the models.

5.6 Results

This section provides the experimental results obtained where the models were compared with the StarGan model in a multi-domain translation task. Three datasets were used for experiments

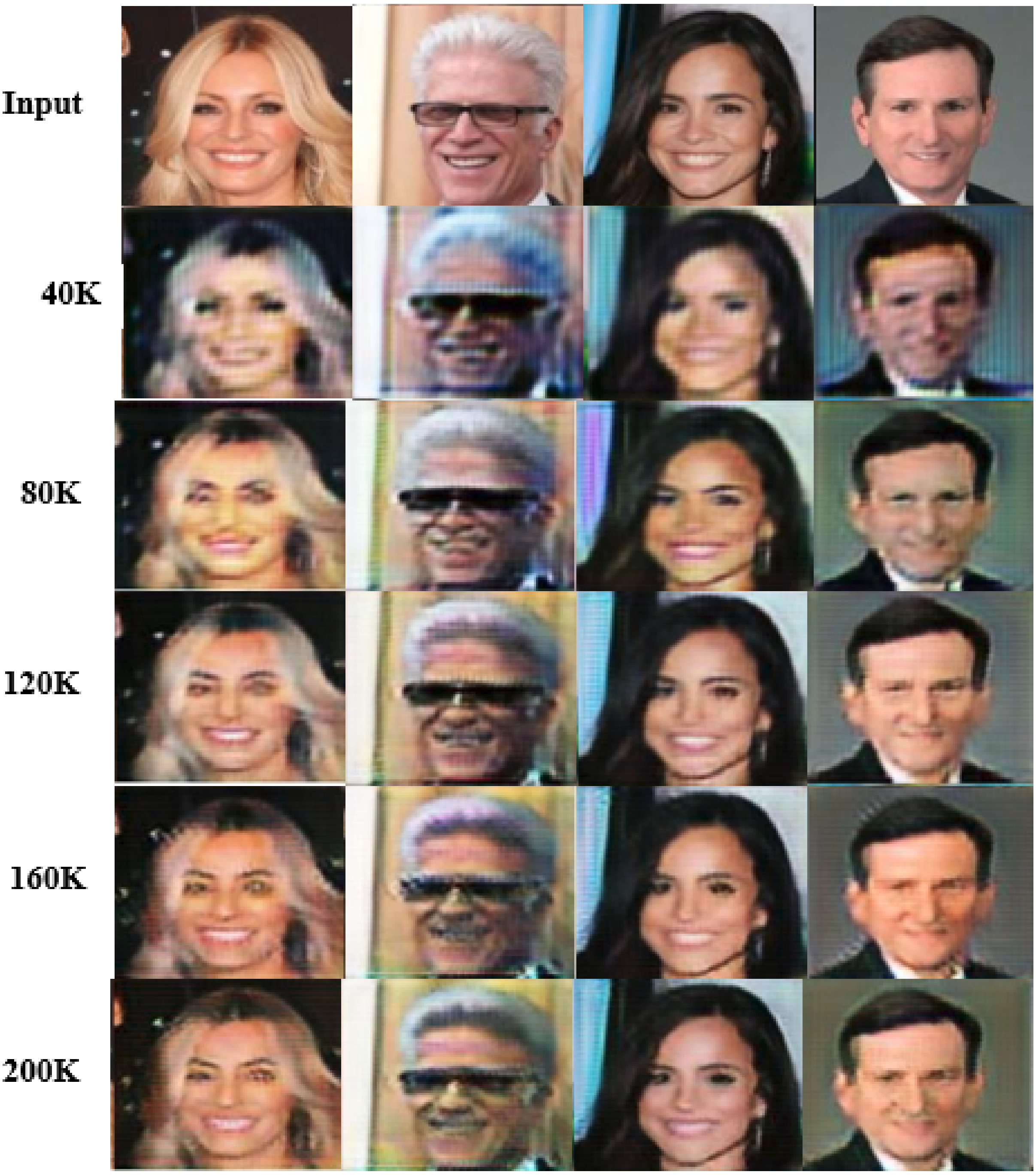


Figure 5.5: The training process of DepthwiseDG shows that the discriminator fails to learn important information since the number of parameters is reduced.

Algorithm 3 Model Training

```
Input: Given a set of real images $\{x_1, x_2, .......\} \backsim p(x)$, $\lambda_{cls} = 1, \lambda_{rec} = 10$
for number of training iterations do
    Sample mini-batch of latent sample $\{z\}$ from latent prior $p_z(z)$
    Randomly sample mini-batch of one sample $\{x_1, x_2, y\}$ from training set.
    Update the parameter $\theta_d$ by descending its stochastic gradient with the Adam optimizer:
    $\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^r_{cls}$
    Update the generator $\theta_g$ by ascending its stochastic gradient with Adam optimizer:
    $\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^f_{cls} + \lambda_{rec}\mathcal{L}_{rec}$
end for
Output Trained generator $G$ and discriminator $D$.
```

and evaluated the quality of the results using the FID. Training the standard StarGan took approximately three days (8.5 Million parameters for the generator and 44 Million parameters for the discriminator). In contrast, training DepthwiseGANs was faster. The DepthwiseG model took one day and 12 hours of training on a model with 1.5 Million parameters for the generator and 44 Million parameters for the discriminator, while training took two days for the DeeperDepthwiseG model, with 5.6 Million parameters for the generator and 44 Million parameters for the discriminator. Nine hours was required for training the DepthwiseDG model with 1.5 Million parameters for the generator and 32 Million parameters for the discriminator.

5.6.1 Results on CelebA

DepthwiseGANs were compared against the StarGan model. Figure 5.7 shows the results of facial attribute transfer on the CelebA dataset using the four models. Results indicate that including depthwise convolutions in the discriminator leads to poor quality image generation. In contrast, restricting DSCs to the generator and increasing the depth yields results indistinguishable from StarGan. The figure shows that using DSCs in both networks degraded images, which appear almost cartoon-like. As a result, it seems that additional learning parameters are required to distinguish between image domains in the discriminator. Figure 5.5 shows the images after a set of training iterations when training DepthwiseDG, it struggled to find significant features in the input data as the discriminator struggled to learn to distinguish between classes. On the other hand, the training process was more reliable for the DeeperDepthwiseG model as the discriminator's parameters were kept from the original architecture, as shown in figure 5.6.

The FID measure in figure 5.8 shows that the convergence of the distribution of images generated by models using DSCs to the training dataset distribution is faster for StarGan, although similar

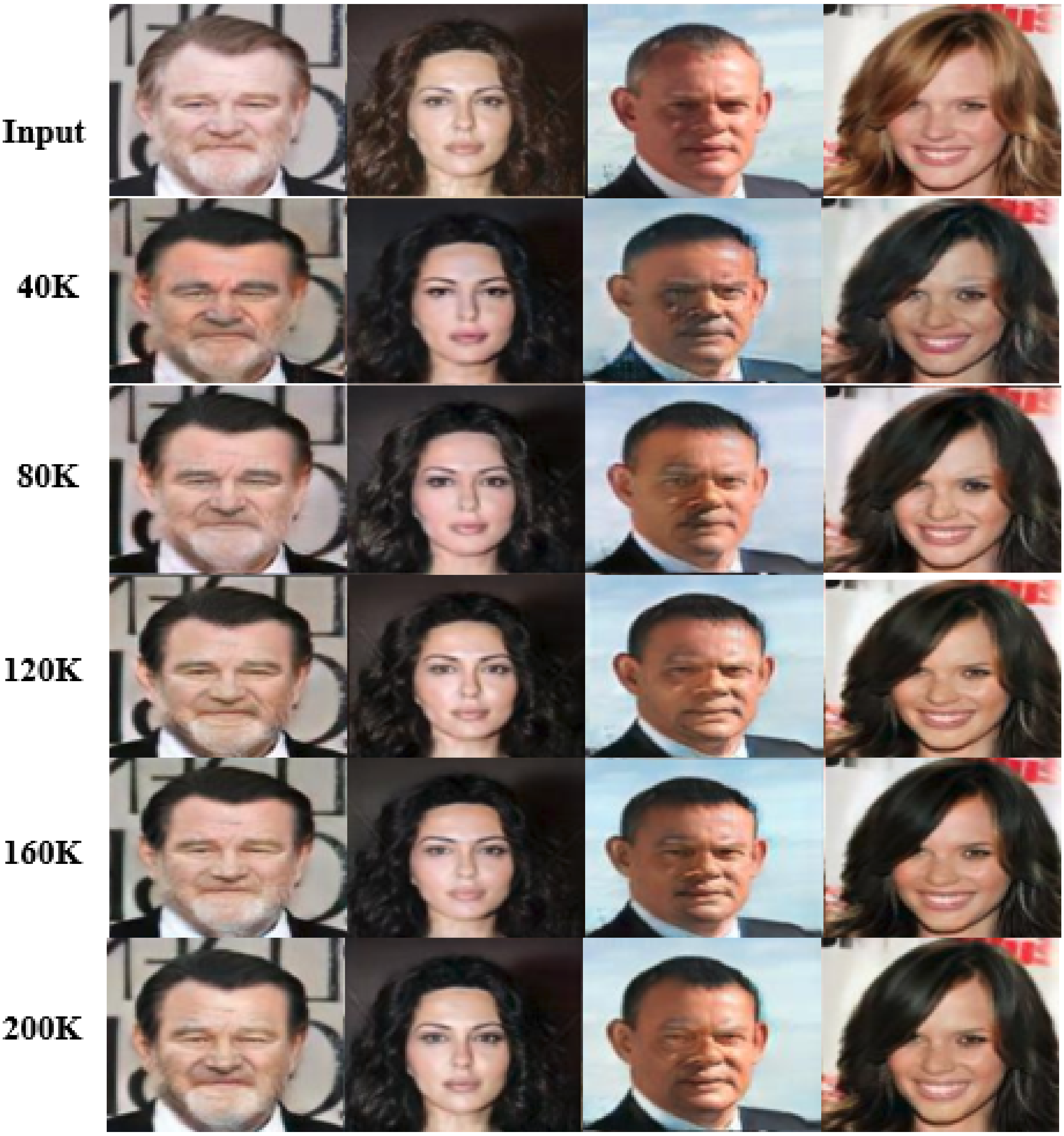


FIGURE 5.6: The discriminator is kept identical to the original architecture, but the number of generator DSCs is increased.

performance is obtained after about 175 epochs. It is clear that deeper models perform better, indicating that the model capacity introduced by additional parameters is important if generated images are to follow a similar distribution to that of the training set.

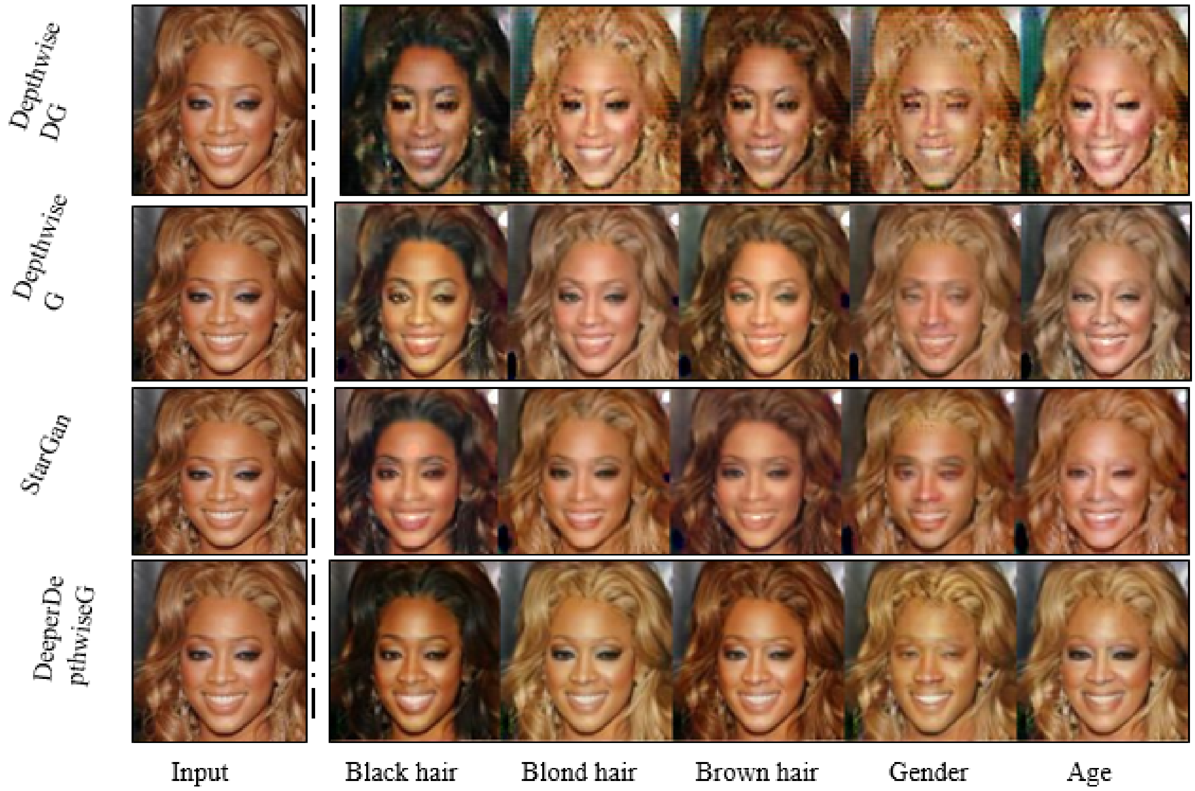


Figure 5.7: Applying StarGan and the three models for image-to-multi domain translation shows that using DSCs on the discriminator can have a negative effect on the results, as shown on top. Adding additional DSC layers to the generator results in realistic images, as shown at the bottom of the figure.

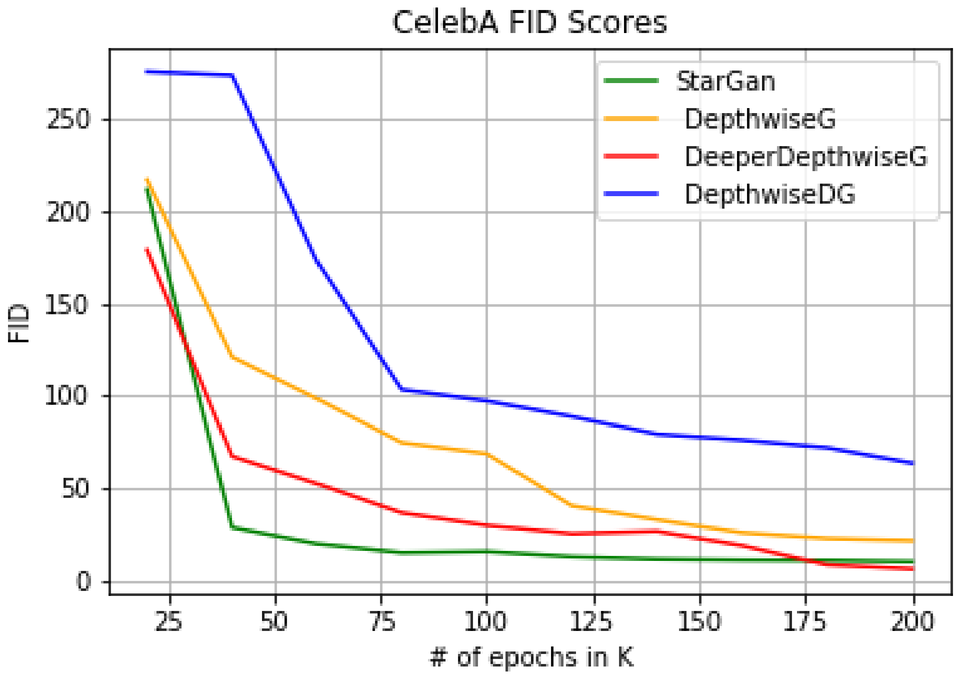


Figure 5.8: Scores for FID on four models that were experimented on and K represents thousands. DeeperDepthwiseG performs similarly to the baseline model. DepthwiseDG performed poorly against all models.

5.6.2 Results on RaFD and Stirling

The models were also on RaRD and Stirling datasets for facial expression transfer. Both the Stirling and RaFD datasets comprise similar facial expressions, captured in different environments. Results illustrated in figure 5.11 show that the DeeperDepthwiseG model generates high-quality images when compared to the original StarGan. The FID scores in figures 5.9 and 5.10 show that the difference between generated and training image distributions is similar for all models. Qualitatively, images produced by the model with fewer parameters have blurry regions, and it appears that the model has limited learning capabilities due to the reduced parameter set. However, the model with more parameters generates better samples, highlighting the importance of model capacity further.

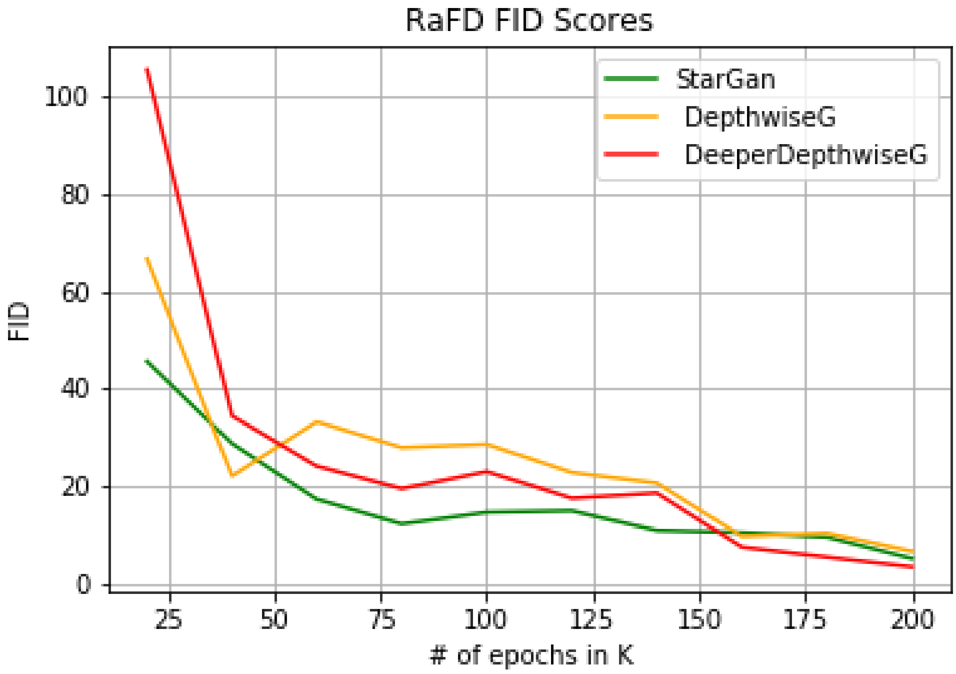


Figure 5.9: FID scores computed for RaFD dataset. DeeperDepthwiseG obtains similar performance to StarGan, and all models exhibit similar convergence after 200 epochs.

The results show that DSCs are more effective when only used on the generator, that they reduce the number of parameters, which speeds up training time, but that deeper models are required to obtain similar performance to standard convolutions.

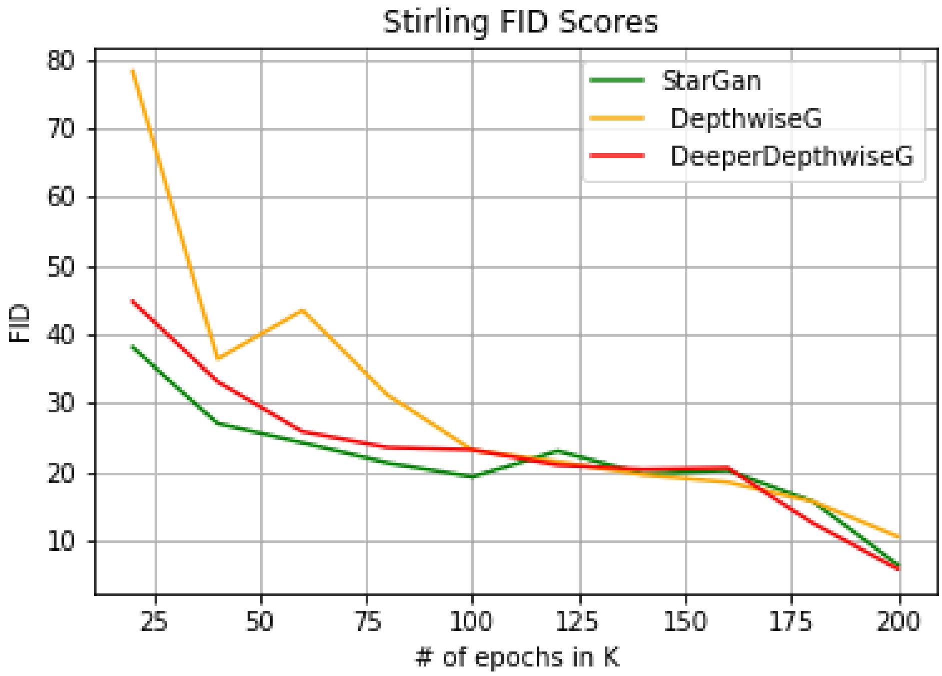


Figure 5.10: The FID scores for the Stirling dataset exhibit similar performance to that obtained on other sets.

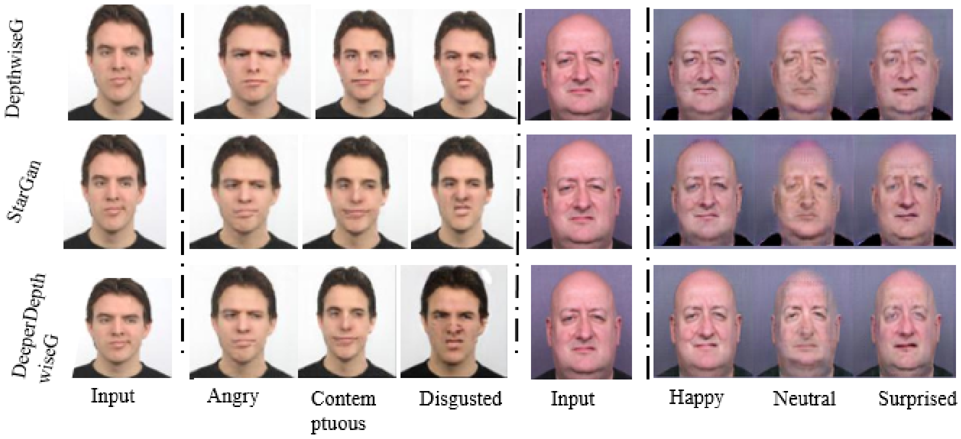


Figure 5.11: Samples generated show that generative models with additional capacity can perform well. The RaFD and Stirling datasets are similar but captured under different lighting conditions.

5.7 Conclusions

This chapter has introduced GANs, which can be used for data augmentation, and to potentially address the intersectional accuracy challenges identified previously. Unfortunately, training GANs is extremely slow. In order to address this, the replacement of standard convolutions was introduced by DSCs. These convolutions can reduce the number of training parameters, which speeds up the training process. However, this technique also has its downfalls, which affect the output results. This was observed when using the DSCs in both the generator and the discriminator, as generated images were blurry. Results presented in this chapter showed that although the GANs performance depends on the depth of the network, careful implementation of DSC can be used to speed up training and to maintain the model's performance.

The generation of a representative dataset for South African contexts could be made possible using GANs in many applications, but this research will only focus on the drowsiness problem. In chapter 6, GANs are applied to the driver drowsiness detection dataset in order to produce a more representative dataset for the South African context. PCA is used as a way to extract features from the data in support of re-sampling from GAN generated images for re-training the model. This process is a form of boosting, where model performance is improved in regions where it fails.

6 | GAN-based bias correction

In the previous chapter, the GAN architecture was introduced as a means to generate synthetic data. GAN training is particularly slow, but it showed that replacing the standard convolutions in the generator with DSCs can improve training speed. The resulting images are very realistic, which shows the potential for GAN to be used for generating datasets that are representative of South African contexts, which comprise a very wide range of races and ethnicities.

This chapter introduces a framework that can be used to reduce bias in training data by generating more realistic images with different facial attributes where the model is failing and for specific ethnical groups. The framework is composed of four primary components. Firstly, a GAN architecture produces synthetic images of individuals with facial attributes that can be used when retraining the network. The second component is the CNN architecture that predicts the state of the driver, while the third component is a PBV that highlights regions where the model is performing well and where it is failing to generalise. Lastly, the sampler targets images where the model is not performing well and searches through the synthetic images to find more images similar to those, which are used for model retraining. This idea is based on the boosting technique where a weak classifier is strengthened by re-weighting failing models. Results show that this framework does reduce bias in the training dataset and improves the performance of driver drowsiness detection algorithms.

A number of boosting approaches have been used in systems for detecting driver drowsiness, typically to improve face detection classifiers. Putta et al. used the Adaptive Boosting (AdaBoost) algorithm as part of the Viola-Jones face detection system for a real-time drowsiness detection system [131]. The AdaBoost algorithm was used for selecting important features (eyes and mouth regions) that can be used for training. In addition, Kong et al. in 2015 proposed a system that aims to improve strategies to detect fatigue in drivers using an improved AdaBoost algorithm [85]. Here, the authors used the AdaBoost algorithm for training and face regions were detected using Haar feature extraction. These faces were then forwarded to PERCLOS for analysis of drowsiness. Li et al. proposed a system that detects driver drowsiness using eye state features extracted using a Haar classifier and AdaBoost [94]. AdaBoost was used to train the cascade and to produce a strong classifier based on the Haar features. Furthermore, Bharambe and Mahajan have also used the AdaBoost algorithm and Haar-like features to detect the driver drowsiness [15].

Unfortunately, these systems mostly focus on improving algorithmic performance, while this work focuses on investigating and correcting dataset sampling bias. The unavailability of representative datasets makes networks generalise poorly on representative test data. This chapter shows that the use of the DrowsyGAN (DGAN) framework can boost the performance of CNNs in African contexts.

6.1 Boosting

In machine learning, boosting is a technique where a weak classifier is improved to be a stronger classifier by using a set of rules. The goal of boosting is to improve a weak classifier by treating it as a black box with a set of rules containing penalties for weak classifiers. There are different types of boosting algorithms, which include AdaBoost, Gradient Tree Boosting (GBoost), and A Scalable Tree Boosting System (XGBoost). These are detailed below.

AdaBoost – is a machine learning approach that was introduced by Freund and Schapire in 1995 [42]. The main idea of the AdaBoost algorithm is to maintain a set of weights over the training set. This weight distribution for the training set is firstly given the same value, but on each iteration, the weights of incorrectly classified training points are increased so that the weak learner is forced to focus on the harder tasks in the training set. The weak classifier is trained using a distribution D_t and is given a weak hypothesis h_t containing an error value. Once the weak hypothesis is given, the algorithm chooses a parameter α_t. The main goal of α_t is to measure the importance that is assigned to h_t. The hypothesis is defined as follows :

$$H(x) = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big), \tag{6.1}$$

GBoost – is one of the most powerful machine learning techniques for building predictive models. This technique was introduced by Friedman in 1999 where the outputs of weak classifiers are combined to produce stronger classifiers with the help of decision trees of fixed sizes [43]. The GBoost algorithm has two primary parameters which are the number of iterations M and the learning rate.

XGBoost – is a scalable tree boosting technique that was introduced by Chen and Guestrin in 2016 and is widely used by data scientists [22]. It uses a more regularised model formalisation for the prevention of overfitting. It is a portable and accurate technique for large-scale tree boosting problems.

6.2 Data Augmentation

Data augmentation is a technique used to increase the size of a training dataset and reduce the chances of overfitting by a network. In computer vision, image datasets are often artificially expanded by the inclusion of modified versions of the original training datasets. The most common way to perform data augmentation is by random parameterised transformations including some of the popular techniques which are described below.

Rotation – a technique that transforms image dimensions by rotating about the image centre. The image dimensions do not remain the same after performing this operation.

Flipping – the original images are flipped horizontally and vertically and this can create two more images per image. Vertical flipping is also the same as 180-degree rotation.

Scaling – this technique is performed on an image by zooming in and out. This method can result in the los of information in the image.

Cropping – a technique that randomly crops the image in different regions. As scaling, this can result in lose of the information in the image.

Unfortunately, as shown in Chapter 4, in the case of driver drowsiness detection, most available datasets are unrepresentative. This makes it hard to test on an African dataset as it contains individuals with facial features that are not included in the training dataset. Applying common data augmentation to the available dataset improves the results by a small amount but fails to address this challenge.

There are advanced data augmentation techniques that can help in the real world scenarios, which may contain a variety of conditions that can not be solved by using standard techniques.

For example, in the case of driver drowsiness detection, there are various conditions to look at such as the complexion of the driver, the time of day, illumination and obstacles. Conditional GANs are used as an advanced way to augment data when there is a variation of data and some knowledge of the specific data to be generated.

There is much work which has been done on using GAN architectures as a data augmentation technique. Rahul Gupta used conditional GANs for a sentiment classification task [54]. He obtained 1.6% and 1.7% improvement over a baseline model which was only trained on real data. The conditional GAN was trained using different strategies including pre-training and noise injection on the training data. Mok and Chung proposed an automatic data augmentation approach that enables machine learning methods to learn from the available annotated samples efficiently [110]. Their architecture consists of a coarse-to-fine generator which captures the manifold of the training sets. Their proposed method was used on Magnetic Resonance Imaging (MRI) images and achieved improvements of about 3.5% over the traditional augmentation approaches that they were compared against. In addition, Wu et al. used multi-scale class conditional GAN to perform contextual in-filling which is used to synthesize lesions onto healthy screening mammograms [188]. For experimentation, three classifiers were compared using Area under the ROC Curve (AUC) method of 0.896 which outperformed the baseline model where the AUC was 0.014. Antoniou et al. used a conditional GAN to augment data to another domain [7]. They named their architecture as DataAugmentation Generative Adversarial Networks (DAGAN), and evaluated its performance on low-data availablity tasks using standard stochastic gradient descent neural network training. It is clear that GANs can be used as a substitute for traditional augmentation techniques, particularly where the data is more sophisticated.

6.3 Proposed Framework

In this section, a novel framework that corrects CNNs trained for prediction using GANs for targeted data augmentation based on a PBV technique that groups faces with similar attributes and highlights where the model is failing is introduced. A sampling method selects faces where the model is not performing well, which are then used to fine-tune the CNNs. Each of these components is discussed below.

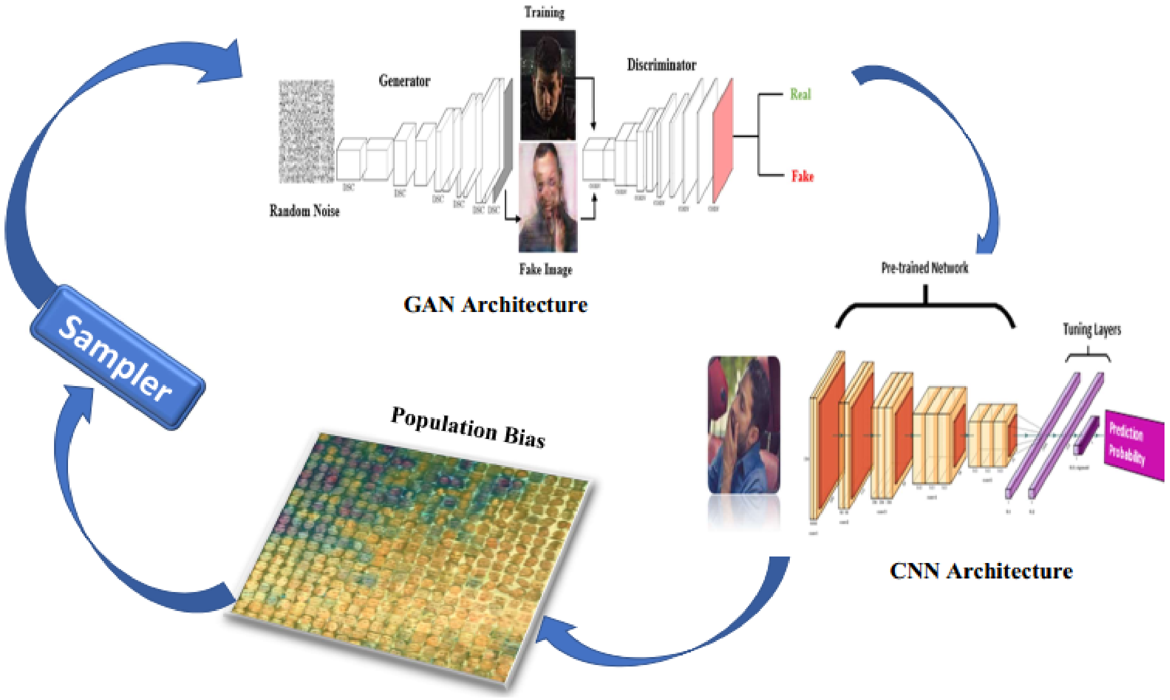


Figure 6.1: The proposed framework with all the components. The first step is generating synthetic data using GAN architecture, which is followed by training the CNN to detect driver state. A population bias visualisation is applied to the testing dataset and highlights where the model is failing to generalise. Where the model is not generalising, those images are then sampled and are used to find similar images from the GAN generated images to continue training the model.

6.3.1 GAN Architecture

The GAN generates realistic images of individuals (drowsy and alert) in these population groups, which are used for retraining the ResNet model used for drowsiness detection with new parameters i.e learning rate and epoch sizes to reduce overfitting and improve the detection accuracy.

The generative network from Choi et al. [25] was adapted for the implementation of the GAN architecture because it has shown impressive results for generating realistic synthetic images in different domains. The architecture is a conditional GAN that is conditioned to translate facial attributes between alert and drowsy across multiple ethnicity groups as shown in figure 6.2. This translation of attributes helps in improving the detection model by supplying images with appropriate features where the drowsiness detector fails to generalise. The generator was modified by replacing the standard convolutions by DSCs [117]. The benefit of this is to have fewer trainable parameters while retaining the performance of the network. Furthermore, the generator consists of a stride size of two for down-sampling and 11 DSCs. Instance normalisation was used [176] instead of batch normalisation for the generator for all layers except the last

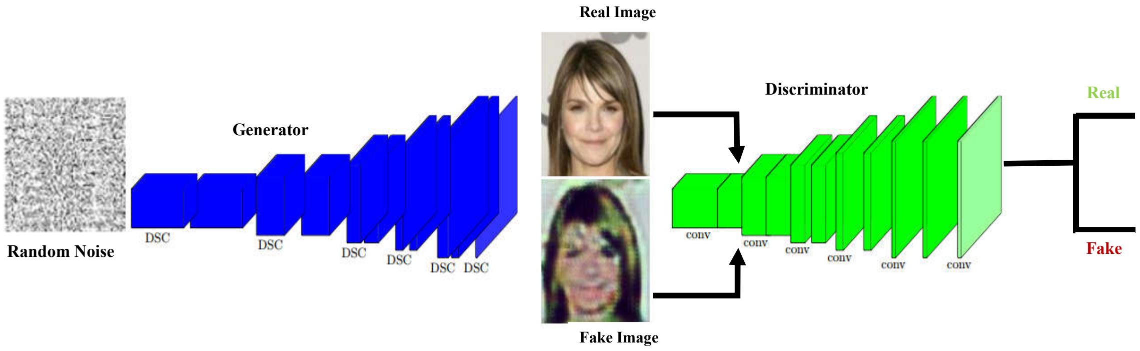


FIGURE 6.2: The GAN architecture that translates images from being alert to sleepy and vice versa.

output layer. For the discriminator network, standard convolutions were retained because the discriminator acts as a classifier and requires greater capacity in order to distinguish between real and fake images. In addition, PatchGANs [93] were adapted for the discriminator network because they make use of a fixed-size patch discriminator that is easily applied to 256 x 256 images.

6.3.2 CNN Architecture

The second component of DGAN framework is the CNN architecture which predicts the state of the driver. A fine-tuned Resnet model is used for classifying whether the driver is drowsy or not. This architecture is chosen because of its depth and the dataset that is pre-trained on. A pre-trained ResNet model which contains 50 layers was used and was originally trained on Canadian Institute For Advanced Research-10 (CIFAR-10), ILSVRC and Common Objects in Context (COCO 2) [57]. The pre-trained model was fine-tuned on the last layers where the prediction function was modified to sigmoid. The ResNet model has about 23 million parameters and after adding the fine-tuning layers there were approximately 25 million parameters where the initial 23 million parameters were frozen and only 2 million were trained.

6.3.3 Population Bias Visualisation

PBV is used to identify the regions where the model is failing to generalise well [156]. This is done by projecting images into a 2-dimensional grid using the PCA technique which locates similar images of people in the same state (alert or drowsy) and with similar complexions. The test dataset is transformed into an orthogonal subspace where axes (Principal Component) align with the directions of maximum variance in the data. For performing the PCA, SVD was used by accepting **X** matrix of images. The details of this process are provided in chapter 4, section 4.3.

6.3.4 Sampler

This step is performed by randomly selecting images according to the error in prediction, termed the failure probability here. The failure probability

$$C_i = |y_i - y_t| \tag{6.2}$$

is calculated by determining the difference between the CNN sigmoid prediction output, y_i and the true label y_t (a binary label encoding drowsy or alert).

This is normalised by the total probability of failure over all N images in the dataset

$$\hat{C}_j = \frac{(1 - C_j)}{\sum_{i=1}^{N}(1 - C_j)} \tag{6.3}$$

The random selection is performed by categorically sampling images without replacement using the weights above. This ensures that images with greater probability of failure are more likely to be sampled and ensures that there are no duplicates of the selected images. For each selected image in the validation set with a probability of failure, similar images in the GAN generated dataset are then selected by finding close matching images in the PCA space. The selected images are then used to continue training the ResNet model to increase classification performance.

6.4 Experiments and Results

This section describes parameters that are used in DGAN framework, starting from training the GAN up to the sampling method. Results are also described in this section.

6.4.1 Training

For the training of the GAN architecture, the Adam optimiser [83] was used with $\beta_1 = 0.3$ and $\beta_2 = 0.6$. A batch size of 32 was also used to help in stabilising the training process. These parameters were chosen because they are widely used. A horizontal flipping data augmentation was applied with a probability of 0.5. More detailed information about the network architecture is provided in tables 6.1 and 6.2. For the discriminator network in table 6.2, the Leaky-ReLU activation function is used, but the generator network uses the ReLU activation function. There is also notation used to represent some of the variables in the GANs including the number of domains (n_d), number of domain labels (n_c), number of of output channels (N), kernel size (K), stride size (S), padding size (P) and instance normalisation. To train the model, a learning rate of 0.0001 for 100 epochs and linearly decayed the learning rate over the next 100 epochs was used. This strategy was to compensate for the fact that the training data was limited. Images were cropped to 256 x 256 and three types of losses were applied. The first loss was for domain classification loss, weighted by 1, the second loss was a reconstruction loss with a default weight value of 10 and lastly a gradient penalty loss was used with a default weight value of 10. All the experiments were carried out on a single Nvidia Tesla K20c GPU and the training took 12 hours.

The ResNet pre-trained network was fine-tuned and learning rates were constantly adjusted as the training continued. Initially, the models were trained on three different datasets (NTHU, DROZY, and CEW) as discussed in section 4.4 and were compared with DGAN framework, as illustrated in the results sections 6.4.3 and 6.4.5.

All images were cropped to 150 x 150 size and standard augmentation techniques (rotation range=60, zoom range=0.6, and horizontal flipping) were performed. The pre-trained ResNet model used with a learning rate of 0.0001, which was then modified for the rest of the experiments from 1e^{-3} to 1e^{-6}, this was performed for each iteration on the framework. The early stopping training strategy was also used to prevent overfitting of the model. Furthermore, the Adam optimiser was used with both $\beta_1, \beta_2 = 0.99$, and a learning rate decay of 0 over each update.

Table 6.1: Generator network architecture information

Part	Input Shape	Output Shape	Layer Information
Down-sampling	$(h, w, 3 + n_c)$	$(h, w, 64)$	DSC-(N64,K7x7,S1,P3),IN,ReLU
	$(h, w, 64)$	$(\frac{h}{2}, \frac{w}{2}, 128)$	DSC-(N128,K4x4,S2,P1),IN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128)$	$(\frac{h}{4}, \frac{w}{4}, 256)$	DSC-(N256,K4x4,S2,P1),IN,ReLU
Bottleneck	$(\frac{h}{4}, \frac{w}{4}, 256)$	$(\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block:DSC-(N256,K3x3,S1,P1),IN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256)$	$(\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block:DSC-(N256,K3x3,S1,P1),IN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256)$	$(\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block:DSC-(N256,K3x3,S1,P1),IN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256)$	$(\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block:DSC-(N256,K3x3,S1,P1),IN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256)$	$(\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block:DSC-(N256,K3x3,S1,P1),IN,ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256)$	$(\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block:DSC-(N256,K3x3,S1,P1),IN,ReLU
Up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256)$	$(\frac{h}{2}, \frac{w}{2}, 128)$	DEDSC-(N128,K4x4,S2,P1),IN,ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128)$	$(h, w, 64)$	DEDSC-(N64,K4x4,S2,P1),IN,ReLU
	$(h, w, 64)$	$(h, w, 3)$	DSC-(N3,K7x7,S1,P3),Tanh

Table 6.2: Discriminator network architecture information

Part	Input Shape	Output Shape	Layer Information
Input Layer	$(h, w, 3)$	$(\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64,K4x4,S2,P1),Leaky-ReLU
Hidden Layer	$(\frac{h}{2}, \frac{w}{2}, 64)$	$(\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128,K4x4,S2,P1),Leaky-ReLU
	$(\frac{h}{4}, \frac{w}{4}, 128)$	$(\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256,K4x4,S2,P1),Leaky-ReLU
	$(\frac{h}{8}, \frac{w}{8}, 256)$	$(\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512,K4x4,S2,P1),Leaky-ReLU
	$(\frac{h}{16}, \frac{w}{16}, 512)$	$(\frac{h}{32}, \frac{w}{32}, 1024)$	CONV-(N1024,K4x4,S2,P1),Leaky-ReLU
	$(\frac{h}{32}, \frac{w}{32}, 1024)$	$(\frac{h}{64}, \frac{w}{64}, 2048)$	CONV-(N2048,K4x4,S2,P1),Leaky-ReLU
Output Layer (D_{src})	$(\frac{h}{64}, \frac{w}{64}, 2048)$	$(\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(N1,K3x3,S1,P1)
Output Layer(D_{src})	$(\frac{h}{64}, \frac{w}{64}, 2048)$	(1,1,n_d)	CONV-(N(n_d),K$\frac{h}{64}$x$\frac{w}{64}$,S1,P0)

6.4.2 Results

The first comparison was with the results of the DGAN framework to those obtained by training on the publicly available dataset using the pre-trained ResNet models. All the parameters were kept the same for all the first experiments, thereafter the learning rate and training epochs were modified to prevent overfitting.

6.4.3 GAN Augmentation Results

Figure 6.3 shows the complexion attribute transfer results on the testing dataset. The testing dataset is comprised of a variety of facial attributes of different ethnicities. This helps to balance all the testing datasets and to produce a more synthetic dataset that covers most population groups. The observation was that it was easy to transfer from light complexion to dark complexion and achieved high-quality images. This was because of the presentation of these attributes in the light-skinned ethnicities as compared with the dark-skinned ethnicities. On the other hand,

Table 6.3: Learning rate - accuracy tradeoff

Learning rates with accuracies	
Learning Rate	Accuracies
1e^{-3}	89.70%
1e^{-4}	96.30%
1e^{-5}	96.91%
1e^{-6}	98.01%

when transferring from dark complexion to light complexion, the images were blurry and the reason was the balance of the races in the dataset. The orientation, light intensity and poses of dark skinned people were limited and generating reasonable results was difficult. To overcome this problem, additional images of the face of the driver in good lighting conditions are required. The scope of this thesis is limited to daylight conditions and do not consider night conditions as they would pose other problems and need specialised cameras such as infra-red cameras. The training time of the GAN model is quite long because there are two networks to be trained and information is shared between them, but by using DSC on the generator network to ease the computational resources and reduce the number of parameters in the generator network, were a reasonable time training time was obtained and retain a reasonable performance quality. In this case, the performance of the network was kept as high as possible to produce images that could be used for training. Using blurry or poor quality images on the retraining of the network could have some influence on the prediction scores. However, these images boosted the performance of the driver drowsiness detection task.

6.4.4 Learning rate Results

Table 6.3. shows the influence of different learning rates on the performance of the model. A learning rate of 1e^{-3} was too high and limited model learning. The best performance was observed from 1e^{-4} to 1e^{-6}. The early stopping strategy was also applied to prevent overfitting which can improve the generalisation on the model. At first, the models were trained with the same number of epochs. It was observed that using the same number of epochs on different learning rates showed traits of overfitting. Therefore, early stopping was used when there was no change in the accuracy.

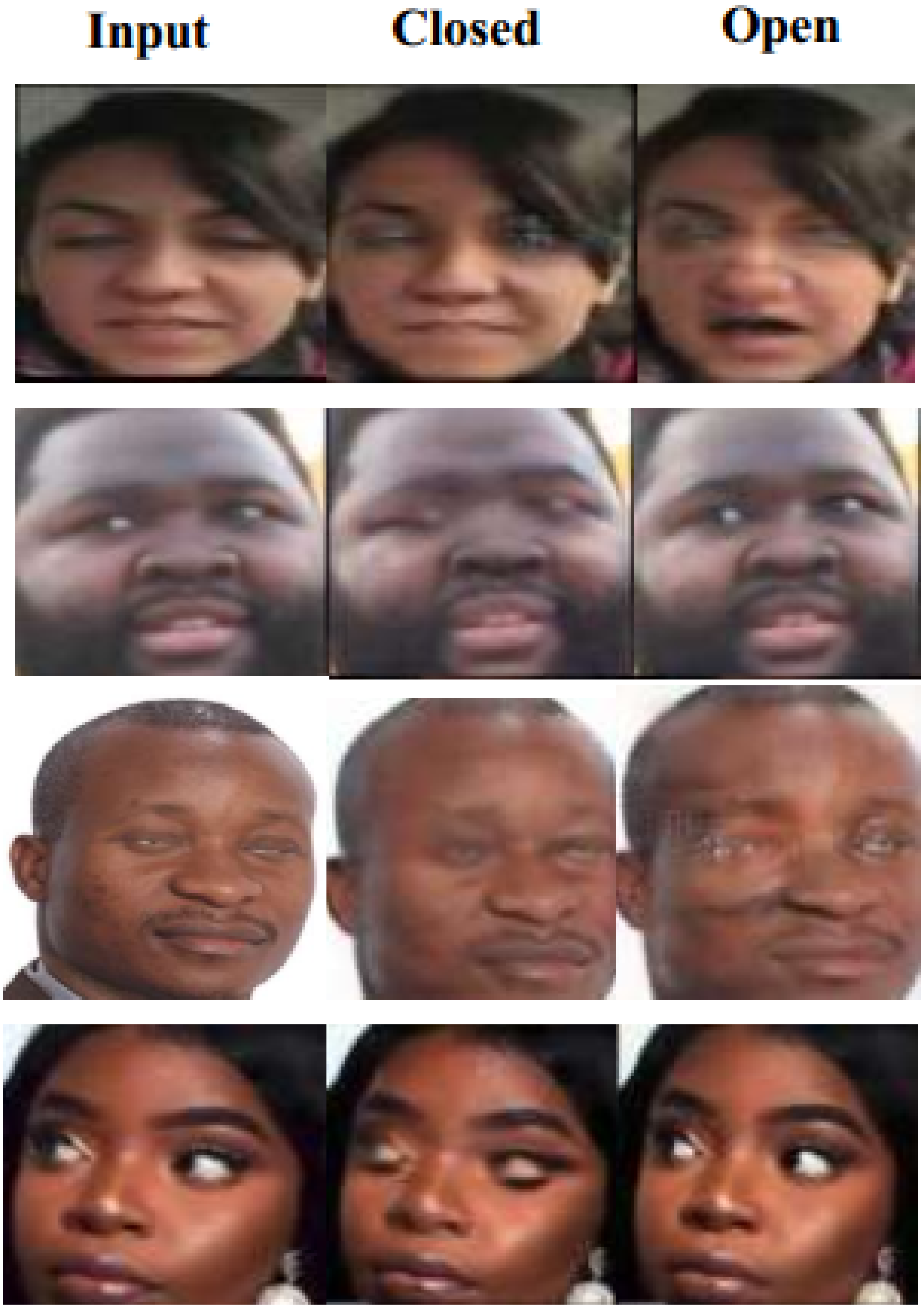


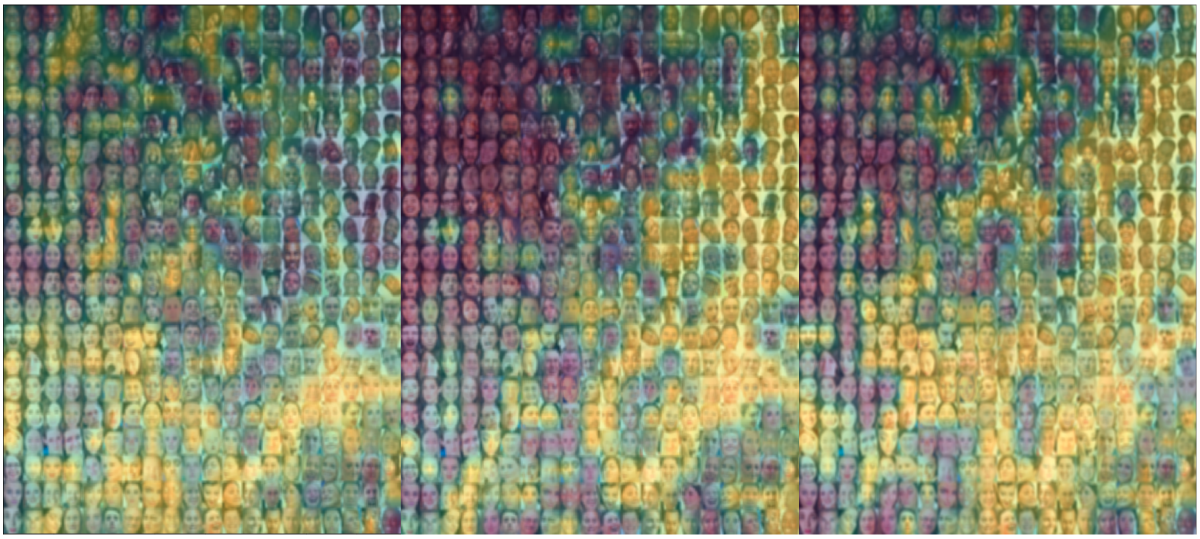
Figure 6.3: The GAN generated images by domain transfer across most ethnicity groups. The generated images show people from different ethnicity groups with eyes open and closed.

6.4.5 Population Bias Visualisation Results

It is demonstrated that the DGAN framework can improve the driver drowsiness detection task by providing synthetic generated images with different attributes (e.g. alert or sleepy) using GAN architecture. This was followed by using the results of thePBV technique to highlight faces where the model failed to generalise. The re-sampling strategy that selects images where the model is failing was performed and PCA was used to extract features and find images with similar attributes in the synthetic data and to retrain the model. In these experiments, the framework was compared to models that are trained on CEW, NTHU, and DROZY datasets.

Figure 6.4 shows the PBV on the three datasets. The first image from figure 6.4 is a model trained on CEW dataset, which provided the most promising results, when compared against the models trained using the other two datasets, with an accuracy range of 67% to 78%. With the pre-trained model on DROZY dataset, most of the dark faces were not predicted correctly and the accuracy rate was between 50.4% to 59.32% which is essentially the same as guessing of whether the driver is drowsy or alert. This makes it difficult to rely on such a model because it would raise false alarms. Furthermore, the dataset that was trained on was solely comprised of light-skinned faces, but it also does not generalise well on light-skinned faces because the images were captured under a limited set of conditions. In addition, the number of people that the model was exposed to was limited, which leads to such high failure percentages. The third image in figure 6.4 shows the results obtained using the model that was pre-trained on the NTHU dataset which performed better than the model trained on the DROZY. Although both datasets were captured in similar environments, the NTHU model was trained on more faces and reached about 55.87% to 65.97% accuracy. In all these models, it is clear that they failed to generalise to darker complexions.

Figure 6.5 shows the results of the DGAN framework which demonstrates the improvement of the driver drowsiness detection performance using the proposed framework. Given enough data, the model was able to generalise and the process of achieving these results is recursive. The left image on figure 6.5 shows the results after a single correction iteration using the framework. This model still performed worse on dark skinned individuals, but had higher accuracy than the models trained on the publicly available dataset. In addition, the model was already generalising on more of the dark-skinned faces. After the seventh iteration, the model reached the highest accuracy of 98.89%, which indicates that with the re-sampling technique can achieve better results when incorporating it with GAN generated data.



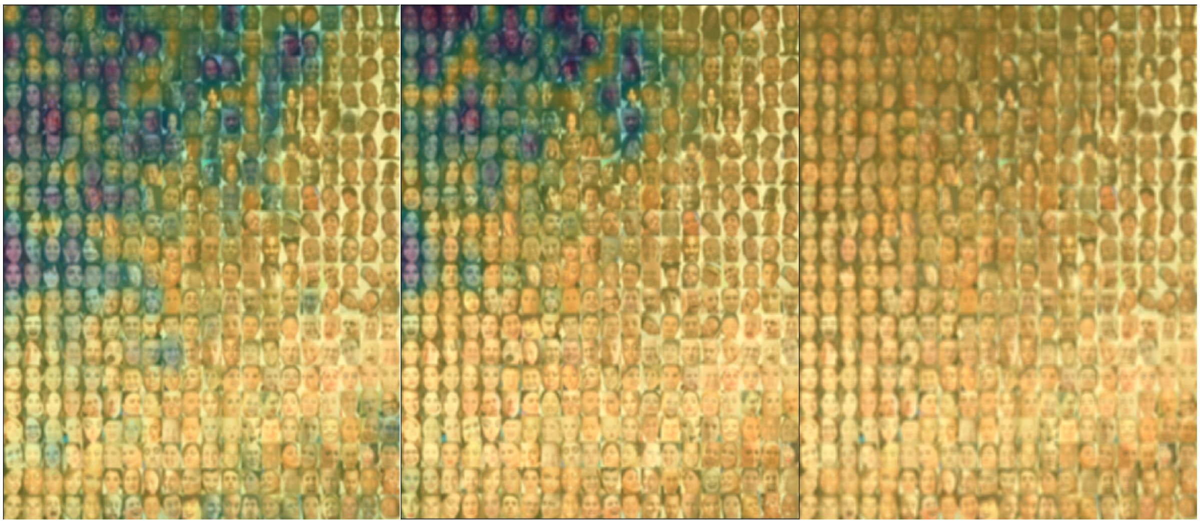
(A) ResNet-CEW (B) ResNet-DROZY (C) ResNet-NTHU-DROZY

FIGURE 6.4: The figure shows images produced using the PBV technique. All the trained models appear to be failing on the population groups on the upper part of the image. The yellow shaded parts indicate where the model performs well, while failures are indicated by the purple shaded parts, which appear mostly on the upper part of the images. The green shaded parts show that the model is also performing well, but with lower probability (0.50 to 0.65).

TABLE 6.4: Model classification accuracy

Detection Accuracy				
Iteration	Original Data	GAN Augmentation (validation)	GAN Augmentation with Sampling (validation)	GAN Augmentation with Sampling (test)
1	56.40%	74.70%	87.65%	85.28%
2	55.70%	76.86%	89.87%	85.97%
3	57.08%	77.91%	89.43%	86.42%
4	52.00%	77.01%	90.04%	87.02%
5	59.76%	78.80%	93.66%	88.83%
6	60.92%	80.51%	94.54%	89.05%
7	60.54%	80.93%	96.75%	91.62%

The models were further compared with baseline experimental results, where random sampling from the augmentation data was performed. It was noted that when the sampling technique was used from the GAN generated images and did not use targeted sampling, the model improvement was limited. Targeted sampling is clearly more effective. Table 6.4 shows the accuracy results from the three experiments which indicate how accuracy is improved with additional iterations of targeted sampling.



(A) First visualisation (B) Second visualisation (C) Seventh visualisation

FIGURE 6.5: The figure shows the progressive work of DGAN framework. The models were fine-tuned and retrained with the images where they failed to generalise. As the models were fine-tuned and changing the learning rate, there was an improvement and the models reached higher probabilities (0.70 to 0.99). At the seventh cycle, the best performance of the model was reached.

6.5 Conclusion

In this chapter, a novel framework that can be used to boost the performance of driver drowsiness detection models by reducing bias in the training dataset was introduced. Here, a GAN produces realistic images that are used when retraining a ResNet model on synthetic data generated based on failure cases in validation data.

Importantly, the proposed approach does not rely on any meta-data or assumptions about the race or ethnicity of individuals in the datasets, which is a commonly used approach to determine algorithmic fairness or bias. Requiring this knowledge is potentially problematic as it tends to rely on subjective and controversial racial classifications.

Furthermore, this chapter has shown that bias in datasets can be addressed by using targeted sampling and GANs. However, this process still requires that some training data be available for various population groups, and does not eliminate the need for good, representative datasets. Rather, the proposed approach is intended to remedy more subtle bias introduced by imbalanced datasets, where images of people from a particular group may be more numerous than that of another group.

7 | Conclusions and future work

7.1 Conclusion

This thesis has focused on highlighting the effects of bias in the training dataset for the task of driver drowsiness detection task. This was followed by the introduction of a process correcting the bias using a PBV framework. This framework's components consist of a GAN architecture for creating synthetic data for a variety of facial attributes and a wide range of races, which is followed by the detection of drowsiness using a CNN model. PBV highlights those regions where the model is failing and the sampler selects those faces and searches for similar faces from the synthetic generated data to retrain the network. This work has shown that publicly available datasets for drowsiness detection do not cover all population groups and different facial attributes, this problem was highlighted when CNNs were trained on a publicly available dataset and tested in the African context. Bias in datasets is a serious problem, which was addressed here. The proposed technique to increase the training dataset through the addition of images with different facial attributes across a wide range of races and ethnicities resulted in a substantial increase in the CNN performance. The GAN architecture was modified and experiments showed that replacing the standard convolutions only on the GANs generator network with DSCs can improve the training time of GANs, while retaining the performance of the network.

The fundamentals around the drowsiness concept and how this can affect driving ability was provided as the foundation to understand which facial attributes are most important. In addition, methods of measuring drowsiness using different approaches, including machine learning techniques were discussed. A meta-analysis was conducted to investigate which machine learning method can be used for better detection of drowsiness. The limitation of the availability of public datasets was also highlighted.

Machine learning, which drives most drowsiness detection systems, fundamentals of machine learning and the building blocks of CNNs, was introduced, along with a discussion of the training methods that help in improving the accuracy of the network.

Finally, a DGAN framework was introduced that boosts the performance of the task of driver drowsiness detection by eliminating bias in the training dataset. This framework corrects the CNN trained for prediction using augmented GAN based images. Here, PBV is used to monitor performance and a sampler component selects GAN images for model tuning. This chapter further highlighted that, with standard augmentation techniques, a model can improve to a certain degree of performance. Using the GAN generated images boosted the CNN performance. Targeted sampling was significantly more effective than training with all GAN images.

7.2 Limitations

The efficiency of CNNs in the task of detecting of driver drowsiness depends solely on the quality and the diversity of the training dataset. There are a number of limitations in this thesis which need to be addressed. The major limitation is that the training was based on images and not on videos and that this application was tested offline, not in real-time. The training images were captured in daylight conditions and nighttime conditions were not tested because this required additional sophisticated specialised equipment such as infrared cameras to capture images.

More importantly, the bias remediation scheme proposed in this work depended strongly on the need for a representative dataset for data augmentation. Furthermore, although the proposed approach can improve on the fairness of a biased classifier the proposed approach provides no fairness guarantees and more subtle biases may still be present in machine learning models.

7.3 Future work

Bias in a training dataset is one of the most crucial problems encountered when training a CNN model because it prohibits the development of more generalised models. This research has focused on behavioural methods to measure the level of drowsiness which were used together with CNNs to predict drowsiness. The challenge of limited training datasets for driver drowsiness was identified and a framework for bias remediation was proposed. The proposed framework showed promising results in eliminating bias in the training dataset.

It should be noted that the introduced framework is not limited to the drowsiness detection task, but can be applied to any task that has a limited dataset and which can be demonstrated in different domains.

Although promising results were obtained, only one strategy was used to measure the level of drowsiness. Future work on building a hybrid system that combines other non-invasive methods can be used to allow for more informed decision making. There is a need for exploring the design of a device to sense levels of drivers drowsiness, which can be placed on the car seat.

Bibliography

[1] M. Ehsan Abbasnejad et al. "Bayesian Conditional Generative Adverserial Networks". In: *CoRR* abs/1706.05477 (2017). arXiv: 1706.05477. URL: http://arxiv.org/abs/1706.05477.

[2] Shabnam Abtahi, Behnoosh Hariri, and Shervin Shirmohammadi. "Driver drowsiness monitoring based on yawning detection". In: *2011 IEEE International Instrumentation and Measurement Technology Conference* (2011), pp. 1–4.

[3] Shabnam Abtahi et al. "YawDD". In: *Proceedings of the 5th ACM Multimedia Systems Conference on - MMSys '14*. New York, New York, USA: ACM Press, 2014, pp. 24–28. ISBN: 9781450327053. DOI: 10.1145/2557642.2563678. URL: http://dl.acm.org/citation.cfm?doid=2557642.2563678.

[4] Paul Russel B Agustin et al. "MATLAB-Based Drowsiness Detection System Using an Array of Sensors and Fuzzy Logic". In: *TENCON 2014 - 2014 IEEE Region 10 Conference* (2014). DOI: 10.1109/TENCON.2014.7022353.

[5] Torbjörn Åkerstedt et al. "The subjective meaning of good sleep, an intraindividual approach using the Karolinska Sleep Diary". In: *Perceptual and motor skills* 79.1 (1994), pp. 287–296.

[6] Ghassan Jasim AL-Anizy, Md Jan Nordin, and Mohammed M Razooq. "Automatic Driver Drowsiness Detection Using Harr Algorithm and Support Vector Machine Techniques". In: *Asian Juornal of Applied Sciences* (2015). DOI: 10.3923/ajaps.2015. URL: http://docsdrive.com/pdfs/knowledgia/ajaps/0000/68098-68098.pdf.

[7] Antreas Antoniou, Amos Storkey, and Harrison Edwards. "Augmenting image classifiers using data augmentation generative adversarial networks". In: *International Conference on Artificial Neural Networks* (2018), pp. 594–603.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)* 70 (2017), pp. 214–223. arXiv: arXiv:1701.07875v3.

[9] Arrivealive. *Decade of Action for Road Safety in SA: Mid- term Report.* 2017. URL: https://www.arrivealive.co.za/United-Nations-Decade-of-Action-for-Road-Safety-2011-2020 (visited on 05/15/2017).

[10] Audi. *Audi A3 Sedan.* 2014. URL: https://www.audi.co.za/za/web/en/models/a3/a3-sportback.html (visited on 06/20/2017).

[11] Am Bagci and R Ansari. "Eye tracking using Markov models". In: *Recognition, 2004. ICPR* 3 (2004), pp. 2–5. ISSN: 10514651. DOI: 10.1109/ICPR.2004.1334654.

[12] Leonard E Baum and Ted Petrie. "Statistical inference for probabilistic functions of finite state Markov chains". In: *The annals of mathematical statistics* 37.6 (1966), pp. 1554–1563.

[13] BCC. *Tesla in fatal California crash was on Autopilot.* 2018. URL: https://www.bbc.com/news/world-us-canada-43604440 (visited on 08/15/2018).

[14] Ibtissem Belakhdar et al. "Detecting driver drowsiness based on single electroencephalography channel". In: *13th International Multi-Conference on Systems, Signals and Devices, SSD 2016* (2016), pp. 16–21. DOI: 10.1109/SSD.2016.7473671.

[15] Snehal S Bharambe and PM Mahajan. "Implementation of real time driver drowsiness detection system". In: *International Journal of Science and Research* 4 (2015), pp. 2202–2206.

[16] Pankti P Bhatt and JA Trivedi. "Various methods for driver drowsiness detection: an overview". In: *Int J Comput Sci Eng (IJCSE)* 9.03 (2017), pp. 70–74.

[17] *BMW 4 Series Coupe : Driver Assistance.* URL: https://www.bmw.co.za/en/all-models/4-series/coupe/2013/driving-assistance-systems.html (visited on 06/06/2017).

[18] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory.* ACM. 1992, pp. 144–152.

[19] Joy Adowaa Buolamwini. *Gender shades: intersectional phenotypic and demographic evaluation of face datasets and gender classifiers.* 2017.

[20] Christian Cajochen, Sarah L Chellappa, and Christina Schmidt. “Circadian and light effects on human sleepiness–alertness”. In: *Sleepiness and human impact assessment* (2014), pp. 9–22.

[21] Mary A Carskadon. “Guidelines for the multiple sleep latency test (MSLT): a standard measure of sleepiness”. In: *Sleep* 9.4 (1986), pp. 519–524.

[22] Tianqi Chen and Carlos Guestrin. “XGBoost : A Scalable Tree Boosting System”. In: (2016).

[23] Thurn Chia Chieh et al. “Development of vehicle driver drowsiness detection system using electrooculogram (EOG)”. In: *2005 1st International Conference on Computers, Communications and Signal Processing with Special Track on Biomedical Engineering, CCSP 2005* (2005), pp. 165–168. DOI: `10.1109/CCSP.2005.4977181`.

[24] In Ho Choi, Chan Hee Jeong, and Yong Guk Kim. “Tracking a driver’s face against extreme head poses and inference of drowsiness using a hidden Markov model”. In: *Applied Sciences* 6.5 (2016). ISSN: 14545101. DOI: `10.3390/app6050137`.

[25] Yunjey Choi et al. “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation”. In: *arXiv preprint* 1711 (2017).

[26] François Chollet. “Xception: Deep learning with depthwise separable convolutions”. In: *arXiv preprint* (2017), pp. 1610–02357.

[27] National Tsuing Hua University Computer Vision Lab. *Driver Drowsiness Detection Dataset*. 2017. URL: `http://cv.cs.nthu.edu.tw/php/callforpaper/datasets/DDD/` (visited on 09/24/2017).

[28] *Detection and Prevention : Drowsy Driving – Stay Alert, Arrive Alive*. URL: `http://drowsydriving.org/about/detection-and-prevention/` (visited on 06/22/2017).

[29] *Digital divide*. 2019. URL: `https://en.wikipedia.org/wiki/Digital_divide` (visited on 05/10/2019).

[30] David F Dinges and Richard Grace. “PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance”. In: *US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006* (1998).

[31] Jeffrey Donahue et al. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

[32] Charlie Drewes. "Electromyography: Recording electrical signals from human muscle". In: *Tested Studies for Laboratory Teaching. Association* (2000), pp. 248–270. URL: http://ableweb.org/volumes/vol-21/12-drewes.pdf.

[33] DriverAssistance. *BMW 4 Series : Driver Assistance*. 2017. URL: https://www.bmw.co.za/en/all-models/bmw-i/i3/2017/connectivity-driver-assistance.html (visited on 06/06/2017).

[34] *Drowsy Driving*. URL: http://sleepcenter.ucla.edu/drowsy-driving (visited on 06/22/2017).

[35] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *Journal of Machine Learning Research* 12 (2011), pp. 2121–2159. URL: http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf.

[36] Kartik Dwivedi, Kumar Biswaranjan, and Amit Sethi. "Drowsy driver detection using representation learning". In: *Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014* (2014), pp. 995–999. DOI: 10.1109/IAdCC.2014.6779459.

[37] Kartik Dwivedi, Kumar Biswaranjan, and Amit Sethi. "Drowsy driver detection using representation learning". In: *Advance Computing Conference (IACC), 2014 IEEE International*. IEEE. 2014, pp. 995–999.

[38] Engineeringnews. *Auto exports earned South Africa R165bn in 2017*. 2018. URL: http://www.engineeringnews.co.za/article/auto-exports-earned-south-africa-r165-billion-in-2017-2018-05-04 (visited on 11/05/2018).

[39] Eurocontrol. *Fatigue and Sleep Management*. 2018. URL: https://www.eurocontrol.int/sites/default/files/publication/files/sleep-mgnt-online-13032018.pdf.

[40] M. A. Figueiredo. "Adaptive Sparseness for Supervised Learning". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 25 (Sept. 2003), pp. 1150–1159. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2003.1227989. URL: doi.ieeecomputersociety.org/10.1109/TPAMI.2003.1227989.

[41] Nikhil .R Folane and R.M Autee. "EEG Based Brain Controlled Wheelchair for Physically Challenged People." In: *International Journal of Innovative Research in Computer and Communication Engineering* 4.6 (2016), pp. 2257–2263. ISSN: 2320-9798. DOI: 10.15680/IJIRCCE.2016..

[42] Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[43] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[44] Loning Fu and Cheng Chi Lee. *The circadian clock: pacemaker and tumour suppressor.* 2003. DOI: 10.1038/nrc1072. URL: https://doi.org/10.1038/nrc1072.

[45] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4 (Apr. 2019), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251. URL: https://doi.org/10.1007/BF00344251.

[46] Anjith George and Aurobinda Routray. “Real-time Eye Gaze Direction Classification Using Convolutional Neural Network”. In: *International Conference on Signal Processing and Communication* (2016), pp. 1–5. arXiv: 1605.05258. URL: http://arxiv.org/abs/1605.05258.

[47] Tzamalouka Georgia et al. “Lifestyle Patterns As Predictors of Drowsy Driving in the capital area of Greece”. In: *Citeseer* 2016.01 (2006), pp. 1–18.

[48] Mohammed Ghazal et al. “Embedded Fatigue Detection Using Convolutional Neural Networks with Mobile Integration”. In: *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. IEEE. 2018, pp. 129–133.

[49] Sayani Ghosh, Tanaya Nandy, and Nilotpal Manna. “Real time eye detection and tracking method for driver assistance system”. In: *Advancements of medical electronics* (2015), pp. 13–25.

[50] gizmodo. *Uber Driver in Fatal Tempe Crash May Have Been Watching The Voice Behind the Wheel.* 2018. URL: https://gizmodo.com/uber-driver-in-fatal-tempe-crash-may-have-been-watching-1827039127 (visited on 06/30/2018).

[51] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems.* 2014, pp. 2672–2680.

[52] Wang Huan Gu et al. “Hierarchical CNN-based real-time fatigue detection system by visual-based technologies using MSP model”. In: *IET Image Processing* 12.12 (2018), pp. 2319–2329.

[53] Ishaan Gulrajani et al. “Improved training of wasserstein gans”. In: *Advances in neural information processing systems.* 2017, pp. 5767–5777.

[54] Rahul Gupta. “Data augmentation for low resource sentiment analysis using generative adversarial networks”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 7380–7384.

[55] Wei Han et al. “Driver Drowsiness Detection Based on Novel Eye Openness Recognition Method and Unsupervised Feature Learning”. In: *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015* September (2016), pp. 1470–1475. DOI: 10.1109/SMC.2015.260.

[56] Young-Joo Han, Wooseong Kim, and Joon-Sang Park. “Efficient Eye-Blinking Detection on Smartphones: A Hybrid Approach Based on Deep Learning”. In: *Mobile Information Systems* 2018 (2018).

[57] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Multimedia Tools and Applications* (2017), pp. 1–17. ISSN: 15737721. DOI: 10.1007/s11042-017-4440-4. arXiv: 1512.03385.

[58] Kaiming He et al. “Identity mappings in deep residual networks”. In: *European Conference on Computer Vision.* Springer. 2016, pp. 630–645.

[59] K Henneberg. “Chapter 14 - Principles of Electromyography”. In: *The Biomedical Engineering Handbook* (2000), pp. 1–11. DOI: doi:10.1201/9781420049510.ch14.

[60] Martin Heusel et al. “Gans trained by a two time-scale update rule converge to a local nash equilibrium”. In: *Advances in Neural Information Processing Systems.* 2017, pp. 6626–6637.

[61] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent”. In: *Coursera Lecture slides* (2012).

[62] Max Hirshkowitz et al. “National sleep foundation’s sleep time duration recommendations: Methodology and results summary”. In: *Sleep Health* 1.1 (2015), pp. 40–43. DOI: 10.1016/j.sleh.2014.12.010. URL: http://dx.doi.org/10.1016/j.sleh.2014.12.010.

[63] E Hoddes et al. “Quantification of sleepiness: a new approach”. In: *Psychophysiology* 10.4 (1973), pp. 431–436.

[64] Md. Yousuf Hossain and Fabian Parsia George. “IOT Based Real-Time Drowsy Driving Detection System for the Prevention of Road Accidents”. In: *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)* 3 (2018), pp. 190–195.

[65] *How Long Does Caffeine Stay in Your System?* URL: https://www.healthline.com/health/how-long-does-caffeine-last (visited on 08/22/2017).

[66] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[67] David H Hubel and Torsten N Wiesel. In: *Diagnostic Cytopathology* 14.2 (1996), pp. 162–164.

[68] Phung Huynh and Yong Guk Kim. "Detection of Driver Drowsiness Using 3D Deep Neural Network and Semi-Supervised Gradient Boosting Machine". In: 10116.April (2017). ISSN: 0302-9743. DOI: 10.1007/978-3-319-54407-6. arXiv: 1603.06937. URL: http://link.springer.com/10.1007/978-3-319-54407-6.

[69] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications". In: *Neural networks* 13.4-5 (2000), pp. 411–430.

[70] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[71] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[72] Rateb Jabbar et al. "Real-time driver drowsiness detection for Android application using deep neural networks techniques". In: *arXiv preprint arXiv:1811.01627* (2018).

[73] Paul Jackson, Alain Muzet, and Adrian Williams. *Fatigue , sleepiness and reduced alertness as risk factors in driving Fatigue , sleepiness and reduced alertness as risk factors in driving.* October. 2004. ISBN: 8248004503.

[74] Heinrich Jiang and Ofir Nachum. "Identifying and Correcting Label Bias in Machine Learning". In: *CoRR* abs/1901.04966 (2019). arXiv: 1901.04966. URL: http://arxiv.org/abs/1901.04966.

[75] Zhuoni Jie et al. "Analysis of yawning behaviour in spontaneous expressions of drowsy drivers". In: *Automatic Face Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on.* IEEE. 2018, pp. 571–576.

[76] Murray W Johns. "A new method for measuring daytime sleepiness: the Epworth sleepiness scale". In: *sleep* 14.6 (1991), pp. 540–545.

[77] Łukasz Kaiser, Aidan N Gomez, and Francois Chollet. "Depthwise Separable Convolutions for Neural Machine Translation". In: *arXiv preprint arXiv:1706.03059* (2017).

[78] Kashmir Hill. *The Secretive Company That Might End Privacy as We Know It.* 2020. URL: `https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html?auth=link-dismiss-google1tap` (visited on 02/10/2020).

[79] Katyanna Quach. *TThe infamous AI gaydar study was repeated – and, no, code can't tell if you're straight or not just from your face.* 2019. URL: `https://www.theregister.co.uk/2019/03/05/ai_gaydar/` (visited on 02/10/2020).

[80] Rupinder Kaur and Karamjeet Singh. "Drowsiness detection based on EEG signal analysis using EMD and trained neural network". In: *Int. J. Sci. Res* 10 (2013), pp. 157–161.

[81] Ki Wan Kim et al. "A Study of Deep CNN-Based Classification of Open and Closed Eyes Using a Visible Light Camera Sensor". In: *Sensors* 17.7 (2017), p. 1534.

[82] Sukwon Kim, Ben D Cranor, and Young S Ryu. "Fatigue : working under the influence". In: *Proceedings of the XXIst Annual International Occupational Ergonomics and Safety Conference* June (2009), pp. 317–322.

[83] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). arXiv: `1412.6980`. URL: `http://arxiv.org/abs/1412.6980`.

[84] Naveen Kodali et al. "How to Train Your DRAGAN". In: *CoRR* abs/1705.07215 (2017). arXiv: `1705.07215`. URL: `http://arxiv.org/abs/1705.07215`.

[85] Wanzeng Kong et al. "A system of driving fatigue detection based on machine vision and its application on smart device". In: *Journal of Sensors* 2015 (2015).

[86] K. Kozak et al. "Evaluation of Lane Departure Warnings for Drowsy Drivers". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50 (2006), pp. 2400–2404. DOI: `10.1177/154193120605002211`.

[87] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems.* 2012, pp. 1097–1105.

[88] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: *Citeseer* (2009).

[89] N Kurian and D Rishikesh. “Real time based driver’s safeguard system by analyzing human physiological signals”. In: *Int. J. Eng. Trends Technol* 4 (2013), pp. 41–45. URL: `http://www.ijettjournal.org/volume-4/issue-1/IJETT-V4I1P206.pdf`.

[90] Oliver Langner et al. “Presentation and validation of the Radboud Faces Database”. In: 907172236 (2010). DOI: `10.1080/02699930903485076`.

[91] Yann LeCun et al. “A theoretical framework for back-propagation”. In: *Proceedings of the 1988 connectionist models summer school.* Vol. 1. CMU, Pittsburgh, Pa: Morgan Kaufmann. 1988, pp. 21–28.

[92] Y LeCun et al. “Efficient backprop in neural networks: Tricks of the trade (orr, g. and müller, k., eds.)” In: *Lecture Notes in Computer Science* 1524.98 (1998), p. 111.

[93] Chuan Li and Michael Wand. “Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks”. In: (2016), pp. 1–17. arXiv: `arXiv:1604.04382v1`.

[94] Xue Li et al. “Driver’s Eyes State Detection Based on Adaboost Algorithm and Image Complexity”. In: *2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA)* (2015), pp. 349–352. DOI: `10.1109/ISDEA.2015.93`.

[95] Ziwei Liu et al. “Deep Learning Face Attributes in the Wild”. In: *ICCV* (2015), pp. 3730–3738.

[96] Lowveld. *The top 3 causes of accidents?* 2017. URL: `http://lowveld.getitonline.co.za/2017/04/12/top-3-causes-accidents/WSOX6WiGPIU` (visited on 05/20/2017).

[97] Atul Luthra. *ECG Made Easy.* Vol. 1. 2015. ISBN: 9788578110796. DOI: `10.1017/CBO9781107415324.004`. arXiv: `arXiv:1011.1669v3`.

[98] Jie Lyu, Zejian Yuan, and Dapeng Chen. “Long-term multi-granularity deep framework for driver drowsiness detection”. In: *arXiv preprint arXiv:1801.02325* (2018).

[99] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[100] Masoud Mahdianpari et al. “Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery”. In: *MDPI* (2018). DOI: `10.3390/rs10071119`.

[101] Maib Suntis. “Safety Bulletin”. In: *MAIB Reports* June (2014), pp. 1–3.

[102] B. N. Manu. “Facial features monitoring for real time drowsiness detection”. In: *Proceedings of the 2016 12th International Conference on Innovations in Information Technology, IIT 2016* (2017), pp. 78–81. DOI: 10.1109/INNOVATIONS.2016.7880030.

[103] Xudong Mao, Qing Li, and Haoran Xie. “AlignGAN: Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks”. In: *CoRR* abs/1707.01400 (2017). arXiv: 1707.01400. URL: http://arxiv.org/abs/1707.01400.

[104] Q. Massoz et al. “The ULg multimodality drowsiness database (called DROZY) and examples of use”. In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV).* Mar. 2016, pp. 1–7. DOI: 10.1109/WACV.2016.7477715.

[105] Luke Mastin. *Sleep - What Is Sleep, How Does Sleep Work, Why Do We Sleep, How Sleep Can Go Wrong.* 2013. URL: https://www.howsleepworks.com/ (visited on 06/01/2017).

[106] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[107] Mercedes-Benz. *Mercedes-Benz Attention Assist.* http://preview.thenewsmarket.com/Previews/NCAP/DocumentAssets/220986.pdf. 2011. (Visited on 06/06/2017).

[108] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[109] Ajay Mittal et al. “Head movement-based driver drowsiness detection: A review of state-of-art techniques”. In: *Proceedings of 2nd IEEE International Conference on Engineering and Technology, ICETECH 2016* (2016), pp. 903–908. DOI: 10.1109/ICETECH.2016.7569378.

[110] Tony C. W. Mok and Albert C. S. Chung. “Learning Data Augmentation for Brain Tumor Segmentation with Coarse-to-Fine Generative Adversarial Networks”. In: *CoRR* abs/1805.11291 (2018). arXiv: 1805.11291. URL: http://arxiv.org/abs/1805.11291.

[111] Nurul Muthmainnah, Mohd Noor, and Salmiah Ahmad. “Analysis of Different Level of EOG Signal from Eye Movement for Wheelchair Control”. In: *Int. J. of Biomedical Engineering and Technology* 11.2 (2013), pp. 175–196.

[112] Taro Nakamura, Akinobu Maejima, and Shigeo Morishima. “Driver Drowsiness Estimation from Facial Expression Features”. In: *Computer Vision Theory and Applications* (2014).

[113] Yuval Netzer et al. “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.* 2011. URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[114] NewRules. *New road rules for SA: What's going on with the speed limit? | Wheels24.* URL: http://www.wheels24.co.za/News/Guides_and_Lists/new-road-rules-for-sa-whats-going-on-with-the-speed-limit-20170313 (visited on 06/26/2017).

[115] Ziyanda Ngcobo. *Over 1,700 people died on SA roads this festive season.* 2017. URL: http://ewn.co.za/2017/01/10/over-1-700-people-died-on-sa-roads-this-festive-season (visited on 05/20/2017).

[116] Nhat M Nguyen and Nilanjan Ray. "Generative Adversarial Networks using Adaptive Convolution". In: *arXiv preprint arXiv:1802.02226* (2018).

[117] M. Ngxande, J. Tapamo, and M. Burke. "DepthwiseGANs: Fast Training Generative Adversarial Networks for Realistic Image Synthesis". In: *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA).* Jan. 2019, pp. 111–116. DOI: 10.1109/RoboMech.2019.8704766.

[118] NHTSA. *Drowsy Driving NHTSA reports.* 2018. URL: https://www.nhtsa.gov/risky-driving/drowsy-driving (visited on 09/29/2018).

[119] *NTHU CVlab - Driver Drowsiness Detection Dataset.* 2016. URL: http://cv.cs.nthu.edu.tw/php/callforpaper/datasets/DDD/ (visited on 09/03/2017).

[120] Halszka Oginska. "Who's not Sleepy at Night? Individual Factors Influencing Resistance to Drowsiness during Atypical Working Hours". In: *Nato science series sub series I life and behavioural science* 355 (2003), pp. 330–335.

[121] Fariborz Omidi and Gebraeil Nasl Saraji. "Non-intrusive Methods used to Determine the Driver Drowsiness: Narrative Review Articles". In: *International Journal of Occupational Hygiene* 8.4 (2016), pp. 186–191.

[122] Craig D Oster. "Proper Skin Prep Helps Ensure ECG Trace Quality". In: *Critical Care* (1998).

[123] Sanghyuk Park et al. "Driver drowsiness detection system based on feature representation learning using various deep networks". In: *Asian Conference on Computer Vision.* Springer. 2016, pp. 154–164.

[124] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. "Deep Face Recognition." In: *BMVC.* Vol. 1. 3. 2015, p. 6.

[125] Leo Pauly and Deepa Sankar. “Detection of drowsiness based on HOG features and SVM classifiers”. In: *Proceedings of 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2015* (2015), pp. 181–186. DOI: 10.1109/ICRCICN.2015.7434232.

[126] Leo Pauly and Deepa Sankar. “Detection of drowsiness based on hog features and svm classifiers”. In: *Research in Computational Intelligence and Communication Networks (ICRCICN), 2015 IEEE International Conference on*. IEEE. 2015, pp. 181–186.

[127] *Perceptrons and Multi-Layer Perceptrons: The Artificial Neuron at the Core of Deep Learning*. 2019. URL: https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/ (visited on 05/03/2019).

[128] T Shiva Prasad and N Raghu Kisore. “Application of Hidden Markov Model for classifying metamorphic virus”. In: *Advance Computing Conference (IACC), 2015 IEEE International*. IEEE. 2015, pp. 1201–1206.

[129] *Psychological Image Collection at Stirling (PICS)*. accessed 2018/09/10. 2018. URL: http://www.pics.stir.ac.uk.

[130] A. Punitha, M. Kalaiselvi Geetha, and A. Sivaprakash. “Driver fatigue monitoring system based on eye state analysis”. In: *2014 International Conference on Circuits, Power and Computing Technologies, ICCPCT 2014* (2014), pp. 1405–1408. DOI: 10.1109/ICCPCT.2014.7055020.

[131] Rohan Putta, Gayatri N Shinde, and Punit Lohani. “Real Time Drowsiness Detection System using Viola Jones Algorithm”. In: *International Journal of Computer Applications* 95.8 (2014).

[132] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: *CoRR* abs/1511.06434 (2015). arXiv: 1511.06434. URL: http://arxiv.org/abs/1511.06434.

[133] Bhargava Reddy et al. “Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks”. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE. 2017, pp. 438–445.

[134] Bhargava Reddy et al. “Real-time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks”. In: *Computer Vision and Pattern Recognition Workshops* (2017). DOI: 10.1109/CVPRW.2017.59.

[135] Andrea Renda. “Ethics, algorithms and self-driving cars–a CSI of the ‘trolley problem’. CEPS Policy Insights No 2018/02, January 2018”. In: *CEPS Policy Insight* 2018/02 (2018).

[136] Corneliu Florea research. *Eye-Chimera Database.* 2017. URL: `http://imag.pub.ro/common/staff/cflorea/cf_research.html` (visited on 09/26/2017).

[137] Maria Rimini-doering et al. “Effects of lane departure warning on drowsy drivers’ performance and state in a simulator”. In: *Proceedings of the Third International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design* (2005), pp. 88–95. URL: `http://trid.trb.org/view.aspx?id=763021`.

[138] *Road Safety Advice for foreigners driving in South Africa.* 2018. URL: `https://www.arrivealive.mobi/road-safety-advice-for-foreigners-driving-in-south-africa` (visited on 09/20/2018).

[139] Road Traffic Management Corporation. “Interim Road Traffic and Fatal Crash Report for the year 2006”. In: January (2007). URL: `www.arrivealive.co.za`.

[140] Timothy Roehrs and Thomas Roth. “Sleep, sleepiness, and alcohol use”. In: *Alcohol research and Health* 25.2 (2001), pp. 101–109.

[141] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[142] Leon Rosenthal, Timothy A Roehrs, and Tom Roth. “The sleep-wake activity inventory: a self-report measure of daytime sleepiness”. In: *Biological Psychiatry* 34.11 (1993), pp. 810–820.

[143] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Neurocomputing: Foundations of Research”. In: ed. by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988. Chap. Learning Representations by Back-propagating Errors, pp. 696–699. ISBN: 0-262-01097-6. URL: `http://dl.acm.org/citation.cfm?id=65669.104451`.

[144] Mehrdad Sabet et al. “A new system for driver drowsiness and distraction detection”. In: *ICEE 2012 - 20th Iranian Conference on Electrical Engineering* (2012), pp. 1247–1251. DOI: `10.1109/IranianCEE.2012.6292547`.

[145] Vandna Saini and Rekha Saini. “Driver Drowsiness Detection System and Techniques : A Review”. In: *International Journal of Computer Science and Information Technologies* 5.3 (2014), pp. 4245–4249. URL: `http://ci.nii.ac.jp/naid/110007970269/`.

[146] Tim Salimans et al. "Improved Techniques for Training GANs". In: *CoRR* abs/1606.03498 (2016). arXiv: `1606.03498`. URL: `http://arxiv.org/abs/1606.03498`.

[147] Lawrence K Saul and Sam T Roweis. "An introduction to locally linear embedding". In: *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html* (2000).

[148] Ramprasaath R Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." In: *ICCV*. 2017, pp. 618–626.

[149] Ankita Shah et al. "Yawning Detection of Driver Drowsiness". In: *International Journal of Recent Development in Engineering and Technology* 2.3 (2014), pp. 137–140. ISSN: 1882-7764. URL: `http://ci.nii.ac.jp/naid/110007970269/`.

[150] Jiaqi Shao et al. "A Lightweight Convolutional Neural Network Based on Visual Attention for SAR Image Target Classification". In: *Sensors* 18.9 (2018), p. 3039.

[151] Uzma Siddiqui and A N Shaikh. "An Overview of Electrooculography". In: *International Journal of Advanced Research in Computer and Communication Engineering* 2.11 (2013), pp. 4328–4330. ISSN: 2278-1021. URL: `www.ijarcce.com`.

[152] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[153] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[154] Hardeep Singh, Jagjit Singh Bhatia, and Jasbir Kaur. "Eye tracking based driver fatigue monitoring and warning system". In: *India International Conference on Power Electronics, IICPE 2010* (2011). DOI: `10.1109/IICPE.2011.5728062`.

[155] Karamjeet Singh and Rupinder Kaur. "Physical and physiological drowsiness detection methods". In: *International Journal of IT, Engineering and Applied Sciences* (2012).

[156] Lindsay I Smith. "A tutorial on Principal Components Analysis". In: *Statistics* 51 (2002), p. 52. ISSN: 03610926. DOI: `10.1080/03610928808829796`. arXiv: `1511.06448`.

[157] *South Africa's shocking road death numbers at highest level in 10 years.* 2010. URL: `https://businesstech.co.za/news/motoring/178275/south-africas-shocking-road-death-numbers-at-highest-level-in-10-years/` (visited on 06/06/2017).

[158] Caio B Souto et al. “Real-time SVM Classification for Drowsiness Detection Using Eye Aspect Ratio”. In: *Probabilistic Safety Assessment and Management* (2018).

[159] Nitish Srivastava et al. “Dropout: A simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[160] G. Stewart. “On the Early History of the Singular Value Decomposition”. In: *SIAM Review* 35.4 (1993), pp. 551–566. DOI: 10.1137/1035134. URL: https://doi.org/10.1137/1035134.

[161] Jane C Stutts, Jean W Wilkins, and Bradley V Vaughn. “Why do people have drowsy driving crashes”. In: *AAA Foundation for Traffic Safety Washington (DC)* 202.638 (1999), p. 5944.

[162] R Subhashini and S.L Veena. “Driver Alertness Based on Eye Blinking and Bio-signals”. In: *International Journal of Advanced Research* 2.3 (2014), pp. 666–670. ISSN: 2320-5407.

[163] Yijia Sun, S Zafeiriou, and M Pantic. “A Hybrid System for On-line Blink Detection”. In: *Forty-Sixth Annual Hawaii International Conference on System Sciences* (2013). URL: http://ibug.doc.ic.ac.uk/media/uploads/documents/yijiablink.pdf.

[164] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[165] Christian Szegedy et al. “Inception-v4, inception-resnet and the impact of residual connections on learning.” In: *AAAI*. Vol. 4. 2017, p. 12.

[166] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2818–2826.

[167] Eyosiyas Tadesse, Weihua Sheng, and Meiqin Liu. “Driver Drowsiness Detection through HMM based Dynamic Modeling”. In: *International Conference on Robotics and Automation* (2014). DOI: 10.1109/ICRA.2014.6907440.

[168] Tambiama Madiega. *EU guidelines on ethics in artificial intelligence: Context and implementation*. 2019. URL: https://www.europarl.europa.eu/RegData/etudes/BRIE/2019/640163/EPRS_BRI(2019)640163_EN.pdf (visited on 02/10/2020).

[169] Wei Ren Tan et al. “ArtGAN: Artwork Synthesis with Conditional Categorial GANs”. In: *CoRR* abs/1702.03410 (2017). arXiv: 1702.03410. URL: http://arxiv.org/abs/1702.03410.

[170] Brian C. Tefft. “The Prevalence and Impact of Drowsy Driving”. In: (2010), pp. 1–15. URL: https://www.aaafoundation.org/sites/default/files/2010DrowsyDrivingReport_1.pdf.

[171] Pallavi M Tekade and S Gawali. “Investigation and new method of no intrusive detection of driver drowsiness”. In: *International Journal of Engineering and Innovative Technology* 1.5 (2012), pp. 210–216.

[172] Joshua B Tenenbaum, Vin De Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *science* 290.5500 (2000), pp. 2319–2323.

[173] S Tereza and Č Jan. “Real-Time Eye Blink Detection using Facial Landmarks”. In: *Proc. Computer Vision Winter Workshop*. 2016.

[174] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. “On catastrophic forgetting and mode collapse in Generative Adversarial Networks”. In: *CoRR* abs/1807.04015 (2018). arXiv: 1807.04015. URL: http://arxiv.org/abs/1807.04015.

[175] *The Closed Eyes in the Wild (CEW) dataset*. URL: http://parnec.nuaa.edu.cn/xtan/data/ClosedEyeDatabases.html (visited on 04/19/2018).

[176] D Ulyanov, A Vedaldi, and VS Lempitsky. “Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv 2016”. In: *arXiv preprint arXiv:1607.08022* (2016).

[177] Paul Viola and Michael J Jones. “Robust Real-Time Face Detection”. In: *International Journal of Computer Vision.* 57.2 (2004), pp. 137–154.

[178] *Visualizing what ConvNets learn*. 2019. URL: http://cs231n.github.io/understanding-cnn/ (visited on 01/20/2019).

[179] Martha Hotz Vitaterna, Joseph S Takahashi, and Fred W Turek. “Overview of circadian rhythms”. In: *Alcohol Research and Health* 25.2 (2001), pp. 85–93.

[180] Gaihua Wang et al. “An multi-scale learning network with depthwise separable convolutions”. In: *Transactions on Computer Vision and Applications* (2018), pp. 1–8. DOI: https://doi.org/10.1186/s41074-018-0047-6.

[181] Di Wen, Hu Han, and Anil K Jain. “Face spoof detection with image distortion analysis”. In: *IEEE Transactions on Information Forensics and Security* 10.4 (2015), pp. 746–761.

[182] Wheels24. *Driving after 5 hours of sleep is akin to drunken driving - study.* 2016. URL: http://www.wheels24.co.za/Road_Trip/On_The_Road/driviving-after-5-hours-of-sleep-is-akin-to-drunken-driving-study-20161214 (visited on 05/17/2017).

[183] Wikipedia. *Ethnic groups in South Africa.* 2018. URL: https://en.wikipedia.org/wiki/Ethnic_groups_in_South_Africa (visited on 10/15/2018).

[184] Wikipedia. *Lane departure warning system.* 2017. URL: https://en.wikipedia.org/wiki/Lane_departure_warning_system (visited on 06/15/2017).

[185] Zbigniew Wojna et al. "The devil is in the decoder". In: *arXiv preprint arXiv:1707.05847* (2017).

[186] World Health Organization. *Global Status Report on Road Safety 2013.* 2013. URL: http://www.who.int/violence_injury_prevention/road_safety_status/2013/en/index.html (visited on 05/29/2017).

[187] Worldometers. *South Africa Population.* 2018. URL: http://www.worldometers.info/world-population/south-africa-population/ (visited on 10/30/2018).

[188] Eric Wu et al. "Conditional Infilling GANs for Data Augmentation in Mammogram Classification". In: *CoRR* abs/1807.08093 (2018). arXiv: 1807.08093. URL: http://arxiv.org/abs/1807.08093.

[189] You Xie et al. "tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow". In: *CoRR* abs/1801.09710 (2018). arXiv: 1801.09710. URL: http://arxiv.org/abs/1801.09710.

[190] Byung-Jun Yoon. "Hidden Markov models and their applications in biological sequence analysis". In: *Current genomics* 10.6 (2009), pp. 402–415.

[191] Fisher Yu et al. "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop". In: *CoRR* abs/1506.03365 (2015). arXiv: 1506.03365. URL: http://arxiv.org/abs/1506.03365.

[192] Matthew D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: *CoRR* abs/1212.5701 (2012). arXiv: 1212.5701. URL: http://arxiv.org/abs/1212.5701.

[193] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision.* Springer. 2014, pp. 818–833.

[194] Bo Zhang, Wenjun Wang, and Bo Cheng. "Driver eye state classification based on cooccurrence matrix of oriented gradients". In: *Advances in Mechanical Engineering* 7.2 (2015). ISSN: 16878140. DOI: 10.1155/2014/707106.

[195] Fang Zhang et al. "Driver fatigue detection based on eye state recognition". In: *Proceedings - 2017 International Conference on Machine Vision and Information Technology, CMVIT 2017* (2017), pp. 105–110. ISSN: 10059830. DOI: 10.1109/CMVIT.2017.25.

[196] C. Zheng et al. "Lane-level positioning system based on RFID and vision". In: *IET International Conference on Intelligent and Connected Vehicles (ICV 2016)*. Sept. 2016, pp. 1–5. DOI: 10.1049/cp.2016.1172.

[197] B. Zhou et al. "Learning Deep Features for Discriminative Localization." In: *CVPR* (2016).

[198] Xuemin Zhu, Wei-long Zheng, and Baa-liang Lu. "EOG-based Drowsiness Detection Using Convolutional Neural Networks". In: *2014 International Joint Conference on Neural Networks (IJCNN)* (2014), pp. 128–134. DOI: 10.1109/IJCNN.2014.6889642.

[199] *ZJU Eyeblink Database*. 2017. URL: http://www.cs.zju.edu.cn/~gpan/database/db_blink.html (visited on 09/20/2017).