DRIVER PHONE USE DETECTION USING VISUAL, AUDIO AND INERTIAL SENSOR PROCESSING

by

Barend Jacobus Viviers

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

October 2019

SUMMARY

DRIVER PHONE USE DETECTION USING VISUAL, AUDIO AND INERTIAL SENSOR PROCESSING

by

Barend Jacobus Viviers

Supervisor(s):	Prof HC Myburgh
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Computer Engineering)
Keywords:	Phone use detection, driver distraction, phone localisation, behaviour classification, talking on phone detection, texting detection, convolutional neural networks

Driver distraction is a major cause of road accidents and fatalities, especially distraction caused by the use of phones while driving. Current driver phone use detection methods can be divided into two broad categories, namely vision and non-vision-based approaches. Several methods need additional hardware infrastructure for the system to function. Approaches requiring little infrastructure will have improved adoption rates. The type of output produced by each method of implementation needs careful consideration. Some methods estimate phone position (i.e. is the phone located in the driver's or passenger's side of the vehicle), while other methods detect real-time instances of phone use (i.e. whether the driver is currently talking on the phone).

Current vision-based methods can only detect a driver talking on the phone while it is held next to the ear. Arguably, even more dangerous phone use behaviour is texting, as it diverts a driver's attention for an extended period. This work focused on the implementation and combination of three driver phone use detection methods. Two of the methods provide phone localisation inside the vehicle. This is helpful, as it indicates if a phone might be in the driver's access area. The first localisation method utilises audio ranging to time the arrival of audio pulses; the second method uses embedded phone

inertial sensors to track the phone from a known reference point. The third method monitors driver behaviour using a camera and identifies instances of phone use. This vision-based approach continually monitors the driver and detects talking on the phone and texting behaviour as it occurs. An approach that combines the phone use detection methods developed is proposed. Phone localisation methods are fused with driver behaviour image classification to create a more accurate and robust system.

Comprehensive experimentation was conducted to test system performance in a wide variety of conditions and circumstances. Experiments were first conducted individually for each method; all methods were then tested together collectively. Collective evaluation involved numerous vehicle trips where factors such as harsh lighting conditions, head pose variations, increase in the ambient noise level and irregular phone pick-ups were tested. Experiments were chosen to evaluate method performance in real-world conditions. All experiments combined accounted for 122 minutes of collected data. This included a total of 7379 samples, 4839 samples were of 'no phone use', 1536 samples were of 'talking on phone' and 1004 samples were of 'texting'. Each sample relates to one second of recorded data.

Results from experimentation show that very accurate localisation and driver behaviour identification can be provided by the methods developed. Audio ranging was the most accurate localisation method. It obtained overall average precision of 94.61% and recall of 96.22%. Phone inertial localisation achieved average precision of 83.50% and recall of 85.59%. The vision-based method that utilised a convolutional neural network (CNN) to classify driver behaviour yielded average precision of 91.47% and recall of 95.04%. CNN image classification combined with audio ranging localisation obtained even higher accuracy when detecting driver phone use behaviour. It obtained overall average precision of 95.89% and recall of 95.29% when classifying 'talking on phone' and 'texting' driver behaviour. A combination of detection methods increases system accuracy and robustness. A comparison of methods developed in this work to those in previous works illustrates that the new implementations provide several benefits and performance increases. The proposed solutions furthered the development of driver phone use detection systems. This contributes to the ultimate goal of lowering road accidents and fatalities caused by driver distraction.

ACKNOWLEDGEMENTS

This work is based on the research supported wholly / in part by the National Research Foundation of South Africa (Grant Number: UID 111723). This research was also supported by Telkom South Africa and Bytes Universal Systems via the Telkom Centre for Connected Intelligence (CCI) at the University of Pretoria.

LIST OF ABBREVIATIONS

AHRS	Attitude and Heading Reference System
CNN	Convolutional Neural Network
CPD	Change Point Detection
CWT	Continuous Wavelet Transform
CZT	Chirp Z-Transform
EKF	Extended Kalman Filter
FFT	Fast Fourier Transform
GGA	Generalized Goertzel Algorithm
HOG	Histogram of Oriented Gradients
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
MEMS	Micro-Machined Electromechanical System
NIR	Near-Infrared
ReLu	Rectified Linear Unit
RMS	Root Mean Square
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise Ratio
STD	Standard Deviation
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
UBI	Usage-Based Insurance

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION **1**
1.1 PROBLEM STATEMENT 1
1.1.1 Context of the problem 1
1.1.2 Research gap 2
1.2 RESEARCH OBJECTIVES AND QUESTIONS 3
1.3 APPROACH 3
1.4 RESEARCH GOALS 4
1.5 RESEARCH CONTRIBUTION AND OUTPUTS 4
1.6 OVERVIEW OF STUDY 4

CHAPTER 2 LITERATURE STUDY **6**
2.1 CHAPTER OVERVIEW 6
2.2 PROPOSED PHONE USE DETECTION METHODS 6
2.2.1 Types of phone use detection methods 7
2.2.2 Vision-based phone use detection methods 7
2.2.3 Non-vision-based phone use detection methods 10
2.3 INERTIAL NAVIGATION AND TRACKING 11
2.3.1 Types of inertial navigation systems 12
2.3.2 Attitude representation 12
2.3.3 Reference frames 13
2.3.4 Inertial sensor motion tracking 14
2.3.5 Sensor error sources 15
2.3.6 Error propagation in inertial tracking systems 16
2.4 DEEP LEARNING IMAGE CLASSIFICATION 16
2.4.1 Hierarchical feature learning 17

2.4.2 Convolutional neural networks . . . 19
2.5 CHAPTER SUMMARY . . . 22

CHAPTER 3 METHODS . . . 23
3.1 CHAPTER OVERVIEW . . . 23
3.2 IN-VEHICLE PHONE LOCALISATION UTILISING AUDIO RANGING . . . 23
3.2.1 Detecting pulse arrival time . . . 24
3.2.2 Time-frequency spectral analysis . . . 25
3.2.3 Audio pulse design . . . 27
3.2.4 Implementation of audio ranging phone localisation . . . 27
3.2.5 Change point detection implementation . . . 28
3.2.6 Gradient change point detection . . . 30
3.3 PHONE INERTIAL LOCALISATION . . . 30
3.3.1 Proposed phone inertial localisation approach . . . 31
3.3.2 AHRS algorithm . . . 35
3.4 CONVOLUTION NEURAL NETWORK-BASED PHONE USE DETECTION . . . 40
3.4.1 CNN architecture . . . 40
3.4.2 CNN hyperparameters . . . 42
3.4.3 Collection and preparation of dataset . . . 44
3.4.4 Training procedure . . . 47
3.5 COMBINATION OF PHONE USE DETECTION METHODS . . . 52
3.6 CHAPTER SUMMARY . . . 54

CHAPTER 4 RESULTS . . . 56
4.1 CHAPTER OVERVIEW . . . 56
4.2 AUDIO RANGING PHONE LOCALISATION RESULTS . . . 56
4.2.1 Experimental design . . . 57
4.2.2 Control experimentation results . . . 59
4.2.3 Natural pose experimentation results . . . 61
4.2.4 Signal noise analysis . . . 64
4.3 PHONE INERTIAL LOCALISATION RESULTS . . . 64
4.3.1 Experimental design . . . 64
4.3.2 Phone pick-up experimentation results . . . 66
4.4 CNN DRIVER PHONE USE CLASSIFICATION EXAMPLES . . . 67

4.5 COMBINED PHONE USE DETECTION METHOD RESULTS 69
4.5.1 Experimental design 69
4.5.2 Visual method classification output 71
4.5.3 Method combination experimentation results 74
4.6 CHAPTER SUMMARY 81

CHAPTER 5 DISCUSSION 82
5.1 CHAPTER OVERVIEW 82
5.2 AUDIO RANGING PHONE LOCALISATION 82
5.3 PHONE INERTIAL LOCALISATION 84
5.4 PERFORMANCE OF COMBINED PHONE USE DETECTION METHODS 85
5.4.1 CNN image classification evaluation 85
5.4.2 Method combination evaluation 86
5.5 SIGNIFICANCE OF RESULTS COMPARED TO PREVIOUS WORKS 88
5.5.1 Non-vision-based method comparison 88
5.5.2 Vision-based method comparison 89
5.6 CHAPTER SUMMARY 92

CHAPTER 6 CONCLUSION 93
6.1 SUMMARY OF WORK CONDUCTED 93
6.2 CONCLUSIONS FROM RESEARCH QUESTIONS 94
6.3 FUTURE WORK 96

REFERENCES 97

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

There has recently been an increase in the number of accidents that are caused by distracted drivers, in particular distraction caused by using a phone while driving. In 2017, 3 166 people were killed in vehicle crashes involving distracted drivers in the United States and there were 434 fatalities (i.e. 14%) as a result of phone use in distraction-affected crashes [1]. There is a growing trend among younger drivers to text while they drive. Drivers under the age of 30 accounted for 53% of distracted drivers using phones [1]. Texting is one of the most dangerous distractions, as it diverts a driver's attention from the road for an extended period of time.

Furthermore, drivers using phones are not inclined to compensate by allowing greater headway or reducing the vehicle's speed [2]. Various countries have introduced laws that attempt to penalise drivers for the use of mobile phones while driving [3], but this has proved to be an ineffective deterrent. The ineffectiveness of these deterrents can mainly be attributed to the difficulty in enforcing the penalties, as law enforcement officials are required to provide direct observation, which is inefficient and time-consuming. A practical application of monitoring driver phone use is usage-based insurance (UBI). A driver's insurance premiums could be adjusted according to the way the vehicle is driven. Drivers who agree not to use their phones while driving could pay a lower premium. Additional information could be provided to UBI systems through analysis of driver phone use.

1.1.2 Research gap

Many previously proposed methods that monitor phone use inside a vehicle require the use of additional infrastructure to function effectively. There is the possibility to development a phone localisation system that utilises minimal infrastructure. It would localise the phone to either the driver's or passenger's side of the vehicle, mainly by using the phone's embedded sensors. Certain assumptions about phone use can be made based on the phone's location. A phone located in the driver's area indicates that it could possibly be used. If there is an agreement between the driver and a separate party stating that the phone may not be located in the driver's area because of possible phone use, this could be enforced using a localisation method. Location information provided by these methods could also be utilised to improve the detection accuracy of methods that directly observe instances of phone use, such as a vision-based method. Work done previously [4, 5] had a similar view of phone localisation and identification of a phone in the driver's area. A system that monitors vehicle dynamics to determine phone location has already been developed [4]. However, this system requires the installation of additional hardware to function.

Vision-based systems previously proposed [6–9] are only able to detect a driver talking on the phone. Another phone use behaviour that is becoming more common is texting or the handling of a phone while looking down. The systems previously proposed are unable to detect this behaviour. Most previous methods have not performed extensive experiments to test method performance in harsh illumination conditions or when the driver exhibits excessive head pose variations. In addition, many tests have been conducted in controlled environments and not during actual vehicle trips, where several external factors may influence accuracy.

There is a possibility of combining phone use detection methods to create a more accurate and robust system. It could be beneficial to combine a phone localisation method with a vision-based method. Phone localisation would provide information as to when the phone is in the driver's area, while the vision-based method could identify specific phone use (i.e. talking on the phone or texting) instances. Current approaches have not attempted to fuse method output in this way before.

1.2 RESEARCH OBJECTIVES AND QUESTIONS

The main objective of the proposed research is to identify and develop driver phone use detection methods that are accurate and capable of functioning in real-world conditions. A system that combines the various methods will also be created. To achieve this objective, previous solutions to the driver phone use detection problem need to be investigated. The investigation will highlight areas where previous methods can be improved and reveal opportunities for the development of new methods. Once the development of driver phone use detection methods has concluded, extensive experimentation to evaluate method performance will be completed. Performance evaluation will include a comparison with existing methods. Research conducted in this work will aim to solve the following research questions:

1. Is it possible to develop a phone localisation system capable of accurately localising a phone to the driver's or passenger's side of a vehicle with minimal infrastructure requirements?
2. Is it possible to develop a vision-based phone use detection method that is capable of managing harsh illumination changes and excessive head pose variations? Furthermore, will such a system be able to detect not just a driver talking on the phone, but also texting?
3. How resilient will the implemented methods be to variable environmental conditions? Will a system that combines both vision and non-vision-based methods be able to obtain greater accuracy and robustness?

1.3 APPROACH

A thorough literature review of various methods that detect driver phone use was completed. Differences between non-vision-based and vision-based methods were studied. The benefits and drawbacks of the various approaches were investigated. Current computer vision and image-processing methods were studied to determine which are most suitable for implementation in a vision-based system.

Development involved the implementation of three different methods. Two methods localised the phone to either the passenger's or driver's side of the vehicle, one method utilised audio ranging, while the other used phone inertial sensors to track the phone's location. Finally, a vision-based system that classifies driver behaviour with the ability to identify a driver talking or texting on a phone was

implemented. A technique that combines the localisation methods with driver behaviour identification was developed.

Once the various methods were designed and implemented, evaluation of these components was performed. Experiments were conducted in different operating environments, where the effect of various human and environmental factors (e.g. varying lighting conditions) was evaluated. Results obtained from different methods were compared and the effectiveness of method combination was determined. Method performance was also compared to the results of previous studies.

1.4 RESEARCH GOALS

The research goal is to evaluate previous driver phone use detection methods and determine possible areas of improvement. The most promising methods need to be identified. Newly developed methods will be compared to previous works by producing a robust set of results. The goal of the results will be to determine the practical viability of the new methods and to show that a combination of detection methods will improve method accuracy and robustness.

1.5 RESEARCH CONTRIBUTION AND OUTPUTS

This work aims to further the development of driver phone use detection systems and provide a detailed analysis of their performance and effectiveness in various operating environments. Successful development and deployment of these systems could reduce road accidents and fatalities caused by driver phone use.

A work in progress conference paper was published in SATNAC 2018; it is titled ‘Towards an Accurate Low-Cost Driver Phone Use Detection System’. A journal article on the research conducted will be prepared, the target journal being IEEE Sensors.

1.6 OVERVIEW OF STUDY

Chapter 2 contains a literature review of material that is relevant to driver phone use detection. This includes an overview of previous detection methods. Chapter 3 details the implementation of three

phone use detection methods. Two methods localise the phone inside the vehicle cabin, while the third vision-based method can identify a driver talking or texting on the phone. An approach to combining detection methods is also explored. Results for each of the methods, as well as method combinations, are presented in Chapter 4. Chapter 5 analyses results obtained in Chapter 4 in more detail and evaluates method performance when compared to previous works. Finally, Chapter 6 concludes the dissertation and summarises the work completed. Each research question is addressed and suggestions for future work are provided.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

This chapter includes information relating to a literature study on previous driver phone use detection methods and subject matter relevant to the research conducted. A discussion of different phone use detection methods previously proposed is presented in Section 2.2. Methods are divided into vision-based and non-vision-based categories. Factors to consider in inertial localisation and tracking applications are described in Section 2.3. The role of deep learning in image classification tasks is examined in Section 2.4.

2.2 PROPOSED PHONE USE DETECTION METHODS

Various technologies currently proposed monitor driver behaviour. A popular aspect of monitoring driver behaviour is detecting driver distraction levels using visual tools and algorithms [10]. Driver distraction may be defined as taking a driver's attention away from vital activities that are required for safe driving; it is increasingly being recognised as a common cause of fatalities and injuries on the road. Non-intrusive methods for monitoring driver distraction are favoured; this has led to the appeal of using vision-based systems. Visual information relating to the behaviour of the driver is collected; visual cues are extracted by monitoring changes in facial behaviour. Another form of detecting driver distraction is identifying specific driver actions that could divert attention, such as phone use. There are several secondary activities that the driver may be doing in addition to the primary task of driving that could distract the driver's attention. Some secondary actions that may lead to driver distraction include drinking, eating, tuning the radio or operating a phone (either talking or texting) [10]. The

sub-category of monitoring driver behaviour specifically related to driver phone use is valuable and provides several practical applications.

2.2.1 Types of phone use detection methods

Detection of driver phone use may be divided into two broad categories. These categories are vision-based and non-vision-based methods. Various approaches have been used in both categories, with varying degrees of accuracy. Several approaches utilise additional hardware infrastructure that needs to be installed inside the vehicle to function properly. The use of existing infrastructure and minimising the use of expensive sub-systems are important factors that need to be considered. Vision-based solutions rely on computer vision, image-processing and machine-learning techniques to track and detect the use of a phone. Non-vision-based methods utilise a wide variety of sensor implementations and hardware solutions to determine phone use.

The type of output produced by different methods needs to be taken note of. Some methods provide phone position estimation inside the vehicle [4, 11–13]. This is useful, as it indicates if the phone is inside the driver's access area. Some methods detect actual instances when a phone is used by the driver (e.g. when talking on the phone) [6–9]. Other methods have used antennas to capture the power generated by a phone to determine when a phone is being used [14]. A method that monitors driving conditions has been developed [15]; it identifies the risk associated with phone use in different scenarios. A method that detects driver phone calls using voice features has previously been implemented [16]. Audio is recorded using a pre-installed microphone, audio is then transmitted and processed by the vehicle's on-board unit.

2.2.2 Vision-based phone use detection methods

Vision-based methods typically involve the use of a camera or vision system. A stand-alone camera is normally placed in a position facing the driver. Vision-based methods usually employ image-processing algorithms or computer vision techniques to analyse the recorded visual information. These approaches are popular because driver behaviour can be tracked in real-time. Four notable works that have utilised vision-based techniques are described [6–9]. A camera is installed inside the vehicle for all methods, except [7]. In [7], the camera is directed at the vehicle's windshield from above. All these methods

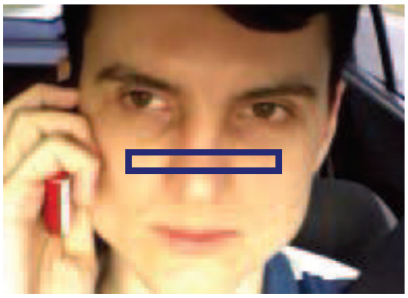
identify drivers talking on the phone; other types of phone use behaviour, such as a driver looking down while texting, cannot be detected. Each of these methods extracted hand-engineered features from training data. Other approaches that have identified driver distraction have utilised gaze tracking to determine if a driver's eyes are off the road [17–19]. These methods could potentially be useful in the detection of driver texting.

2.2.2.1 Hybrid vision system

A hybrid vision system for driver phone use detection was developed [9]. The hybrid system utilises a pattern recognition system for classification and a movement detection system for parameter selection. Prediction calculations are divided into time intervals of three seconds, meaning immediate classifications are not provided. A visual representation of the preprocessing and segmentation stages of the pattern recognition system is shown in Figure 2.1. This method required the selection of suitable threshold values. It also relied on segmentation using skin pixel values, which caused confusion when the drivers' faces were the same colour as their clothing or when their shoulders were visible in the frame. Instances where the driver was talking on the phone were identified with 83.81% accuracy and cases of no phone use obtained an accuracy of 92.74%.

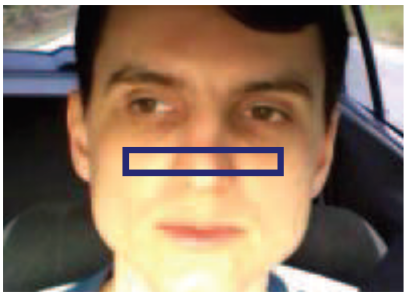
2.2.2.2 Phone use from high occupancy vehicle/high occupancy tolling near-infrared images

A near-infrared (NIR) camera system directed at the windshield of the driver has been developed for phone use detection [7]. Firstly, the driver's face region is localised within the windshield image. A region of interest is then classified around the driver's face to determine whether a mobile phone is being held. The NIR cameras are installed to manage high occupancy vehicle and tolling lanes. The utilisation of NIR cameras means that the system can be operated at night. Once face detection has been done, an area around the driver's face is extracted and classified to identify driver phone use.



(a) Cropped image (with phone present).

(b) Binary image (with phone).



(c) Cropped image (without phone).

(d) Binary image (without phone).

Figure 2.1. The preprocessing stage of the algorithm is shown in (a) and (c); the blue rectangles show samples of skin used for segmentation. The results of segmentation are shown in (b) and (d), where the pixels in the red region are counted. Taken from [9], © 2016 IEEE.

2.2.2.3 Facial landmark tracking

A vision-based approach was investigated that automatically detects when a phone is held near a driver's ear [6]. The location of the driver's facial landmarks is tracked to extract a crop of the desired region. Features are extracted and classified using the cropped region to determine driver mobile phone use. Figure 2.2 shows the process by which a cropped region is extracted after the user's facial landmarks have been located. The highest classification rate achieved using this approach was 93.86%.

Figure 2.2. A cropped region of interest that is extracted to determine the presence of a phone. Taken from [6], © 2015 IEEE.

2.2.3 Non-vision-based phone use detection methods

Non-vision-based methods commonly utilise sensors embedded in phones to detect dangerous driving behaviour [4, 11–13, 15]. Some methods require the use of additional hardware components and infrastructure that function in conjunction with the phone. Most non-vision-based methods cannot provide direct classification of driver phone use instances, but rather present insight into the phone's location.

2.2.3.1 Sensing vehicle dynamics

A technique that utilises the embedded sensors in a phone to estimate vehicle dynamics and determine driver phone use was developed [4]. Vehicle dynamics are sensed using embedded phone sensors to determine changes in centripetal acceleration. The centripetal acceleration varies depending on the position where it is measured in the vehicle. The acceleration measured by the phone is compared with a reference point to determine if the phone is located on the right or left side of the vehicle. This system requires the driver to navigate several corners before it becomes accurate. The phone sensor readings cannot be used directly to represent the vehicle dynamics, as they are pose-dependent. The system can achieve classification accuracy of 90%, which increases to 95% when sensing from multiple turns are combined.

2.2.3.2 Leveraging vehicle stereo system

A driver phone use detection system that leverages the existing stereo system in a vehicle was developed [11]. An acoustic ranging approach is followed where a series of customised high-frequency beeps are played from the phone through the vehicle stereo system using a Bluetooth network. The beeps are played at different time intervals across the left and right speakers. A change-point detection algorithm is used to calculate beep arrival time. The phone's distance from the vehicle's centre is estimated and a passenger or driver classification is made. The objective of localisation is to determine whether a phone is in the driver's area and therefore capable of being used while driving. Figure 2.3 shows an illustration of the steps involved in the system. The system achieved a classification accuracy of 90%, which improved to 95% in some calibration configurations.

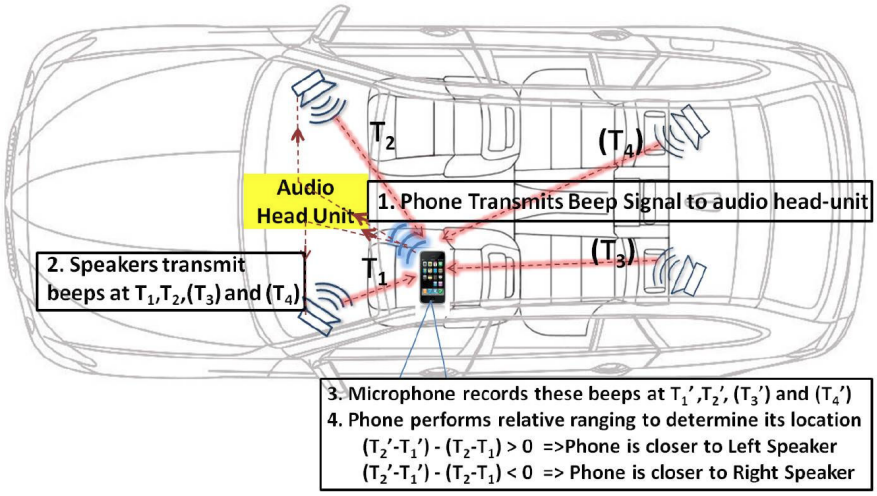


Figure 2.3. Illustration of the steps involved to determine phone position. Taken from [11], © 2012 IEEE.

2.3 INERTIAL NAVIGATION AND TRACKING

Inertial navigation refers to the use of gyroscopes and accelerometers to measure rotational and translational movement with respect to an inertial reference frame [20]. A new phone localisation approach, which utilises a phone's inertial sensors and limited additional infrastructure, is proposed. Relevant aspects relating to inertial navigation and motion tracking are examined. The collection and transmission of phone inertial sensor measurements fall in the field of vehicle telematics. This

includes categories such as monitoring road conditions, driver classification and transportation mode classification [21].

Inertial navigation and motion tracking would typically use relative localisation techniques through which an object's position is tracked relative to its initial location. Absolute localisation techniques might still be used in these systems to improve the accuracy of a location estimate because of relative localisation's tendency to drift. Inertial sensors have high sampling rates that provide accurate pose estimates over short time intervals, but drift over longer periods. These sensors can be combined with sensors that have lower sampling rates that do not drift over time to provide better pose estimates [22]. For pose estimation specifically, inertial sensor measurements can be combined with cameras [23–25], ultra-wideband methods [26–28] and global navigation satellite systems [29, 30] to correct relative measurements. Relative localisation is usually more readily available because it does not rely on external signals that need to be received.

2.3.1 Types of inertial navigation systems

Previously, inertial navigation systems (INSs) used stable platform technology through which inertial sensors were mounted on a stable platform that was isolated from the rotational motion of the vehicle. The mechanical complexity of these platforms has been removed in modern systems. Instead, the sensors are rigidly attached to the body of the host vehicle [20]. This approach is referred to as strapdown inertial navigation and provides several benefits, including reduced size, lower cost and better reliability.

2.3.2 Attitude representation

The attitude or orientation of the body with respect to the chosen reference frame may be represented in three different ways. The three attitude representations are:

1. Direction cosines: A 3×3 direction cosine matrix is used where the columns represent the unit vectors of the body axes projected along the reference axes. A measurement vector represented in the body axes can be translated to the reference axes by multiplying the vector with the direction cosine matrix.

2. Euler angles: Transforming a body from one coordinate plane to another involves three successive rotations about different axes. Owing to the presence of singularities for pitch angles of $\pm 90^{\circ}$, Euler angles are not commonly used in practice.
3. Quaternions: Transforming from one coordinate frame to another involves a single rotation about a vector that is defined in the reference frame. The quaternion is a four-element vector, where each element is a function of its orientation and the magnitude of rotation.

2.3.3 Reference frames

A reference frame refers to the set of axes to which sensor measurements are referenced [20]. The reference frames used in phone motion tracking are shown in Figure 2.4. The phone sensor frame (x_p, y_p, z_p) is the reference frame of the moving phone inside the vehicle. The phone's inertial measurements are resolved in this frame. When the phone is held in its default position the x-axis is horizontal and points to the right-hand side, while the y-axis is vertical and points upwards. The z-axis points outwards from the screen's face. The earth frame (x_e, y_e, z_e) rotates with the earth. Its origin is at the centre of the earth and its axes are fixed with respect to the earth. The navigation frame (x_n, y_n, z_n) is the local geographic frame where navigation takes place. In the intended application, the position and orientation of the phone sensor frame with respect to the navigation frame need to be computed. In most applications the navigation frame can be defined as stationary with respect to the earth frame; only when sensors move over large distances will the navigation frame be moved and rotated along the earth's surface.

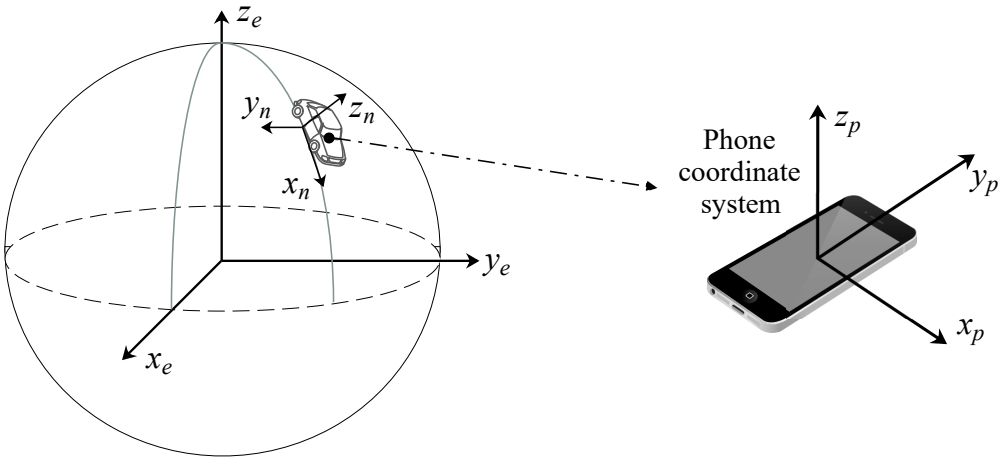


Figure 2.4. Coordinate system reference frames.

2.3.4 Inertial sensor motion tracking

Modern applications that require low-cost inertial sensors have promoted the development of micro-machined electromechanical system (MEMS) sensors. These sensors have eliminated many issues that previously prevented the use of inertial sensors in applications. Low cost, low power consumption and small size have made the use of these sensors more attractive and commonplace. Many smartphones have embedded MEMS sensors owing to the reduction in cost and size. Generally, smaller size leads to lower sensitivity and an increase in noise, but effective compensation techniques have resulted in better measurement accuracy [20].

Inertial navigation typically refers to the use of accelerometer and gyroscope measurements to determine the velocity and position of a vehicle, but inertial sensors can also be used for a wide variety of motion tracking applications. Current applications utilising an inertial measurement unit (IMU) for motion tracking have focused mainly on fields such as pedestrian dead-reckoning and indoor localisation [31–33], pen-tracking for writing recognition [34, 35], inertial navigation [36] and motion tracking [37, 38]. The trend of phones having embedded IMUs has opened up new opportunities for the everyday use of this technology, as it is more readily available. Commercial inertial measurement units are a cost-effective technology that can be used in motion tracking applications where optical technologies are unsuitable [38].

Acceleration measurements obtained from an accelerometer provide the specific force applied to the sensor. These measurements can be used to calculate velocity and position over time by performing successive integrations of acceleration over a period. Gyroscopic sensors can be used to measure the rotational motion of a body with respect to an inertial reference frame. This is also helpful in determining the orientation of accelerometers at any time. Given the rotational information, acceleration measurements can be translated into the correct reference frame before being integrated. The attitude and heading of the body with respect to the reference frame can be deduced from gyroscope measurements. This attitude and heading information can then be used to align accelerometer measurements with the desired reference frame. Once accelerations have been correctly aligned, double integration will provide the velocity and position of the body in the reference frame. While gyroscopes provide measurements of a body's turn rate, accelerometers are unable to separate acceleration with respect to inertial space and acceleration caused by the presence of a gravitational field. This is an important factor that needs consideration. Gyroscopes do not need to consider earth's gravity when calculating

the turn rate, but accelerometers require the removal of earth's gravity from measurements to obtain the true acceleration that is applied to the body. Therefore, the effect of the earth's gravitational force first needs to be removed from the aligned and corrected accelerometer measurements before integration. The process whereby inertial measurements are integrated to provide attitude and position estimates is often known as dead-reckoning and is shown in Figure 2.5.

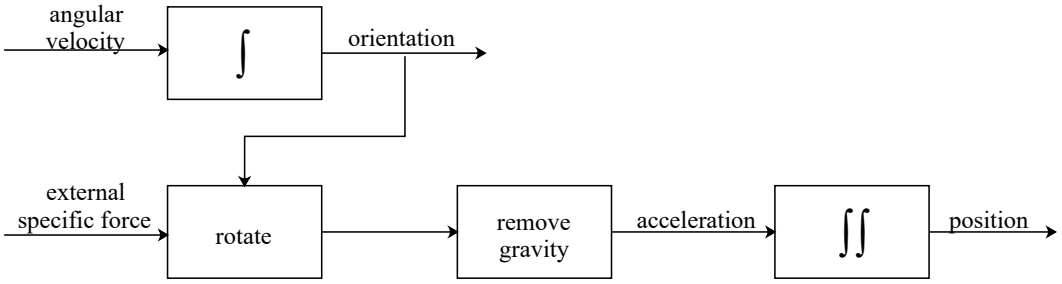


Figure 2.5. Process of determining attitude and position estimates from inertial sensors. Taken from [26], © 2015 IEEE.

2.3.5 Sensor error sources

A wide range of error sources is found in gyroscopes and accelerometers, especially MEMS-based sensors such as those found in phones. The average IMU error observed during a period of time mainly depends on the quality of the sensor used. IMU quality can be described as commercial, tactical, navigational or strategic and is affected by the type of inertial sensor technology being used [39]. A simulation that only considered random error sources with the device in a stationary position found commercial-grade IMUs to perform worst, with the highest drift, while strategic-grade IMUs performed best [39]. A drift of 1 cm was observed after 4 seconds for the commercial-grade IMU, while the strategic-grade IMU had the same amount of drift in 300 seconds. In practice, drift will grow at a more rapid rate owing to additional error sources.

The most common error sources present in gyroscopes and accelerometers include bias, sensor noise, temperature effects and bias instability [36]. When a sensor experiences a certain physical input, it will output a measurement that is offset by a bias. For a gyroscope, the bias is the average sensor output when no rotation is taking place. The constant bias of a gyroscope can be estimated by determining the long-term average of its output; it can then be compensated for by subtracting the bias from the output. A sensor that measures a constant signal will always have some degree of noise present in the

measurement. Changes in temperature will cause movement in the sensor's bias. Sensor bias does not stay constant over time because of flicker noise experienced by the electronics.

2.3.6 Error propagation in inertial tracking systems

The most important property to be considered in the calculated position of an INS is the rate at which drift errors accumulate. Drift mostly originates from the double integration of errors present in accelerometer measurements. Over extended periods a greater proportion of drift can be attributed to errors in the orientation, which arise from errors in angular velocity measurements (i.e. gyroscope measurements).

Even small orientation errors create an error in the estimated position that grows rapidly. Tilt errors are usually introduced through small bias errors in gyroscope measurements. These tilt errors grow linearly with time, resulting in positional drift that will grow proportional to t^3, where t refers to tracking duration [33].

2.4 DEEP LEARNING IMAGE CLASSIFICATION

Machine-learning systems are increasingly being used for tasks such as object detection in images, speech-to-text conversion, grouping news items and selecting the most relevant results in search queries. Traditional machine-learning systems were restricted in their capability to process data in a raw form. Building a machine-learning or pattern recognition system relied on expertise in the field of study and special engineering to design feature extraction methods that could transform raw data (e.g. image pixel values) into a suitable internal representation or feature vector. This feature vector forms part of a learning sub-system, such as a classifier, that classifies patterns in the input [40]. The choice of feature vector has an enormous effect on the performance of the machine learning algorithm. For many tasks, it is difficult to determine what the best features to extract are.

Deep learning is a sub-field of machine learning, which in turn is a sub-field of artificial intelligence. The relationship between these fields of study can be seen in Figure 2.6. Artificial intelligence refers to methods and techniques that enable computing devices to mimic human intelligence using logic to solve problems. Machine learning is a subset of artificial intelligence whereby a machine can use raw

data and acquire its knowledge through the extraction of patterns [41]. Deep learning is the subset of machine learning composed of algorithms that allow a machine to train itself and perform certain tasks, such as image classification or speech recognition, by presenting the network with a large amount of data.

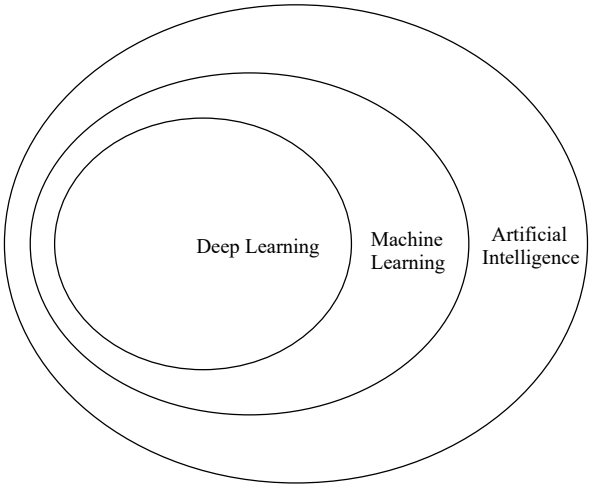


Figure 2.6. Venn diagram depicting the relationship between deep learning, machine learning and artificial intelligence. Adapted from [42], © 2018 IEEE.

Feature learning is a collection of techniques that enables a system to be given raw input values and automatically learn features and representations required for classification and detection tasks. Deep-learning methods are feature learning methods with multi-level representations. Representations are obtained by components that transform the representation at a specific level (beginning with the raw input) into a higher, more abstract representation. Complex functions may be learned by combining multiple such transformations. For classification tasks, higher levels of representation can be learned to enhance discriminating features in the input that are important and suppress non-relevant variations. For images, discriminating patterns learned in lower levels might represent corners and edges, while patterns learned in higher levels are more abstract, which is beneficial when distinguishing between classes. The vital aspect of deep learning is that these levels of discriminating patterns or features are not hand-engineered, but acquired automatically from a learning procedure [40].

2.4.1 Hierarchical feature learning

Machine and deep learning may be categorised into three types of learning: supervised learning, unsupervised learning and semi-supervised learning. Supervised learning is the most common form of

machine learning. Data that are representative of the classification task are presented to the system and used for training. The given training data are used to create a model or classifier through a training procedure where model predictions are made on input data and corrected when wrong. Model predictions are adjusted by modifying internal parameters (weight vectors). An objective function that measures the error between the output and the ground truth provided is computed. The training procedure continues until the model error rate obtains the desired accuracy or a certain number of iterations have passed. In contrast to supervised learning, unsupervised learning uses training data with no associated labels, meaning model predictions cannot be corrected when wrong. Semi-supervised learning has access to training data where only a fraction of it is labelled. The labelled training data are analysed and used to provide a label for each of the unlabelled data points, which can then be used as additional training data.

Many current machine-learning applications utilise linear classifiers in addition to hand-engineered features. Image recognition tasks require the input-output function to be insensitive to irrelevant scale, viewpoint, occlusion and illumination variations of the object. Linear classifiers require a good feature extractor to limit the effect of these variations; the classifier should be able to discriminate important aspects of the image, but be invariant to aspects that are irrelevant, such as changes to the object's orientation. Traditionally, hand-engineered features that are relevant to a specific problem would be designed. This may be avoided by learning good features automatically with a general-purpose learning procedure (e.g. deep neural networks) [40]. Networks can now be trained with many layers that are capable of hierarchical learning, where lower layers learn basic concepts and more abstract patterns are learned in higher layers.

The aim of the hand-engineered features is to encode shape (Hu moments [43], Zernike moments [44]), texture (local binary patterns [45], Haar-like features [46], Haralick texture [47]) and colour (colour histograms, colour correlograms [48]). Other approaches that describe the most distinct and interesting regions of an image are keypoint detectors (FAST [49], Harris [50], DOG [51]) and local invariant descriptors (SURF [52], SIFT [51], ORB [53], BRIEF [54]). A method that performs well at detecting objects in images when the position and orientation of the object are not considerably different from the images on which the classifier was trained is the histogram of oriented gradients (HOG) [55]. A great deal of research has focused on HOG and its variants, including exemplar support vector machines (SVMs) [56] and the deformable parts model [57], both of which are computationally expensive. In each of these cases, the algorithm was designed to quantify a specific aspect of the image (e.g.

colour, shape, texture etc.). The hand-engineered algorithms are applied to image pixels to obtain feature vectors describing the contents of the image. Feature vectors acquired from feature extraction are then fed as inputs to machine-learning models. Deep learning, more specifically convolutional neural networks (CNNs) function in a different way. Features are not learned from hand-engineered algorithms, but are instead automatically learned from the training procedure. Lower layers encode simple concepts, while more abstract patterns are learned in higher layers. This hierarchical feature learning approach completely removes the need for hand-designed feature extraction [58]. An image is provided and the pixel intensities are used as input to the CNN. Features are then extracted from the image utilising a series of hidden layers. Each layer builds upon the previous layer hierarchically. In contrast to traditional machine-learning algorithms where a performance plateau is reached despite a greater amount of training data, neural network classification accuracy increases as the depth of the network increases.

2.4.2 Convolutional neural networks

CNNs are one of the most popular and well known deep learning models used in computer vision applications, particularly for image classification. CNNs are end-to-end models, meaning features are learned internally to distinguish between classes, unlike traditional feature extraction and machine-learning approaches that require hand-engineered features. A data-driven approach is followed where example images relating to each class are used to teach the algorithm the fundamental differences between class categories. Four steps are required to create the classification system [58]. Firstly, the dataset relating to the classification problem is acquired. The second step involves splitting the acquired data into training, validation and testing sets. Thirdly, the network is trained using a learning procedure that attempts to discover what class categories look like by making predictions on the training set and correcting itself when wrong. Lastly, the network's performance is evaluated on the validation and test sets. In traditional feature-based image classification systems, an additional step is taken between steps two and three, namely feature extraction. This step usually implements hand-engineered algorithms (e.g. local binary patterns [45] and HOG [55]) to quantify the contents of the image. CNNs avoid this step, as features are learned inside internal layers that allow differentiation between object classes. Filters inside the hidden layers of the network are learned to distinguish between image classes.

2.4.2.1 Loss function and optimisation algorithm

The process of training a CNN can be described as parameterised learning, because the model summarises data with a fixed set of parameters. Two key components of parameterised learning are the scoring function and the loss function. Mapping of input data to output class labels is done by the scoring function. The ultimate goal of training a network is to find the optimal set of parameters that minimise some loss function. The loss function quantifies the performance of a scoring function (i.e. whether it correctly classifies input data points). Loss is minimised by adjusting parameters such as the weight vectors, using an optimisation algorithm.

The optimisation algorithm calculates a gradient vector for each weight vector that indicates by how much the loss would increase or decrease when a weight is incremented by a small amount. Following this computation, the weight vector is adjusted in the opposite direction to the gradient vector. A negative gradient vector indicates the direction of steepest descent in the loss landscape. The loss will be lower when the gradient vector is near a minimum [40]. Gradient descent is a commonly used optimisation algorithm in CNNs. For a given model and set of training inputs, the outputs, errors and average gradient for these inputs are computed. The weight vectors are then adjusted accordingly. This procedure continues for many iterations in the training set, until the average of the loss function stops decreasing. Learning valuable multi-stage feature extractors with little prior knowledge was thought to be infeasible. In particular, it was thought that simple gradient descent would get caught in poor local minima (i.e. weight configurations where no small change would decrease the objective function error). In general, local minima are not a serious problem; the system reaches solutions of similar quality regardless of the initial conditions presented [40].

2.4.2.2 Network layers

Some of the most basic building blocks used in CNNs include convolutions, normalisation, pooling, activation functions and fully connected layers. The convolutional layer is the core building block of a CNN. It consists of many learnable filters that are all applied to the input vector. Each filter is a matrix of $k \times k$ weights, where each weight is a learnable parameter. In a CNN, each neuron is only connected to a local region in the input volume called the local receptive field. The vectors produced by a convolutional neuron are passed through an activation function yielding a feature map.

The process whereby feature maps are produced is shown in Figure 2.7. An activation function often used in CNNs is rectified linear units (ReLUs). They are normally placed after fully connected and convolutional layers, but may also be used before layers as a pre-activation. Pooling is usually applied after several convolutional layers to reduce the spatial dimensions of the vector, thus lowering the number of parameters in the network. A larger stride may also be used in convolutional layers to reduce the representation size instead of using pooling layers. The normalisation of input data and the output of convolutional layers are common. Fully connected layers have neurons connected to all activations in the previous layer, similar to feedforward neural networks. Fully connected layers are always located at the end of the network.

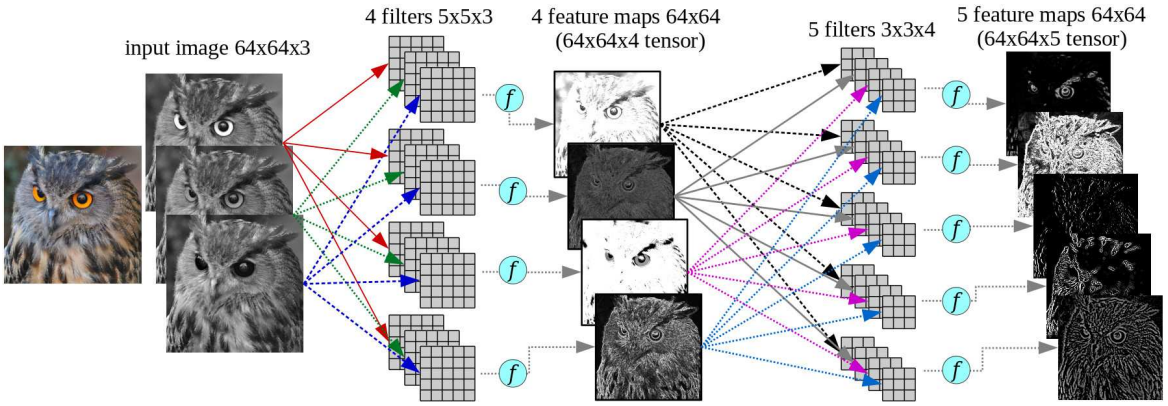


Figure 2.7. Illustration of feature maps produced from convolutional layers. The feature maps are stacked and presented as input to the next layer. The light blue circle denotes the activation function. Taken from [59], © 2017 IEEE.

2.4.2.3 Transfer learning

Transfer learning refers to the concept of using pre-trained models to learn new patterns from data on which these were not originally trained. The pre-trained model is used as a starting point for a different classification task. Models trained on large-scale image datasets such as ImageNet [60] have learned a great number of discriminating features for many separate object classes. Two main methods are used when performing transfer learning on networks. Networks can be utilised as feature extractors, through which new data are propagated through the network until reaching an arbitrary layer. The activations from this layer are then treated as feature vectors. The second method is fine-tuning networks by removing the head of the network and adding new fully connected layers to the head of the CNN. Weights from the newly added layers are then fine-tuned to recognise new object classes.

2.5 CHAPTER SUMMARY

A detailed review of previously developed driver phone use detection methods was completed. Methods can be divided into two broad categories, namely vision-based and non-vision-based approaches. It was found that the type of output produced by the implementation of each method needs careful consideration. Some methods estimate phone position, while others detect real-time instances of phone use. Brief descriptions of the most notable works were given. The most important aspects to consider for inertial navigation and motion tracking were then examined. This is required for the development of the new phone inertial localisation method. Finally, the role of deep learning in image classification tasks was evaluated.

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

This chapter presents the design and implementation of three driver phone use detection methods. Two methods provide localisation of the phone inside the vehicle, while the third method detects actual instances of phone use behaviour. The two phone localisation techniques rely on audio ranging and phone inertial position estimation and are presented in Sections 3.2 and 3.3 respectively. In the context of the problem, localisation refers to the identification of an area where the phone is located; full localisation is not performed. In Section 3.4 a CNN is designed and trained to correctly classify driving behaviour, which includes a driver talking on a phone or texting. Finally, a method to combine the three detection methods is described and proposed in Section 3.5 to produce a more robust and accurate driver phone use detection system.

3.2 IN-VEHICLE PHONE LOCALISATION UTILISING AUDIO RANGING

The fundamental principle upon which the approach is based relies on acoustic ranging and is a variation of a previously proposed method [11]. Audio pulses are played through the vehicle stereo system at specific intervals. The pulse arrival time at the phone will vary depending on its location in the vehicle. The exact onset of pulse arrival needs to be calculated, along with timing differences. Specific methods used for calculating these variables vary in precision and complexity.

Audio played through the vehicle stereo system comes in the form of high-frequency pulses. These pulses were chosen because humans find it difficult to perceive high-frequency sound and music played through the radio typically has lower frequency content. Two separate pulse frequencies are selected,

with the same frequency always played through the same channel (left or right speaker). This allows a particular pulse with its associated frequency to be assigned to either the left or right vehicle speaker. In practice the first pulse is played on a particular channel, followed by a short delay when no sound is played. The second pulse is then played on the opposite channel. The phone's microphone records audio segments to be processed. Challenges can arise due to differences in vehicle stereo systems. The frequency response of the speakers, the location of speakers and the number of speakers that are available are all factors that could differ from one vehicle to another.

3.2.1 Detecting pulse arrival time

Detecting accurate arrival times of signal pulses would be very difficult in the time-domain because of difficulty in distinguishing exact pulse onset. Signals represented in the time-domain only show the time-amplitude relationship. The most distinctive properties and features of the signal are often found in the frequency content of the signal. The frequency-domain representation of a signal shows which frequencies are present in the signal and at what magnitude. One of the most commonly used methods for transforming a signal from the time-domain to the frequency-domain is the fast Fourier transform (FFT), which provides a frequency-amplitude representation of the signal.

Frequency resolution is dependent on the relationship between the sampling rate of the recorded sound and the FFT length. The FFT resolution is $\frac{f_s}{N}$, where f_s is the sampling frequency and N is the FFT length [61]. The highest most common sampling rate found on phone microphones is 44.1 kHz. This sampling rate is fixed. The Nyquist-Shannon sampling theorem states that the greatest detectable frequency is 22.05kHz, i.e. half the sampling frequency. Pulses are limited to a single distinct frequency, meaning that wideband frequency analysis like that provided by the FFT is not necessary. Two techniques used in narrow-band spectrum analysis is the Chirp Z-transform (CZT) and the Generalized Goertzel algorithm (GGA) [62]. These techniques allow for computing the spectrum of a signal in a narrow band at a fine resolution.

However, the FFT, CZT and GGA all have a drawback in that time information is not available in the transformed signal. These methods provide frequency information about the signal, meaning that they indicate the amplitude of each frequency component that is present in the signal, but do not provide time intervals at which each of these frequency components occurs.

3.2.2 Time-frequency spectral analysis

The Fourier transform is not an appropriate analysis method for non-stationary signals, as it provides no information regarding time intervals when spectral components appear. For the case when time localisation of spectral components is required, a time-frequency representation of the signal is needed. A standard method for evaluating a time-varying signal is the short-time Fourier transform (STFT) [63]; another method is the wavelet transform. The wavelet transform allows for high-frequency components to be examined with sharper time-resolution than low-frequency components [64].

Figure 3.1 shows different domain representations of a recorded signal. Both wavelet transform and STFT are represented in the time-frequency domain. STFT provides fixed resolution across all times, while wavelet transform has variable resolution. Wavelet transform uses larger rectangles for low frequencies where precise frequency resolution is required and narrow rectangles for high frequencies where accurate time localisation is required. High-frequency components are better resolved in time, while low-frequency components are better resolved in frequency (i.e. it is easier to locate a particular high-frequency component in time than a low-frequency component). Low-frequency components are the opposite; they are located with better accuracy in frequency compared to high-frequency components. There is a trade-off between frequency and temporal resolution for the wavelet transform; higher temporal resolution comes at the cost of lower frequency resolution.

The only difference between the STFT and an FFT is that the signal is divided into segments small enough to be considered stationary when performing the STFT. A window length is chosen that breaks the signal up into smaller, equal parts. Because the window is of finite length, only a portion of the signal is covered, resulting in poorer frequency resolution. A window of infinite length provides an FFT with perfect frequency resolution, but no time information is available. A narrower window provides better temporal resolution at the cost of frequency resolution. If the target frequencies are sufficiently separated, it is advantageous to sacrifice frequency resolution for temporal resolution.

The wavelet transform analyses the signal using an approach called multi-resolution analysis. It analyses the signal at different resolutions, depending on the frequency. At high frequencies, it produces good temporal resolution and poor frequency resolution, while it produces poor temporal resolution and good frequency resolution at low frequencies. This is the preferred outcome for signals that typically have high-frequency spectra in short durations, while low-frequency spectra last for

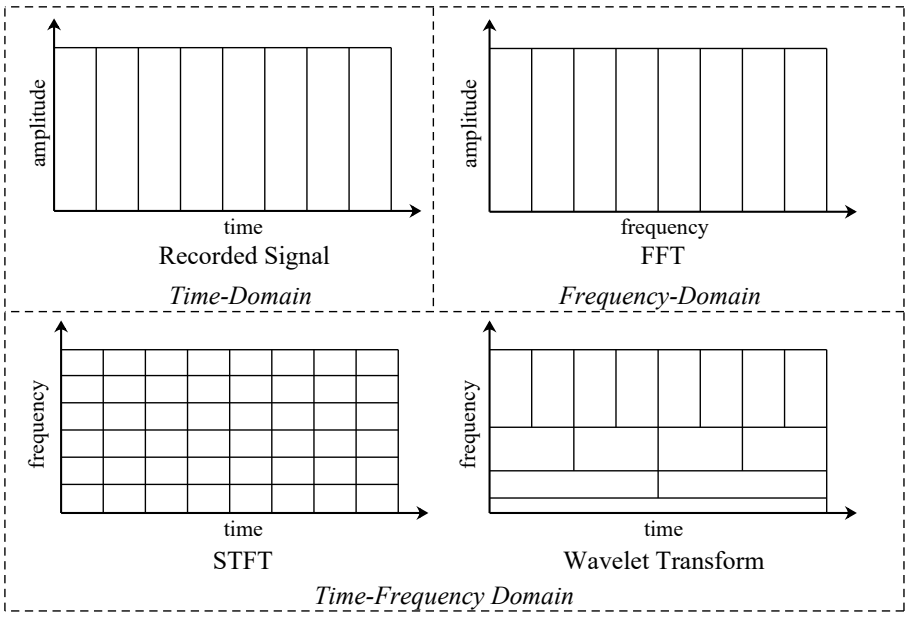


Figure 3.1. Tiling diagrams for different domain representations of a signal. The time domain contains no information regarding frequency, while the frequency domain contains no time information. For time-frequency representation, STFT provides a fixed resolution at all times, while the wavelet transform provides a variable resolution.

longer durations. The transformed signal is a function of the scale and translation parameters; the transforming function is known as the mother wavelet. Two examples of a wavelet function are the Mexican hat wavelet and the Morlet wavelet. The wavelets act as the window function and are scaled (dilated or compressed) and shifted versions of the mother wavelet.

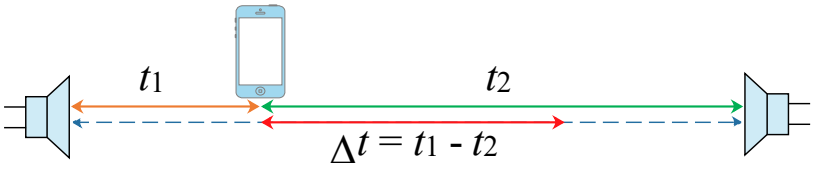
One aspect of wavelet analysis that requires consideration is the choice of a wavelet function. Factors to be considered when choosing a wavelet function include a complex or real wavelet function, wavelet width and the wavelet shape [65]. A complex wavelet function returns both amplitude and phase and is better suited for recognising oscillatory behaviour. A real wavelet function captures only a single component and is used to detect peaks and discontinuities. A narrow wavelet function will provide good temporal resolution but poor frequency resolution, while a wider function will have poor temporal resolution but good frequency resolution. Wavelet shape depends on the features present in the time-series and should reflect these features [65].

The approach previously proposed [11] used the STFT to filter the recorded signal around the pulse

frequencies. As discussed, by making use of the wavelet transform, improved temporal resolution is achievable.

3.2.3 Audio pulse design

The frequency of audio pulses played over the vehicle stereo are chosen such that these pulses are not easily noticeable to the driver and fall in a frequency range where noise (e.g. music, wind, engine sound) has little effect. Pulse frequencies of 16 kHz and 18 kHz were chosen. A sequence for playing pulses is followed; at first, a pulse of 16 kHz is played on only the left channel; this is followed by an interval of silence across both channels; a pulse of 18 kHz is then played on only the right channel. The duration of both pulses and that of the interval of silence are known and it is therefore possible to determine the arrival time of pulses at the phone's microphone. The time of arrival is used to calculate a distance and location estimate. An illustration of how the difference in pulse arrival time is calculated is shown in Figure 3.2. The time at which the pulse from the left channel arrives is denoted by t_1 and the time at which the pulse from the right channel arrives is denoted by t_2. The difference between t_1 and t_2 is denoted by Δt. The sign and magnitude of Δt determine phone distance and position.



if $\Delta t > 0$: Phone on right side of vehicle
if $\Delta t = 0$: Phone in middle of vehicle
if $\Delta t < 0$: Phone on left side of vehicle

Figure 3.2. Illustration of how difference in arrival time between pulses is calculated.

3.2.4 Implementation of audio ranging phone localisation

Audio ranging localisation algorithms are implemented in Matlab. A diagram of program flow is shown in Figure 3.3. At first, audio recorded from the phone microphone is imported. The exact onset of audio pulses in the signal is unknown, therefore recorded audio is segmented and analysed sequentially to determine the pulse presence in each segment. Audio segments are filtered by performing a continuous wavelet transform (CWT); only frequencies closest to the target pulse frequencies are considered. After

filtering, approximate pulse regions are identified and extracted. The first pulse region is extracted by locating the highest peak in the transformed signal. The remaining pulse region is located using the known sample interval between pulses. Pulse onset is identified once both pulses have been located and a region of interest has been extracted. The exact pulse onset is identified using a change point detection (CPD) algorithm. The actual time difference between pulses, which is a known constant, is compared with the calculated value to determine a location estimate. The sign and magnitude resulting from the difference calculation between pulse arrival times indicate the distance the phone is located from the centre of the vehicle. A distance value can be calculated because the speed of sound is known.

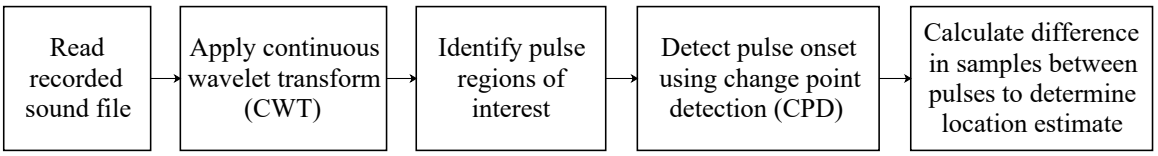
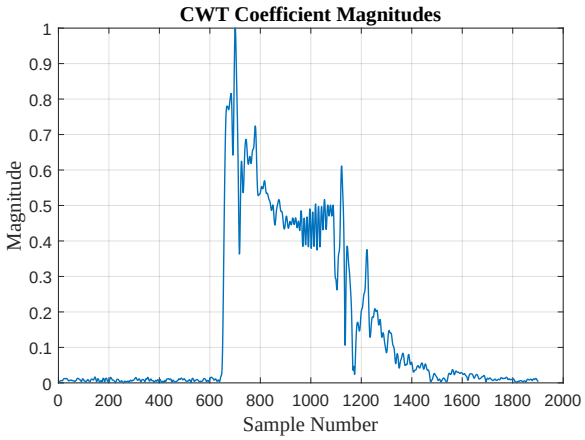


Figure 3.3. Flow diagram of application implementation.

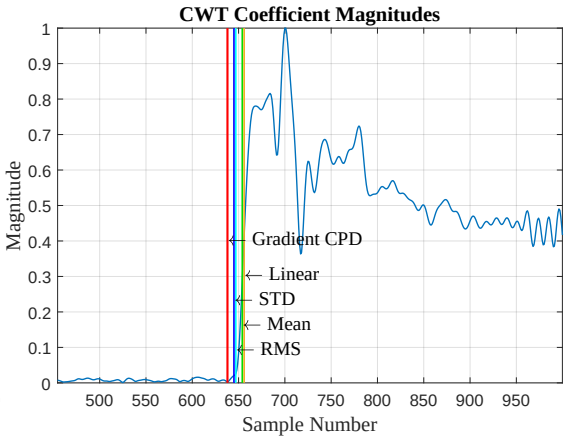
3.2.5 Change point detection implementation

CPD can be regarded as the problem of finding abrupt changes in data when a certain aspect of the time series data has been altered [66]. Unsupervised methods are attractive because no prior training data is required; typically statistical features of the data are used for detection. The choice of statistical features in the built-in Matlab function include mean, which detects changes in the mean; root mean square (RMS), which detects changes in the RMS level; standard deviation (STD), which detects changes in the STD using the Gaussian log-likelihood and linear [67], which detects changes in the mean and slope [68]. Results obtained by calculating the onset of pulses using these built-in algorithms were not satisfactory. A CPD scheme specifically designed to detect the onset of the unique pulse signal was developed. Because the pulse signal waveform has distinguishable characteristics, it is more beneficial to design a CPD algorithm satisfying the specific needs of the intended application.

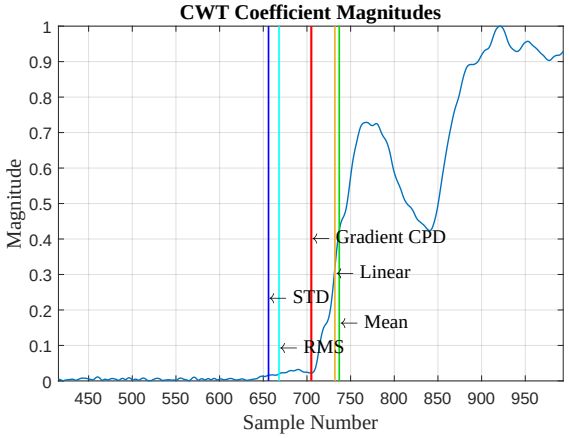
Figure 3.4 shows examples of CPD locations for different methods. The built-in Matlab CPD methods shown in the figures include linear, STD, mean and RMS, while the tailored CPD scheme is referred to as gradient CPD. Only the CWT coefficient magnitudes relating to the target pulse frequencies are shown. The magnitudes relating to other frequencies are omitted, as they are unimportant. Figure 3.4(a) shows an ideal pulse waveform with very little noise present. Figure 3.4(b) shows an enlarged



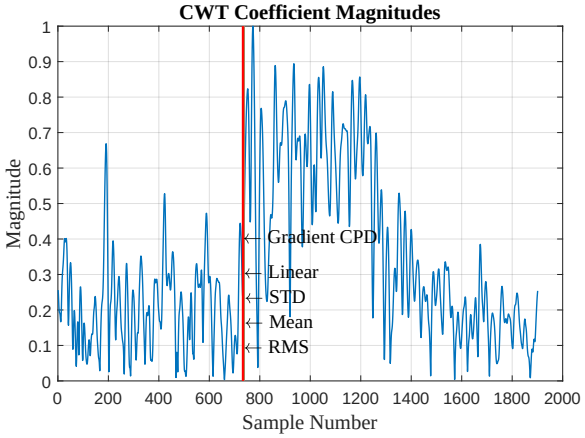
(a) Ideal pulse waveform, without change points.



(b) Ideal pulse waveform from 3.4(a) enlarged with change points indicated.



(c) Onset locations for different detection methods.



(d) Pulse waveform with a great deal of noise before signal onset.

Figure 3.4. Examples of pulse signal waveforms, with the location of different CPD methods indicated. Gradient CPD: red line, Linear: yellow line, STD: blue line, Mean: green line, RMS: cyan line.

version of the same waveform, with the change points indicated. In this case Matlab's RMS and STD features perform similarly to the custom scheme, where the change point is correctly identified; mean and linear features detect signal onset too late. Figure 3.4(c) shows an example where only the newly designed gradient CPD method detects the correct pulse onset. Other methods detect it either too early or too late. Figure 3.4(d) shows an example with a high noise level before signal onset. All detection methods correctly identified the start of the pulse.

3.2.6 Gradient change point detection

The newly developed gradient CPD algorithm calculates the area under the curve at various horizontal signal intercepts and selects the intercept point with the largest relative area. The area under the curve is calculated at a fixed width. Relative area refers to the fact that each successive area is divided by the previous area calculation. The change point is derived from the intercept with the largest relative area. Once an intercept point is chosen, the signal slope is traced downwards until the signal gradient is close to zero. This point is the most likely start of a signal pulse, as there is a rapid increase in the waveform slope. Figure 3.5 illustrates this process visually. No filtering around target pulse frequencies is applied, as this suppresses the desired sharp increase in signal waveform. A performance comparison between gradient CPD and Matlab's built-in CPD methods is presented in the results section.

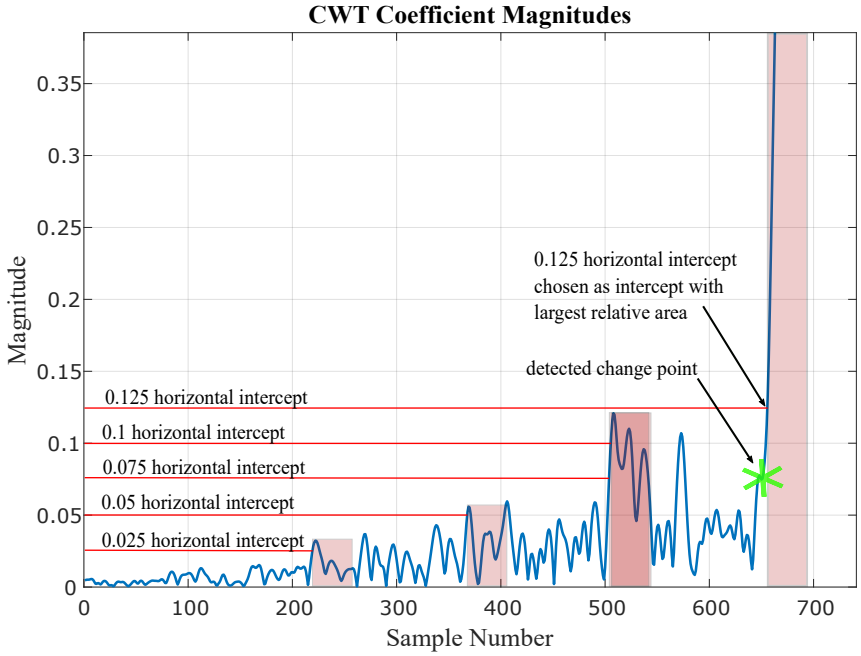


Figure 3.5. Illustration of gradient CPD.

3.3 PHONE INERTIAL LOCALISATION

A phone localisation system using embedded phone sensors was developed. It aims to determine the direction of movement and distance of a phone when it is removed from a known reference point inside a vehicle. This provides an indication of movement towards the driver's or passenger's side of the vehicle. The objective of localisation is to determine whether a phone is in the driver's area and

therefore possibly being used while driving. Phone inertial tracking provides a localisation method that does not require complex additional infrastructure to be installed in the vehicle; only the user's phone and a stationary reference point are used.

Accurate phone position estimation with an IMU unit is difficult to achieve because of the considerable drift caused by sensor errors that need to be compensated for. Algorithms previously used for position tracking and estimation in pedestrian and pen-tracking applications can be employed in the in-vehicle environment. The main components used include motion identification and segmentation, zero velocity updates and compensating for drift caused by integration. Sensor errors and error propagation in INSs have a significant influence on the accuracy of the estimates that can be achieved. The phone has to be tracked from a known reference point to determine the direction of travel. A magnetic phone holder is used as a reference; the magnetometer found in a phone's embedded IMU identifies when a large magnetic field is present. A large magnitude is measured in the magnetic field vector when the phone is close to the reference location. This magnitude reduces as the phone moves further away from the reference location, such as when it is picked up. It is assumed that a magnetic phone holder will be placed in a known location inside the vehicle and that the phone will be placed on the mount while driving.

3.3.1 Proposed phone inertial localisation approach

To date, no known inertial tracking method has been developed that tracks a user's phone relative to a reference point in a vehicle. An approach that positions a phone with respect to an accelerometer that is vehicle-fixed was previously proposed [69]. This approach requires measurements from at least two phones or one phone and an external vehicle-fixed accelerometer. The approach proposed here uses a single phone and a physical accessory in the form of a magnetic phone holder as a reference. Although none of the previous position tracking methods has been directly applied to estimate phone position in a vehicle, the algorithms and techniques used could still be adapted and prove useful for the intended application. In the proposed approach, inertial motion tracking is performed, which refers to the use of embedded phone accelerometers and gyroscopes to determine the position of a phone inside a vehicle with respect to a reference frame.

Inertial sensor alignment in the case of phone position estimation refers to the process through which

the orientation of phone sensor axes is determined relative to the reference axis system. The reference axis system is the space in which phone motion takes place; in this case, it is the inside of a vehicle or the local geographic frame. It is important to determine the initial attitude of the phone in the alignment process and to initialise the velocity and position to appropriate values (the velocity is set to 0 m/s and the position is set to the origin). The goal of the angular alignment procedure is to calculate the quaternion parameters or direction cosine matrix that characterises the relationship between inertial sensor axes and the chosen reference frame (e.g. local geographic frame).

Inertial sensors are mounted similar to a strapdown system, but instead of being mounted directly on the vehicle, these are only semi-fixed because the phone can be removed from the fixed platform at any instant. Alignment in the vehicle takes place while the phone is semi-fixed to the platform, meaning that the phone position will not change unless it is removed from the platform. This means that alignment can be treated as if it were done on a fixed platform. Full calibration of phone sensors is not performed, as it requires several rotations and movements using specialised equipment. It is not practical for each user to calibrate the specific phone before the system can be used. The proposed approach will aim to achieve acceptable motion tracking without calibrating phone sensors first. While no calibration is performed, the attitude and heading reference system (AHRS) is still initialised and allowed to converge.

Phone inertial position tracking will not make use of a full INS, since full inertial tracking of a vehicle is not required. However, the phone's embedded inertial measurement unit and an attitude and heading reference system will be used to compute the direction and distance of phone travel to provide a rough position estimate. The building blocks for phone inertial movement tracking are shown in Figure 3.6.

Phone movement with respect to its initial position in the reference frame is calculated. Phone direction of movement is divided into a 3D grid system, where the phone is classified as moving into a particular octant section. Octant numbering with the corresponding coordinate axes signs is shown in Figure 3.7. A typical phone pick-up would translate into moving the phone from the origin into one of the eight octant sections. A reference point located somewhere on the centre console would realistically mean that the phone could only move into octant sections numbered 3, 4, 7 and 8. Movement into sections 3 or 4 would indicate a driver pick-up as the phone moved to the right, while movement into sections 7 or 8 would indicate a passenger pick-up as the phone moved to the left.

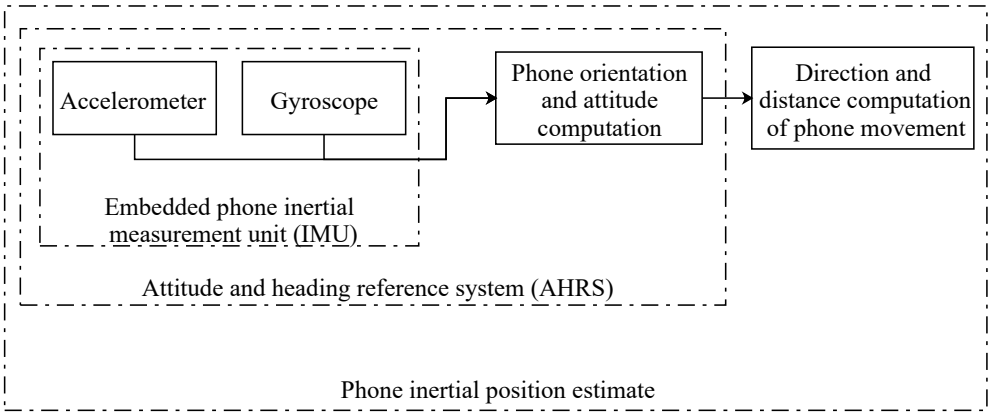


Figure 3.6. Phone inertial tracking building blocks.

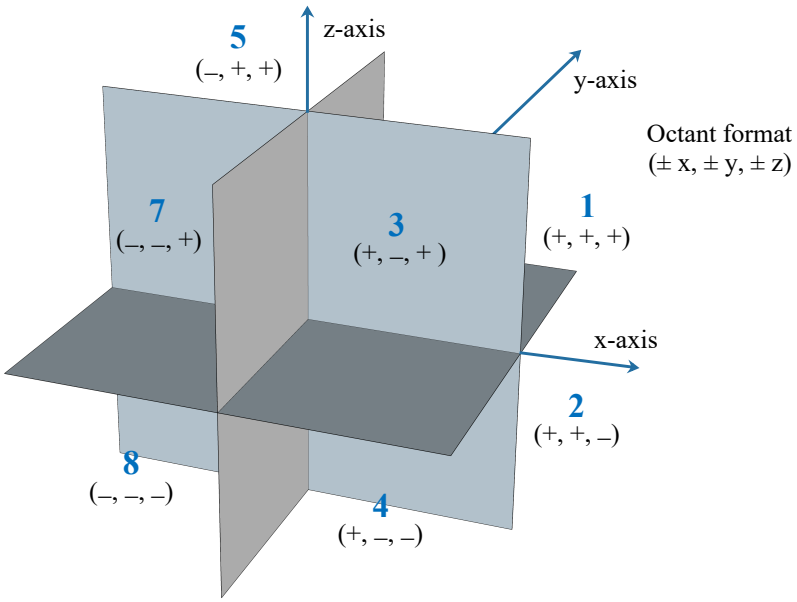


Figure 3.7. Octant numbering with corresponding coordinate axes signs.

An essential component of phone motion tracking is having an accurate orientation estimate of phone sensors. This is required to ensure that acceleration due to gravity is removed from each of the three axes, producing a true linear acceleration estimate. Integrating phone gyroscope measurements provides information about the orientation of the phone and its sensors. After subtracting gravity, acceleration can be double integrated to obtain a position estimate for the phone. To subtract gravity, the orientation of the phone needs to be known. Phone orientation indicates the gravity contribution to each accelerometer axis. Phone orientation and position estimation are closely linked and without an accurate orientation approximation, the position estimate will suffer. Figure 3.8 shows the process of

using inertial sensors to determine a position estimate for the phone. Even small errors in measured acceleration and angular velocity have a large impact on the calculated phone orientation and position, particularly when only using inertial sensors. Inertial sensors are often supplemented with additional sensors and models to improve orientation and position estimates. In the proposed approach only the embedded phone inertial sensors are used; no signals are received from external sources.

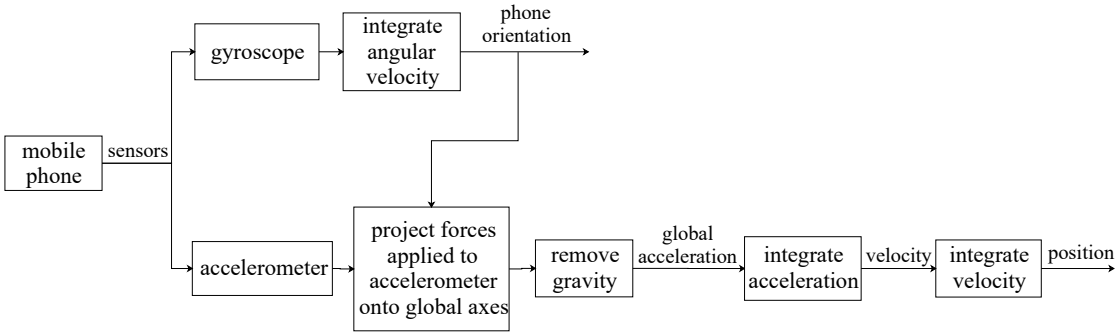


Figure 3.8. Phone inertial dead-reckoning.

Several essential steps in the proposed approach need to be completed for accurate phone position estimation in the vehicle. A magnetic pad is used as a reference point in the vehicle; whenever the phone is lifted from the pad, motion tracking is initiated. The first step is identifying the removal of a phone from the magnetic pad. This starts the motion tracking process. Removal of the phone from the magnetic pad is detected using the phone's magnetometer, which measures magnetic field strength. Considering the phone is semi-fixed to the magnetic pad, a large magnitude in magnetic field strength is measured when the phone is near the pad. The magnitude of measured magnetic field strength reduces significantly as the phone moves further away from the magnetic pad. A low magnitude signifies that the phone has been removed from the reference point and needs to be tracked. An upper and lower magnitude threshold is used to detect the phone being picked up and put down from the magnetic pad. Upper and lower threshold values of 500 μT and 200 μT are used respectively, these values were determined empirically.

The system originally made use of zero-velocity updates that would reset the phone's velocity to zero during periods of little acceleration to reduce the effect of drift. Discontinuities created by forcing the velocity signal to zero would then be compensated for. Use of zero-velocity updates with compensation during pick-up was later removed from the algorithm. It has the undesirable effect of changing velocity directions during the pick-up motion, which results in incorrect direction classifications. Instead, the

phone is tracked for a user-defined period (e.g. 0.8 s) after being lifted from the magnetic pad. Drift error in measurements during this short period is not enough to affect tracking results negatively. Thus, initial phone motion is used to classify the direction and distance of phone travel. Phone sensor data (accelerometer and gyroscope measurements) are processed through the AHRS algorithm to compute a phone orientation estimate. Phone acceleration is rotated from the sensor frame into the local navigation frame. Gravity is removed from accelerometer readings to obtain a linear acceleration approximation. Gravity is removed by either subtracting the gravity constant, which in the earth's case is 9.81 from the Z-axis, or high-pass filtering re-oriented readings from the Z-axis to remove the DC value. The filter method was chosen, as it produces the best results. The linear acceleration approximation is integrated to yield phone velocity and phone velocity is integrated once again to produce the phone position for each of the three sensor axes. The progression of accelerometer measurements from raw phone output to a position estimate is shown in Figure 3.9. A driver or passenger phone pick-up classification is made by observing the direction of travel in different coordinate planes. Phone movement in each of the three coordinate planes is shown in Figure 3.10. The combined distance travelled in all three axes is also shown. The sign of axes indicates the octant section (see Figure 3.7) into which the phone moved, while the position vector magnitude provides an approximation of the distance travelled inside the octant section.

3.3.2 AHRS algorithm

Accelerometers have good static features, but poor dynamic features, meaning that measurements are noisy, but accurate over long periods. Gyroscopes have good dynamic features but poor static features, meaning that measurements are accurate over short periods, but drift over time. When estimating orientation, accelerometers have desirable low-frequency properties, while gyroscopes have desirable high-frequency properties. An AHRS algorithm is required to combine the measurements from different sensors and minimise the drawbacks of each. The AHRS is essential in the estimation of phone position, as the effect of gravity first needs to be removed before linear acceleration can be acquired. Two popular filters often used in AHRS algorithms are Kalman and complementary filters.

A particular Kalman filter, the extended Kalman filter (EKF), is frequently used to calculate state estimates in non-linear state-space models. It is also applicable in the application of orientation

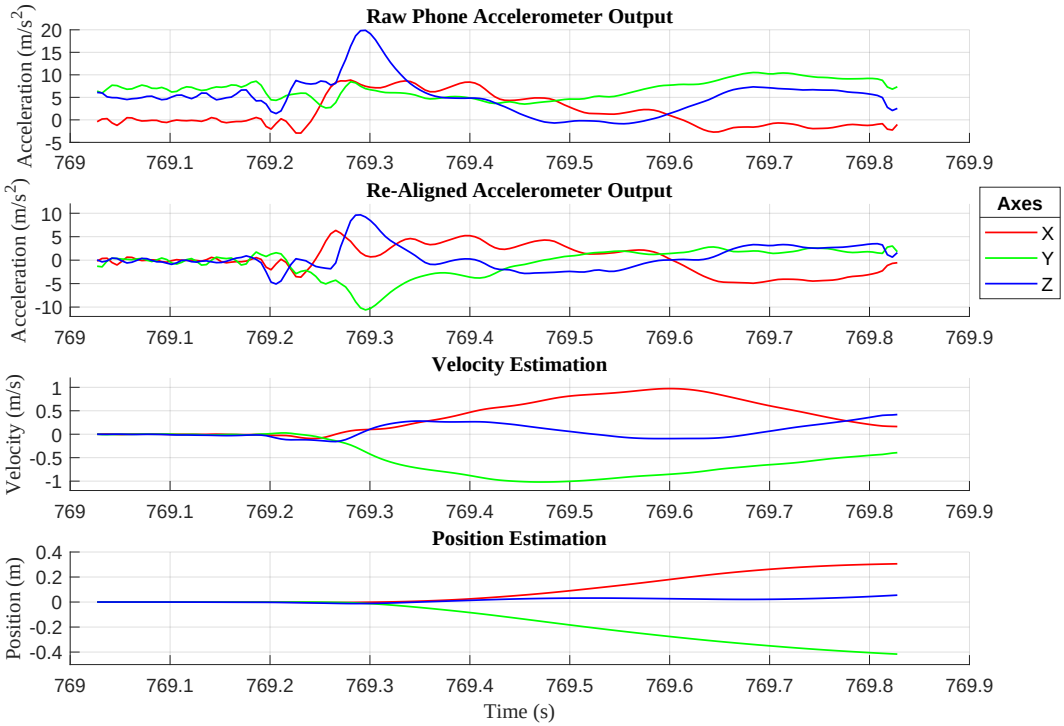


Figure 3.9. Progression of phone accelerometer measurements into a position estimate.

estimation. An alternative to using EKFs for estimating orientation is the complementary filter. Complementary filters rely on the fact that both accelerometers and gyroscopes provide sensor orientation information. Because accelerometers have desirable low-frequency properties, their measurements can be filtered using a low-pass filter to isolate these features. Similarly, because gyroscopes have desirable high-frequency properties, their measurements can be high-pass filtered to isolate these features. Complementary filters make use of the best properties of each sensor to provide a more accurate orientation estimate. Two well-known complementary filters are the gradient descent-based complementary filter [70] and the explicit complementary filter [71]. Both algorithms are available online [72]. A performance analysis that compares orientation estimation by EKFs and complementary filters was completed on experimental and simulated data [22]. It was found that orientation estimates are expected to be more accurate in the pitch and roll than the heading. All the tested algorithms were able to produce good orientation estimates. Both the EKF and complementary filter were found to have linearisation issues when large corrections to orientation estimates are required.

The gradient descent-based complementary filter [70] is chosen for phone orientation estimation in the

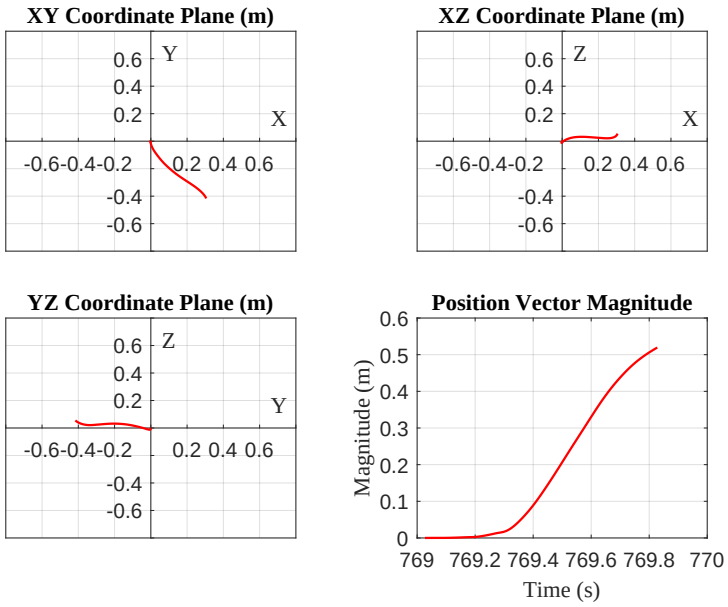


Figure 3.10. Coordinate planes used to calculate phone direction and distance of travel.

proposed approach. It allows for much simpler implementation, a reduced sampling rate and simpler parameter tuning compared to a Kalman-based solution [73]. Equations used to describe the filter and its operation were previously provided [73]. The most important equations are selected and presented. A quaternion describing an orientation ${}^{A}_{B}\hat{\boldsymbol{q}}$ can be defined by

$$ {}^{A}_{B}\hat{\boldsymbol{q}} = [q_1 \quad q_2 \quad q_3 \quad q_4], \tag{3.1} $$

where the notation of leading super- and sub-scripts denotes the relative frames of orientation. A notation of ${}^{A}_{B}\hat{\boldsymbol{q}}$ describes the orientation of frame B relative to frame A. The quaternion product, denoted by $\otimes$, can be used for compound orientations.

The measurements obtained from a tri-axis gyroscope can be arranged into the vector ${}^{S}\boldsymbol{\omega}$ described by

$$ {}^{S}\boldsymbol{\omega} = [0 \quad \omega_x \quad \omega_y \quad \omega_z]. \tag{3.2} $$

Accelerometer measurements can be arranged into the vector ${}^{S}\hat{\boldsymbol{a}}$ described by

$$ {}^{S}\hat{\boldsymbol{a}} = [0 \quad a_x \quad a_y \quad a_z]. \tag{3.3} $$

The filter uses a quaternion representation, where the orientation of the sensor, ${}^{S}_{E}\hat{\boldsymbol{q}}$, is defined by a quaternion such as the example

$$ {}^{S}_{E}\hat{\boldsymbol{q}} = [q_1 \quad q_2 \quad q_3 \quad q_4]. \tag{3.4} $$

Equations relating to orientation calculation from angular rate (gyroscope) and vector observations (accelerometer) are described. The equations relating to the fusion of orientation calculations are also shown.

3.3.2.1 Gyroscope

Orientation of the earth frame relative to the sensor frame at time t, ${}^{S}_{E}\boldsymbol{q}_{\omega,t}$ can be calculated using equations

$$ {}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t} = \frac{1}{2} {}^{S}_{E}\hat{\boldsymbol{q}}_{est,t-1} \otimes {}^{S}\boldsymbol{\omega}_t \quad \textit{and} \tag{3.5} $$

$$ {}^{S}_{E}\boldsymbol{q}_{\omega,t} = {}^{S}_{E}\hat{\boldsymbol{q}}_{est,t-1} + {}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t}\Delta t, \tag{3.6} $$

where the variable ${}^{S}\boldsymbol{\omega}_t$ represents the angular rate at time t, Δt is the sampling period and ${}^{S}_{E}\hat{\boldsymbol{q}}_{est,t-1}$ is the previous orientation estimate.

3.3.2.2 Accelerometer

The variable ${}^{S}\hat{\boldsymbol{a}}_t$ represents the measured acceleration at time t. The gradient of the solution surface and its Jacobian can be calculated using equations

$$\nabla \boldsymbol{f} = \boldsymbol{J}_g^T({}^{S}_{E}\hat{\boldsymbol{q}}_{est,t-1})\boldsymbol{f}_g({}^{S}_{E}\hat{\boldsymbol{q}}_{est,t-1}, {}^{S}\hat{\boldsymbol{a}}_t), \tag{3.7}$$

$$\boldsymbol{f}_g({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{S}\hat{\boldsymbol{a}}) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 - q_3q_4) - a_y \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - a_z \end{bmatrix} \textit{ and} \tag{3.8}$$

$$\boldsymbol{J}_g({}^{S}_{E}\hat{\boldsymbol{q}}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix}. \tag{3.9}$$

3.3.2.3 Filter fusion algorithm

An orientation estimate of the sensor frame relative to the earth frame, ${}^{S}_{E}\boldsymbol{q}_{est,t}$ is obtained using equations

$${}^{S}_{E}\boldsymbol{q}_{est,t} = {}^{S}_{E}\hat{\boldsymbol{q}}_{est,t-1} + {}^{S}_{E}\dot{\boldsymbol{q}}_{est,t}\Delta t, \tag{3.10}$$

$${}^{S}_{E}\dot{\boldsymbol{q}}_{est,t} = {}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t} - \beta {}^{S}_{E}\hat{\boldsymbol{q}}_{\varepsilon,t} \quad \textit{and} \tag{3.11}$$

$${}^{S}_{E}\hat{\boldsymbol{q}}_{\varepsilon,t} = \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|}, \tag{3.12}$$

where the variable ${}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t}$ refers to the rate of change of orientation measured by the gyroscope, β is the filter gain rate and ${}^{S}_{E}\hat{\boldsymbol{q}}_{\varepsilon,t}$ is the estimated error.

The filter gain, β, represents gyroscope measurement errors that are zero mean. The error sources could include calibration errors, sensor noise, sensor misalignment and sensor non-orthogonality. Based on previous experimentation [73], it was determined that the optimal value for β in their application is 0.033, with an initial value of 2.5 used during the initialisation period to ensure filter

convergence. Using higher filter gains during the initialisation period could improve filter convergence from initial conditions. The minimum acceptable value for β is determined by gyroscope measurement errors.

A filter gain of 0.5 was found to work best with the proposed application. The initial filter gain value was set to 5 to allow for filter convergence. The filter is allowed to converge using the first few seconds of phone sensor measurements.

3.4 CONVOLUTION NEURAL NETWORK-BASED PHONE USE DETECTION

Identification of driver phone use is an image classification problem, as the different driving behaviours relating to phone use need to be classified. Traditional image classification systems would usually implement hand-engineered algorithms for feature extraction. CNNs are chosen because they operate as end-to-end models, meaning features are automatically learned internally to distinguish between classes. Focus is placed on detecting drivers talking on their phones and texting, but the system could be extended to track additional driving-related behaviour, such as eating, drinking or smoking. Previous [6, 7, 9] vision-based phone use detection methods focused only on identifying driver's talking on their phones. Design of an effective network requires selection of a suitable architecture and hyperparameters.

3.4.1 CNN architecture

Network architecture is vital in ultimately determining model accuracy. Network capacity needs to be chosen such that the underlying features of classes can be learned, while still obtaining good performance on images it was not trained on. Some of the most popular architectures designed for image classification of the ImageNet challenge [60] are AlexNet [74], VGG [75], Residual Networks (ResNet) [76] and GoogLeNet [77]. These architectures have been trained on many images from 1000 object classes. Pre-trained models with these network architectures have learned a vast array of discriminating features.

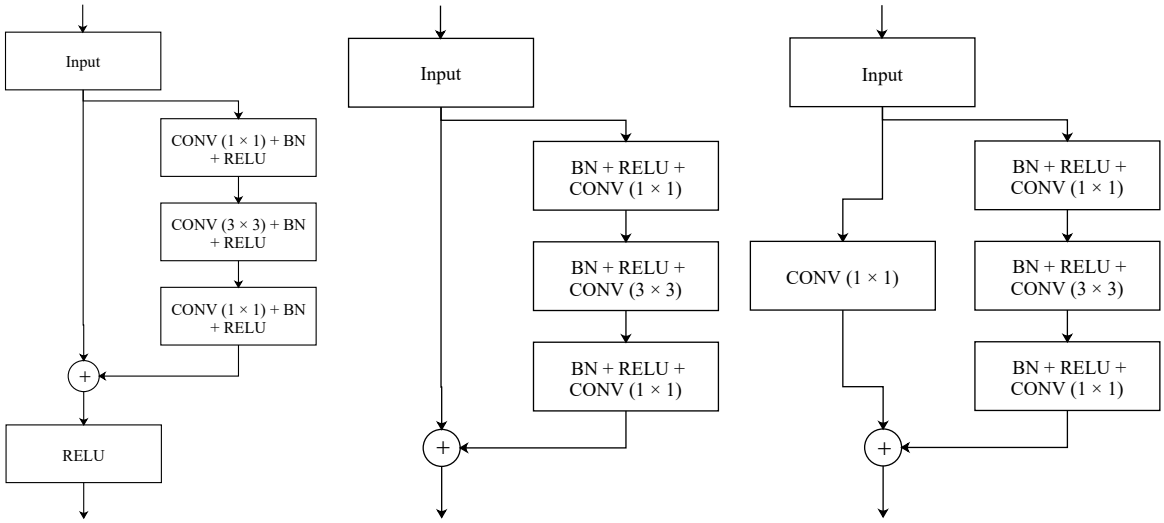
3.4.1.1 Residual Networks

Current state-of-the-art CNNs tend to make use of micro-architectures that act as building blocks and fit into the overall macro-architecture of the network. The inception and residual modules used by GoogLeNet [77] and ResNet [76] respectively are examples of micro-architectures. Micro-architectures allow networks with greater depth to train faster and more efficiently. ResNet makes limited use of max pooling operations to reduce spatial dimension; instead it uses convolutions with strides that are greater than one. The full implementation of ResNet only has one max pooling layer at the start of the network to reduce spatial dimensions.

The residual module relies on identity mappings, which refers to the process of adding the original input to the output of a series of operations. Different types of residual modules are shown in Figure 3.11. The bottleneck residual module is shown in Figure 3.11(a). It has two branches, which split from the input. No processing on input data is performed on the left branch; it connects directly to an addition operation at the bottom of the module. The right branch of the module performs a series of convolutions, batch normalisations and activations before being added to the original input. The first and last convolution layers use filters of size 1×1, while the second convolutional layer uses filters of size 3×3. The number of filters learned in the first two convolutional layers is a fourth of those learned in the final layer. Volume size is reduced during the first 1×1 and 3×3 convolutional layer. The final 1×1 convolutional layer applies four times the number of filters, which increases spatial dimensions once more. The residual framework enables the training of networks with much greater depth [58]. A more optimal layer-ordering scheme that provides additional accuracy was found and is called pre-activation [76]. The residual module with pre-activation is shown in Figure 3.11(b). The activation at the very end of the module is removed and the batch normalisation and activation layers on the right branch are re-ordered to appear before the convolutional layer. Each of the convolutional layers uses a stride of 1. The final output is the addition operation. The final output has the same volume size as the original input. Another variation of the residual module that reduces the dimensionality of the volume is shown in Figure 3.11(c). A new convolutional layer is added to the left branch. Dimensionality is reduced by setting the stride of the new convolutional layer and the second convolutional layer on the right branch to 2. This variant of the pre-activation module is usually placed before several consecutive normal pre-activation modules.

A newly defined variant of ResNet is implemented to train the network on the collected dataset. The

trained network will attempt to distinguish accurately between classes of no phone use, talking on a phone and texting.



(a) Bottleneck residual module.

(b) Pre-activation residual module.

(c) Residual module with dimensionality reduction.

Figure 3.11. Different types of residual modules. CONV refers to a convolutional operation with filter size specific in brackets, BN refers to a batch normalisation operation and RELU refers to a rectified linear unit activation function.

3.4.2 CNN hyperparameters

Hyperparameter options include the selection of a loss function with a specified amount of regularisation penalty. An optimisation algorithm and associated learning rate need to be chosen. The number of epochs that the network will train has to be selected, ensuring that overfitting and underfitting do not occur. Regularisation techniques that allow the network to generalise to images outside the training set also have to be considered.

3.4.2.1 Loss function

Two of the most common loss functions currently being employed are cross-entropy loss used by the Softmax classifier and hinge loss used by the multi-class SVM classifier. Previously the use of mean squared error was popular, but cross-entropy loss greatly improved the performance of models with

Softmax and sigmoid outputs. These models suffered from slow learning rates when used with mean squared error [41]. A benefit that the Softmax classifier has over SVM is that it provides probabilities for each class label, while hinge loss used by the SVM classifier provides the margin between class labels. This makes the output of Softmax classifiers easier to interpret. The Softmax classifier with cross-entropy loss uses a scoring function f to map data points, x_i, to output class labels, y_i. The scoring function f is defined by

$$f(x_i, W) = Wx_i\,, \tag{3.13}$$

where x_i is the i-th data point and W is the weight matrix with bias term included. The scoring function can be abbreviated to s and the score obtained for the j-th class via the i-th data point is

$$s_j = f(x_i, W)_j\,. \tag{3.14}$$

Cross-entropy loss for a single data point is defined by

$$L_i = -log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right), \tag{3.15}$$

where the logarithm is base e. The cross-entropy loss for the entire dataset is computed as

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i\,. \tag{3.16}$$

Cross-entropy loss is selected as the loss function to be used during network training. The output provided by the Softmax classifier in the form of probabilities makes interpretation of network confidence in a class label easier. If the loss function is used as is, a possible problem might arise. There could be many similar weight matrices (W) for which the model correctly classifies the training set. The solution is to add a regularisation penalty, $R(W)$, that discourages individual weights from becoming too large. The most common regularisation penalty used is L2 regularisation (weight decay) [59]. Cross-entropy loss with L2 regularisation included becomes

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i + \lambda \sum_i \sum_j W_{i,j}^2\,, \tag{3.17}$$

where λ is the regularisation strength. It controls the magnitude by which weights are allowed to grow. L2 regularisation or weight decay is applied in the network to reduce overfitting and improve the accuracy of the validation and testing sets.

3.4.2.2 Optimisation algorithm

The choice of optimisation algorithm is important, as it will be a large factor in determining network performance regarding the loss and accuracy achieved. Gradient descent is the most common method used for minimising loss. The standard gradient descent algorithm only updates network weights once per epoch, thus causing slow convergence. All training examples also have to be loaded into memory to compute the gradient, which might not be possible for large datasets. Stochastic gradient descent (SGD) is a modification of the standard method, where weights are updated multiple times per epoch. Training data are split into batches with weight updates applied to each of the batches. SGD is one of the optimisation algorithms used in network training. Gradient descent algorithms are controlled by the learning rate parameter; it determines what the step size will be in the direction of the gradient. A learning rate that is too high will cause weight updates to be erratic in the loss landscape and the network will not be able to learn any patterns. If the learning rate is too low, it will take an excessive number of iterations to reach an acceptable loss value. Other frequently used optimisation algorithms include an adaptive learning rate (Adadelta) [78], adaptive gradient (Adagrad) [79] and Adam [80].

3.4.3 Collection and preparation of dataset

An essential component that needs consideration before any network training process can take place is the collection of appropriate image data that are representative of the classification task. Initially, network training was done to distinguish only between a normal face (representing the no phone use class) and a person talking on a phone. Images of faces were taken from face databases such as the colour FERET database [81], siblingsDB [82] and University of Essex Face Recognition Data [83]. Examples of faces from the colour FERET database are shown in Figure 3.12. Image databases of people talking on a phone are not freely available, so these images had to be gathered manually. Images of people talking on a phone were gathered from Getty images. These images were saved and cropped individually, making it a very time-consuming process. Models trained on these particular datasets could not generalise to new image data and performed poorly. This was because there was a training data versus real-world data mismatch. While images of people talking on phones had varying backgrounds around the face region, images of the face class did not have representative backgrounds. Images from the face class were taken in a controlled environment with a single background colour

(a) Sample 1 **(b)** Sample 2 **(c)** Sample 3 **(d)** Sample 4

Figure 3.12. Colour FERET database [81] samples.

and no clutter around the face region. The images do not mimic conditions that would be found when testing the actual system. An additional test was completed where faces from people talking on phones were cropped into two halves (i.e. images were split down the middle of the nose). The one half contained the side of the face with a phone up against the ear, while the other half contained the opposite side of the face with no phone, forming the second class. This formed two classes that included real-world backgrounds and improved model performance, as the model had better generalisation ability. A downside of using this approach is that both the left and right sides of the face have to be classified.

These experiments show the importance of having data that are representative of the classification task. Without the correct image data, the model cannot generalise and perform accurately on new data from real-world conditions. New images had to be collected that accurately represent conditions that would be present when testing the model in the real world. An additional dangerous form of driving behaviour that involves the use of a phone while driving was added to the two established classes. It detects possible driver texting. Texting could also be detected by tracking the driver's gaze, but this would require algorithms specific to gaze tracking to be implemented. The inclusion of a third texting class in the network provides a simpler solution. Detecting driver phone use can be described as behaviour classification and not object classification, because specific driving-related behaviours are identified rather than objects. It is a fine-grained classification task, as there is little distinction between image classes. All image classes contain a person's face, the only difference being variations in behaviour that the person exhibits. Because of the similarity between classes, the machine-learning

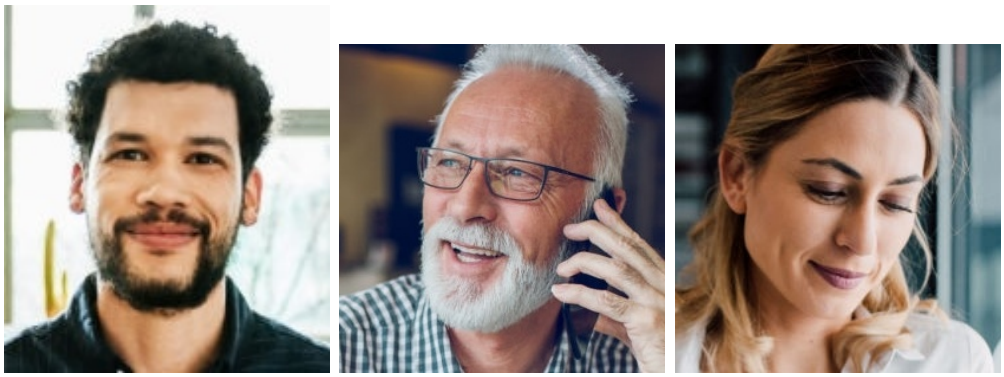
(a) Person standing. **(b)** Person talking on a phone. **(c)** Person texting.

Figure 3.13. Sample of images collected for different classes from Getty Images.

model needs to learn extremely discriminating features to be able to distinguish between classes. A sufficient amount of image data is required for training and testing. Insufficient data would cause the network to overfit more easily, thus negatively affecting its generalisation ability.

Gathering enough image data for each class by individually saving and pre-processing the images would require a significant amount of time. Some steps involving the collection and pre-processing of image data into the appropriate form were automated. An image scraping tool [84] was used to download images consecutively from a list of web pages provided. Only images relevant to the three classes were stored for further processing. Images of people talking on a phone, people texting and standard faces were collected from Getty images. Additional images of normal faces were taken from the Labeled Faces in the Wild [85] database. There is a wide variety of poses and backgrounds for each class. Images of people talking on a phone with the phone held up to the right or left ear were collected. Samples of images collected for each of the three classes are shown in Figure 3.13. A total of 9 987 images were collected: 3 705 of the images are of standard faces with no phone use, 3 532 of the images are of people talking on a phone and 2 750 of the images are of people texting. These images still need to be processed by only extracting the face and a small region around the face. In previous experiments, the cropping process was done by hand, but this is unfeasible owing to the large amount of new image data. A script is used to automate the cropping process. The result of only extracting the region of interest from the samples is shown in Figure 3.14. The script first identifies the face; the face together with a user-specified region around the face is then extracted and saved. A region around the face is included to ensure that the phone and the person's hand are included in the captured image.

All collected and processed images are stored with their corresponding class labels. The collected



(a) No phone use class. **(b)** Talking on a phone class. **(c)** Texting class.

Figure 3.14. Cropped versions of sample images collected from Getty Images.

dataset is divided into three: a training set, a testing set and a validation set. The training set contains 70% of image data, the testing set 20% and the validation set 10%. Image data are shuffled before splitting; data are also split in a stratified fashion to ensure class balance is preserved. The training set is used during network training for the model to learn the underlying patterns. Network performance during training is evaluated using the testing set. The validation set is used to find the best set of hyperparameters.

3.4.4 Training procedure

Python is selected as the programming language to develop and test trained network models. It is used in conjunction with the Keras library and TensorFlow backend. OpenCV is also used to perform various image-processing tasks. Some common steps need to be followed irrespective of the method or model before actual network training can commence. Pre-processing operations are applied to images. Operations include resizing images to the correct network input size. Images are resized while keeping the aspect ratio consistent. This is important to ensure that the visual presentation of the image is not altered. Another pre-processing operation to normalise image data is mean subtraction per image channel. The mean value of each channel (red, green and blue) is subtracted from the image to centre the data around zero. When training a pre-trained model on new image data, the images are pre-processed in the same manner as the original network. Weights and parameters learned in the network have adapted to work with image data in a particular format. The collected dataset has a class imbalance. To combat this, a weight associated with each class is calculated. Weighting increases

the per-instance loss by a larger weight when observing under-represented classes. The collected dataset has 3 705 images of faces, 3 532 images of a person talking on a phone and 2 750 images of a person texting. The class weight array is defined by $[1.0 \quad 1.048 \quad 1.347]$, meaning every instance of texting is treated as 1.347 instances of faces. For example, during training a misclassification of the texting class will result in a loss that is 1.347 times higher than a misclassification of the faces class. A model with the selected architecture and hyperparameters is compiled. A log file is used to continually monitor a plot of the loss and accuracy for both training and testing sets. This is useful to identify potential network overfitting. Finally, network training can commence for the specified number of epochs.

3.4.4.1 Pre-trained model design

The first network design involved fine-tuning pre-trained networks, which is a type of transfer learning. Pre-trained networks have learned a large number of discriminating features on the ImageNet challenge [60]. These networks can be trained to recognise classes on which they were not originally trained. The network architecture is modified to enable parts of the network to be retrained. The final set of network layers is removed and replaced with a new set of fully connected layers. The network is initialised with weights learned on the ImageNet dataset. At first, old network layers are made untrainable, meaning weights in these layers cannot be changed. Once the new set of fully connected layers has started to learn patterns, additional layers in the body of the network are made trainable at a low learning rate.

The first network architecture to be fine-tuned is VGG16 [75]. The final set of fully connected layers is removed and a new set is added after the last pooling layer. The new set of fully connected layers contains 256 nodes, followed by a dropout layer. Finally, a Softmax layer is added to output the confidence in each of the class labels. The new head is trained for 20 epochs at a learning rate of 0.001. After 20 epochs an accuracy of 94.51% is achieved on the test set. The last set of convolutional layers is made trainable and training is continued for another 15 epochs at the same learning rate. A final accuracy of 97.51% is achieved on the testing set. Network training on the VGG16 architecture is time-consuming, with each epoch taking on average 56 minutes to complete. This high time burden inhibits effective tuning of hyperparameters. Good accuracy is obtained by the network, but training and classification speed are slow.

ResNet [76] and MobileNetV2 [86] architectures trained on the ImageNet dataset are also fine-tuned. Similar to the VGG16 architecture, the head of the network is removed and replaced with a new set of fully connected layers. The fine-tuned ResNet architecture adds a pooling layer followed by a fully connected layer (256 nodes) and a Softmax layer to the head of the network. The fine-tuned MobileNetV2 architecture is the same, but a dropout layer is inserted between the fully connected layer and the Softmax layer. An accuracy of 88.93% is obtained for the fine-tuned ResNet network, while the fine-tuned MobileNetV2 network obtains an accuracy of 91.62% on the test set. Networks that have been fine-tuned using ResNet and MobileNetV2 architectures are unable to perform as well as the VGG16 architecture. This could be due to incorrect hyperparameter selection or an unsuitable head being added to the existing architecture.

3.4.4.2 ResNet variation design

Sufficient training data are available to train a newly defined network that should be able to achieve good accuracy. A variation of the full ResNet architecture ([58]) trained on the ImageNet challenge is proposed. Residual modules shown in Figures 3.11(b) and 3.11(c) are utilised in the network. The full network architecture is shown in Figure 3.15. Input to the network are images of size $224 \times 224 \times 3$. Initial layers in the network normalise the input by applying a batch normalisation layer followed by a single convolutional layer that learns 5×5 filters. The spatial dimensions of the volume are reduced through the pooling layer. Initial layers are followed by four individual components, each containing a series of residual modules that are stacked. Each component contains a single dimensionality reduction residual module, followed by several pre-activation residual modules. Stacked residual modules are followed by the network head. The volume size is reduced using a pooling layer instead of fully connected layers. The pooling layer is followed by a fully connected layer with three nodes, one for each of the class labels. Finally, a Softmax activation function is applied to output the final probabilities for each class label.

3.4.4.3 First network design iteration

The number of filters learned in each residual module component, as well as the first convolutional layer, needs to be specified. The first architecture iteration learns 64 filters in the first convolutional layer; the four residual module components learn 64, 128, 256 and 512 filters respectively. Training is

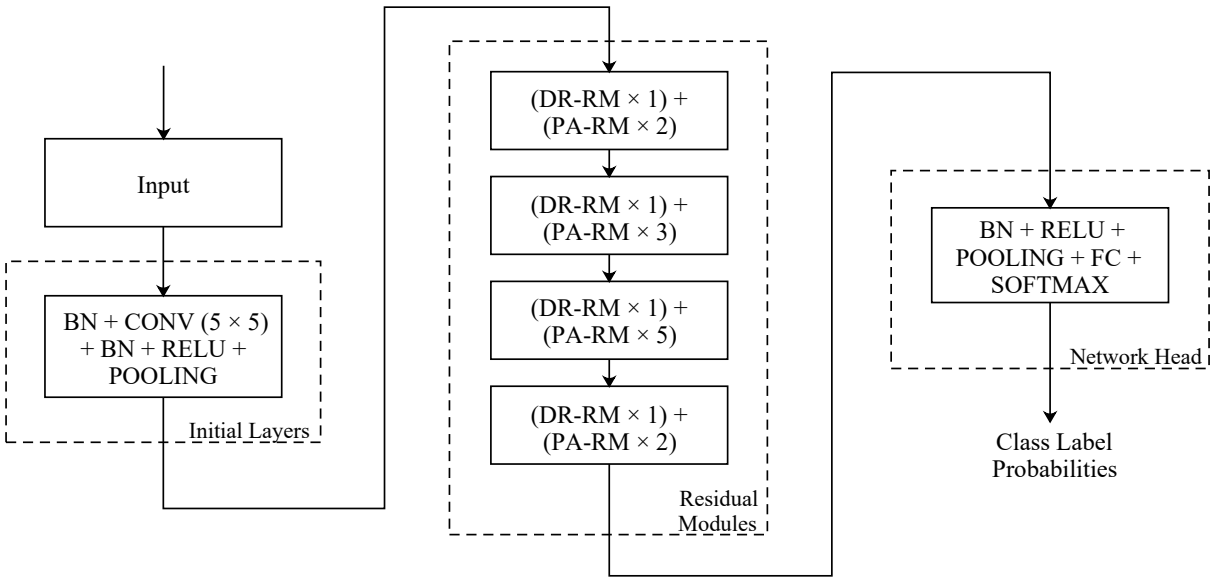


Figure 3.15. Network architecture used to train ResNet variation. CONV refers to a convolutional operation with filter size specified in brackets, BN refers to a batch normalisation operation, RELU refers to a rectified linear unit activation function and POOLING refers to a pooling operation applied to data. DR-RM is a dimensionality reduction residual module (shown in Figure 3.11(c)) and PA-RM is a pre-activation residual module (shown in Figure 3.11(b)).

started with SGD as the optimiser and a momentum term of 0.9. Appropriate adjustment of the learning rate is a vital factor in ultimately determining network performance. Three different approaches to adjusting the learning rate are tested. The approach with the corresponding optimisation algorithm and accuracy is shown in Table 3.1.

Table 3.1. Performance obtained on first iteration of architecture using different learning rate approaches and optimisation algorithms. A momentum term of 0.9 is used for all SGD optimisers.

Learning Rate Approach	**Optimisation Algorithm**	**Accuracy Obtained on Test Set**
Manual adjustment of learning rate.	SGD	97.93%
	Adam	97.82%
Learning rate decayed with polynomial function.	SGD	97.26%
Learning rate reduced on plateau of loss.	SGD	97.72%

Even though all three approaches perform similarly, adjusting the learning rate manually when loss and accuracy have stagnated with SGD performs best. Use of a different optimiser such as Adam did not provide a performance benefit. Network performance progression on the training and test set using SGD with manual adjustment of the learning rate is shown in Figure 3.16. A learning rate of 0.1 is used until epoch 46. It is then lowered to 0.01 until epoch 87. Test loss and accuracy do fluctuate using an initial learning rate of 0.1, but test loss does show a continued downward trend without diverging substantially from training loss. Initial fluctuations in test loss and accuracy are due to the high learning rate, causing erratic network behaviour on the loss landscape. A form of learning rate decay could be used to stabilise training, but this might also result in lower final accuracy, as seen in Table. 3.1. Lowering the learning rate on epoch 46 results in an immediate improvement in both loss and accuracy. There is a slight divergence between training and test loss, but the gap is maintained. No overfitting occurs, as test loss does not increase during training and there is no substantial deviation between test and training loss. A second lowering of the learning rate to 0.001 was tested, but this did not improve performance.

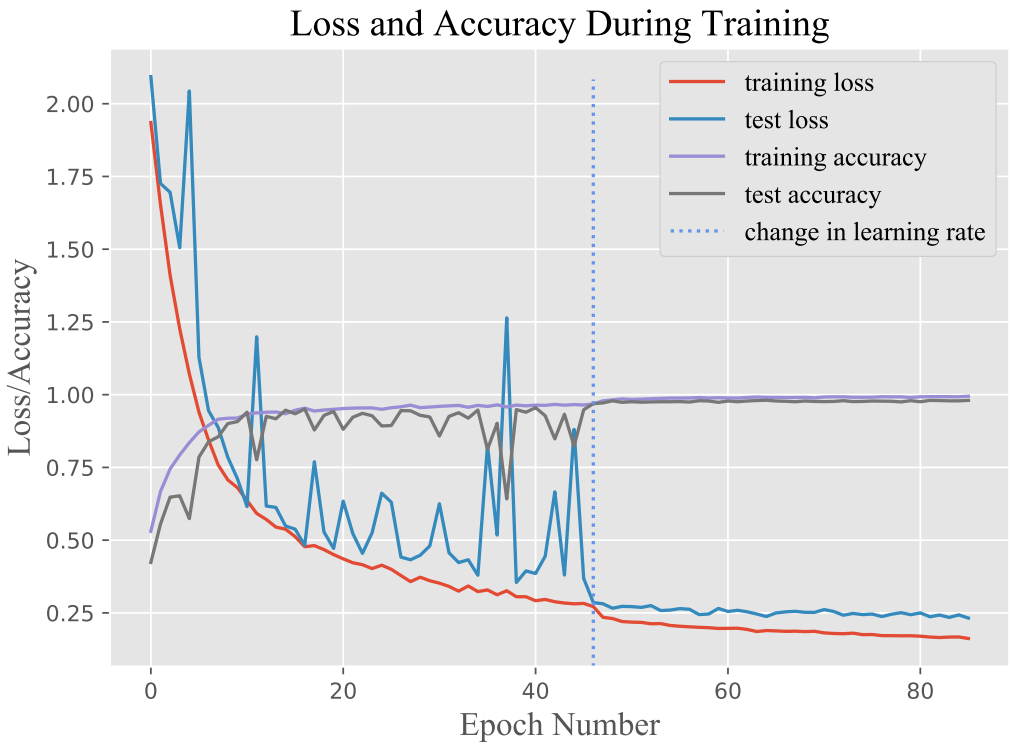


Figure 3.16. Loss and accuracy progression during network training for first iteration of network architecture. The learning rate is 0.1 until epoch 46; it is then lowered to 0.01 until epoch 87.

The performance of the trained model on the validation set for the first architecture iteration is shown in Table. 3.2. Precision and recall [1] with the corresponding number of samples in each class are shown. None of the images in the validation set is used to train the network or tune hyperparameters.

Table 3.2. Network performance on validation set for first network architecture iteration.

Class	**Precision (%)**	**Recall (%)**	**Num. of Samples**
No phone use	98.32	97.78	360
Talking on phone	98.54	99.41	339
Texting	97.31	96.93	261
Average / Total	98.06	98.04	960

3.4.4.4 Second network design iteration

An additional network design is tested that has the same architecture, but fewer filters are learned. Fewer filters have the effect of lowering training time and improving classification speed, as there are fewer trainable parameters. There is the possibility that simplification of the network through the reduction in the number of filters may affect performance negatively. The new network design learns 32 filters in the first convolutional layer; the four residual module components learn 32, 64, 128 and 256 filters respectively. The lower filter count for this iteration means the number of trainable parameters is reduced by a factor of 3.94. The network model size for the first iteration is 12.0 Mb on disk and 3.4 Mb for the second iteration. SGD is used as the optimiser with manual adjustment of the learning rate, as these parameters obtained the best performance in the first iteration. While the use of fewer filters did not have a significant impact on network performance, the test set achieved 97.67% accuracy; its generalisation ability on footage captured in real-world environments suffered. Ultimately the first network iteration was chosen to conduct experiments.

3.5 COMBINATION OF PHONE USE DETECTION METHODS

Two of the three methods developed provide phone localisation inside the vehicle. These methods indicate whether a phone is in the driver's or passenger's reachable area. The third method detects actual instances of driver phone use in real-time through image classification. It would be beneficial

[1] Precision and recall definitions are provided in Section 4.5.1.

to combine localisation and image classification methods to create a more robust and accurate phone use detection system. The process of combining sensory data that result in better information can be described as a sensor fusion problem. This particular case of fusion falls under decision or high-level fusion, as a decision is taken based on knowledge of the perceived situation. Methods typically used for decision fusion include fuzzy logic, statistical methods and voting. None of these were used for method combination, instead, a custom solution was implemented.

The output from image classification and phone localisation classification methods cannot be combined directly, as they represent two different aspects of the perceived situation. The configuration is complementary, as the sensors do not depend on one another directly, but they can be combined to provide a more complete picture of the classification output. Phone use during driving is usually not sporadic, but rather involves the use of a phone over a continuous period where the driver is either texting or talking on the phone. Audio ranging and inertial phone localisation methods cannot directly determine phone use, but they do provide valuable information as to periods when driver phone use might have occurred (i.e. indicate when a phone is in the driver's area). Talking on a phone or texting detections using the image classification method developed is then only considered during periods when one of the localisation methods have provided a positive indication of being in the driver's area. Positive phone use detections from image classification that fall outside the identified periods most probably represent false positive detections and are not considered. This solution for combining methods can be seen as a custom decision fusion approach.

Combination of localisation methods was considered. However, due to the output of the localisation methods being binary (i.e. either 'in the driver's area' or 'not in driver's area'), the algorithm would always favour the most accurate method. If more than two localisation methods were available, a final classification could be made by assigning weights according to each method's accuracy.

The output format and rate for each of the three methods developed are different, meaning that they are incompatible. Output format and rate are modified to ensure compatibility when method combination takes place. Method algorithms are updated to accept long continuous stretches of data and provide corresponding classifications at a specified sampling rate. Each method is configured to produce a classification output every 1 second. There are also timing differences between methods that are compensated for. During experimentation, each method is started independently at a slightly different absolute time. Method output would be misaligned in time if compared directly, therefore the output

is aligned and the time at which the last method is started will be the absolute start time for all three methods. Audio ranging and phone inertial localisation produce a classification output of 1 when the phone is in the driver's area and 0 otherwise. Image classification using the CNN classifies video footage for six frames every second. Each frame generates a confidence score in each of the class labels (e.g. no phone use, talking on phone or texting) as a percentage. The confidence percentages for each class label are averaged over each second and the class label with the highest average is assigned the classification for that particular second. Final image classification produces an output value of 0 for no phone use, 1 for talking on the phone and 2 for texting.

3.6 CHAPTER SUMMARY

In this chapter, the design and implementation of three driver phone use detection methods were presented. Two of the methods are phone localisation techniques, while the third method detects instances of phone use by classifying images of the driver.

In Section 3.2 the first localisation method that uses audio ranging was presented. Several considerations that should be taken into account when detecting pulse arrival time were discussed. Time-frequency spectral analysis of the signal using a wavelet transform was chosen as the method to detect pulse arrival time. A suitable audio pulse design was also presented. CPD methods to identify the exact onset of signal pulses were discussed and a new CPD method specifically designed for audio ranging was developed and presented.

The second phone localisation method was provided in Section 3.3. The embedded inertial measurement unit inside the phone was used to track the phone from a known reference point. Accelerometer and gyroscope measurements were processed through an AHRS algorithm to re-orient phone measurements to the local geographic frame. Compensated accelerometer readings were integrated twice to obtain a direction and distance estimate.

A CNN that classifies images into one of three driving behaviour classes was presented in Section 3.4. Network architectures that are currently being used widely were examined and further detail on residual networks was provided. Network hyperparameters that were considered and tuned were presented. The dataset collection and preparation process, as well as the network training procedure

that was followed, was specified. The trained network can accurately distinguish between classes of no phone use, talking on the phone and texting for the validation dataset. Experimentation will verify network performance in the real world.

An approach to combining the phone use detection methods developed was proposed in Section 3.5. Phone localisation methods are fused with driver behaviour image classification to create a more accurate and robust system. The experimental procedure to evaluate each of the methods individually and in combination are described in Chapter 4.

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

This chapter details the results obtained from experiments conducted on phone use detection methods that were developed. Results for audio ranging localisation in both control and natural pose experiments are presented in Section 4.2. Different CPD methods are tested in noisy and noise-free environments. Phone inertial localisation accuracy is evaluated in Section 4.3. Classification examples of captured video frames using the trained CNN are shown in Section 4.4. More comprehensive experimental results relating to CNN image classification are presented in Section 4.5. Detailed results for the combination of detection methods are also presented in Section 4.5. These results are analysed together because both require comparison with the generated ground-truth values and simultaneous analysis will allow for improved comparison.

4.2 AUDIO RANGING PHONE LOCALISATION RESULTS

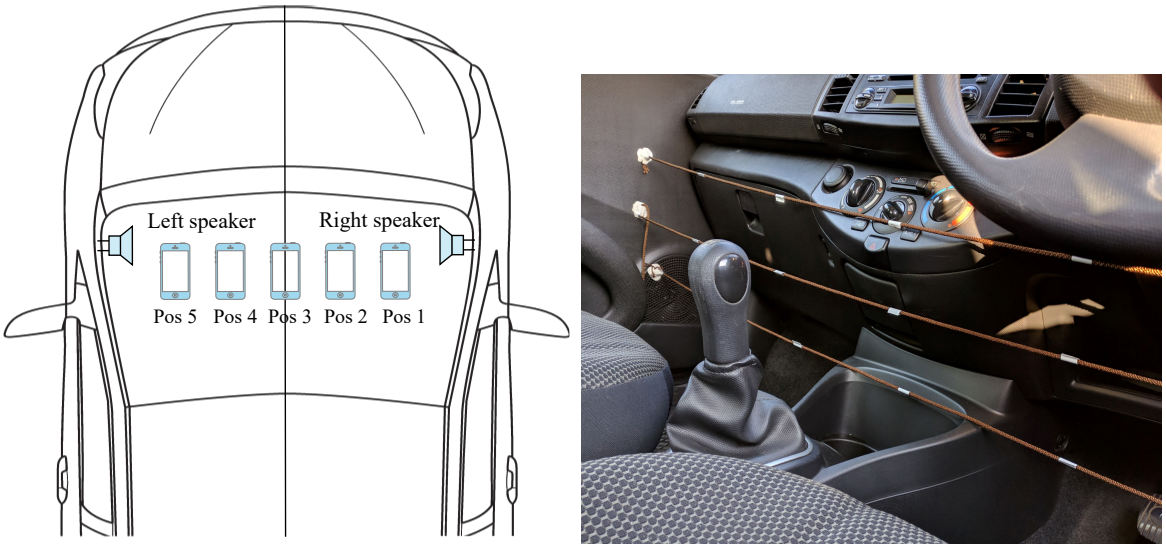
A OnePlus 3T smartphone was used for microphone audio recordings. Three different vehicles with varied stereo systems were used, namely a Mercedes ML, Nissan Micra and VW Polo. Control and natural pose experiments were conducted for each vehicle. The results of both control and natural pose experiments are analysed for different CPD methods in noise-contaminated and noise-free environments. A signal noise analysis is performed to determine the noise level that was observed for music merged with signal pulses.

4.2.1 Experimental design

Two different sets of experiments were completed for each vehicle. Control experiments determine algorithm effectiveness and accuracy (i.e. the underlying principle of the method is tested). The control experimentation procedure involved placing the phone in five different positions directly in line with the vehicle's speakers. Figure 4.1(a) shows phone placement positions during control experimentation. Control experiments provide a real indication of system accuracy regarding distance measurements. Since the phone is precisely in line with the speakers, the distance calculated at each position can be directly compared to the actual distance measurement and the error determined. A change in height of the measurement plane and its effect on distance measurements was also observed for the Nissan Micra. Two additional measurement planes at different heights were constructed, as shown in Figure 4.1(b). Measurement results were only recorded for the case where the speakers were directly in line for the remaining two vehicles (lowest measurement plane on Figure 4.1(b)). Two different audio files, each containing 10 pulses, were recorded on a CD. Each control position and natural pose number thus produced 10 distance computations, one for each pulse. Averaged distance computations are compared with the expected outcome. The first file had no noise present, while the second file had noise in the form of music merged with pulses. Control and natural pose experiment recordings were taken for both audio files (noise and noise-free) to determine the impact of noise. Noise and noise-free measurements are recorded for each of the five control experiment positions and each of the five natural poses.

Natural pose experiments determine if correct phone locations can be predicted when a user is seated and the phone is held or placed in different poses. A description of the poses the phone was held in during natural pose experiments is shown in Table 4.1. The phone was held or placed in each of these poses for both the left- and right-hand sides of the vehicle while a person was present. Classification accuracy was tested by placing and holding the phone in pose numbers 1 to 4. In this case, classification refers to selecting the correct vehicle side (either passenger's or driver's side) for different pose numbers. Pose number 5 was not tested in this case because the phone had been placed in the centre console, meaning that a passenger or driver classification could not be made.

Speaker configuration between the vehicles vary. The Mercedes ML has four speakers at the front, two on each side of the vehicle. A bass speaker is located near the bottom of the door for low frequencies and a tweeter speaker is located in the upper part of the door for higher frequencies. The Nissan Micra



(a) Control experiment positions.

(b) In vehicle control experiment setup.

Figure 4.1. Illustration of control experiment setup.

Table 4.1.

Description of poses phone is held or placed in, during natural pose experiments.

Pose Number	**Pose Description**
1	Phone is held next to the user's right ear. This is a typical pose when a user is conversing on a phone.
2	Phone is held next to the user's left ear. This is a typical pose when a user is conversing on a phone.
3	Phone is in front of the user's hand, with the user looking at the screen. Typical pose when texting.
4	Phone is placed in the side of the door compartment, with the phone microphone facing upwards.
5	Phone is placed in the centre console of the vehicle.

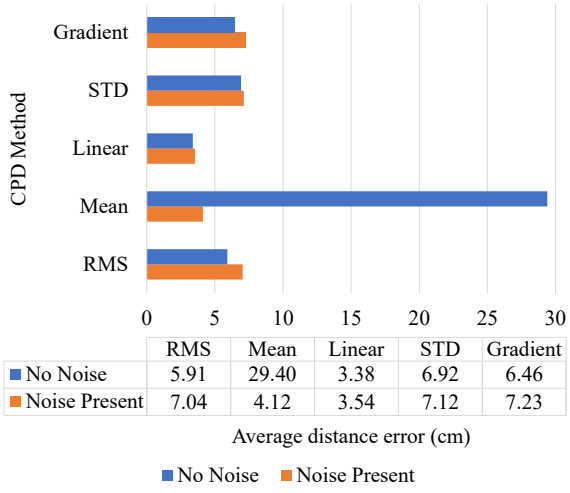
only has two speakers at the front of the vehicle, one on each side. Similar to the Mercedes, the VW Polo also has two speakers on each side at the front of the vehicle.

4.2.2 Control experimentation results

The distance error between the actual and calculated distance measurement is shown for each vehicle, both in the presence of noise and without noise, in Figure 4.2. For each figure the results of multiple CPD methods are shown. Matlab functions employed include STD, linear, mean and RMS, while the newly designed method is referred to as gradient. Figures 4.2(a), 4.2(b) and 4.2(c) show the distance error averaged across all positions (i.e. distance error averaged across the five control phone positions) for each individual vehicle. Figure 4.2(d) shows distance error averaged for each CPD method over all vehicles.

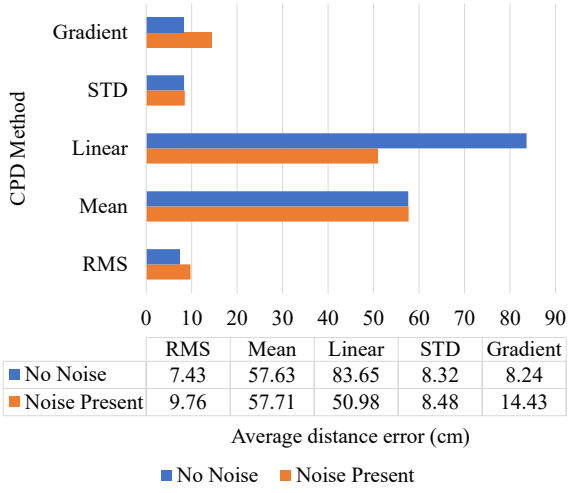
The distance error averaged across all vehicles was lowest for the STD, RMS and gradient methods. The linear CPD method had the lowest distance error in only the Nissan Micra, while it produced high distance errors in other vehicles. The presence of noise had little effect on distance error, especially for the STD, RMS and gradient methods. On average, the linear and mean methods had considerably higher distance errors for noise-free cases. The combined distance error for the STD, RMS and gradient methods was only separated by 0.59 cm in noise-free environments. In environments with noise, the separation was 2.27 cm.

The STD of ten distance measurements is calculated for each control position. STDs are averaged across positions for each vehicle. The final STDs for each CPD method are calculated by averaging computations for individual vehicles together. The combined average STD in distance measurements is shown in Figure 4.3(a). The custom gradient method had the lowest deviation for both noise and noise-free cases. The linear CPD method was the only case where the STD was higher in a noise-free environment. Figure 4.3(b) shows the average distance error per control position for the gradient CPD method; errors are once again averaged across all vehicles. There is no significant distance error variation between positions. In a noise-free environment, the largest error was in position 1 with an error of 8.19 cm. In the presence of noise the largest error was in position 4 with an error of 12.97 cm. In all positions, distance errors decreased in noise-free environments.



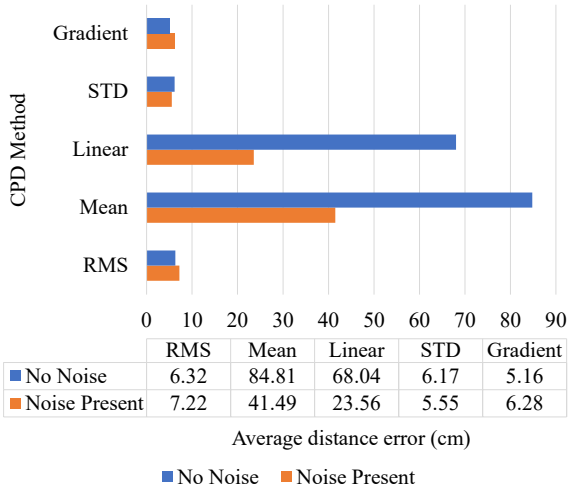
	RMS	Mean	Linear	STD	Gradient
No Noise	5.91	29.40	3.38	6.92	6.46
Noise Present	7.04	4.12	3.54	7.12	7.23

(a) Nissan Micra average distance error.



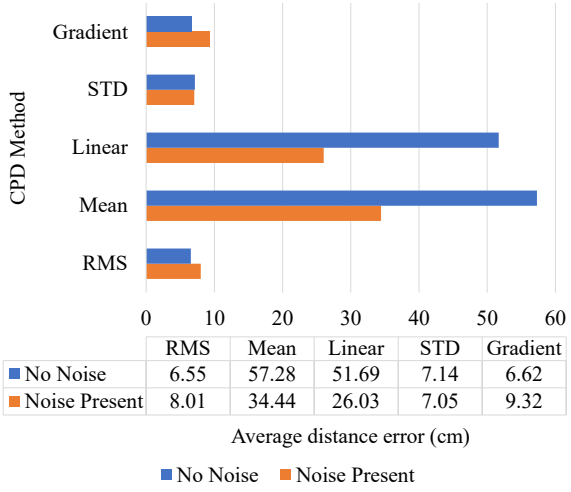
	RMS	Mean	Linear	STD	Gradient
No Noise	7.43	57.63	83.65	8.32	8.24
Noise Present	9.76	57.71	50.98	8.48	14.43

(b) VW Polo average distance error.



	RMS	Mean	Linear	STD	Gradient
No Noise	6.32	84.81	68.04	6.17	5.16
Noise Present	7.22	41.49	23.56	5.55	6.28

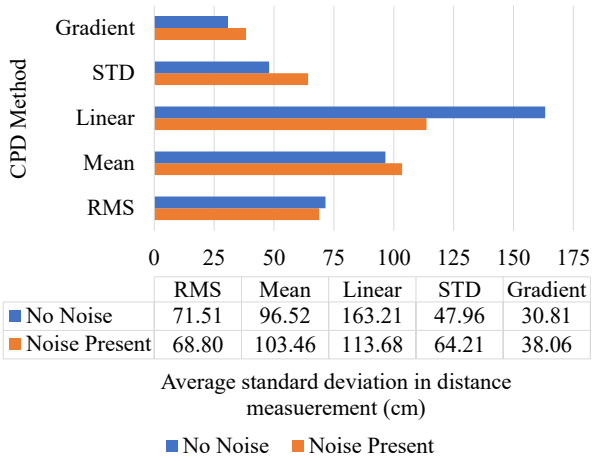
(c) Mercedes ML average distance error.



	RMS	Mean	Linear	STD	Gradient
No Noise	6.55	57.28	51.69	7.14	6.62
Noise Present	8.01	34.44	26.03	7.05	9.32

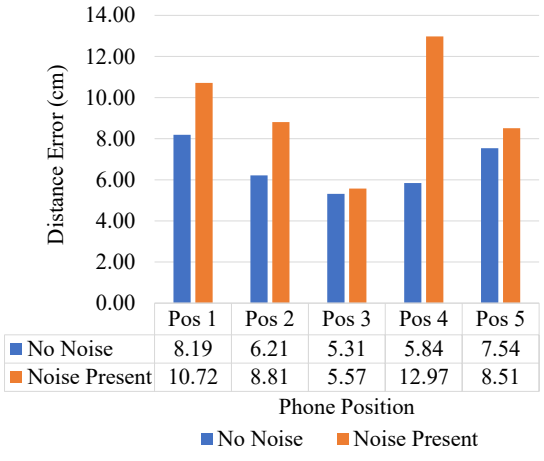
(d) Combined distance error averaged across all vehicles.

Figure 4.2. Control experiments average distance error for different CPD methods.



	RMS	Mean	Linear	STD	Gradient
No Noise	71.51	96.52	163.21	47.96	30.81
Noise Present	68.80	103.46	113.68	64.21	38.06

(a) Combined average STD between distance measurements.



	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5
No Noise	8.19	6.21	5.31	5.84	7.54
Noise Present	10.72	8.81	5.57	12.97	8.51

(b) Combined gradient CPD distance error per position.

Figure 4.3. Vehicle control experimentation graphs.

Two additional measurement planes were constructed to test the impact of a height change on distance measurements in the Nissan Micra. There was a height change of 11 cm between planes, but this did not have a significant effect on the average STD or distance error, especially for gradient CPD. Results for all three measurement planes are shown in Table 4.2.

Table 4.2. Average STDs and distance errors for different measurement levels in Nissan Micra.

Change Point Detection Method:	No Noise:						Noise Present:					
	Level 1		Level 2		Level 3		Level 1		Level 2		Level 3	
	Average STD (cm)	Average Distance Error (cm)	Average STD (cm)	Average Distance Error (cm)	Average STD (cm)	Average Distance Error (cm)	Average STD (cm)	Average Distance Error (cm)	Average STD (cm)	Average Distance Error (cm)	Average STD (cm)	Average Distance Error (cm)
RMS	81.49	5.91	81.12	7.35	78.92	7.62	47.74	7.04	68.78	8.09	110.33	7.66
Mean	48.92	29.40	60.14	46.90	41.07	13.34	33.64	4.12	5.35	5.99	21.44	6.07
Linear	101.80	3.38	102.60	38.89	24.14	5.95	100.78	3.54	5.93	5.76	30.81	5.17
STD	42.64	6.92	22.90	7.51	64.78	7.58	49.14	7.12	54.55	8.01	90.29	7.43
Gradient	4.72	6.46	4.82	6.81	4.10	8.05	9.54	7.23	3.56	7.62	6.03	7.97

4.2.3 Natural pose experimentation results

Natural pose results are averaged across all three vehicles. The difference in distance measured between signals with and without noise are shown in Figure 4.4 for various poses described in Table 4.1. Each pose number corresponds with the appropriate location number. Results are shown for both left- and right-side measurements, as well as different CPD methods. RMS, STD and gradient methods

demonstrate minor differences in the distance measurement. Mean and linear CPD methods display a larger variance between noise and noise-free measurements.

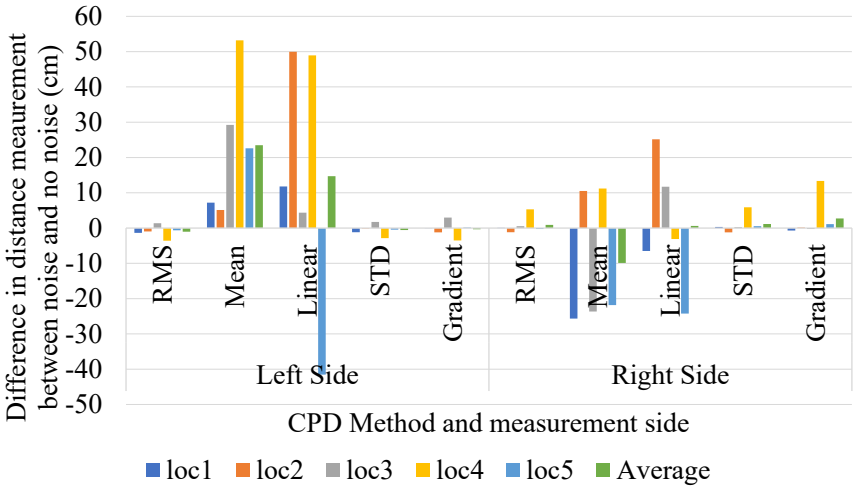


Figure 4.4. Combined difference in distance measurements for noise and no noise.

Figure 4.5 shows the average STD in distances calculated for measurements recorded on the left- and right-side of the vehicle for different noise conditions. The standard deviation between the 10 distance measurements for a particular pose number are averaged. Standard deviations for all pose numbers are then averaged together across all three vehicles. On average, gradient CPD had the lowest measurement deviation for noise and noise-free cases. The gradient method was only slightly outperformed by the STD method in the noise-free case on the left side of the vehicle. While the STD and RMS methods performed well in noise-free environments, all Matlab methods had a high deviation for distance measurements in the presence of noise. With noise present, gradient CPD had an average STD of 55.52 cm lower than the next best method, which was the STD method.

Figure 4.6 shows the percentage of natural poses that were correctly identified for different vehicle sides, noise conditions and CPD methods. A right- or left-side vehicle classification is made based on the sign of the calculated distance. Classifications are made while the phone is held in natural pose numbers 1 to 4. Number 5 is not included because it is at the centre of the vehicle and a right or left classification cannot be made. The number of classifications correctly identified is expressed as a fraction of the total number of classifications. The gradient CPD method performed best in all circumstances, with performance remaining consistent in all noise conditions. An overall classification percentage of 97.7% was achieved. For Matlab-based methods, STD and RMS performed best, while

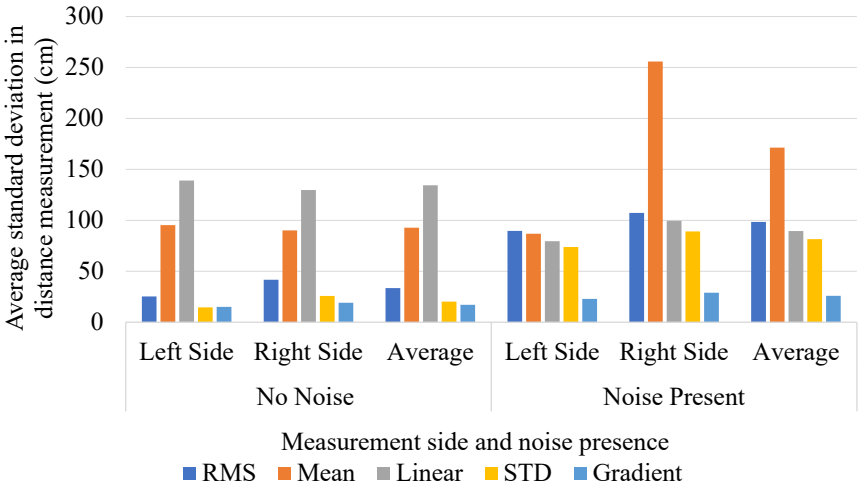


Figure 4.5. Combined average STD for different measurement sides.

the mean and linear methods performed worst. No significant performance benefit was observed between noisy and noise-free environments.

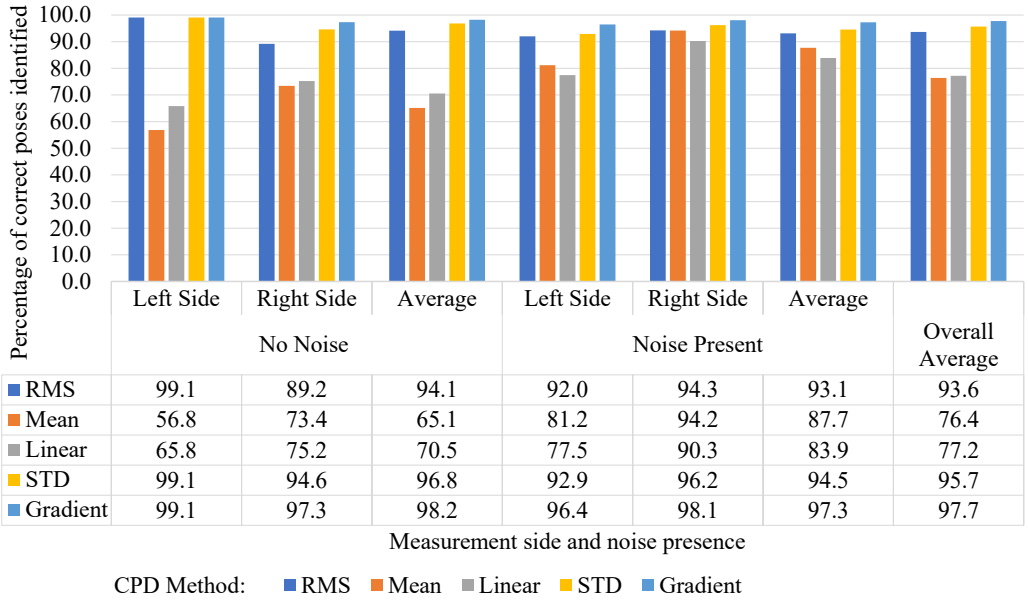


Figure 4.6. Combined percentage of correct classifications for natural poses.

An additional pose where the phone was placed in the user's right pocket was also investigated. When placed in the pocket, excessive noise is introduced to the microphone recording, resulting in multiple audio pulses not being identified. Once the phone is taken out of the pocket, as in cases of phone use, accurate location classifications will be possible.

4.2.4 Signal noise analysis

The signal-to-noise ratio (SNR) of the audio file with pulses merged with music is shown in Table 4.7. The SNR was calculated for each vehicle by recording the signal (pulse) and noise (music) separately at three different locations (left, middle and right side of the vehicle). At each location the SNR was calculated for ten different pulses; results were then averaged across the three locations for a particular vehicle. The reference SNR was calculated from the original audio file without being played over the stereo. In all cases, the signal power was smaller than noise power. Measured SNR did not seem to have a significant impact on calculated distance accuracy. The large SNR observed in the VW Polo indicates why larger control distance errors were measured in this vehicle. Lower SNRs in the Nissan Micra and Mercedes ML resulted in the lowest control distance errors being observed for these vehicles. A reason for Mercedes ML SNR being lower than the reference is that SNR was calculated at different positions and averaged. Different positions produce varied SNRs, which could be lower than the reference when averaged. In-person, it was difficult to recognise the pulses, especially when mixed with music. High pulse frequencies were selected to make discernment difficult for humans.

Location	SNR (dB)
Reference	-0.72
Nissan Micra	-1.86
VW Polo	-6.70
Mercedes ML	-0.57
Vehicle Average	-3.04

Figure 4.7. Signal-to-noise ratio measured in each vehicle along with a calculated reference.

4.3 PHONE INERTIAL LOCALISATION RESULTS

Phone sensor readings were logged using the Android application developed on a OnePlus 3T smartphone. Phone pick-ups were completed while driving a single vehicle in a private estate. Method performance was calculated by analysing phone pick-ups from the reference location.

4.3.1 Experimental design

The performance of the proposed phone inertial localisation approach was analysed using experiments conducted inside a vehicle while driving. A phone was attached to a magnetic holder that acted as a

reference point and could be removed at any time. The magnetic phone holder is shown in Figure 4.8. While driving, the phone could be picked up by either the driver or the passenger; the performance of both sets of outcomes were tested during experimentation. Each phone removal from the reference point was classified as a movement into an octant section. If the phone moved into an octant on the right-hand side it was classified as being in the driver's area, while movement into an octant section on the left-hand side was classified as not being in the driver's area. An Android application was developed that logs phone sensor (accelerometer, gyroscope and magnetometer) readings and records these readings in an output file. Output files were analysed in Matlab using the proposed approach to determine the performance of the algorithm.

(a) Magnetic phone holder.

(b) Phone attached to magnetic phone holder.

Figure 4.8. Setup used for phone reference point.

Considering the octant sections in Figure 3.7 and the selected reference point shown in Figure 4.8, the phone could realistically only move into octant sections 3 or 4 for a driver pick-up and sections 7 or 8 for a passenger pick-up. During experimentation a total of 100 phone pick-ups were completed; 50 pick-ups were from the driver and 50 from the passenger. Of the 50 driver pick-ups, 25 were into octant section 3 and 25 were into section 4. The same phone pick-up assignment was completed for the passenger side and octant sections 7 and 8. Method performance was calculated by analysing whether the correct octant section (i.e. one of the eight octant sections) and phone area (i.e. in driver's area or not in driver's area) had been classified. Precision and recall [1] were calculated for both the octant section and phone area classification.

[1] Precision and recall definitions are provided in Section 4.5.1.

4.3.2 Phone pick-up experimentation results

Inertial localisation results are shown in Table 4.3. The table is divided into two classification tasks. The first shows octant section classification performance (i.e. whether the algorithm correctly identified the octant section into which the phone moved), while the second shows phone area classification performance (i.e. whether the algorithm correctly indicated if the phone moved into the driver's area). Precision, recall and the number of samples are indicated for both classification tasks.

Octant section classification shows similar precision and recall percentages in most cases. Octant sections 3 and 4 imply movement to the driver's side, while sections 7 and 8 imply movement to the passenger's side. Precision for both sides improved in sections where the movement was in a downward direction (i.e. octant sections 4 and 8). Average precision of 57.59% and recall of 50.00% were achieved. Selecting the correct octant section out of a possible eight is a more difficult classification task than simply classifying the phone area.

Average precision and recall for phone area classification were almost the same at approximately 91.00%. This indicates that a large number of positive samples had been identified and few of those identified included false positives. Improved precision and recall percentages were achieved because each area classification included four octant sections, resulting in an easier classification task.

Table 4.3. Phone inertial localisation classification results.

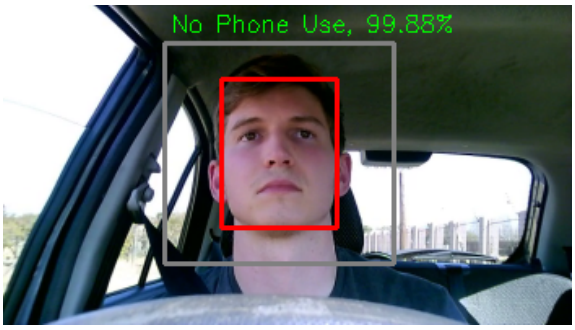
	Class	Precision (%)	Recall (%)	Num. of Samples
Octant section classification	Octant section 3	52.00	52.00	25
	Octant section 4	63.64	56.00	25
	Octant section 7	50.00	48.00	25
	Octant section 8	64.71	44.00	25
	Average / Total	57.59	50.00	100
Phone area classification	Not in driver's area	91.84	90.00	50
	In driver's area	90.20	92.00	50
	Average / Total	91.02	91.00	100

4.4 CNN DRIVER PHONE USE CLASSIFICATION EXAMPLES

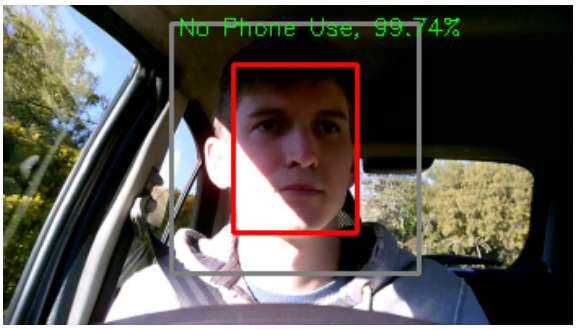
The network trained on the collected dataset is evaluated on footage captured inside the vehicle in real-world conditions using an external webcam. The network's generalisation ability is tested, since there is variation between the collected dataset and images captured inside the vehicle. Images from the collected dataset contain a variety of environments in the background, while those captured in the vehicle only have in-vehicle environments. Only examples of captured frame classifications are shown; a complete analysis of network performance is described in Section 4.5. CNN image classification and method combination both require comparison with the manually labelled ground-truth values, making simultaneous evaluation logical.

Examples of frames captured during different driving behaviours are shown in Figures 4.9 and 4.10. Frames are classified using the trained network architecture. The red box shows the face region identified using an OpenCV face detection function; the grey box shows the extracted region, which is passed as input to the network. The class, as well as network confidence in the class, is displayed in green text above the grey box. Classifications of each class that the network correctly identified are shown in Figure 4.9. The network can accurately distinguish between classes in challenging conditions. While talking on the phone, a variety of hand and head poses were observed. Head pose and tilt variations were also present while texting. Several different lighting conditions inside the vehicle cabin were experienced during experimentation.

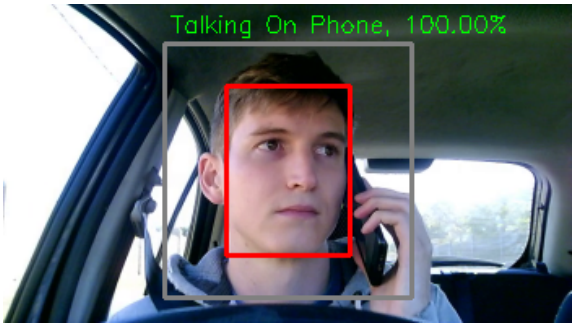
Examples of incorrect network classifications are shown in Figure 4.10. Severe head turns were occasionally classified as texting. Differences between 'no phone use' and 'texting' classes are very subtle, creating a higher chance of misclassification between these classes. Intense light shining through the back window would sometimes cause the network to miss 'talking on phone' classifications. A driver scratching an ear could falsely label this as a 'talking on phone' instance. At times sunlight would directly enter the camera lens, causing extreme sun flare on captured frames. During these periods no face detection was possible, thus driving behaviour classification could also not be performed.



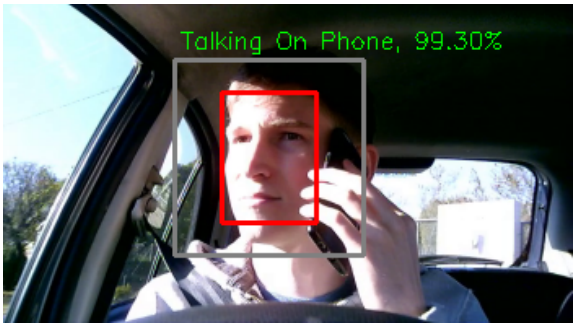
(a) No phone use classification



(b) No phone use classification

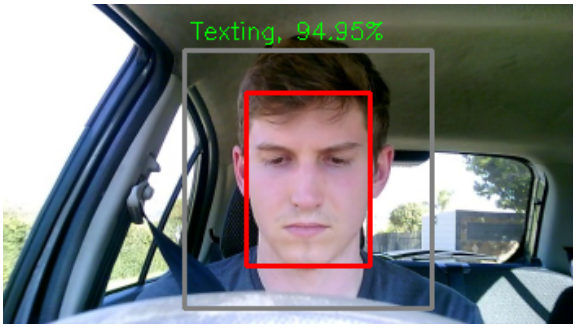


(c) Talking on phone classification

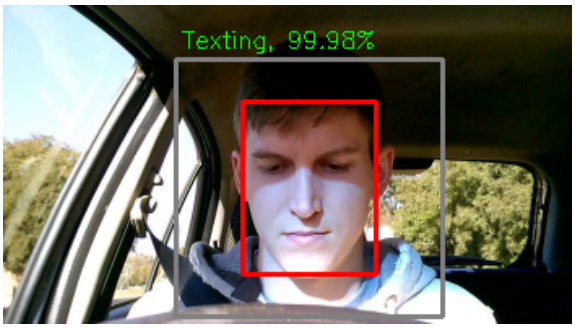


(d) Talking on phone classification

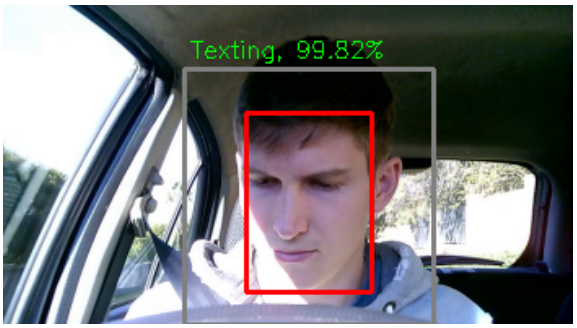
(e) Talking on phone classification



(f) Texting classification

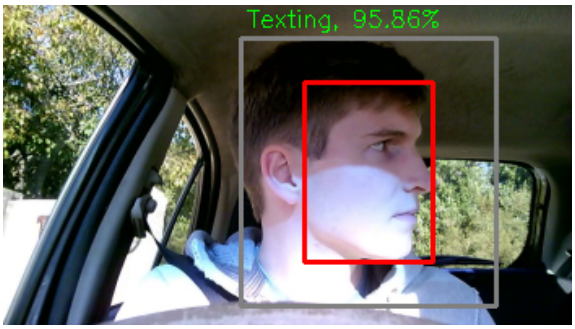


(g) Texting classification

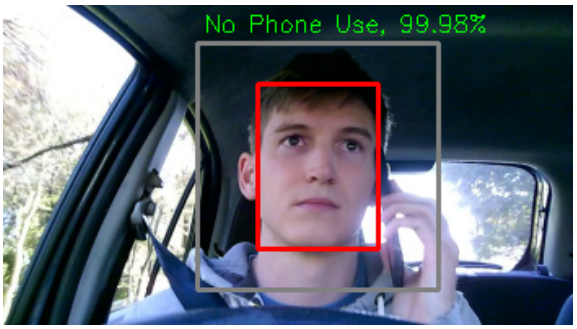


(h) Texting classification

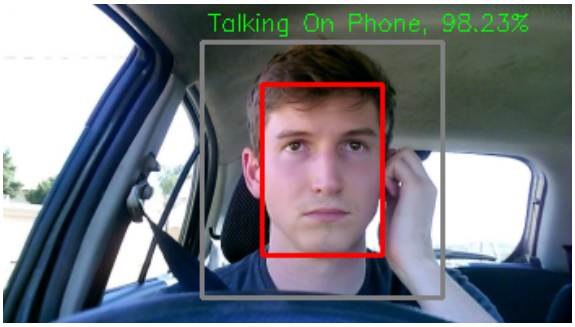
Figure 4.9. Samples of classes correctly classified. Frames are taken from video footage that was captured. The class label and network confidence in the class are displayed in green text.



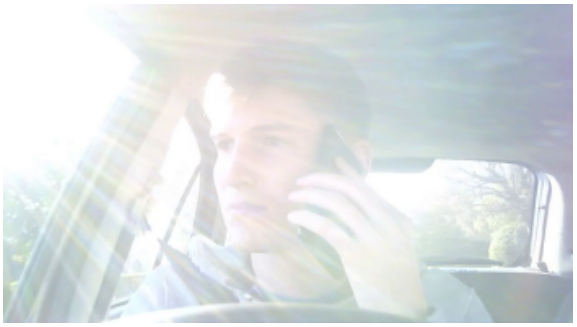
(a) Texting classification



(b) No phone use classification



(c) Talking on phone classification



(d) No classification

Figure 4.10. Samples of classes incorrectly classified. Frames are taken from video footage that was captured. The class label and network confidence in the class are displayed in green text.

4.5 COMBINED PHONE USE DETECTION METHOD RESULTS

The evaluation of CNN image classification and method combination are related and very similar. A collective performance evaluation is completed, as it is beneficial. Examples demonstrating method classification output as time progresses are provided. Examples are selected to visually illustrate certain aspects that affect method performance and are shown in Section 4.5.2. Output traces for all experiments are not shown, but results and metrics obtained for all tests are shown in tabular form in Section 4.5.3.

4.5.1 Experimental design

The procedure for capturing measurements from all three methods requires recordings from two cameras to be triggered simultaneously. The primary camera captures a front-on view of the driver's

face, while the secondary camera captures a view of the phone attached to the magnetic phone holder. A secondary camera is needed to ensure that the exact timing of phone pick-up and return can be recorded. Two separate ground-truth values are required to calculate method performance: a localisation ground-truth to compute audio ranging and phone inertial localisation performance and an image classification ground-truth to compute CNN image classification performance.

A ground-truth value is generated for audio ranging and phone inertial localisation through manual observation of timing from the secondary camera. Intervals where the phone is stationary on the magnetic pad are classified as 0 or 'not in driver's area', while intervals where the phone has been picked up are classified as 1 or 'in driver's area'. Classifications are recorded in intervals of 1 second. Comparison of localisation ground truth values and output from the two localisation algorithms indicate method accuracy. A second ground-truth value is generated by manually observing footage captured from the primary camera. Each second of footage is classified as 0 ('no phone use'), 1 ('talking on phone') or 2 ('texting'). CNN image classification accuracy can be computed by directly comparing method output with the corresponding ground-truth value. Combined method performance is also calculated through comparison with the image classification ground truth.

Method performance is verified by conducting multiple experiments and adjusting factors such as light conditions, driver and passenger phone pick-ups, irregular phone pick-ups, blocking of the phone's microphone, increase in ambient noise and lowering of the volume at which audio pulses are played. Adjustable factors are chosen such that specific and combined method performance can be observed when severely affected either through human interference or environmental conditions. Light conditions include extreme sunlight shining into the vehicle cabin and on the driver's face from several angles during the early morning. There are also occasions when the sun shines directly into the primary camera: mid-afternoon sunlight with good light conditions and less intense sunlight shining into the cabin and late-afternoon sunlight during sunset where light intensity is much lower. Data recording and measurement applications for each method are started independently. Video footage to be classified by the CNN is captured by a webcam plugged into a laptop. An Android application that records microphone audio on the phone and the application that logs phone sensor measurements is also initiated. A vehicle was driven by a single operator in an estate on private roads for experiments that involved testing method combinations.

The chosen metrics to evaluate classifier performance are precision, recall and the respective equally

weighted average per metric. Classes that contain phone use (i.e. talking on phone and texting) will occur less frequently than normal driving behaviour (i.e. no phone use), but these classes are equally, if not more important. Therefore, a macro or equally weighted average is chosen because it gives equal weight to all classes and is not weighted according to the number of samples in each class. This is beneficial, as it highlights the performance of infrequent classes.

Precision refers to a classifier's ability not to label as positive a sample that is actually negative. High precision indicates that most of the predicted labels are correct. Precision is the ratio

$$Precision = \frac{TP}{TP+FP}, \tag{4.1}$$

where TP is the number of true positive samples and FP is the number of false-positive samples. Precision is a critical metric, as it shows if a classifier is falsely accusing a driver of using his phone. Recall or true positive rate is a measure of the classifier's ability to find all positive samples. A classifier with high recall detects most of the positive samples. It is defined by the ratio

$$Recall = \frac{TP}{TP+FN}, \tag{4.2}$$

where TP is the number of true positive samples and FN is the number of false-negative samples. Ideally, analysis of performance data should return high scores for both precision and recall, indicating that most positive samples are detected and there are few false positives. However, in the case of detecting driver phone use, precision is of greater importance. Making accurate predictions regarding phone use is more important than simply detecting all instances of phone use.

4.5.2 Visual method classification output

Classification output for individual methods and method combinations are shown visually in Figures 4.11, 4.12 and 4.13. Classification output as time progresses is represented visually as signal traces. All methods produce classification output once every second, making comparison easy. The correlation between method output and ground-truth values indicates how closely a method matches the true classification. The first two method outputs correspond to audio ranging and phone inertial localisation. A ground-truth value for localisation is depicted in orange. A classification value of 0 indicates that the phone is not in the driver's area, while a value of 1 indicates that the phone is in the driver's area. CNN image classification outputs values of 0 (no phone use), 1 (talking on phone) or 2 (texting).

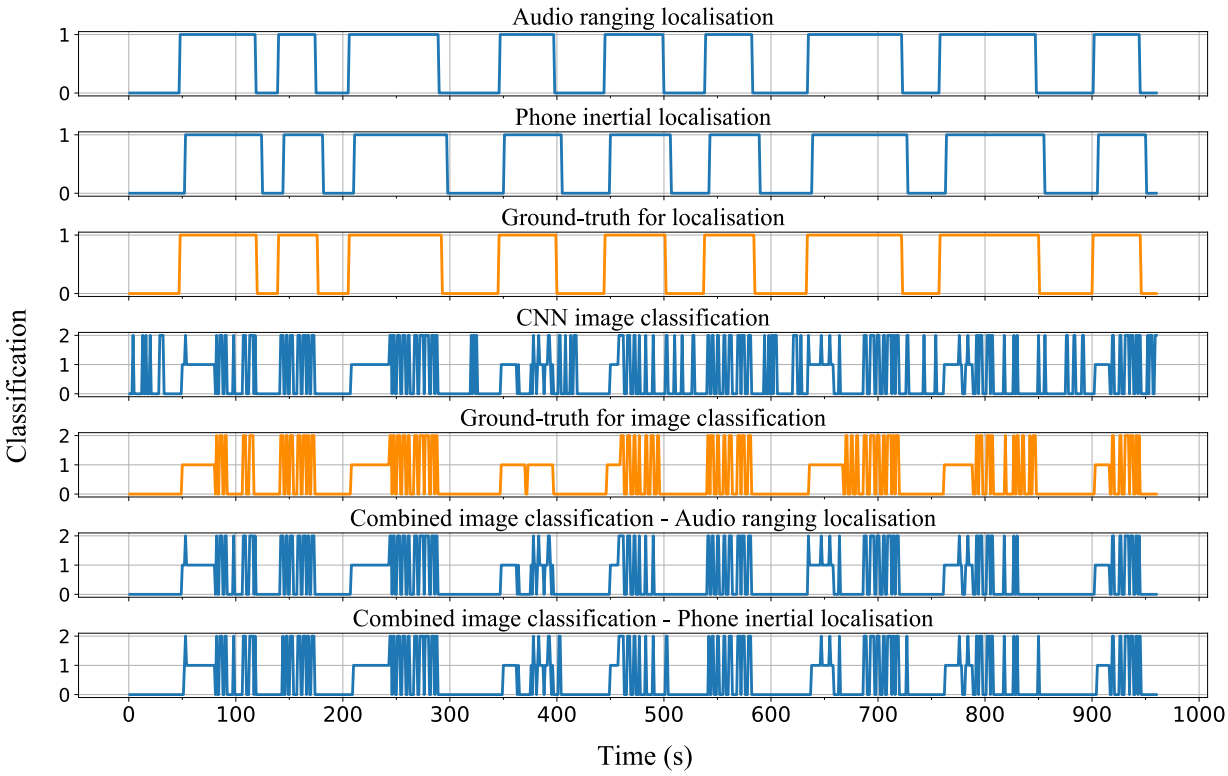


Figure 4.11. Example of method classification output for harsh lighting conditions.

Ground-truth values for image classification are also illustrated in orange. Method combinations are demonstrated next. Firstly, CNN image classification is combined with audio ranging localisation and then phone inertial localisation. Method combinations are also compared with the image classification ground-truth.

Figure 4.11 shows method classification output during harsh lighting conditions. Difficult light conditions are expected to have a detrimental effect on method performance, particularly on CNN image classification. Audio ranging and phone inertial classifications match the corresponding ground-truth values closely; each of the nine phone use segments is detected. Most phone use instances, whether it be talking on the phone or texting, are detected by the trained network. However, several false positive detections are present. Some examples of false positives for CNN image classification can be seen between times 0 to 50 seconds and 500 to 650 seconds. Almost all false positives are misclassification of the texting class. Image classification combination with localisation methods has a major beneficial effect where many of the false positive classifications can be removed.

Figure 4.12 shows method output during experiments with irregular phone pick-ups. Irregular pick-ups

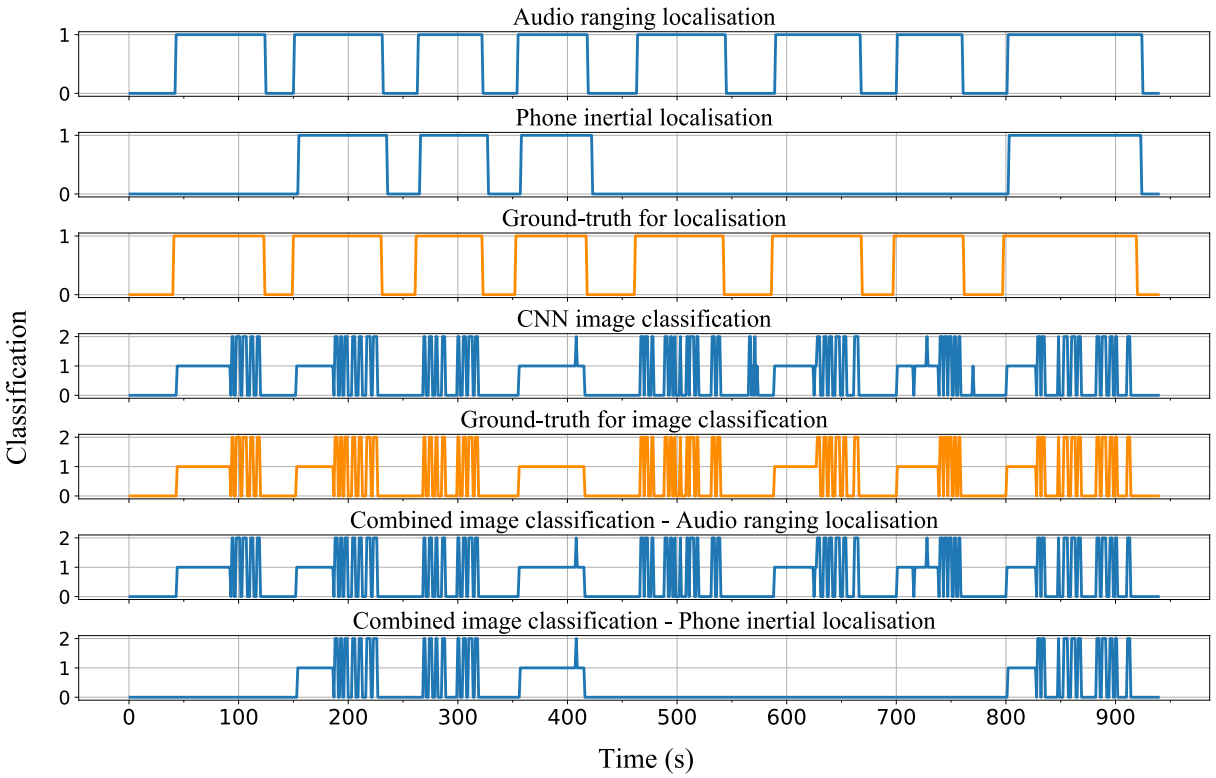


Figure 4.12. Example of method classification output for irregular phone pick-ups.

involve removal of the phone from the reference point in an erratic fashion. The initial movement direction might not correspond with the movement direction required for the destination (e.g. a driver phone pick-up might involve a motion to the passenger's side first before being brought to the driver's side). The figure shows all phone use segments correctly identified by audio ranging localisation. Four of the eight segments are not detected by phone inertial localisation. Image classification detects nearly all instances of phone use, but there are still false positive detections. Image classification combined with audio ranging localisation removes many of the false positive detections. Large sections of phone use detections are omitted from image classification combined with phone inertial localisation. This is due to the four phone use segments not being identified.

Figure 4.13 shows method classification output during experiments where phone microphone recordings were constrained. Audio ranging localisation exhibits erratic behaviour. It correlates closely with ground-truth values, but there are intermittent 'in driver's area' sections that are missed. Phone inertial localisation identified all eight of the 'in driver's area' segments. CNN image classification once again corresponds closely to the ground-truth values with the addition of several false positive classifications. Despite sporadic audio ranging localisation, many incorrect phone use classifications are filtered out

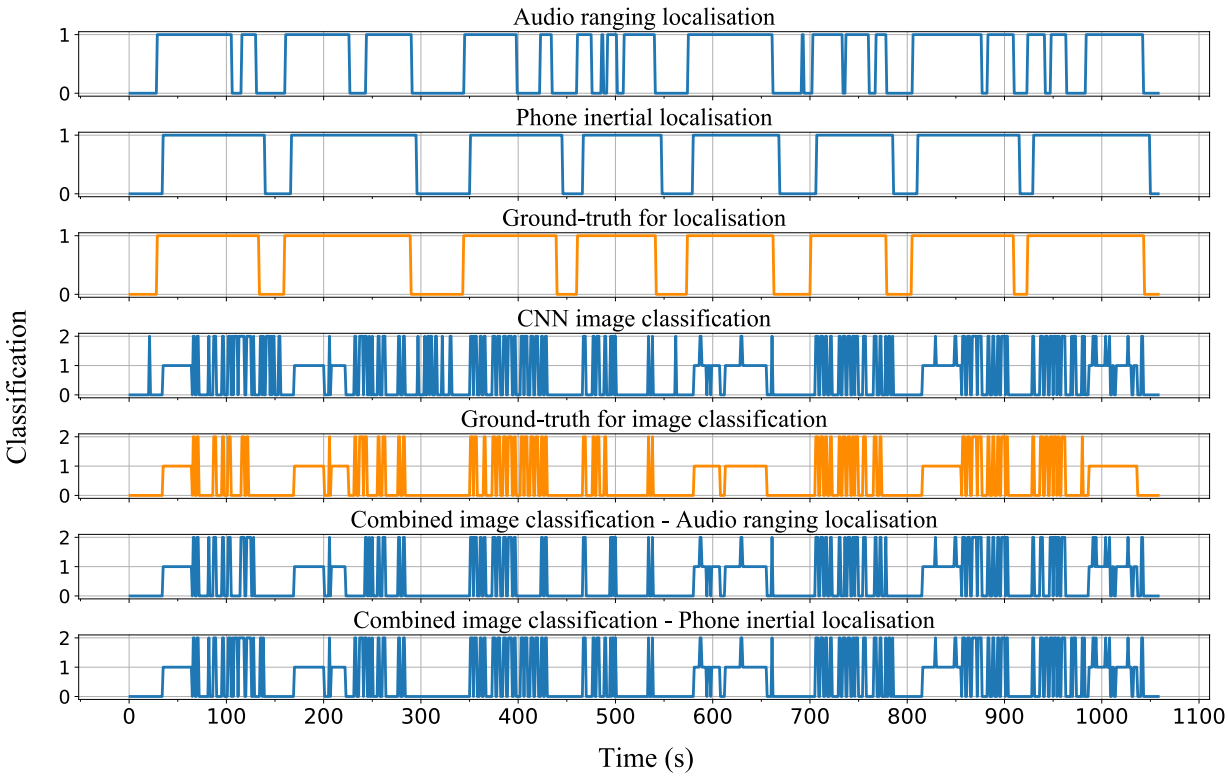


Figure 4.13. Example of method classification output when covering phone microphone.

by this method combination. However, cases of actual phone use are also omitted, such as those in the time frame between 400 and 420 seconds. Image classification combined with phone inertial localisation also removes many false positive detections and does not omit as many cases of actual phone use. Nonetheless, some false positive detections are still included, such as those in the time frame between 960 and 980 seconds. These false positive detections are not present in combined audio ranging because of the sporadic localisation output.

4.5.3 Method combination experimentation results

Results obtained for all tests are shown in tabular form. Precision and recall percentages, as well as the number of samples corresponding to each class, are provided. Localisation methods only have two classes, classification of the phone in the driver's area or not in the driver's area. CNN image classification and method combinations show metrics for three classes. Experiments are conducted such that specific method performances could be negatively affected. CNN image classification is tested under different lighting conditions. Phone inertial localisation is tested through irregular phone

pick-ups. Audio ranging localisation is tested by increasing the ambient noise level, covering the phone microphone and reducing the pulse volume. A test involving driver and passenger phone pick-ups was also conducted. The collective results for all tests are provided and include a total of 122 minutes of classification results.

Experiments to test the effect of different lighting conditions, especially on CNN image classification, are shown for individual tests in Table 4.4 and collectively for all light tests in Table 4.5. Individual tests shown in Table 4.4 are separated into morning, mid-afternoon and late-afternoon lighting. The most severe lighting conditions occurred during the morning. The best conditions were experienced during mid-afternoon and relatively good lighting was observed during late afternoon. Driver face visibility inside the vehicle cabin was highest during mid-afternoon testing. 'In driver's area' precision was slightly higher than 'not in driver's area' for audio ranging localisation in all three tests. The same was true for phone inertial localisation. Precision and recall percentages remained very similar across all three experiments; precision and recall of approximately 97.00% and 98.00% were achieved for audio ranging localisation. Phone inertial localisation had similar precision and recall percentages for the morning and mid-afternoon. Averages for both metrics decreased during the late afternoon because two phone use segments were misclassified as not being in the driver's area. The main objective of these tests was to observe the effect of lighting condition changes on CNN image classification performance. Average precision and recall increased from morning to late afternoon and again from late afternoon to mid-afternoon. The most notable improvement was in texting performance. Precision in the worst lighting conditions was 66.03%; this increased by over 20% to 87.56% in the best lighting conditions. Method combination of image classification with audio ranging localisation showed substantial increases in precision and recall percentages for all three experiments. Precision performance improved particularly for cases of texting phone use. During the morning, texting precision was improved by 23.58% through a combination of CNN image classification with audio ranging localisation. An increase in average precision and recall was observed for all three experiments compared to only employing CNN image classification. Image classification combined with phone inertial localisation generally improved average precision and recall percentages for all three experiments, except in the case of recall in the late afternoon. In this case, recall decreased from 96.48% to 83.19%. This was due to phone inertial localisation not identifying two of the eight 'in driver's area' segments. Positive phone use instances during these periods were not considered, thus reducing the recall.

Table 4.4. Method classification results during different lighting conditions.

		Morning			Mid-afternoon			Late afternoon		
	Class	**Precision (%)**	**Recall (%)**	**Num. of Samples**	**Precision (%)**	**Recall (%)**	**Num. of Samples**	**Precision (%)**	**Recall (%)**	**Num. of Samples**
Audio ranging localisation classification	Not in driver's area	94.99	100.00	379	94.92	99.72	356	95.82	100.00	367
	In driver's area	100.00	96.56	581	99.86	97.39	729	100.00	96.41	446
	Average / Total	97.49	98.28	960	97.39	98.56	1085	97.91	98.21	813
Phone inertial localisation classification	Not in driver's area	87.89	88.13	379	90.45	90.45	356	73.89	94.82	367
	In driver's area	92.24	92.08	581	95.34	95.34	729	94.44	72.42	446
	Average / Total	90.07	90.10	960	92.89	92.89	1085	84.16	83.62	813
CNN image classification	No phone use	91.88	90.20	602	99.50	96.93	619	99.61	95.17	538
	Talking on phone	100.00	80.00	200	99.29	96.54	289	99.40	94.29	175
	Texting	66.03	87.34	158	87.56	99.44	177	75.19	100.00	100
	Average / Total	85.97	85.85	960	95.45	97.64	1085	91.40	96.48	813
Combined image classification - Audio ranging localisation	No phone use	92.57	99.34	602	99.36	99.84	619	99.63	98.88	538
	Talking on phone	100.00	80.00	200	100.00	96.19	289	100.00	94.29	175
	Texting	89.61	87.34	158	95.14	99.44	177	87.72	100.00	100
	Average / Total	94.06	88.89	960	98.16	98.49	1085	95.78	97.72	813
Combined image classification - Phone inertial localisation	No phone use	90.52	98.34	602	97.30	98.87	619	88.94	98.70	538
	Talking on phone	100.00	74.50	200	99.63	92.04	289	100.00	66.86	175
	Texting	85.99	85.44	158	92.59	98.87	177	84.85	84.00	100
	Average / Total	92.17	86.09	960	96.51	96.59	1085	91.26	83.19	813

Collective results for all lighting conditions combined are shown in Table 4.5. All three experiments combined provided 47 minutes of classification results. Audio ranging localisation produced high average precision and recall percentages of 97.59% and 98.39%. Phone inertial localisation had lower averages for both, but the precision percentage for 'in driver's area' was still high at 94.06%. CNN image classification achieved relatively high percentages for precision and recall, but texting precision was low at 76.24%. Average precision, particularly texting precision, was drastically improved with method combination; a 15.15% and 12.3% improvement for the two respective method combinations was obtained for texting.

Experiments to test irregular phone pick-ups are shown in Table 4.6. Two sets of measurements were recorded. The first set was recorded during the morning and included 12 minutes of data recording, while the second set was recorded during mid-afternoon and included 16 minutes of data recording. There were a total of 16 driver phone pick-ups; after each pick-up a series of phone use instances occurred. Audio ranging localisation correctly detected all 16 phone use segments. High precision and recall percentages of 96.30% and 97.15% were achieved. Phone inertial localisation only identified seven of the 16 phone use segments; a visual illustration of phone use segments not identified by phone inertial localisation is shown in Figure 4.12. This negatively affected 'not in driver's area' precision and 'in driver's area' recall. Precision for 'not in driver's area' was low because it contained false

Table 4.5. Method classification results for all lighting experiments combined.

	Class	Precision (%)	Recall (%)	Num. of Samples
Audio ranging localisation classification	Not in driver's area	95.24	99.91	1102
	In driver's area	99.94	96.87	1756
	Average / Total	97.59	98.39	2858
Phone inertial localisation classification	Not in driver's area	83.18	91.11	1102
	In driver's area	94.06	88.44	1756
	Average / Total	88.62	89.77	2858
CNN image classification	No phone use	96.90	94.09	1759
	Talking on phone	99.51	90.96	664
	Texting	76.24	95.17	435
	Average / Total	90.88	93.41	2858
Combined image classification - Audio ranging localisation	No phone use	97.00	99.37	1759
	Talking on phone	100.00	90.81	664
	Texting	91.39	95.17	435
	Average / Total	96.13	95.12	2858
Combined image classification - Phone inertial localisation	No phone use	92.29	98.64	1759
	Talking on phone	99.81	80.12	664
	Texting	88.54	90.57	435
	Average / Total	93.55	89.78	2858

positives from the nine missed phone use segments. Recall for ‘in driver’s area’ was very low because numerous positive phone use segments were not identified. However, most importantly, precision for ‘in driver’s area’ was high at 94.01%. Precision and recall percentages for CNN image classification were high. Texting recall was almost 100%, but precision was only 84.91%. This shows that almost all occurrences of texting were detected, but some detections were incorrectly labelled as texting. Image classification combined with audio ranging localisation produced an increase in metric performance for all classes. Method combination with phone inertial classification only improved ‘no phone use’ recall and most importantly ‘talking on phone’ and ‘texting’ precision. Recall percentages for phone use behaviours dropped dramatically because of the unidentified phone use segments.

Experiments to test audio ranging localisation in challenging environmental conditions are shown in Table 4.7. In total, 34 minutes of data were recorded, with 16 phone use segments during this time. Audio ranging localisation detected most of the ‘in driver’s area’ instances in the segments, but detection behaviour was more erratic. A visual example of this erratic behaviour is shown in Figure 4.13. However, ‘in driver’s area’ precision was still 99.55%. All 16 phone use segments were identified by phone inertial localisation; an average of approximately 88.00% was achieved for both precision and

Table 4.6. Method classification results for irregular phone pick-up experiments.

	Class	Precision (%)	Recall (%)	Num. of Samples
Audio ranging localisation classification	Not in driver's area	93.76	97.99	598
	In driver's area	98.83	96.30	1054
	Average / Total	96.30	97.15	1652
Phone inertial localisation classification	Not in driver's area	48.72	95.15	598
	In driver's area	94.01	43.17	1054
	Average / Total	71.36	69.16	1652
CNN image classification	No phone use	98.70	96.47	1020
	Talking on phone	98.11	93.56	388
	Texting	84.91	99.18	244
	Average / Total	93.91	96.40	1652
Combined image classification - Audio ranging localisation	No phone use	98.73	99.41	1020
	Talking on phone	99.73	93.56	388
	Texting	92.72	99.18	244
	Average / Total	97.06	97.38	1652
Combined image classification - Phone inertial localisation	No phone use	73.45	99.80	1020
	Talking on phone	100.00	37.89	388
	Texting	91.60	44.67	244
	Average / Total	88.35	60.79	1652

recall. Only the texting precision percentage was relatively low at 74.79% for CNN image classification. Even with the volatile behaviour of audio ranging localisation, its method combination still improved precision for both phone-related behaviours. Texting precision was improved by 9.34% and average precision by 2.77%. Phone inertial combination also improved texting and average precision by 6.02% and 2.00% respectively.

Results for an experiment that included driver and passenger phone pick-ups are shown in Table 4.8. The recording was done during the late afternoon, when 13.4 minutes of measurement data were recorded. There were six driver phone pick-ups, each followed by phone use segments, and six passenger phone pick-ups. Audio ranging and phone inertial localisation both had high precision and recall for all classes. Both methods correctly identified the six ‘in driver’s area’ segments. The six passenger phone pick-ups were also correctly identified as ‘not in driver’s area’ for both methods. Except for texting precision (70.33%), all other precision and recall percentages for CNN image classification were very high. Method combinations with both localisation methods significantly improved texting precision. An increase of respectively 28.13% and 26.59% was observed for the two methods.

Table 4.7. Method classification results in constrained audio pulse experiments.

	Class	Precision (%)	Recall (%)	Num. of Samples
Audio ranging localisation classification	Not in driver's area	71.45	98.88	534
	In driver's area	99.55	86.23	1532
	Average / Total	85.50	92.55	2066
Phone inertial localisation classification	Not in driver's area	81.94	82.40	534
	In driver's area	93.85	93.67	1532
	Average / Total	87.89	88.03	2066
CNN image classification	No phone use	99.17	94.38	1389
	Talking on phone	97.72	92.79	416
	Texting	74.79	100.00	261
	Average / Total	90.56	95.72	2066
Combined image classification - Audio ranging localisation	No phone use	96.87	97.98	1389
	Talking on phone	98.97	92.79	416
	Texting	84.13	87.36	261
	Average / Total	93.33	92.71	2066
Combined image classification - Phone inertial localisation	No phone use	97.96	96.62	1389
	Talking on phone	98.93	88.70	416
	Texting	80.81	100.00	261
	Average / Total	92.56	95.11	2066

Table 4.8. Method classification results for driver and passenger phone pick-up experiment.

	Class	Precision (%)	Recall (%)	Num. of Samples
Audio ranging localisation classification	Not in driver's area	97.79	100.00	532
	In driver's area	100.00	95.57	271
	Average / Total	98.90	97.79	803
Phone inertial localisation classification	Not in driver's area	95.50	95.68	532
	In driver's area	91.48	91.14	271
	Average / Total	93.49	93.41	803
CNN image classification	No phone use	100.00	96.13	671
	Talking on phone	100.00	98.53	68
	Texting	70.33	100.00	64
	Average / Total	90.11	98.22	803
Combined image classification - Audio ranging localisation	No phone use	100.00	100.00	671
	Talking on phone	100.00	98.53	68
	Texting	98.46	100.00	64
	Average / Total	99.49	99.51	803
Combined image classification - Phone inertial localisation	No phone use	98.97	99.85	671
	Talking on phone	100.00	89.71	68
	Texting	96.92	98.44	64
	Average / Total	98.63	96.00	803

Table 4.9. Method classification results for all conducted experiments combined.

	Class	Precision (%)	Recall (%)	Num. of Samples
Audio ranging localisation classification	Not in driver's area	89.65	99.31	2766
	In driver's area	99.56	93.13	4613
	Average / Total	94.61	96.22	7379
Phone inertial localisation classification	Not in driver's area	73.21	91.18	2766
	In driver's area	93.80	79.99	4613
	Average / Total	83.50	85.59	7379
CNN image classification	No phone use	98.35	94.96	4839
	Talking on phone	98.68	92.45	1536
	Texting	77.37	97.71	1004
	Average / Total	91.47	95.04	7379
Combined image classification - Audio ranging localisation	No phone use	97.74	99.07	4839
	Talking on phone	99.65	92.38	1536
	Texting	90.29	94.42	1004
	Average / Total	95.89	95.29	7379
Combined image classification - Phone inertial localisation	No phone use	89.69	98.47	4839
	Talking on phone	99.55	72.20	1536
	Texting	86.87	82.37	1004
	Average / Total	92.04	84.35	7379

Results for all experiments combined are shown in Table 4.9. In total, 122 minutes of measurement data were classified, which included 62 phone ‘in driver’s area’ segments. Audio ranging localisation achieved high-performance metric percentages, particularly ‘in driver’s area’ precision, which obtained 99.56%. Phone inertial localisation did not achieve the same high performance, but good performance was still produced with ‘in driver’s area’ precision of 93.80%. Except for texting precision (77.37%), CNN image classification of driver behaviour obtained high precision and recall percentages. Recall percentages were high for all classes, indicating almost all positive samples were identified. Very high ‘talking on phone’ precision (98.68%) demonstrates that there were very few false positive classifications when the driver was talking on the phone. Precision percentages for phone use related behaviour increased in both method combinations, therefore false positive detections were reduced. Audio ranging combination had a 12.92% increase in precision for texting, while phone inertial combination showed a 9.5% increase. Recall percentages for phone inertial method combination decreased for the ‘talking on phone’ and ‘texting’ classes. This was mainly due to ‘in driver’s area’ localisation segments that were misclassified. Only 51 out of 62 driver phone pick-ups were correctly identified by phone inertial localisation.

4.6 CHAPTER SUMMARY

In this chapter, the results for individual methods as well as method combinations were presented. The experimental design used to evaluate performance was provided for each method. Section 4.2 examined audio ranging localisation and included both control and natural pose experiments. Experiments were conducted in noisy and noise-free environments. Control experiments demonstrated that accurate distance computations could be generated, even in the presence of noise. A phone was held and placed in common positions for the driver's and passenger's sides of the vehicle. These natural pose experiments showed that precise classification of the general phone area is possible. Overall, the newly developed gradient CPD algorithm performed best. Music mixed with signal pulses did not have a significant impact on method accuracy. Phone inertial localisation results for a series of phone pick-ups were presented in Section 4.3. Accurate phone area classification was provided by the method, but classification of octant sections was inferior.

Classification results on samples of captured video frames are examined in Section 4.4. Samples were classified using the trained CNN. Network classification output showed that it can function in a variety of adverse lighting conditions with complex head poses and tilt angles. Conditions where misclassification might occur were demonstrated with a few examples. Comprehensive experimental results relating to CNN image classification and method combinations were presented in Section 4.5. Visual examples of method classification output as a function of time were provided. Examples were selected to illustrate certain aspects that affect method performance. Results and metrics obtained in a variety of test conditions were presented in tabular form. Audio ranging localisation achieved very high performance levels, even in difficult conditions. Phone inertial localisation did not perform as well, but still managed accurate phone area localisation. CNN image classification functioned well despite demanding and adverse lighting conditions. Accurate detection of texting was the most challenging. Image classification combined with localisation methods significantly improved precision of phone use behaviours by removing false positives.

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

In this work, the author aimed to develop and evaluate different driver phone use detection methods. Chapter 4 described the procedure used to evaluate the performance of each method. This chapter provides an interpretation and discussion of the significance of results obtained in Chapter 4. In Section 5.2, audio ranging phone localisation results are examined. Differences in CPD method performance are discussed and the effect of noise on system accuracy is evaluated. Phone inertial localisation results are described in Section 5.3. A review of results for CNN image classification and method combinations is presented in Section 5.4. A discussion on the approach used to evaluate method combinations is also provided. Finally, the significance of results compared to previous works is presented in Section 5.5.

5.2 AUDIO RANGING PHONE LOCALISATION

Control experimentation results showed that the fundamental algorithm approach could deliver accurate distance estimates. The newly designed gradient CPD method provided good accuracy compared to available Matlab functions. It had the lowest STD, irrespective of noise conditions, implying distance calculations remained mostly constant, with few abrupt changes. STD provides a measure of the variability between individual distance measurements. A lower value is ultimately better because it means that the CPD method provides more consistent distance measurements. Low STD between distance measurements with the phone held in the same position shows that algorithm output is consistent. Unlike Matlab CPD methods, this method is specifically designed to detect pulse onset in signals with certain characteristics. Distance errors observed were typically under 10 cm. Considering

the minimum vehicle width was 133 cm (Nissan Micra), this distance error is adequate for estimating approximate phone location. In noise-free environments distance errors were 6.62 cm averaged across all vehicles (shown in Figure 4.2(d)). Some of the measurement error can be attributed to human error when constructing and labelling measurement points. Additional error would also be introduced if the phone's microphone is not held precisely at the measurement point during capture. Taking into consideration distance error, STD and different noise conditions, gradient CPD provided the best overall performance.

Natural pose experiments confirmed that accurate phone localisation could be achieved and that gradient CPD is the best method to determine signal onset. Gradient CPD once again obtained the lowest STD between distance measurements regardless of the presence of noise. Small differences in distance measurement between cases of different noise levels illustrated the limited effect noise has on distance measurements (shown in Figure 4.4). High gradient CPD classification percentages, 97.7%, proved that precise localisation was achievable in typical phone use circumstances (shown in Figure 4.6). Negative SNR values were calculated for all vehicles when noise was present. Distance errors only increased slightly for the VW Polo, which had the highest SNR.

Collectively, evaluating results from all audio ranging localisation experiments shows that linear and mean CPD methods perform worst, while the two best performing Matlab methods are STD and RMS. However, gradient CPD outperforms Matlab methods in almost all circumstances. The main algorithm objective is to provide a location estimate of the phone to determine if it is in the driver's access area. The algorithm achieved this with high accuracy even in the presence of noise. Control and natural pose experiments were both adept at functioning in high-noise environments with relatively high SNRs.

This localisation approach utilises already existing vehicle infrastructure, removing the need for additional hardware installations. A drawback of this approach is that it occupies the vehicle stereo system while playing signal pulses. Consequently, music on the radio cannot be played simultaneously. A method for merging music with pulses will have to be developed or audio devices producing pulses external to the stereo system will have to be installed. An important benefit of this localisation approach is that it provides continuous monitoring of phone location. Phone inertial localisation only estimates location after a phone pick-up. A single incorrect classification would therefore result in a section of the trip with wrong location measurements.

5.3 PHONE INERTIAL LOCALISATION

Phone inertial localisation experimentation examined octant section and phone area accuracy by classifying 100 separate phone pick-ups. Phone area classification obtained high precision and recall percentages, but the more complex octant section classification did not perform as well. It is a more complex classification task to select one out of a possible eight octant sections than classifying phone movement into the passenger's or driver's access area. Identification of octant sections might not be particularly accurate, but the main method objective is to determine if the phone moved into the driver's area. It is able to do this with high accuracy of 91.00% (shown in Table 4.3).

Several external acceleration forces associated with driving activities exist. These include vehicle turning, acceleration, deceleration and driving on uneven roads. In addition to external acceleration forces, typical sensor error sources also contribute to drift, which can become uncontrollable if tracked for a long period without correction. Because of these factors, it is important to classify the initial phone motion when identifying a passenger or driver phone pick-up. Phone inertial tracking is not as accurate when initial phone movement from the reference point does not match the movement to the destination. For example, if the final phone destination is to the right, but the initial phone pick-up motion is to the left first, then localisation classification will be negatively affected.

A drawback of this method is that continued phone location tracking after removal from the reference point is not possible. Only phone pick-up motions are classified, meaning a segment of the trip could be misclassified until the phone is returned to the reference point. An advantage of this localisation approach is that no changes to vehicle infrastructure are necessary; only the magnetic phone holder, which acts as a reference point, is utilised. Low infrastructure requirements mean that it could easily be deployed to many vehicles and devices. A benefit phone inertial localisation has over audio ranging localisation is that it will function in the same way for all vehicles; the vehicle stereo system will not be a determining factor in method performance. Some vehicles, like the VW Polo, exhibited larger distance errors than the two other vehicles owing to the influence of the stereo system. The selected reference point is a constraint in this approach, as the method implementation will change according to the location of the reference point. This could be solved by selecting from a list of pre-defined reference points.

5.4 PERFORMANCE OF COMBINED PHONE USE DETECTION METHODS

Results obtained for CNN image classification are examined in isolation first and the implications of the results are discussed. Evaluation of results that include method combinations are then described.

5.4.1 CNN image classification evaluation

Results of the CNN in varied lighting conditions show that the trained network can regularise well to data on which it was not trained. It performed especially well in cases with the sunlight across the driver's face; few examples of this were encountered in the training data. Extreme light conditions still limit classification performance, but reasonable accuracy was nevertheless obtained during these periods (shown in Table 4.4). During periods of low light conditions such as late afternoon, shadows cast over the driver's eyes can trigger a driver 'texting' detection. In instances where the driver's face could not be identified, a classification of 'no phone use' was made. This would occur occasionally when sunlight entered the camera directly. Improved performance can be achieved if a larger number of in-vehicle environment backgrounds are included in the training images.

In all tests the trained CNN network is better at detecting accurate 'talking on the phone' driver behaviour compared to 'texting'. The primary reason for this is the similarity between the 'no phone use' and 'texting' classes. Most misclassifications occur between these two classes, as there are very subtle changes in behaviour that separate them, the main difference being a slight downward tilt of the head. For all cases, during CNN experimentation, average recall percentages were higher than the corresponding precision percentages. However, this was only due to low 'texting' precision percentages affecting the overall average. In nearly all cases 'no phone use' and 'talking on phone' precision percentages were higher than the respective recall percentages, while the opposite was true for 'texting'. For 'texting', false positive detections were produced more regularly compared to other classes.

Results from all tests combined provide a good indication of performance, as the tests include 122 minutes of classification data in a variety of circumstances (shown in Table 4.9). Considering these results, 'talking on phone' precision is very high (98.68%), while 'texting' precision is relatively lower (77.37%). During experimentation, improved lighting conditions during mid- and late afternoon

resulted in a large increase in precision and recall percentages for all three classes. Improvement was especially noticeable in 'texting' performance, with a large gain in precision performance. Precision can be regarded as a more meaningful performance metric compared to recall. Greater importance should be placed on having as few false phone use detections as possible. The balanced accuracy score, which is the same as the equally weighted average recall score, is reasonably high at 95.04%. It indicates overall method accuracy and avoids inflated scores, which are common in imbalanced datasets.

The camera point of view is an important factor in determining classification accuracy. Footage captured from a camera at a sideways angle did not perform well. For accurate classification the face has to be captured from the front. The network was not trained on images captured from unorthodox angles. Subjecting the network to images of phone use behaviour captured from different angles and vantage points will make it more resilient to changes in camera placement.

5.4.2 Method combination evaluation

Testing different methods in combination allowed individual and collective analysis. Experiments were conducted to replicate conditions as these would be found in real-world circumstances. Evaluation of method combinations is completed on a per-second basis, meaning a classification is produced each second for all three methods. Maintaining good performance on continuous classifications is more difficult than averaging the performance of single classifications together. This was the case when audio ranging and phone inertial localisation methods were tested individually. Continuous evaluation is more difficult because methods are exposed to a wide range of external factors. Singular tests can sometimes mask external factors that may influence method performance. For audio ranging localisation these factors could include additional noise sources that may compromise microphone recordings, such as noise generated by the vehicle engine and driving on the road, wind noise from the cooling system and open windows or partially blocking the phone microphone while holding it (this includes noise generated by blocking and unblocking the microphone). Phone inertial localisation tested individually produced results by examining the number of correct phone pick-ups, but a single misclassification of phone area in continual evaluation results in an entire segment of localisation classifications being incorrect.

Method classification output is directly related to the corresponding ground-truth value. This can affect performance especially for localisation methods and method combinations. A small misalignment in time can cause many incorrect classifications as a result of not being perfectly synchronised. Method combination phone use instances that should have been included could be discarded because of not falling inside the 'in driver's area' localisation segment. An example of this phenomenon is present in Figure 4.11 at approximately 900 seconds; a second or two of 'talking on phone' classification is discarded because phone inertial localisation is delayed by a small margin. This problem mainly affected phone inertial localisation performance.

In general, it was very difficult to affect audio ranging localisation negatively. Even when attempting to cover the microphone and introducing external noise, audio pulses could still be identified. Collectively considering all tests (shown in Table 4.9), a high precision percentage was achieved, especially for 'in driver's area', with 99.56%. Generally, precision and recall percentages for phone inertial localisation were lower than audio ranging localisation. This is mainly due to incorrect classifications when the phone is picked up, because a single incorrect classification results in an entire segment being misclassified. A visual illustration of this is shown in Figure 4.12. Another reason is a slight delay between the method output and the ground-truth value. Phone inertial localisation normally trails the ground-truth value by about 2 or 3 seconds (meaning it is triggered with a lag of about 2 seconds). This is most probably caused by small timing errors during the method alignment process. The accumulation of these small errors contributes to several classifications being labelled as incorrect. Phone inertial localisation functions best when the initial pick-up motion is in the same direction as the final destination. The most important metric is 'in driver's area' precision; it collectively obtained 93.80% (shown in Table 4.9), showing that phone inertial localisation still performs well. Precision for localisation methods was higher for 'in driver's area' than 'not in driver's area'. This indicates that localisation methods rarely produced false accusations of the phone being in the driver's access area. Lower recall percentages show that some true positive samples were not identified.

The two localisation methods indicate whether a phone is in close proximity to the driver, while CNN image classification detects actual instances of phone use behaviour. Localisation combined with CNN image classification should provide even more accurate identification of phone use behaviour. Experimentation results prove that method combination does improve performance, particularly for 'talking on phone' and 'texting' phone use behaviour. In extreme lighting conditions during the morning, 'texting' was improved by 23.58% when combined with audio ranging localisation and

by 19.96% when combined with phone inertial localisation (Table 4.4). Audio ranging localisation mostly outperforms phone inertial localisation when combined with CNN image classification. This is expected, as localisation using audio ranging is more accurate and provides continuous location updates, but good performance is still achieved by inertial localisation.

Different sets of experiments where the effect of human interference and/or environmental conditions were tested show that precision for 'talking on phone' and 'texting' phone use behaviour was improved in all cases that involved combination with a localisation method. Method combination considerably reduces the number of false positives; it creates a more robust system capable of functioning in difficult environments and provides greater confidence that detections are true cases of phone use behaviour. Analysing results from all experiments combined (shown in Table 4.9) shows that method combinations can detect driver phone use with great accuracy in a variety of conditions.

5.5 SIGNIFICANCE OF RESULTS COMPARED TO PREVIOUS WORKS

The significance of results when compared to results obtained in similar works are discussed. The performance of works that are similar to the methods developed is compared. Advantages and disadvantages associated with each method are described. Direct performance comparison between methods in different research works is not suitable, because experimentation is rarely conducted in the same manner. This is often a result of datasets not being publicly available, or such datasets not having been created previously.

5.5.1 Non-vision-based method comparison

Work previously done in [11] also produced a driver's or passenger's phone area classification utilising audio ranging. It obtained measurement errors that were within 2 cm and classification accuracy of 95% for a particular phone in a vehicle and 87% for a different phone in another vehicle. The audio ranging method developed with the associated gradient CPD approach obtained an average distance error of 6.62 cm in noise-free environments. Distance error was averaged over three vehicles in five positions for each vehicle. As stated in Section 5.2, a few additional centimetres of error could be attributed to human error during experimentation. Natural pose experiments achieved overall classification accuracy of 97.7%. Audio ranging localisation obtained precision and recall percentages of 94.61% and 96.22%

respectively during a cumulative two hours of varied tests. Good classification results for these tests are not as easily achievable, as explained in Section 5.4.2. The previous work made use of an STFT to filter signal pulses around target frequencies, whereas the method developed in this work utilises a CWT to filter the signal. Benefits of using CWT are discussed in Section 3.2 - it can provide improved time-frequency resolution. The method developed in [11] can localise the phone to the front and back seats, while only front seat localisation is provided in this work.

To date, no known method that localises a phone based on tracking phone inertial movement from a reference point has been developed. It is a unique solution that requires very little additional hardware (only a fixed reference point); embedded phone inertial sensors are primarily used. As shown in experimental results, it can achieve average precision and recall percentages of 83.50% and 85.59% respectively over two hours of varied tests. It also obtained approximately 91% for precision and recall when classifying phone location from 100 phone pick-ups. Other methods that also utilise phone sensors exist [4, 15]. Another method which also localises the phone to the driver's or passenger's side uses phone sensors to calculate vehicle dynamics [4]; acceleration experienced by the phone will vary according to its position. This method achieved 90% classification accuracy. However, it requires installation of an additional acceleration measuring device that acts as a reference.

5.5.2 Vision-based method comparison

Vision-based phone use detection methods are appealing because phone use instances can be identified as soon as they occur. Four notable works that utilised vision-based techniques are analysed [6–9]. All methods except [7] have a camera installed inside the vehicle. The method in [7] has a camera directed at the vehicle from above. A summarised comparison of these methods, along with the methods developed in this work, is shown in Table 5.1.

In [6], raw pixels and HOG features were manually extracted. The trained CNN in this work is an end-to-end model that automatically learns discriminating features. Benefits of using a CNN over hand-engineered features such as HOG are described in Section 2.4. Frames where no faces could be detected were discarded in [6]. In this work all frames were classified, even if a face could not be detected (in this case it was classified as 'no phone use'). Discarding frames is advantageous to performance metrics because cases of phone use that should have been detected are omitted. Test data

in [6] did not include harsh illumination conditions or excessive head pose variations. Only ‘talking on phone’ driver behaviour was detected in [6]. The methods developed in this work were able to detect ‘talking on phone’ and ‘texting’ driver behaviour.

Hand-engineered features were also extracted in [9]. Features included the percentage of hand and moment of inertia. This method required the selection of suitable threshold values. It also relied on segmentation using skin pixel values, which caused confusion when drivers’ faces were the same colour as their clothing or when their shoulders were visible in the frame. Instances where the driver was talking on the phone were identified with 83.81% accuracy and cases of no phone use obtained an accuracy of 92.74%. Continuous, per-second driver behaviour classification was not performed as in the case of the methods in this work. Only ‘talking on phone’ driver behaviour could be detected by [9].

All other methods extracted hand-engineered features from the training data. Optimal features that provide the most distinguishable characteristics are not known. A more intuitive approach was followed in this work, where features were learned automatically by the CNN. Methods proposed in previous works often rely on numerous interconnected vision systems, which can complicate the system unnecessarily. No known previous work has combined a phone localisation approach with a vision-based approach to improve accuracy and robustness. The combination of these methods provides a significant performance benefit, as shown in the results. The prominence of driver texting is increasing, yet none of the other listed vision-based methods is capable of detecting this driver behaviour.

Table 5.1. Comparison of vision-based methods for detecting driver phone use.

Algorithm	**Approach**	**Test Procedure**	**Performance**
Seshadri *et al* [6]	Tracking driver facial landmarks; extract a region of interest; region is classified using trained classifiers; raw pixel and HOG features used to train classifiers. Only ‘talking on phone’ behaviour detectable.	Test data consisted of 13 023 frames (9 288 frames of ‘no phone use’ and 3 735 frames of ‘talking on phone’); equates to 14.5 minutes of footage at 15 FPS; 30 subjects were captured; harsh illumination and excessive head pose variations were not encountered.	AdaBoost classifier with HOG features obtained 93.86% classification accuracy.

Continue on the next page

Table 5.1. Comparison of vision-based methods for detecting driver phone use (cont.).

Algorithm	Approach	Test Procedure	Performance
Berri *et al* [9]	Pattern recognition system used for classification; movement detection system determines pattern recognition parameters. Threshold values and buffer sizes need to be determined manually. Only 'talking on phone' behaviour detectable.	Test data consisted of five videos recorded in real-world conditions.	Driver 'talking on phone' accuracy of 83.81%, 'no phone use' accuracy of 92.74%.
Zhang *et al* [8]	Features collected from the driver's face, mouth and hand. Features provided as input to a hidden conditional random fields model. Only 'talking on phone' behaviour detectable.	104 video segments of unknown length were captured from 11 subjects. Illumination conditions and head pose variations are unknown. Figures provided have no harsh lighting conditions or excessive pose variations in test data.	Accuracy of 91.20% achieved when suitable window size and hidden state parameters were chosen.
Artan *et al* [7]	Images of vehicle captured from above. Driver's face region identified from windshield image. Image descriptors extracted from a region around the face and a classifier trained. Only 'talking on phone' behaviour detectable. Method not suitable for use inside a vehicle.	1 500 images from a public roadway were collected and analysed.	Achieved accuracy of 86.19%
Methods developed in this work	CNN based on ResNet architecture was constructed and trained. Discriminating features are learned automatically. 'Talking on phone' and 'texting' driver behaviour detectable. Image classification performed by CNN was also combined with localisation methods that were developed.	Test data included 122 minutes of captured footage, test data included harsh illumination conditions and excessive head pose variations. A single subject was used during experimentation.	Audio ranging localisation combined with CNN image classification achieved the best performance: 'talking on phone' precision: 99.65%, recall: 92.38%; 'texting' precision: 90.29%, recall: 94.42%.

5.6 CHAPTER SUMMARY

This chapter discussed the results that were obtained during evaluation of the various methods. Results for each detection method were interpreted and the advantages and disadvantages associated with each method approach were examined. The most important factors that can influence method performance were highlighted. Methods developed in this work were compared with similar previously developed methods where applicable and the significance of results concerning the driver phone use detection research field was described. A table that summarises the performance obtained from various vision-based methods, including methods developed in this work, was provided.

CHAPTER 6 CONCLUSION

6.1 SUMMARY OF WORK CONDUCTED

Road accidents and fatalities as a result of driver distraction during phone use is a serious concern. The use of phones while driving recklessly limits a driver's attention on the primary task of driving. Phone use while driving is becoming more common in society, therefore a system that can accurately detect and warn the driver of this reckless behaviour is necessary. This includes not just detecting a driver talking on the phone, but also cases of texting. Infrastructure requirements associated with each detection method need to be considered. Requirements that are too demanding will limit the method's adoption rate. Drivers could be incentivised to install these systems (e.g. by offering a lower insurance premium for not using a phone while driving). This would ultimately lower phone usage while driving and reduce fatalities.

This work set out to address three main research questions relating to the detection of driver phone use, namely: 1. Is it possible to develop a phone localisation system capable of accurately localising a phone to the driver's or passenger's side of a vehicle with minimal infrastructure requirements? 2. Is it possible to develop a vision-based phone use detection method that is capable of managing harsh illumination changes and excessive head pose variations? Furthermore, will such a system be able to detect not just a driver taking on the phone, but also texting? 3. How resilient will the implemented methods be to variable environmental conditions? Will a system that combines both vision and non-vision-based methods be able to obtain greater accuracy and robustness? The methodology for answering these questions included a literature study which, reviewed methods that had previously been proposed.

This work focused on the implementation and combination of three phone use detection methods. Two

of the methods provide phone localisation inside the vehicle, while the third vision-based method monitors driver behaviour and identifies instances of phone use. Phone localisation is helpful, as it indicates if a phone might be in the driver's access area. The vision-based method continually monitors the driver and detects talking and texting behaviour in real-time.

Comprehensive experimentation was conducted to test system performance in a wide variety of conditions and circumstances. Experiments were first conducted individually for each method. All methods were then tested together collectively. Collective evaluation involved numerous vehicle trips where factors such as harsh lighting conditions, head pose variations, increase in the ambient noise level and irregular phone pick-ups were tested. Experiments were chosen to evaluate method performance in real-world conditions. All experiments combined accounted for 122 minutes of collected data.

6.2 CONCLUSIONS FROM RESEARCH QUESTIONS

The first research objective was to determine if the development of a phone localisation system capable of accurately localising a phone to the driver's or passenger's side of a vehicle with minimal infrastructure requirements was possible. A new localisation method was implemented, which uses a fixed reference point and phone inertial sensors to track phone movement when the phone is picked up. It then classifies the phone as being on either the passenger's or driver's side of the vehicle. A localisation method previously proposed [11] was also implemented; some alterations to the original approach were made. Both these methods have minimal infrastructure requirements. Phone inertial localisation only utilises a magnetic pad and inertial phone sensors, while audio ranging localisation only uses the vehicle's stereo system and the phone microphone. Advantages and disadvantages relating to each method approach were provided in Chapter 5. Accurate localisation can be performed by both methods, as shown in individual and collective experimental results.

The second research objective was to determine if a vision-based phone use detection method could be implemented and could function in harsh illumination conditions and with excessive head pose variations. The system also had to detect not just talking on the phone behaviour, but also texting. A CNN model was successfully trained to detect both phone use behaviours in a variety of conditions. Real-world experiments were conducted with sunlight entering the vehicle from multiple angles (experiments were conducted during the morning, mid-afternoon and late afternoon). The driver

displayed a wide range of head poses during experimentation. Precision is the most important evaluation metric, as it indicates the number of false positives. High precision percentages were achieved from the cumulative 122 minutes of captured video footage; 'talking on phone' yielded precision of 98.68%, while 'texting' precision of 77.37% was achieved. Texting precision is lower owing to its high similarity to the 'no phone use' class. Previous vision-based methods [6–9] were only able to detect a driver talking on the phone. Detection of texting is important, because it is the most dangerous distraction, as it diverts a driver's attention for an extended period of time.

The third research objective was to determine if a combination of methods is justifiable by producing greater accuracy and robustness. The resilience of implemented methods to variable environmental factors also had to be determined. Each method achieved good performance when evaluated individually, but significant performance benefits were provided when localisation methods were combined with the vision-based method. In harsh illumination conditions during the morning, 'texting' precision was improved by 23.6% when combined with audio ranging localisation and by 20.0% when combined with phone inertial localisation. For collective evaluation of all 122 minutes of data, CNN classification combined with audio ranging improved 'texting' from 77.37% to 90.29%; method combination with phone inertial localisation improved 'texting' from 77.37% to 86.87%. In both combinations, 'talking on phone' precision was increased to almost 100% (an increase of approximately 1%). All methods were able to function with high accuracy even when numerous environmental and human factors were altered.

The proposed solutions furthered the development of driver phone use detection systems. Driver mobile phone use is a great problem and will remain one until effective phone detection and prevention systems are implemented. The proposed solutions attempt to contribute to the solutions that have already been developed. Results from experimentation show that very accurate localisation and driver behaviour identification can be provided by the methods developed. Audio ranging was the most accurate localisation method; it obtained overall average precision of 94.61% and recall of 96.22%. CNN image classification combined with audio ranging localisation achieved the highest accuracy when detecting driver phone use behaviour. It obtained overall average precision of 95.89% and recall of 95.29%. A comparison of methods developed in this work to those in previous works illustrates that the new implementations provide several benefits and performance increases.

6.3 FUTURE WORK

Detection of driver phone use forms part of the field of detecting driver distraction. The vision-based method can currently detect a driver talking on the phone or texting. Additional driver behaviours can be included in the system. This could improve the accuracy of current phone use behaviour, as fewer false positive detections will be reported. Additional behaviour could include a driver drinking water, eating, smoking or other common actions performed while driving. The vision-based method can be made more robust to changes in camera placement by including images of phone use behaviour captured from different angles in the training dataset of the model. If images captured from unorthodox angles are not included in the training dataset, the CNN will not be able to interpret these images when they are provided as input. Detection of 'talking on phone' and 'texting' phone use behaviour is currently not possible at night. An NIR camera can be used to capture images in the dark; however, all images used in network training will have to be in the NIR format. Another approach could be followed where method combinations are only used during periods of adequate light, while localisation methods are solely used in low-light conditions. Lastly, further experimentation can be completed with more subjects to verify the method performance observed in this work.

REFERENCES

[1] U.S. Department of Transportation, "Traffic safety facts research note: summary of statistical findings. Distracted Driving in Fatal Crashes, 2017," 2019. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812700

[2] J. K. Caird, C. R. Willness, P. Steel, and C. Scialfa, "A meta-analysis of the effects of cell phones on driver performance," *Accident Analysis & Prevention*, vol. 40, no. 4, pp. 1282–1293, Jul. 2008.

[3] N. Dragutinovic and D. Twisk, "Use of mobile phones while driving - effects on road safety," *SWOV Institute, Leidschendam*, May 2005.

[4] Y. Wang, Y. J. Chen, J. Yang, M. Gruteser, R. P. Martin, H. Liu, L. Liu, and C. Karatas, "Determining driver phone use by exploiting smartphone integrated sensors," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1965–1981, Aug. 2016.

[5] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, Jun. 2013, pp. 41–54.

[6] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2015, pp. 35–43.

[7] Y. Artan, O. Bulan, R. P. Loce, and P. Paul, "Driver cell phone usage detection from HOV/HOT NIR images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2014, pp. 225–230.

[8] X. Zhang, N. Zheng, F. Wang, and Y. He, "Visual recognition of driver hand-held cell phone use based on hidden CRF," in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, Jul. 2011, pp. 248–251.

[9] R. Berri, F. Osório, R. Parpinelli, and A. Silva, "A hybrid vision system for detecting use of mobile phones while driving," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2016, pp. 4601–4610.

[10] A. Fernández, R. Usamentiaga, J. L. Carús, and R. Casado, "Driver distraction using visual-based sensors and algorithms," *Sensors*, vol. 16, no. 11, p. 1805, Oct. 2016.

[11] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing driver phone use with acoustic ranging through car speakers," *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1426–1440, Sep. 2012.

[12] C. Bo, X. Jian, X.-Y. Li, X. Mao, Y. Wang, and F. Li, "You're driving and texting: detecting drivers using personal smart phones by leveraging inertial sensors," in *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*. ACM, Sep. 2013, pp. 199–202.

[13] H. Park, D. Ahn, M. Won, S. H. Son, and T. Park, "Poster: Are you driving?: non-intrusive driver detection using built-in smartphone sensors," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. ACM, Sep. 2014, pp. 397–400.

[14] J. M. Rodríguez-Ascariz, L. Boquete, J. Cantos, and S. Ortega, "Automatic system for detecting driver use of mobile phones," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 673–681, Aug. 2011.

[15] Y. Li, G. Zhou, Y. Li, and D. Shen, "Determining driver phone use leveraging smartphone sensors," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 16 959–16 981, 2016.

[16] T. Song, X. Cheng, H. Li, J. Yu, S. Wang, and R. Bie, "Detecting driver phone calls in a moving vehicle based on voice features," in *The 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*. IEEE, Apr. 2016, pp. 1–9.

[17] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2014–2027, Aug. 2015.

[18] M. Kutila, M. Jokela, G. Markkula, and M. R. Rue, "Driver distraction detection with a camera vision system," in *2007 IEEE International Conference on Image Processing*, vol. 6, Sep. 2007, pp. VI – 201–VI – 204.

[19] M. Sodhi, B. Reimer, J. Cohen, E. Vastenburg, R. Kaars, and S. Kirschenbaum, "On-road driver eye movement tracking using head-mounted devices," in *Proceedings of the 2002 symposium on Eye tracking research & applications*, Mar. 2002, pp. 61–68.

[20] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.

[21] J. Wahlström, I. Skog, and P. Händel, "Smartphone-based vehicle telematics: A ten-year anniversary," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2802–2825, Oct. 2017.

[22] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *arXiv preprint arXiv:1704.06053*, 2017.

[23] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–536, 2007.

[24] J. D. Hol, T. B. Schön, H. Luinge, P. J. Slycke, and F. Gustafsson, "Robust real-time tracking by fusing measurements from inertial and vision sensors," *Journal of Real-Time Image Processing*, vol. 2, no. 2-3, pp. 149–160, 2007.

[25] A. Martinelli, "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.

[26] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultrawideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293–1303, 2015.

[27] S. Pittet, V. Renaudin, B. Merminod, and M. Kasser, "UWB and MEMS based indoor navigation," *The Journal of Navigation*, vol. 61, no. 3, pp. 369–384, 2008.

[28] J. A. Corrales, F. Candelas, and F. Torres, "Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, Mar. 2008, pp. 193–200.

[29] J. D. Hol, "Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS," Ph.D. dissertation, Linköping University Electronic Press, 2011.

[30] O. Maklouf and A. Adwaib, "Performance evaluation of GPS/INS main integration approach," *World Acad Sci Eng Technol Int J Mech Aerosp Ind Mechatron Eng*, vol. 8, no. 2, pp. 476–484, 2014.

[31] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, "Position estimation approach by Complementary Filter-aided IMU for indoor environment," in *2013 European Control Conference (ECC)*. IEEE, Jul. 2013, pp. 4208–4213.

[32] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in *WISP 2009 IEEE International Symposium on Intelligent Signal Processing*. IEEE, Aug. 2009, pp. 37–42.

[33] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th International Conference on Ubiquitous Computing*. ACM, Sep. 2008, pp. 114–123.

[34] W.-C. Bang, W. Chang, K.-H. Kang, E.-S. Choi, A. Potanin, and D.-Y. Kim, "Self-contained spatial input device for wearable computers," in *Proceedings of the Seventh IEEE International Symposium on Wearable Computers*. IEEE Computer Society, Oct. 2003, p. 26.

[35] C. C. Tsang, "Error reduction techniques for a mems accelerometer-based digital input device," Ph.D. dissertation, The Chinese University of Hong Kong, 2008.

[36] O. J. Woodman, "An introduction to inertial navigation," *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, vol. 14, p. 15, 2007.

[37] J. Du, "Signal processing for MEMS sensor based motion analysis system," Ph.D. dissertation, Mälardalen University, 2016.

[38] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, "Survey of motion tracking methods based on inertial sensors: a focus on upper limb human motion," *Sensors*, vol. 17, no. 6, p. 1257, 2017.

[39] E. Foxlin *et al.*, "Motion tracking requirements and technologies," *Handbook of Virtual Environment Technology*, vol. 8, pp. 163–210, 2002.

[40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[42] D. Lukac, M. Milic, and J. Nikolic, "From artificial intelligence to augmented age an overview," in *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*. IEEE, May 2018, pp. 100–103.

[43] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[44] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[45] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[46] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, Dec. 2001, pp. I–I.

[47] R. M. Haralick, K. Shanmugam *et al.*, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, no. 6, pp. 610–621, 1973.

[48] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 1997, pp. 762–768.

[49] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, vol. 2. IEEE, Oct. 2005, pp. 1508–1515.

[50] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Fourth Alvey Vision Conference*, vol. 15, no. 50. CiteSeer, 1988, pp. 147–151.

[51] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2. IEEE, Sep. 1999, pp. 1150–1157.

[52] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[53] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Nov. 2011, pp. 2564–2571.

[54] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *European Conference on Computer Vision*. Springer, 2010, pp. 778–792.

[55] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, Jun. 2005, pp. 886–893.

[56] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *2011 IEEE International Conference on Computer Vision*. IEEE, Nov. 2011, pp. 89–96.

[57] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[58] A. Rosebrock, *Deep Learning for Computer Vision with Python*, 2nd ed. PyImageSearch.com, 2019.

[59] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazare, T. Bui, and J. Collomosse, "Everything you wanted to know about deep learning for computer vision but were afraid to ask," in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*. IEEE, Oct. 2017, pp. 17–41.

[60] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[61] S. Scholl, "Exact Signal Measurements using FFT Analysis," University of Kaiserslautern, Tech. Rep., 2016. [Online]. Available: https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/4293

[62] P. Rajmic, Z. Prusa, and C. Wiesmeyr, "Computational cost of chirp Z-transform and Generalized Goertzel algorithm," in *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, Sep. 2014, pp. 1004–1008.

[63] P. Sarin and P. Dabas, "Time-Frequency Spectral Analysis of Single-Channel Earthquake Data for P-Wave Detection." *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 7, no. 2, pp. 893–895, 2016.

[64] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[65] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis," *Bulletin of the American Meteorological Society*, vol. 79, no. 1, pp. 61–78, 1998.

[66] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 339–367, 2017.

[67] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[68] MathWorks, "Find abrupt changes in signal - MATLAB findchangepts." [Online]. Available: https://www.mathworks.com/help/signal/ref/findchangepts.html

[69] J. Wahlström, I. Skog, P. Händel, and A. Nehorai, "Imu-based smartphone-to-vehicle positioning," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 139–147, Jun. 2016.

[70] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE, Jun. 2011, pp. 1–7.

[71] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[72] x-io Technologies Limited, "Open source IMU and AHRS algorithms." [Online]. Available: http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/

[73] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," University of Bristol, Tech. Rep., 2010.

[74] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[75] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[77] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.

[78] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[79] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, Jul. 2011.

[80] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[81] P. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[82] T. F. Vieira, A. Bottino, A. Laurentini, and M. De Simone, "Detecting siblings in image pairs," *The Visual Computer*, vol. 30, no. 12, pp. 1333–1345, Dec 2014. [Online]. Available: https://doi.org/10.1007/s00371-013-0884-3

[83] University of Essex, "Face recognition data." [Online]. Available: https://cswww.essex.ac.uk/mv/allfaces/

[84] S. Tadwalkar, "scrapingimages," 2018. [Online]. Available: https://github.com/sushrutt12/scrapingimages

[85] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, Oct. 2007.

[86] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2018, pp. 4510–4520.