

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

Multimodal Machine Learning for Intelligent Mobility

by
Aubrey James Roche

Doctoral Thesis

**Submitted in partial fulfilment of the requirements for the award of Doctor of
Philosophy**

Institute of Digital Technologies, Loughborough University London

© Aubrey James Roche

May 2020

Dedicate to my partner Sádhbh & my son Ruadhán
– *Nunc scio quid sit amor, semper gratiam habebo.*

Ever tried. Ever failed.
No matter. Try Again.
Fail again. Fail better.
(Samuel Beckett)



**Certificate of Originality
Thesis Access Conditions and Deposit Agreement**

Students should consult the guidance notes on the electronic thesis deposit and the access conditions in the University’s Code of Practice on Research Degree Programmes

Author: Jamie Roche

Title: Multimodal Machine Learning for Intelligent Mobility

I Jamie Roche, 2 Herbert Tec, Herbert Road, Bray, Co Wicklow, Ireland, “the Depositor”, would like to deposit Multimodal Machine Learning for Intelligent Mobility, hereafter referred to as the “Work”, once it has successfully been examined in Loughborough University Research Repository

Status of access OPEN / ~~RESTRICTED~~ / ~~CONFIDENTIAL~~

Moratorium Period: 3 years 3 months years, ending 13 / Dec 2019

Status of access approved by (CAPITALS)

Supervisor (Signature) VARUNA DE SILVA

School of: Loughborough University London

Author's Declaration *I confirm the following:*

CERTIFICATE OF ORIGINALITY
This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work therein has been submitted to this or any other institution for a degree

NON-EXCLUSIVE RIGHTS
The licence rights granted to Loughborough University Research Repository through this agreement are entirely non-exclusive and royalty free. I am free to publish the Work in its present version or future versions elsewhere. I agree that Loughborough University Research Repository administrators or any third party with whom Loughborough University Research Repository has an agreement to do so may, without changing content, convert the Work to any medium or format for the purpose of future preservation and accessibility.

DEPOSIT IN LOUGHBOROUGH UNIVERSITY RESEARCH REPOSITORY
I understand that open access work deposited in Loughborough University Research Repository will be accessible to a wide variety of people and institutions - including automated agents - via the World Wide Web. An electronic copy of my thesis may also be included in the British Library Electronic Theses On-line System (EThOS).

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

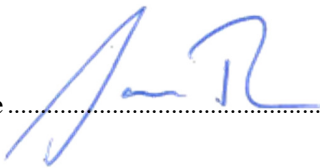
I understand that once the Work is deposited, a citation to the Work will always remain visible. Removal of the Work can be made after discussion with Loughborough University Research Repository, who shall make best efforts to ensure removal of the Work from any third party with whom Loughborough University Research Repository has an agreement. Restricted or Confidential access material will not be available on the World Wide Web until the moratorium period has expired.

- That I am the author of the Work and have the authority to make this agreement and to hereby give Loughborough University Research Repository administrators the right to make available the Work in the way described above.
- That I have exercised reasonable care to ensure that the Work is original, and does not to the best of my knowledge break any UK law or infringe any third party’s copyright or other Intellectual Property Right. I have read the University’s guidance on third party copyright material in theses.
- The administrators of Loughborough University Research Repository do not hold any obligation to take legal action on behalf of the Depositor, or other rights holders, in the event of breach of Intellectual Property Rights, or any other right, in the material deposited.

The statement below shall apply to ALL copies:

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Restricted/confidential work: All access and any copying shall be strictly subject to written permission from the University Dean of School and any external sponsor, if any.

Author's signature  Date 13/12/2019.....

user’s declaration: for signature during any Moratorium period (Not Open work):			
<i>I undertake to uphold the above conditions:</i>			
Date	Name (CAPITALS)	Signature	Address

Abstract

Scientific problems are solved by finding the optimal solution for a specific task. Some problems can be solved analytically while other problems are solved using data driven methods. The use of digital technologies to improve the transportation of people and goods, which is referred to as intelligent mobility, is one of the principal beneficiaries of data driven solutions. Autonomous vehicles are at the heart of the developments that propel Intelligent Mobility. Due to the high dimensionality and complexities involved in real-world environments, it needs to become commonplace for intelligent mobility to use data-driven solutions. As it is near impossible to program decision making logic for every eventuality manually. While recent developments of data-driven solutions such as deep learning facilitate machines to learn effectively from large datasets, the application of techniques within safety-critical systems such as driverless cars remain scarce.

Autonomous vehicles need to be able to make context-driven decisions autonomously in different environments in which they operate. The recent literature on driverless vehicle research is heavily focused only on road or highway environments but have discounted pedestrianized areas and indoor environments. These unstructured environments tend to have more clutter and change rapidly over time. Therefore, for intelligent mobility to make a significant impact on human life, it is vital to extend the application beyond the structured environments. To further advance intelligent mobility, researchers need to take cues from multiple sensor streams, and multiple machine learning algorithms so that decisions can be robust and reliable. Only then will machines indeed be able to operate in unstructured and dynamic environments safely. Towards addressing these limitations, this thesis investigates data driven solutions towards crucial building blocks in intelligent mobility. Specifically, the thesis investigates multimodal sensor data fusion, machine learning, multimodal deep representation learning and its application of intelligent mobility. This work demonstrates that mobile robots can use multimodal machine learning to derive driver policy and therefore make autonomous decisions.

To facilitate autonomous decisions necessary to derive safe driving algorithms, we present an algorithm for free space detection and human activity recognition. Driving these decision-making algorithms are specific datasets collected throughout this study. They include the Loughborough London Autonomous Vehicle dataset, and the Loughborough London Human Activity Recognition dataset. The datasets were collected using an autonomous

platform design and developed in house as part of this research activity. The proposed framework for Free-Space Detection is based on an active learning paradigm that leverages the relative uncertainty of multimodal sensor data streams (ultrasound and camera). It utilizes an online learning methodology to continuously update the learnt model whenever the vehicle experiences new environments. The proposed Free Space Detection algorithm enables an autonomous vehicle to self-learn, evolve and adapt to new environments never encountered before. The results illustrate that online learning mechanism is superior to one-off training of deep neural networks that require large datasets to generalize to unfamiliar surroundings.

The thesis takes the view that human should be at the centre of any technological development related to artificial intelligence. It is imperative within the spectrum of intelligent mobility where an autonomous vehicle should be aware of what humans are doing in its vicinity. Towards improving the robustness of human activity recognition, this thesis proposes a novel algorithm that classifies point-cloud data originated from Light Detection and Ranging sensors. The proposed algorithm leverages multimodality by using the camera data to identify humans and segment the region of interest in point cloud data. The corresponding 3-dimensional data was converted to a Fisher Vector Representation before being classified by a deep Convolutional Neural Network. The proposed algorithm classifies the indoor activities performed by a human subject with an average precision of 90.3%. When compared to an alternative point cloud classifier, PointNet[1], [2], the proposed framework outperformed on all classes.

The developed autonomous testbed for data collection and algorithm validation, as well as the multimodal data-driven solutions for driverless cars, is the major contributions of this thesis. It is anticipated that these results and the testbed will have significant implications on the future of intelligent mobility by amplifying the developments of intelligent driverless vehicles.

Acknowledgements

First and foremost, I would like to thank my supervisors, Varuna De Silva and Ahmet Kondo, for their unwavering support, invaluable research insights, ideas, good humour and help in times of crisis. It has been hugely rewarding being a member of the Institute of Digital Technologies, not least because of the fantastic team they have put together: thanks to everyone for the discussions, arguments, and adventures along the way!

My partner Sadhbh and my son Ruadhán deserve special recognition: I would not have made it this far without the two of them. They have made my time at Loughborough University London and our life on water more enjoyable. Finally, I would like to thank my family, who have been there to support me from the beginning, always encouraging me when I needed to escape from London, no matter how infrequently it was.

Monday, 4 May 2020

Jamie Roche

Contents

THESIS ACCESS CONDITIONS AND DEPOSIT AGREEMENT	III
ABSTRACT	V
ACKNOWLEDGEMENTS.....	VII
LIST OF FIGURES	XI
LIST OF TABLES.....	XVII
LIST OF ABBREVIATIONS	XVIII
CHAPTER 1 MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY	1
1.1 INTRODUCTION	1
1.2 CONTEXT & MOTIVATION	4
1.3 OBJECTIVES	7
1.4 CONTRIBUTIONS	8
1.5 PUBLICATIONS	9
1.6 THESIS OUTLINE.....	10
CHAPTER 2 DATA-DRIVEN MACHINE INTELLIGENCE	12
2.1 INTRODUCTION	12
2.2 IMMINENT NERVOUS ACTIVITY A TEMPLATE FOR ARTIFICIAL INTELLIGENCE ..	14
2.2.1 NEURAL PLASTICITY	15
2.3 FUNDAMENTALS OF MACHINE LEARNING.....	17
2.3.1 MACHINE LEARNING PARADIGMS AND METHODS.....	19
A. SUPERVISED LEARNING	21
B. SEMI-SUPERVISED LEARNING	22
C. UNSUPERVISED LEARNING.....	24
2.3.2 REGRESSION & THE SUPPORT VECTOR MACHINE	25
A. SUPPORT VECTOR KERNEL	28
B. HYPERPLANES	29
2.3.3 THE ARTIFICIAL NEURAL NETWORKS	29
A. FEEDFORWARD NEURAL NETWORKS	30
B. CONVOLUTIONAL NEURAL NETWORKS	31
C. ACTIVATION FUNCTIONS	36
D. LOSS FUNCTION	39
E. BACKPROPAGATION.....	40
2.4 MULTIMODAL MACHINE LEARNING.....	41
2.5 APPLICATIONS OF MACHINE LEARNING IN INTELLIGENT MOBILITY.....	42
2.6 SUMMARY.....	44
CHAPTER 3 ARCHITECTURE FOR AN INTELLIGENT MOBILE ROBOT	47
3.1 INTRODUCTION	47
3.2 KEY INTELLIGENT MOBILITY DEVELOPMENTS.....	49

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

3.3 INTELLIGENT MOBILITY DATASETS.....	52
3.3.1 AUTONOMOUS VEHICLE DATASETS	52
3.3.2 HUMAN ACTIVITY RECOGNITION DATASETS	54
3.4 PROBLEM DEFINITION	56
3.5 TECHNICAL PARAMETERS OF AN AUTONOMOUS PLATFORM	57
3.5.1 PLATFORM & DATASET REQUIREMENTS.....	58
3.6 AN AUTONOMOUS MOBILE ROBOTIC PLATFORM	60
3.6.1 THE SENSING LAYER.....	62
3.6.2 THE APPLICATION LAYER	64
A. ULTRASOUND-BASED CONTROL	66
B. CAMERA-BASED CONTROL	67
C. OPERATOR-BASED CONTROL	67
3.6.3 EXPERIMENTAL SETUP	71
3.7 SENSOR DATA REPRESENTATIONS	73
3.7.1 RGB DATA	73
3.7.2 RGB-D DATA	74
3.7.3 POINT CLOUD DATA	75
3.7.4 ULTRASONIC DEPTH DATA	75
3.8 DATA COLLECTION	76
3.8.1 LOUGHBOROUGH AUTONOMOUS VEHICLE DATASET.....	77
3.8.2 LOUGHBOROUGH HUMAN ACTIVITY RECOGNITION DATASET	81
3.9 SUMMARY	84
CHAPTER 4 A SELF-EVOLVING FREE SPACE DETECTION MODEL	86
4.1 INTRODUCTION.....	86
4.2 PROBLEM DEFINITION	87
4.2.1 STATE-OF-THE-ART FREE SPACE DETECTION	88
4.2.2 MOTIVATIONS AND REQUIREMENTS	89
4.3 GEOMETRIC ALIGNMENT OF SENSOR DATA.....	91
4.4 SELF-EVOLVING FREE SPACE DETECTION	93
4.4.1 IMAGE BASED FREE SPACE DETECTION	96
4.4.2 ULTRASOUND-BASED FREE SPACE DETECTION	96
4.4.3 ONLINE ACTIVE LEARNING FOR FREE-SPACE DETECTION	97
4.5 RESULTS AND DISCUSSION.....	98
4.5.1 DATASET.....	98
4.5.2 PERFORMANCE.....	100
A. ONLINE ACTIVE ML PERFORMANCE.....	102
B. DEEPLABV3+ PERFORMANCE.....	103
4.5.3 VISUAL RESULTS	104
4.6 SUMMARY	107

CHAPTER 5 A MULTIMODAL FISHER VECTOR NETWORK FOR HUMAN ACTIVITY RECOGNITION109

5.1 INTRODUCTION 109

5.2 PROBLEM DEFINITION..... 111

5.2.1 3D POINT CLOUD MACHINE LEARNING..... 112

5.2.2 MOTIVATION AND CONTEXT 114

5.3 POINT CLOUD REPRESENTATION LEARNING FOR ACTIVITY RECOGNITION..... 115

5.3.1 OBJECT DETECTION NETWORK..... 115

5.3.2 GEOMETRIC ALIGNMENT OF SENSOR DATA 116

5.3.3 3DMFV CLASSIFICATION NETWORK 120

A. FISHER VECTORS.....121

B. CLASSIFICATION OF FISHER VECTOR REPRESENTATION.....122

5.4 RESULTS AND DISCUSSION 123

5.4.1 DATASET 124

5.4.2 MFV NET PERFORMANCE 124

A. 2D DETECTION NETWORK PERFORMANCE125

B. 3D CLASSIFICATION NETWORK PERFORMANCE.....127

5.4.3 POINTNET CLASSIFICATION NETWORK PERFORMANCE..... 128

5.4.4 VISUAL RESULTS 129

5.4.5 LIMITATIONS & COMPARISON 132

5.5 SUMMARY..... 134

CHAPTER 6 CONCLUSION135

6.1 INTRODUCTION 135

6.2 CONTRIBUTIONS 137

6.2.1 ARCHITECTURE FOR AN INTELLIGENT MOBILE ROBOT..... 137

6.2.2 A SELF EVOLVING FREE SPACE DETECTION MODEL 138

6.2.3 A MULTIMODAL FISHER VECTOR NETWORK FOR HUMAN ACTIVITY RECOGNITION 138

6.3 BENEFITS AND IMPLICATIONS OF THIS RESEARCH 138

6.4 FUTURE WORK 139

APPENDIX A: LBOROLDNAV DATASET SAMPLE142

APPENDIX B: LBOROLDNHAR DATASET SAMPLE.....143

APPENDIX C: TECHNICAL DRAWINGS144

BIBLIOGRAPHY148

List of Figures

Figure 1: Two neurons side by side showing the Nucleus, Axon, Synapse, Dendrites, Neurotransmitters, and the Synaptic Cleft [71]. 13

Figure 2: Some different terms commonly used in AI, Rules-Based Systems, ML Representation Learning, DL and Bayesian Networks. 18

Figure 3: Three ML paradigms discussed in this chapter. Most of the research in this thesis is focused on supervised learning with a small diversion into semi-supervised learning. 20

Figure 4: Shows the process of online active ML. At first, the algorithm uses little data to make a classification. As a new instance is observed, the active Learning algorithm queries it. The online Learning Algorithm updates the Classifier..... 23

Figure 5: Shows linearly separable data, a decision boundary, Support Vectors, and the Margin. In this case, the data falls into one of two Classes. 27

Figure 6: Shows a feedforward neural network is an ANN in which connections between the nodes do not form a cycle. The feedforward neural network was the first and most straightforward type of artificial neural network..... 30

Figure 7: (a) Showing the Red layer of the true colour image of the number four. (b) Shows the flattening of the layer out into a single array lining up one column after another..... 31

Figure 8: (a) Shows the original data of image four. (b) Shows the reduced horizontal dimensions of the image data still retains the spatial arrangement. 32

Figure 9: (a) & (b) Show the image data after having been acted on by the down sampling factor (1 0.3) and (0.5 1), respectively. In both cases, the dimensionality has reduced the image data from a 17 by 4 matrix to a 17 by 3 matrix. 33

Figure 10: (a) Shows the original image of the number four. (b) Shows the same image with the down sampling factor acting on both the horizontal and vertical data points. Note the overall reduction in dimensionality on the image. 34

Figure 11: The generalized architecture of a CNN showing the three main elements of CNN; Convolutional Layers, Pooling Layers, and the Output Layer. Note the process that occurs between the different layers of the network [152]...... 35

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

Figure 12: Shows the down sampling process when using a Max Pooling layer filter with dimensions of 2 by 2 acting on a 4 by 4 sample of data from an image representation.	35
Figure 13: Shows the Binary Step activation function [157]......	36
Figure 14: Shows the Bipolar Step activation function [157]......	36
Figure 15: Shows the Sigmoid activation function [160].	38
Figure 16: Shows the ReLU activation function [161]......	38
Figure 17: The Regression Loss, Binary Classification, Multi-class Classification loss function and their subdivisions [167]......	39
Figure 18: Shows the sequence of steps in the Gaussian process resolution matching algorithm. On the left side of the image is the high-resolution camera data, and on the right is the lower resolution LiDAR data [116].	42
Figure 19: Shows the scalable, multi-layer context mapping and recognition system for the autonomous platform. They are depicting the sensing layer, a multi-layered context representation, the data analysis layer, and the application layer.	57
Figure 20: Loughborough University London autonomous platform indicating the location of the range and optical sensors used to collect the LboroLdn AV and LboroLdnHAR datasets. The Kinect sensor, missing from this image, is positioned on the handlebar.	61
Figure 21: Proposed framework for the platform showing the controllers, actuators, depth, optical, and telemeter sensors. All-optical and proximity data was logged to the laptop. Telemetry data was timestamped and logged to an SD card.	62
Figure 22: Steering assembly showing both servos motors the right and left tie rods linking to the handlebars. The link to the handlebars is through an additional two tie rods. Both servo motors are driven from the same signal – PWM – from the Arduino.....	65
Figure 23: Shows the ultrasonic sensor array assembly, detailing the six HC-SR04 ultrasonic sensors and a section of the autonomous platform headset. The HC-SR04 were at an angel of 5°, 25°, and 45° either side of the longitudinal axis.....	65
Figure 24: Ultrasound collision avoidance dictates navigation and driver policy for the autonomous platform for data collection. Decisions are hard programmed based on the proximity of objects relative to the ultrasonic sensor array.	66

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

Figure 25: The ultrasound-based control architecture for collision avoidance. Objects within the range are logged and acted upon, influencing the driver policy of the platform. 68

Figure 26: The camera-based control architecture for colour detection. Currently, the platform understands the colour of red green and amber. 69

Figure 27: The operator-based control architecture for manual override. The user is the only element of driver policy that exerts control over the collision avoidance policy – in the ultrasounds-based controller – and the colour governance in the camera-based controller. .. 70

Figure 28: Platform setup indicating the location of the sensors relative to the front axle. The location of each sensor is measured relative to the ground and the front axle of the platform. Except for the Wansview cameras, all sensors were on the centreline. 71

Figure 29: Platform setup indicating the horizontal FoV of the radar, the LiDAR, and the ultrasonic sensor array. The LiDAR provides midfield depth data, the radar provides far-field depth data, and the ultrasonic sensor array provides near filed depth data. 72

Figure 30: The primary data collection routes were traversed between 28th May 2018 and to 1st October 2018, and 1st November 2019 to 30th January 2020. The traversable distance was a total of 1.2 km over four different locations, Route 1 Lesney Avenue, Route 2 BT 77

Figure 31: Montage of six images taken on 28th May 2018, illustrating the diverse range of buildings and short-term lighting changes encountered by the platform. 79

Figure 32: Illustrates two different views from a single location under different environmental condition. Note environmental conditions are limited for this release of the dataset as most of the data was logged over the summer months. 80

Figure 33: Traversals for different environmental conditions for the LboroLdnAV dataset. Factors influencing the route, include events on campus and driver policy. The license and permits were granted for three months between 28th May and 1st October 2018. 80

Figure 34: The layout of furniture and the position of the autonomous platform during the collection of the LboroLdnHAR dataset. The data was collected throughout the 17th and 18th August 2018. In both cases, the subjects start and finish the activity with a T pose. 82

Figure 35: Montage of 6 images taken on 17th and 18th August 2018. The images illustrate six of the different activities performed during the data collection period. Carrying a box, pushing a board, running, sitting at a desk, standing, and walking while texting. 83

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

Figure 36: Illustrates the plan view of the sensor setup showing the location of the ultrasonic sensor Array and the wide-angled 360Fly camera. It should be noted that α denotes the range between the ultrasonic sensor array and the object, it is not the perpendicular distance.	91
Figure 37: Illustrates the elevated view of the sensor setup showing the location of the ultrasonic sensor array and the wide-angled 360Fly camera.	91
Figure 38: The proposed FSD pipeline. We first train an SVM on a small quantity of data. As new data becomes available, we quarry the robust sensor stream as to its class.	93
Figure 39: The proposed framework for FSD utilizing supervised and semi-supervised ML. This technique uses online active ML methods to self-learn free space and evolve as it encounters new data.	94
Figure 40: Shows the results of the ultrasound OGMap matching Scenario 1 in Figure 41. In this image, the white rays emanating from the testbed's position at the zero coordinates correspond to free grid points, while the black grid points indicate occupied space.	97
Figure 41: A subset of the data used during the comparison between the proposed framework and DeepLabV3+. From the top left corner: (a) Scenario 1, (b) Scenario 2, (c) Scenario 3, (d) Scenario 4, (e) Scenario 5, (f) Scenario 6, (g) Scenario 7, (h) Scenario 8, (i) Scenario 9 and (j) Scenario 10. Scenario details reported in Table 10.	99
Figure 42: Confusion Matrix for the online active ML framework. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified.	103
Figure 43: Confusion Matrix for DeepLabV3+ framework. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified.	104
Figure 44: Visual results of the Semi-supervised and Fusion approaches to FSD and DeepLabV3+ FSD. From the top-down. Scenario 1, Scenario 2, Scenario 3 and Scenario 4. Table 16 details the scenarios depicted in Figure 44. Images in the first column indicate the output of the proposed self-evolving FSD framework. Images in the second column indicate the output of DeepLabV3+.	106
Figure 45: Depicts the MfV Net HAR pipeline. Given RGB data, we first detect a subject and generate an ROI. Each RGB ROI is translated and aligned onto the 3D point cloud. The	

corresponding 3D ROI is then passed onto the classifier before deciding what activity is being performed. 111

Figure 46: Shows the PointNet Architecture. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net to facilitate the segmentation the components that construct the object its classifying [1]. 113

Figure 47: The proposed HAR architecture. They are depicting all the elements in the Multimodal Fisher Vector Network. We first leverage the power of a CNN to detector objects and propose an ROI. The geometrically aligned and translated Point Cloud data allows us to identify the corresponding 3D ROI in the 3D sample. This segmented ROI is converted into a modified FV representation before being passed onto the classification network. The resultant class is overlaid on the object detector image proposed by the object detector network. 115

Figure 48: Illustrates the plan view of the setup showing the position of the Kinect and LiDAR sensors relative to the autonomous platform. 117

Figure 49: Illustrates the elevation view of the setup showing the position of the Kinect and LiDAR sensors relative to the autonomous platform. 117

Figure 50: Illustrates the plan view of the setup showing the position of the RGB and LiDAR sensors relative to the LboroLdn autonomous testbed. 118

Figure 51: Illustrates the elevated view of the setup showing the position of the RGB and LiDAR sensors relative to the LboroLdn autonomous testbed. 118

Figure 52: (a): A Scenario showing the subject carrying a box. Overlay with the 2D ROI, the translated and aligned Point Cloud data. (b) Shows the Point Cloud data with corresponding 3D ROI overlay. 120

Figure 53: Shows three 2D points superimposed on a 2D Gaussian on the left of the image. In the centre of the image is the FV, and on the right is the FV representation for the three points. The vector in the centre of the image indicates the rate of curvature for the data..... 121

Figure 54: (a) Shows the spherical Gaussians superimposed on the Point Cloud data. (b) Shows the modified FV representation of the GMM for the Point Cloud data..... 122

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

Figure 55: The average precision of the object detector portion of the proposed framework showing the trade-off between precision and recall. A high area under the curve represents both high recall and high precision – as we can see here..... 126

Figure 56: The log-average miss rate of the object detector portion of the proposed framework indicating the quality of detection of the object detector. 126

Figure 57: Confusion matrix for the 3D classifier portion of the proposed framework. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified. 128

Figure 58: Confusion matrix for PointNet. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified..... 129

Figure 59: A subset of the data used to assess the visual performance of the proposed MfV Net. From the top: Scenario 1, Scenario 2, Scenario 3 and Scenario 4. Table 19 details the scenarios depicted in Figure 59. Images in the first column indicate the output of the Object Detector. The centre column shows the segmented Point Cloud data. The final column shows the output of the network and the associated class identified for the activity being performed. 131

Figure 60: Montage of 48 traversals of the same location on 22nd May 2018, illustrating the diverse range of images, short-term lighting and weather changes encountered by the testbed when collecting the LboroLdnAV dataset. 142

Figure 61: Montage of 48 scenarios performed by different subjects captured on 17th June 2018 and 18th June 2018, illustrating the diverse range of activities in the LboroLdn HAR dataset. 143

Figure 62: Isometric View of the Autonomous Platform showing the quad bike chassis, electronic rack, ultrasonic array, steering assembly and MSI laptop..... 144

Figure 63: Shows the ultrasonic sensor array. Top left showing side elevation. Top right showing the isometric view. Bottom Left showing plan view. Bottom right front elevation. 145

Figure 64: Shows the Steering assembly. Top left showing side elevation. Top right showing the isometric view. Bottom Left showing plan view. Bottom right front elevation. 146

Figure 65: Shows the Power Electric Rack. Top left showing side elevation. Top right showing the isometric view. Bottom Left showing plan view. Bottom right front elevation. 147

List of Tables

TABLE 1: REVIEWED AV DATASETS.....	53
TABLE 2: REVIEWED HAR DATASETS.....	55
TABLE 3: PROXIMITY SENSORS SUMMARY	63
TABLE 4: OPTICAL SENSORS SUMMARY	63
TABLE 5: TELEMETRY SENSORS SUMMARY.....	63
TABLE 6: DESCRIPTION OF THE LBOROLDNAV DATASET	79
TABLE 7: SUMMARY STATISTICS FOR THE LBOROLDNAV DATASET	79
TABLE 8: DESCRIPTION OF THE LBOROLDNHAR DATASET	82
TABLE 9: SUMMARY STATISTICS FOR THE LBOROLDNHAR DATASET	82
TABLE 10: SCENARIO DETAILS DEPICTED IN FIGURE 41	99
TABLE 11: SEMANTIC SEGMENTATION NETWORKS MEAN IOU	101
TABLE 12: DATASET METRICS FOR THE ONLINE ACTIVE ML FRAMEWORK.....	102
TABLE 13: CLASS METRICS FOR THE PROPOSED ONLINE ACTIVE ML FRAMEWORK.....	102
TABLE 14: DATASET METRICS FOR DEEPLABV3+.....	103
TABLE 15: CLASS METRICS DEEPLABV3+	103
TABLE 16: SCENARIO DETAILS DEPICTED IN FIGURE 44	105
TABLE 17: 2D DETECTION NETWORK PERFORMANCE.....	125
TABLE 18: 3D CLASSIFICATION NETWORK PERFORMANCE.....	127
TABLE 19: POINTNET CLASSIFICATION NETWORK PERFORMANCE.....	129
TABLE 20: SCENARIO DETAILS DEPICTED IN FIGURE 59	132

List of Abbreviations

AV.....	Autonomous Vehicle
AVSR.....	Audio Visual Speech Recognition
DL.....	Deep Learning
DMV.....	Department of Motor Vehicles
DUC.....	Dense Up-sampling Convolution
EKF.....	Extended Kalman Filter
ERC.....	European Research Council
ESR.....	Electronic Scanning Radar
FMCW.....	Frequency Modulated Continuous Wave
FoV.....	Field of View
FPS.....	Frames Per Sec
FSD.....	Free Space Detection
FTN.....	Flock Traffic Navigation
FV.....	Fisher Vector
GMM.....	Gaussian Mixture Model
GNSS.....	Global Navigation Satellite System
GPU.....	Graphical Processing Units
GUI.....	Graphical User Interface
HAP.....	Human Action Prediction
HAR.....	Human Activity Recognition
HD.....	High Definition
HDC.....	Hybrid Dilated Convolution
HFoV.....	Horizontal Field of View
HOG.....	Histogram of Oriented Gradients

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

HRI	Human-Robotic Interaction
HSV	Hue Saturation and Variance
IMU	Inertia Measurement Units
IoT	Internet of Things
IoU	Intersection Over Union
IP	Internet Protocol
IR	Infra-Red
ITS	Intelligent Transport Systems
LboroLdnAV	Loughborough London Autonomous Vehicle
LboroLdnHAR	Loughborough London Human Activity Recognition
KITTI	Karlsruhe Institute and Toyota Technological Institute
LDR	Light Dependent Resistor
LiDAR	Light Detection and Ranging
LUTZ	Low-carbon Urban Transport Zone
MfV Net	Multimodal Fisher Vector Network
ML	Machine Learning
MSI	Micro-Star International
NHTSA	National Highway Transportation and Safety Administration
OGMap	Occupancy Grid Map
NTU	Nanyang Technological University
PNAS	Proceedings of the National Academy of Sciences
PWM	Pulse Width Modulation
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
ReLU	Rectifier Linear Unit
RF	Radio Frequency

MULTIMODAL MACHINE LEARNING FOR INTELLIGENT MOBILITY

RGB	Red Green Blue
ROI.....	Region of Interest
RPM.....	Revolutions Per Minute
RPN.....	Region Proposal Network
SAE.....	Society of Automotive Engineers
SD	Secure Digital
SLAM	Simultaneous Localization and Mapping
SVM.....	Support Vector Machine
TfL	Transport for London
VFoV	Vertical Field of View
VIAC.....	VisLab Intercontinental Autonomous Challenge
VW.....	Volkswagen

Chapter 1 Multimodal Machine Learning for Intelligent Mobility

1.1 Introduction

Artificial Intelligence (AI) and Machine Learning (ML) have been used to beat masters at chess, poker aficionados and Go grandmasters – one of the world's most complex games [3], [4]. IBM Watson has even won the Gameshow Jeopardy, and now a different version of Watson has been trained to design personalized cancer treatments for patients [5]. AI is the development of algorithmic systems capable of performing tasks that typically require human intelligence [6]. ML, on the other hand, is a subset of AI in which computers use big data to learn how to do a particular task [6]. Both AI and ML are used to do things that humans have spent many years perfecting, and the machines are beating us hands down. However, as good as we are at teaching AI to perform complex tasks, we are terrible at getting machines to learn even the most basic, childlike skills [4], [7].

Moravec's Paradox¹ is the problem of being able to perform complex tasks but not simple ones. Named after Hans Moravec, who studied this issue in the 1980s, he seemed to

¹ Moravec Paradox states that, contrary to popular belief, high-level reasoning requires less computation than low-level unconscious cognition. [8]

have a pretty straightforward solution – make a program that thinks like a child [8]. On the most basic level, this is Moravec's Paradox. We do not know how to program general intelligence, although we are great at getting AI to do a singular task. For example, most toddler level skills, such as facial recognition, involves learning new things before transferring them to another context. Getting a computer to do this is a primary goal of AI, we called this concept, Artificial General Intelligence (AGI).

In 1988 Hans Moravec said that evolution is the reason it is so hard to achieve AGI [8]. His point was that the things that seem easy for us are the result of thousands of years of evolution. So even though most kids can quickly tell the difference between blue and yellow, a friend and a stranger, these are not simple skills. They only seem simple to us because our species have spent thousands of years refining them.

For example, learning to drive, one will always remember the early experiences of being behind the wheel. Whether it is memories of the driving instructor, the first lesson, theory or practical test, the memory is pronounced. The habits humans form, or driver policy we develop are a direct result of the rules of the road, and the experience gained. These rules are not instructions on how to behave but rather a group of statutory instruments setting out what one may or may not do. While this is sufficient to get started, it is only through the experience of driving on the road that one develops a policy to which we adhere. Of course, we can all agree that learning to drive is a trailing experience; however, Moravec theorized that it is not as difficult as learning to talk, walk, or understand the terrain around us.

This is no different for Intelligent machines where driver policy refers to the decision-making capability of the vehicle under various situations. The goal is to solve complex tasks consisting of high dimensional data input. Unfortunately, it has proven hard to construct sophisticated agents that are capable of driving a vehicle with human-like performance [9]. Although these agents can perform specific tasks very well, they cannot perform a range of functions.

Part of the problem is that computer science is relatively young, and the study of AI is even younger! Therefore, it is ambitious to think humans would have figured out AGI already. Then again, Moravec did not believe this was the only issue; he also thought that researchers were approaching the problem the wrong way [8]. In the 1980s, researchers into AI were mainly working from the top-down, trying to copy the mental process of fully formed human brains [10]. Moravec believed that the most successful approach would be to work

from the bottom up. In other words, instead of building a complex brain from scratch, he thought that machines should mimic evolution.

Like nature, Moravec thought it best to start small and then add complexity to the system, all the while challenging these systems to adapt. Moravec, amongst others, thought how the human brain performs tasks could be studied and then applied to machines [10], [11]. Self-evolution is what happens in online active learning [12], [13], where the machine queries new data as it comes in before reforming its understanding of what it is seeing. Moravec's theory appears to be a solution that works, and the more that researchers base their AI on the human brain, the smarter these machines will get.

Although Moravec's paradox was conceived 30 years ago, it is still very relevant today. For example, the current state-of-the-art in AI is excellent at solving "narrow" competencies or singular tasks, humans, on the other hand, are good at pretty much everything [14]. In an interview, Dr Sean Holden of Cambridge University stated that "Most AI researchers don't try to solve the whole problem because it is too hard. They take some specific problem and do it better" [14]. Dr Holden is not alone in this thinking. In [15], researchers reported on the fact that the majority of the application of AI in today's workplace uses virtually no abstract thought. Workers in positions that work to a fixed set of rules are being replaced at an astonishing rate [16]. Conversely in the creative industries, where abstract thought is prerequisite, peoples livelihood are as secure as they ever have been [16].

Neural networks are systems that can teach themselves to recognize patterns – they are modelled on the way human brains learn. When something new is learned, our brain strengthens the connection between neurons [17]. It does this by adding more connection so that the brain can process more signalling molecules. Over time these connections develop and grow stronger and will, in all likelihood, stay this way for some time. The connection can get overwritten with something more helpful. Neural networks mimic the circuits in the human brain [4]. They start with some basic framework about how to do a task. Then they practice that task with labelled test data to refine and optimize the connection between artificial neurons [18]. As a tool, Neural Networks are not perfect – they can only perform the task, they trained for, and the connections they form, adjust every time they re-train [19]. However, they are a big step towards AGI [20], although fully operational AGI is quite some distance away.

One approach to solving this problem takes inspiration from the way the human brain does not just use whatever neurons are available [21]. Instead, it activates different sets of neurons for different tasks and leaves some neurons alone [22], [23]. This process is achieved

using Dropout – where some neurons in the Neural Network are switched off – or an activation function that only responds to specific values. In a 2018 study published by Proceedings of the National Academy of Sciences (PNAS), researchers showed that it was possible to do this in a Neural Network too [24]. Making one task activate one set of neurons, and another task activate another. By combining this approach with previous research methods, these researchers were able to program a Neural Network that achieves 90% accuracy on over 500 different tasks [23]. The greater understanding we attain about the method in which these networks refine connections, the better this branch of AI is going to get at performing various tasks. The challenge is getting the system to learn from more than just one example.

Not every task in ML has a massive dataset for a network to sort through; if a program is going to think like a human, it must start grasping the rules that govern it. While some tasks can be solved using big data, some must be learned from a relationship linking to data types. While it might not seem as obvious, these systems take design cues from our brains. When a system is learning from a robust sensor stream, it can help us deduce what another sensor sees. By building similar systems in AI, we hope to encourage the systems to keep learning and adapting their understanding of the surroundings [25].

A program with general intelligence should be able to process multiple kinds of data sources and be able to learn new rules. As well as taking advantage of data sources, AGI should be able to take advantage of multiple ML algorithms that perform different tasks that contribute towards a single goal [26]. Multimodal ML is a multi-disciplinary research field with one of its earliest applications in the field of Audio-Visual Speech Recognition (AVSR) [27]. Similar to sensor data fusion in the sense that it uses multiple sensor modalities, Multimodal ML uses multiple ML algorithms, either in parallel or in series to improve performance [28]. While not quite AGI, this process allows different ML algorithms to perform multiple tasks in a similar way to people. Multimodal ML helps one modality to affect the identification of another modality, allowing use of complementary data [26].

1.2 Context & Motivation

Intelligent mobility is one of the most relevant applications of AI. Frequently used in many different sectors from Agriculture [29] to Medicine [30] and Finance [31], it is likely to have the greatest impact on the transport sector in the short-term [32]. Worldwide there is an average of 3,300 road deaths a day. In the UK alone, over the 11 years, from 1999 to 2010,

there were more than 3 million road traffic incidents [33]. In 2015 Transport for London (TfL) reported 25,193 casualties took place at signal-controlled urban junctions [34], [35].

Increasing degrees of automation – from semi-manual to fully computer-controlled vehicles – already exist or are being added to vehicles to improve safety, reduce the number of driver tasks, and improve fuel efficiency. The technology and sensors integral to self-driving, or vehicle autonomy, already impact the way humans drive. Some believe that as Autonomous Vehicle (AV) or robots become ubiquitous, traffic incidents and fatalities will reduce [36], [37]. While it is possible to distinguish between the different systems – driverless car, AV, or Intelligent mobile platform – they are in effect all robots. To that end, these terms are used interchangeably throughout this research.

Autonomous driving technology has developed at a rapid pace, and will, with all probability, continue to do so for the foreseeable future. Already, the first steps towards hands-free driving are evident, for example, Parking Assist by Volkswagen (VW) and Parktronic by Mercedes-Benz [38], [39]. While the majority of the sensor technologies used in automation are well-established – Light Detection and Ranging (LiDAR), Near Field Vision, Radar, and ultrasonic rangefinders – they primarily function independently, triggering a response rather than making decisions based on the data they observe [38], [39]. Fusing the sensor data used in such technologies should enable autonomous robots to create multi-layered virtual maps with real-time context information. From these real-time virtual maps, the robots can make more holistic decisions.

Given the high dimensionality and complexity of optical sensor data, it is crucial to examine ML algorithms to continue the progress already made in Intelligent mobility. Deep Learning (DL) – a subset of ML – is one approach to solving problems in this field using enormous datasets. Principally, data is passed through a function-approximator using a deep multi-layer Neural Network to learn features and make a classification [40]. DL has gained much notoriety because of two significant advancements. Firstly, improvements in hardware, specifically Graphical Processing Units (GPU), have facilitated computers to increase the bandwidth and quantity of data they process. Secondly, the availability of more and more large-scale labelled datasets – which are used for training and verification of feature learning networks. That is not to say that DL is flawless. Some significant shortcomings are limited knowledge about the internal workings of Neural Network [41] and sufficient amounts of training data [42], [43]. When either of these scenarios are met the network becomes unreliable and prone to errors.

Contrary to this, humans, or more specifically infants, have innate knowledge upon which they build their understanding of the world. This intuition helps them voraciously learn and adapt to situations never encountered. Many scientists working in the field of AI argue that most human skills are learned, and therefore machines can learn them without the need for pre-loaded rules [44]. But lately, there are a growing number of researchers attempting to encode machines with a bit of common sense [45], [46].

The latest trend in ML is DL. Deep Neural Networks – a collection of simple function-approximators – loosely modelled on neurons in the brain, adjust weights and bias as they are presented with more and more data. The results are astonishing, and credit where credit is due, Deep Neural Networks can perform remarkable tasks. From facial recognition [47] to classifying human activity [48], these networks produce incredible results. However, Neural Networks require thousands of training samples to identify the necessary features to form the associations needed to make a classification. Even then, they can produce some embarrassing errors and dangerous mistakes [49]. For example, ML algorithms can play classic Atari games like space invaders with superhuman skills. But by adding one or two random pixels to the playing screen, the player's avatar becomes a sitting duck [50]. By comparison, an infant can see an image once and instantly recognize it in another context.

Different research groups are trying to categorize human instinct before encoding into AI [51]. These frameworks sit somewhere between pure ML and hard programmed systems. One research team developed ML algorithms that emulate interaction networks [52]. Interaction Networks embed a rule that relationships exist between objects. For example, researchers in [53] embedded some basic knowledge about relationships before getting a Neural Network to segment the region of an image containing specific geometric shapes. Like the way a baby parses the world into groups with some underlying knowledge, or how they use a sense, like touch, to learn about something. In tests, once the ML algorithm learned physical properties like gravity and the specific relationships to a falling string, its ability to predict its behaviour increased dramatically [48]. While human-like AI is a little way off, these latest attempts to artificially reproduce common sense bring closer the possibility of creating machines that can fully interact with the world the way humans do, machines that start as an infant and progress to learn like a child.

1.3 Objectives

Traditional methods used in computer science require manual programming of specific tasks; however, for real-world perception tasks, this is not always possible. This is especially the case in computer vision, such as object or scene recognition. Of course, we have some well-defined problems, such as edge detection, and some that are more challenging, such as recognizing intricate relationships between different elements in an image [52]. In general terms, ML is a technique or a set of methods for automated analysis of structure in data [54]. It can be broken down into three learning paradigms, Unsupervised Learning, Semi-Supervised Learning and Supervised Learning. ML is very similar to data mining, but the focus is more on autonomous machine performance, rather than enabling humans to learn patterns from the data [55].

While there have been significant developments in ML, monumental challenges remain to enable real-world engineering systems to be enriched with data-driven systems. For example, most of the DL systems require massive amounts of data to achieve adequate generalization capability. In some instances, collecting enormous amounts of data is practically impossible. On the other hand, many critical engineering systems require multiple layers of safety, before new data-driven algorithms are assimilated into their operation, and most importantly, before the human operators can be replaced. Despite recent developments, ML is far from reaching the level of human perception and cognitive ability we hold. We can define perception as the interpretation of data to make a decision, and cognition as wisdom, or knowledge of an occurring event. In the medium to short-term, human intelligence must be the benchmark to compare the performance of ML algorithms. Hence it needs to be at least as good as humans are at performing perception and cognitive tasks [56]–[58].

Intelligent mobility is the use of advanced technology to improve the way humans and objects are mobilized. The AVs that drive without the need for human intervention is a vital element of Intelligent mobility. The AVs will be one of the first mass-market application of intelligent robotics in the world. Broadly speaking, the objectives of the research fall under perception tasks and cognitive skills. For example, Free Space Detection (FSD), one of the contributions of this research, is a perception task, which is the detection of traversable space for an AV and largely regarded as a cornerstone of automated driving and human locomotion [59], [60].

On the other hand, robots such as AV should be aware of humans around it before making decisions. Human Activity Recognition (HAR), another contribution of this research, is a cognitive skill that classifies well-defined movements of a human agent to determine what activity they are performing [61]. This leaves us with a vast scope to address, and many problems to solve, if we want to apply ML to Intelligent mobility, and most importantly if we were to make AVs intelligent like humans.

Specifically, the high-level research objectives of this thesis are:

1. To design and develop a data collection mechanism to investigate a wide spectrum of environments that are to be catered by Intelligent mobility applications, such as indoor spaces and pedestrianized areas.
2. To explore ML algorithms that are capable of adapting to new environments and data streams with little or no training.
3. To investigate methods of leveraging multiple heterogeneous data streams (multimodal sensor data) to make robust decisions in safety-critical autonomous systems.

The hypothesis of this research is to see if: it is possible to make autonomous systems safer and more intelligent with algorithms that are capable of adapting to new environments by leveraging multiple heterogeneous data streams to make robust decisions.

1.4 Contributions

To achieve the objectives listed above, this thesis presents multiple contributions to the academic literature. The contributions specifically focus on the autonomous (driverless) vehicle technology, and all the experiments and discussions are based on several applications of Intelligent mobility. The participation of this thesis are as follows:

1. Throughout the course of this research and in the quest to prove the research hypothesis, many datasets were reviewed. These datasets were deemed unsuitable for the project's requirements because they lacked the correct sensor data modality, did not log data from unstructured surroundings or were recorded in outdoor environments. Therefore, a means of collecting specific data was needed. The first contribution of this thesis is to develop an autonomous platform as an open-source

experimental framework for data gathering, sharing, and experimental validation for driverless vehicle technology.

2. Using the autonomous platform, we developed two novel multimodal datasets collected with data-driven algorithm development and experimental validation in mind. Firstly, the Loughborough London Autonomous Vehicle (LboroLdnAV) Dataset is a dataset gathered from unstructured indoor and pedestrianized outdoor environments, annotated with 7 object classes collected using seven different perception sensors. Secondly, the Loughborough London Human Activity Recognition (LboroLdnHAR) Dataset is a Multimodal open-source dataset collected indoors, using three different sensors and annotated with 9 classes of human activity
3. A self-evolving FSD algorithm is developed, which leverages the relative uncertainty of different sensors as a utility to automatically label new data (active learning) and re-learn the data-driven model whenever new data streams are encountered (online learning).
4. Knowing what human agents are doing in their environment is crucial for safe decision-making by AVs. A Multimodal Fisher Vector Network, which is a type of deep CNN, is proposed as a new methodology for the classification of different human activities leveraging both Red Green Blue (RGB) camera data and the Point Cloud data that are gathered from LiDAR sensor.

1.5 Publications

Several publications in peer-reviewed conferences and journals have been made as a result of the contributions presented in Section 1.4. The following are the journal articles or conference papers that are about to be published or currently under review and directly influenced this thesis:

1. **ROCHE, J., DE SILVA, V., KONDOZ, A., 2019.** A Multimodal Perception Driven Self-Evolving Autonomous Vehicle. IEEE Trans Cybernetics. 2019 (Resubmitted for Review April 2020).

2. **ROCHE, J.**, DE SILVA, V., HOOK, J., MOENCKS, M., KONDOZ, A., 2019. Multimodal Modal ML for Human Activity Recognition with Applications to Intelligent Mobility. IEEE Trans. on Industrial Informatics 2020 (Submitted May 2020).

The following are the journal articles and conference proceedings that are published, co-authored and partly influenced this thesis:

1. DE SILVA, V., **ROCHE, J.**, KONDOZ, A., 2018. Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots. Sensors, 18 (8), 2730.
2. DE SILVA, V., **ROCHE, J.**, SHI, X, KONDOZ, A., 2018. IoT driven ambient intelligence architecture for indoor Intelligent mobility. IEEE 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress Athens, Greece, 12-15 August 2018, pp.451-456.
3. MOENCKS, M., DE SILVA, V., **ROCHE, J.**, & KONDOZ, A., 2019. Adaptive Feature Processing for Robust Human Activity Recognition on a Novel Multimodal Dataset. ArXiv, abs/1901.02858., in Robotics and Autonomous Systems (Resubmitted for Review March 2020).

The final publication list is a conference paper that contributed little to this thesis other than some cross over areas of the relevant review material:

1. **ROCHE, J.**, DE SILVA, V., KONDOZ, A., 2018. A cognitive framework for object recognition with application to autonomous vehicles. IN: IEEE Computing Conference, London, United Kingdom, 10-12 July 2018.

1.6 Thesis Outline

The organization of this thesis is as follows. The current chapter introduces the context, motivation, and objectives of this thesis, as well as the original contributions.

Chapter 2 provides background into the biological fundamentals of nervous activity in the human brain, before presenting a review of the fundamentals of ML, Multimodal ML and applications of ML in Intelligent mobility.

Chapter 3 presents research on a perception driven AV for data collection. In this Chapter, the current available AV and HAR datasets are reviewed. The requirements for the development of the test platform and the experimental setup are set out. Finally, this chapter finishes with a description of the sensor data representation and the datasets gathered during this research – LboroLdnAV dataset & LboroLdnHAR dataset.

Chapter 4 presents research on a self-evolving FSD model. In this chapter, the problems with the current state-of-the-art in FSD are identified, followed by an explanation of the geometric alignment of the sensor data representations. Immediately after this, the algorithmic frameworks for FSD are presented with an explanation of the different learning paradigms used in this process. Finally, Chapter 4 finishes off with some results of the proposed framework before moving on to a summary of the findings.

Chapter 5 presents research on a 3-Dimension (3D) Multimodal Fisher Vector Network (MfV Net) for HAR. In this chapter, problems with the current state-of-the-art in HAR are identified, followed by an explanation of the geometric alignment of the sensor data. Immediately after this, the algorithmic framework for HAR is presented with an explanation of the different methods of representation learning. Chapter 5 finishes off with some results of the proposed network before moving on to a summary of the findings.

Chapter 6 concludes this thesis and presents some suggestions for future work. Appendix A details the LboroLdnAV Dataset, while Appendix B details the LboroLdnHAR Dataset. Appendix C presents some technical drawings describing the autonomous platform.

Chapter 2 Data-Driven Machine Intelligence

2.1 Introduction

Many cognitive pathways are employed to survey a visual scene before judgment, and associated action is made [62]. During this period, objects in the sensory range are identified and classified. Posterior and occipital lobes are critical in linking the visual map with reality and, therefore, in determining the location of an object [63]. The ear, neck and extra-ocular muscles contribute to this ability to geo-locate [64]. These muscles and auditory abilities are responsible for maintaining the link between reality and visual perception [65]. For example, when the ears hear a sound, the head and eyes move with respect to the body. The input from eyes and ears are required to locate the object of interest. Once the location of an object is identified, the visual information is compared to memories stored in the temporal lobes – bringing about recognition of the objects. The occipital and temporal lobe regulates decisions regarding the recognized object [66]. Commonly referred to as the two-stream hypothesis, the link between these parts of the brain is responsible for all visual processing [67], [68].

The brain is a system with multiple and distinct components performing specific tasks for the body and mind. The neurobiology of vision has its genesis in the Occipital Lobe. The Occipital Lobe is a cluster of densely packed neurons, located towards the rear of the brain just above the brainstem. When the Occipital Lobe becomes stimulated, dopamine, and other neurotransmitters flood the nucleus accumbens and the central nervous system forcing the hippocampus to recall memories relating to the objects they see [69]. For example, dopamine, a neurotransmitter that helps control the reward and pleasure centres of the brain, muscle

innervation, and emotional response – enables humans not only “see rewards, but to take action to move towards them” [69], [70].

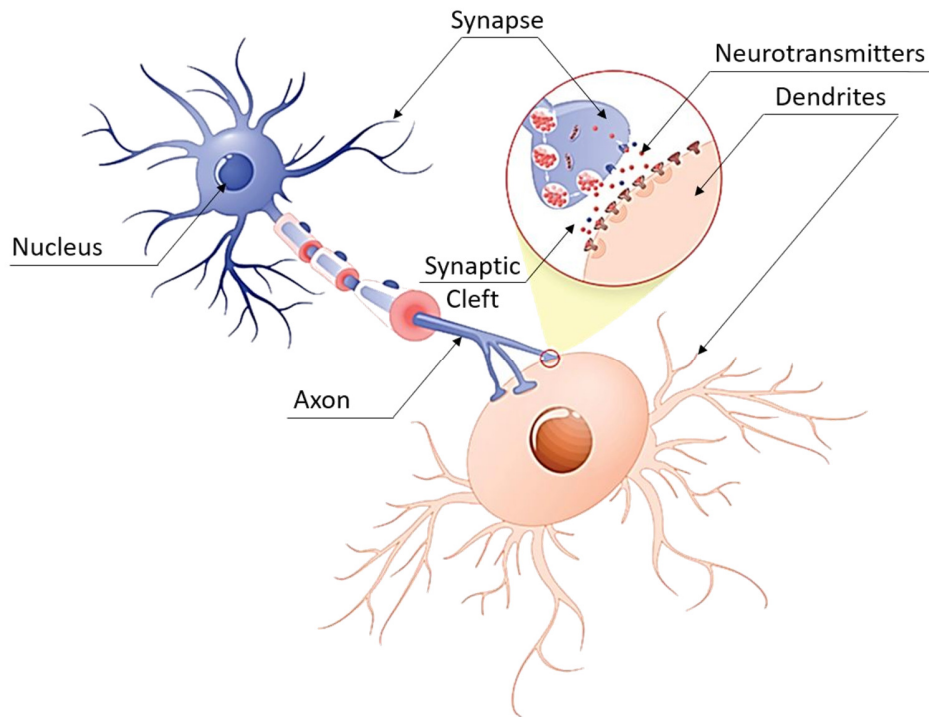


Figure 1: Two neurons side by side showing the Nucleus, Axon, Synapse, Dendrites, Neurotransmitters, and the Synaptic Cleft [71].

Although no physical connection exists between synapse and dendrites – the start and the end of neurons – signals are transmitted with the assistance of neurotransmitters and biochemical markers [72]. When a signal needs to pass across the synaptic cleft – the space between synapse and dendrites – neurotransmitters change the permeability of the cell wall [73]. This change in permeability allows cellular fluid to flow into the cerebral space mixing with cerebrospinal fluid – thus inducing an action potential that passes along the axon from neuron to neuron [70]. This arrangement is depicted in Figure 1, and depending on the biochemical marker and nerve type, the body and brain react in different ways. This relationship between neurotransmitters and biochemical markers, coupled with the location of brain components and nerve groupings, is the result of millions of years of evolution. It has allowed humankind to survive and controls how the body perceives objects it encounters continually. It is what makes humans who we are. It is what scientists are trying to emulate when they research AI [4].

Typically, what is regarded as AI nowadays is an Artificial Neural Network (ANN). An ANN is a computer algorithm that imitates the biological function of the human brain [74]. It contains virtual neurons that arrange in interconnected layers. Each artificial neuron passes on information by performing calculations in a similar way to the human brain. In the Neural Network, the neurons are just numbers in the code with typical values between 0 and 1. The connections between neurons are often referred to as weights. These weights describe how much the information from one-layer matters to the next. The values of the neurons (bias) and the weights are the free parameters of the network. In training the network, we want to find those values of these parameters so that loss can be minimized – often referred to as the loss function. Technically it is an optimization problem that the Neural Networks solve [75]. In such optimization problems, the Neural Network works through all combinations and learns the pattern in the data using Backpropagation. Learning through Backpropagation means that as the network returns a result that is not particularly good, the weights and bias are changed. Neural Networks learn from failure through a combination of forward and backpropagation. While it is nowhere near AGI, it certainly shows progress in the right direction.

This chapter is organized as follows; Section 2.2 reports on immanent nervous activity before moving onto the fundamentals of ML in Section 2.3. In this section, we shift from the biological to the artificial, reporting on the different learning paradigms and the ML techniques used during this research. In Section 2.4, we report on Multimodal Machine Learning before moving onto Section 2.5, where we applications of ML in Intelligent mobility. In Section 2.6, we conclude this chapter with a summary.

2.2 Imminent Nervous Activity a template for Artificial intelligence

In 1848 Phineas Gage was working as part of a crew cutting a railroad in the American bedrock of Vermont. Gage was using a tamping iron to pack explosives when tragedy struck [76]. The tamping iron - roughly 1 meter long, 3 cm in diameter and 6 kilograms in weight – ignited the explosives and shot the iron rod through Gage's left cheek, into his brain before the rod exited his skull and landed several meters away [77]. When Gage presented to Dr John Harlow - blinded in his left eye and suffering traumatic brain injury – Gage is reported to have remained lucid enough to tell the doctor, "Here is business enough for you" [76], [77].

Such injury inevitably caused some notoriety for Gage amongst locals, but it was Dr Harlow who etched the incidence into the history books. Treating Gage for some time after the incident, Dr Harlow reported that his friends found him "no longer Gage." "The balance

between his intellectual faculties and animal propensities seemed gone." "He could not stick to plans, uttered the grossest profanity, and showed little deference for his fellows" [77], [78]. It was not long until the construction company – that previously regarded him as a "model foreman" – refused him employment [76]–[80]. Medical practitioners and psychologists argued that this was evidence that the brain was "localized" or "plastic" [77], [78]. To an extent, modern neuroscience tells us that both are correct – as the brain is regarded as being both plastic and compartmentalized.

Fast forward nearly 100 years, and research by Canadian psychologist Donald Hebb resulted in what is now known as Hebbian learning. Hebbian Learning describes how the brain undergoes neural plasticity following a traumatic incident or a period of neural adaptation [81]–[83]. His work considered events experienced by people such as Gage, and he formed the hypothesis of learning based on the mechanism of neural plasticity [83]. Building on Donald Hebb's work, Warren McCulloch and Walter Pitts researched threshold logic, which is one of the cornerstones of modern Neural Nets. The work looked at two distinct approaches to biological processing in the brain and the applications of neural nets [74].

Donald's view on how the brain wires and rewires itself is what all contemporary neural nets do. How weights and bias are assigned to inputs to reaffirm a connection or build a new, more robust link is not a new idea [83]. People have been discussing concepts like this since Gage had that unfortunate accident. More recently, however, neural nets have gone through a revival due in part to projects like Google's Deep Mind, Tesla Self-Driving Cars, and Toyota's billion-dollar AI research Investment. These projects are expanding on McCulloch and Pitts' work with one key difference - they are expanding neural nets by adding hidden layers, stacking them on top of each other, and calling them Deep Net's [84], [85].

2.2.1 Neural Plasticity

Neuroscience is a biological science that is concerned with the function of the brain and the nervous system. Throughout our life, we are shaped by experiences that not only change our behaviour but also alter how we think. Exactly how this happens is not entirely understood, but one crucial mechanism is the physical changes and connections in our brain. The changing and shaping of connections in our brain is referred to as neuroplasticity. Donald Hebb was amongst the first people to describe this process. In his book "The Organization of Behaviour; A Neuropsychological Theory," he wrote his now-classic Hebb's postulate [11]. Hebb's postulate states that when two neurons fire at the same time, the connection between them is strengthened – becoming more likely to fire again in the future. When two neurons

fire in an uncoordinated manner, the connection between them weakens – becoming more likely to act independently in the future. Simply Put - neurons that fire together, wire together, - and neurons that fire apart, wire apart [11].

The brain can be viewed as a vast interconnected circuit with millions of different paths for the electrical currents to flow. Some of these paths can accommodate higher current allowing the electrons to move without restriction. These paths represent a human, established way of thinking, feeling, and doing. Every time we think, feel, or do something in the same way, we strengthen this path making it more robust, allowing an action potential to move without restriction. As a result, it becomes quicker and easier for the signals in our brain to travel this way [82].

By contrast, if a path is damaged, unused, or is not well constructed in the first place, we start to use a different pathway. If we keep using that new pathway and continue to use this route more and more - this new way of thinking, feeling, or doing becomes automatic. In the meantime, the old pathway gets less and less use, eventually weakening [11]. In other cases, it may be possible to repair or rebuild blocked pathways. This process of rewiring the brain by strengthening existing pathways, making new ones, weakening old ones, and repairing broken ones is neuroplasticity in action.

When we apply the neuroscience of learning to how the brain works, we can modify four variables to maximize the retainment of knowledge. Summarized as attention, generation, emotion, and spacing [86]. For us to be able to learn something, we need to be able to pay attention to it. If we can minimize the distractions, then we can maximize the learning, and we will not forget it. We need to encourage the learner to generate meaningful connections and associations with what they already know – helping the learner to make that connection to previously learned patterns and the broader context [86]. If we attach emotion to learning – and help motivate the learner with rewards – they are much more likely to remember the information at a later date.

It is essential to understand that neuroplasticity is not good or bad – it is just what the brain does. Neuroplasticity can result in significant changes, like when a child learns to cross the road safely, or when adults learn a new set of skills. On the other hand, it can result in unhelpful changes in the brain, when someone learns an unnatural way of thinking or a bad habit – like Mr Gage above.

2.3 Fundamentals of Machine Learning

Learning is the acquisition of knowledge or skills and is one of the most basic features of intelligence [87]. Learning enables an agent – biological or artificial – to perform a task more efficiently than the rest of the population [87]. Since the development of Hebb’s postulate in the late 1940s, many descriptions have been developed to suit various AI topologies [88]. In 1959, Arthur Samuel defined AI as a "*Field of study that gives computers the ability to learn without being explicitly programmed*" [89]. The concept is relatively old, but it has gained much popularity in recent times in the scientific community. The simple reason for this is that until recently, the data needed to train AI was not available.

Nowadays, there is a considerable increase in available and useable data. In fact, with the abundant number of digital assistants and talking computers, it would be easy to think the AI revolution is already here. When Google launched its flagship home assistant in 2016, CEO Sundar Pichai said computing is moving from the mobile world into an AI world. It is commonplace for companies like Google, Tesla, and Facebook to promote AI as breaking new ground. Whereas those at the forefront of research point out that there is much work to do and claim there are many key challenges to overcome before the real AI revolution begins – what we are experiencing is merely the illusion of AI [90]–[92]. Figure 2 depicts the relationship between the different AI techniques and some of the standard terms frequently used in these systems.

As noted, ML is a subset of AI which enables the computer to act and make data-driven decisions to carry out a specific task. ML came into existence in the early 1990s. It shifted focus from the symbolic approaches it had inherited from AI and moved towards methods borrowed from statistics and probability theory [4], [93], [94]. These algorithms are designed in a way that they can learn and improve over time when exposed to new data. In ML, a machine retains information and becomes smarter over time. However, unlike a human, it is not susceptible to things like short-term memory loss, information overload, sleep deprivation, and distractions.

Consider the problem of determining if an image contained a cat or a dog. When only considering the physical appearance between a cat and a dog, the difference can be a little grey. Of course, one could say that a cat has pointy ears, and dogs have floppy ears, but those rules are not universal. Between tail length, fur texture, and colour, there are many options to categories.

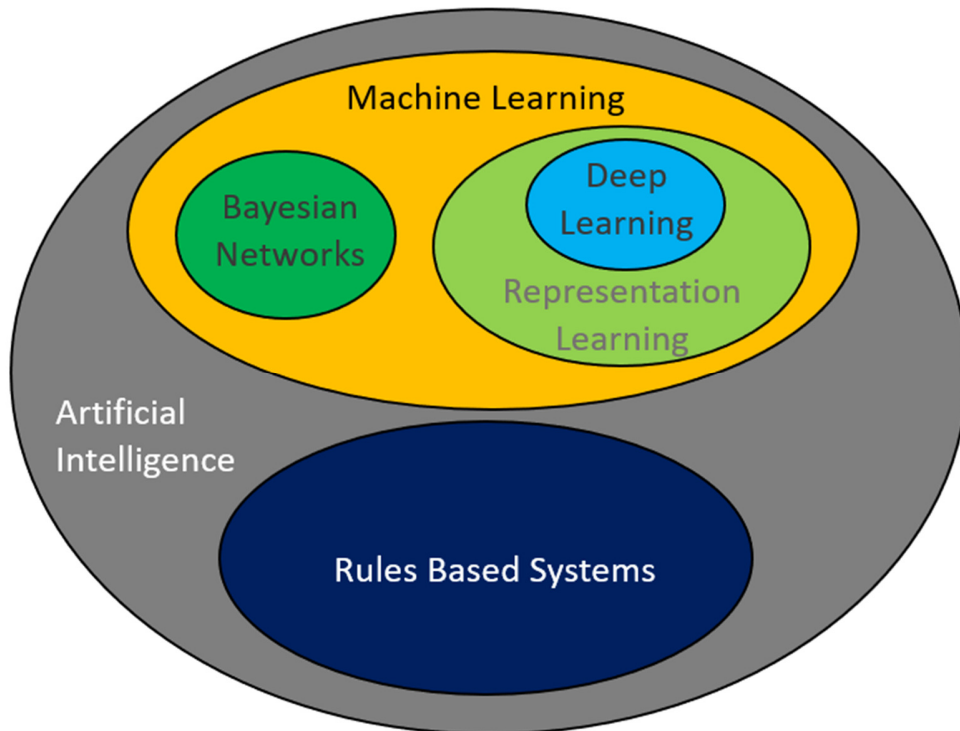


Figure 2: Some different terms commonly used in AI, Rules-Based Systems, ML Representation Learning, DL and Bayesian Networks.

In the case of traditional computer programming, this means a lot of tedious rules that someone would have to write manually to help a computer to spot the differences between cat and dog. Conversely, ML is about making a machine learn like humans, and like any child, that means they must learn through experience. With ML, programs analyse thousands of examples to build an algorithm. It then tweaks the algorithm based on if it achieves its goal or not. Just like the cognitive development of a child, over time, the algorithm gets smarter and better at solving problems. That is how machines like IBM Watson can diagnose cancer, compose classic symphonies, or crush Ken Jennings in jeopardy [95].

Some algorithms mimic the way the human brain is structured with Neural Networks. DL harnesses the power of Deep Neural Networks. It merely takes the data connections of all the artificial neurons and adjusts them according to the data pattern. The structure of a Deep Neural Network is mostly the same as a Neural Network; there is an input layer, an output layer, and connected hidden layers [84], [85]. The primary function of Deep Neural Network is to perceive an input before performing a complex calculation. The result is an output that can be used to classify and solve a problem with many different categories.

Although DL is a popular choice, it is not the only type of ML algorithm available [40]. There are many types; Logistical regression, Support Vector Machine (SVM), and Naïve Bayes. For example, if we want to design a classifier to predict whether the weather will be good or bad, and all we know are the environmental conditions, the date, temperature, and atmospheric pressure. A high score indicates that the weather will be bad, and a low score suggests the weather will be good. In either case, it is difficult to tell the weather knowing the date or temperature alone. Identifying a parameter like atmospheric pressures alongside date and temperature would return more accurate results. In this case, the classifier would assign greater importance to atmospheric pressure than those assigned to date or temperature.

Typically, these types of classifiers are used when the output gets categorized into at least two groups. While the differences between each ML algorithm can be subtle, it is essential to distinguish them as separate entities part of the same family. This is because they are best suited to different applications. Akin to the compartmentalized and interconnected structure of the human brain – each circuit has a different task.

2.3.1 Machine Learning Paradigms and Methods

The most common forms of ML are Supervised, Unsupervised, and Semi-Supervised Learning. While Supervised Learning is a relatively well-established form of ML, Unsupervised Learning and Semi-Supervised Learning, are regarded by some as still in its infant years. Although the more recent learning paradigms are gaining in popularity, most of the research utilizes Supervised ML methods. Figure 3 depicts three ML paradigms discussed in this Chapter and some examples of the different types of learning.

ML of today differs from ML of the past. Born of pattern recognition and the theory that computers can learn without being programmed to perform specific tasks – things have changed quite a bit. Nowadays, data is everywhere and generated at an astonishing rate making the iterative aspect of ML more relevant. Complementing the rise of ML with advances in computing power has led to an enormous increase in intelligent mobility research. Even though new algorithms are published almost every few months [25], [96]–[101], these advances are failing to solve real-world problems. This is due in part to the high dimensionality of the working environment. That is not to say that the scientific community has not produced valuable contributions, but rather the reliability of the techniques used in intelligent mobility lacks confidence.

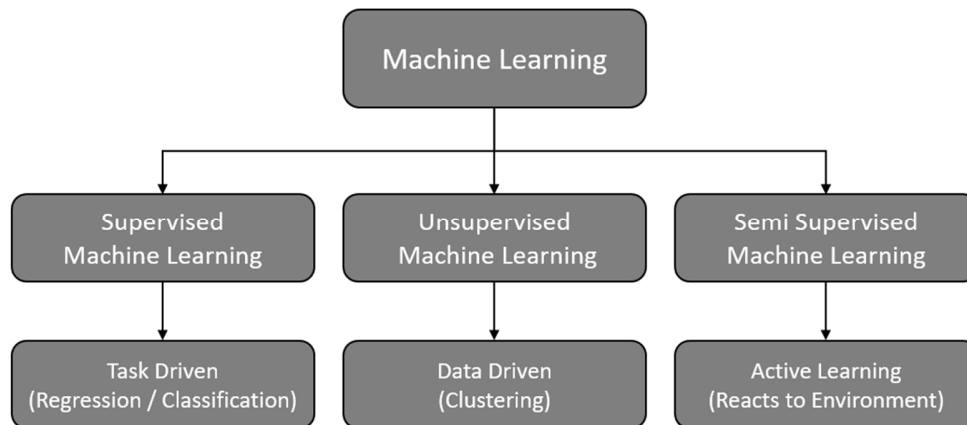


Figure 3: Three ML paradigms discussed in this chapter. Most of the research in this thesis is focused on supervised learning with a small diversion into semi-supervised learning.

One of the exciting debates about ML is how safe do AV have to be before they are let free on the roads [102]. On one side, people are saying that these vehicles need to be perfectly safe [103]. When one thinks about this, it is somewhat strange, considering that people use cars that are inherently unsafe at any speed [104]. Another school of thought is to get them out there, and over time they will learn how to be better, and we will get safer vehicles.

This argument can be reinforced by the fact that administrative bodies such as the Society of Automotive Engineers (SAE) and the National Highway Transportation and Safety Administration (NHTSA) have partnered up to develop a definition of autonomy [105]. The definition has five varying layers, starting after zero automation: Level One – Driver Assistance, Level Two – Partial Automation, Level Three – Conditional Automation, Level Four – High Automation, and Level Five – Complete Automation [106]. The first three levels still require a human operator, and the last two levels have high automation and do not require an operator at all. Many vehicles have standard Level One features, such as radar cruise control. Whereas, a more advanced vehicle, say a Tesla, is generally regarded to have high-Level Two automation. In all cases, they require the full attention of the operator, and can only operate in specific driving scenarios or structured environments. These systems do not make data-driven decisions.

There is a big jump between the lower levels and Level Four and Five. In the higher levels, machines should make decisions based on the data they have learned and the information they encounter at any point in time. The majority of commercial applications currently in use – where agents that have to act and perceive the world – have little or no data-driven decision-making involved [107]. In simpler terms, the action is not learned, and the

agent makes decisions based on rules rather than deriving a data-driven policy to make decisions.

The march towards a future where these systems underpin most of society's decision-making infrastructure is underway. So, we must understand the principles that will help us engineer for reliability. While it is ok for ML algorithms to misclassify a recommendation on YouTube, driving a car has a greater degree of gravitas. The reliability of the associated sensors limits the precision during operation. Data-driven ML plays a vital and integral part of solving some of the safety problems. Traditional ways of programming are not suitable for environments that need high dimensionality. As a result, researchers must rely on the different ML paradigms to fill the gap. Although it will not answer every question, they are the best tool available, given the complex problems that need to be solved.

A. Supervised Learning

In 1958 a psychologist called Frank Rosenblatt was inspired to create a single artificial neuron to perform binary classification tasks. His goal was to teach a machine to classify a single shape under his supervision [108]. Regarded as seminal work into Supervised Learning. Rosenblatt built a machine and wired it to a 400-pixel camera – he called it a Perceptron [109]. His experiments consisted of showing the machine images containing a triangle or a 'not triangle.' Depending on what each pixel saw, it would send a different electric signal to the Perceptron. If the total charge were above the threshold, it would send a signal to turn on a light, therefore indicating it saw a triangle. When the electric charge was too weak to hit the threshold, the light would not turn on. Rosenblatt used yes and no buttons to train the machine under supervision. Every time Rosenblatt pressed the no button, the machine would adjust the charge sent to the synapse of the Perceptron, and this changed the machine's threshold levels. This process improved the chances of the machine getting things right the next time; hence, the term Supervised Learning.

Rosenblatt's Perceptron used a simple stepwise activation function as a decision boundary. Suffering from one shortcoming – it only learned when it got things wrong, and it started from scratch without any prior knowledge of what it was meant to learn. Fast forward to the 1990s, and ML shifts from a symbolic approach to a data-driven approach [93], [94]. Since researchers began creating an algorithm for computers to analyse large amounts of data, numerous tools have been developed. One such development is the SVM. Conceived by Vladimir Vapnik, it is regarded as one of the most useful tools in modern statistical ML. The SVM combines fundamental concepts and principles related to learning, well-defined problem

formulation, and self-consistent mathematical theory [110]. Arguably, it is one of the best approaches to predictive learning, and it compares favourably to other, more empirical methodologies based on instinctive, asymptotic, and biological arguments.

The primary objective of Supervised Learning is to discover the pattern linking the inputs X to the outputs Y , when $D = \{(X_i, Y_i)\}_{i=1}^N$ given a dataset D of size N – with labelled input-output pairs [111], [112]. In the simplest terms, each input value has a dimension vector of numbers that represent the data we want the ML algorithm to learn. Commonly referred to as features, the dimensional vector is the understanding of X the algorithm develops during training. From the perspective of Y , the output, can be anything, but the assumption is that it matches a categorical or nominal variable from the dataset used during training.

When the output Y is categorical, the ML algorithm is regarded as a classification task, and when Y is a real value, the algorithm is known as regression. In classification, the primary goal is to learn the pattern linking the inputs X to the outputs Y , where $Y \in \{1, \dots, C\}$ and C is the number of classes [113]. If C equals two, the classification task is regarded as being a binary classification problem, and when C is greater than 2, it is viewed as being a multi-class classification problem. In a multi-class classification problem, the class can belong to two or more groups. When we use the term classification, we mean multi-class classification problem – unless stated otherwise. It can formalize this problem by making a function approximation where we assume $y = f(X)$ for an unknown function f . If the objective is to learn the function given a labelled training set followed by making predictions, we can describe the predictions in terms of f as $\hat{y} = \hat{f}(X)$. In this case, the objective is not just to learn the training data and identify the pattern, but to make a prediction on data not encountered before. While somewhat misleading, it is called generalization, and refers to the generalization of data varieties and not the ability of the classifier to solve general problems.

B. Semi-Supervised Learning

Semi-supervise ML is a combination of supervised and unsupervised ML methods. Online active learning is the amalgamation of online ML and active ML. Generally regarded as two distinct and separate paradigms of ML. For ease of description and because of their application in a later section of this thesis, the two styles have been combined, as depicted in Figure 4, and classified as a form of semi-supervised ML. Many researchers are already exploring both these fields of science and their applications to obstacle avoidance using a monocular camera [114], estimating depth from monocular imagery [115] and free space

classification using fused monocular and LiDAR sensors [116]. Traditional methods of ML are useful for interpreting sensor streams, but they need large amounts of annotated data to learn. Furthermore, while conventional ML algorithms work well in one area, they often do not generalize to new situations never seen before [117].

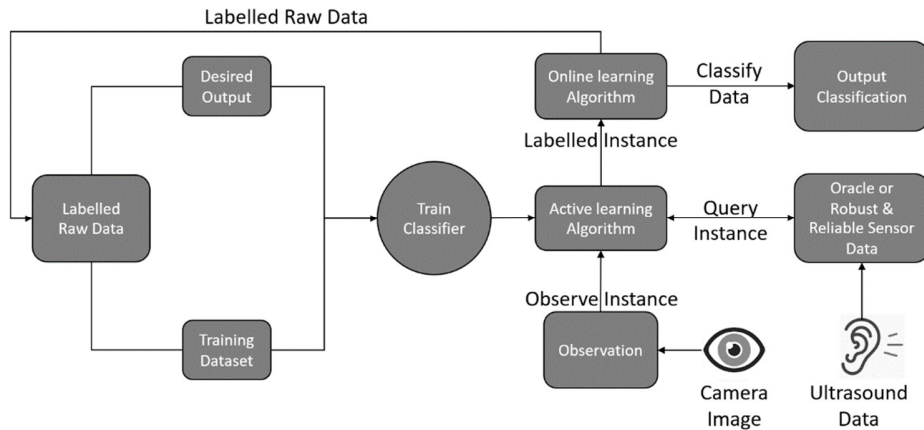


Figure 4: Shows the process of online active ML. At first, the algorithm uses little data to make a classification. As a new instance is observed, the active Learning algorithm queries it. The online Learning Algorithm updates the Classifier.

A solution to these issues is online active learning, where the intelligent mobile robot self learns and adjusts its perception of the surroundings as it encounters new data. The idea is to use a robust sensor stream to self-learn and improve the relative uncertainty of the new data it encounters. To understand online active ML, we examine both components independently. However, that requires an understanding of two different learning paradigms. Online ML is a form of supervised ML. online learning describes a technique where labelled data – that has just become available is added in sequential order to the dataset to update the classifier at each step [118]. In this technique, the classifier evolves forever, adding the new data it encounters and modifying the algorithm over time.

Online ML is used where it is impossible to train the algorithm on the entire dataset – referred to as out-of-core algorithms. In terms of online learning, the information has not become available yet, and therefore it is not possible to update the classifier with the new data until it arrives.

Conversely, active ML is a form of semi-Supervised ML that sits somewhere between Supervised ML and Unsupervised ML. In traditional methods of active ML, the algorithm queries the user for the label of the data it has just encountered [119]. Active ML is especially useful in situations where data is in abundance but is not labelled. It usually requires some labelled data to start, and then the user assists the algorithm as the new unlabelled data becomes

available. When combined with other ML methods, such as online ML, it becomes a powerful tool.

C. Unsupervised Learning

Not quite as prolific as Supervised Learning, Unsupervised Learning has an equally extensive and illustrious history. There is the distinct, Hebbian learning with its long-lasting influence over Neural Networks and ML paradigms [11]. And then the less apparent concepts by David Marr, who demonstrated an approach to brain function, grounded in an in-depth knowledge of the biological facts of the human mind. Marr's work showed how the Purkinje cells of the cerebellar cortex played a vital role in the mind's central pattern generator. The central pattern generator is crucial to the initiation of human locomotion and plays a vital role in Unsupervised Learning [120]. His work demonstrated how the human body develops patterns for performing specific tasks and can be regarded as the blueprint for Unsupervised Learning.

Perhaps a less abstract but most notable contribution was from Geoffrey Hinton into a way of learning called the Boltzmann Machine [121]. The Boltzmann machine incorporated many concepts from statistics that now dominate density estimation methods and clustering [122]. The Boltzmann machine is a symmetrical Neural Network, where neurons make arbitrary decisions about being on or off. In doing so, they learn features in the data by composing binary vectors that map the data into either 0 or 1.

More recently, the concept of clustering began to take hold. One technique of clustering partitions data into groups according to some criteria, transforming the data points to a higher dimensionality feature space [112]. Placing boundaries between the different groups – similar to an SVM – creates a divide between the higher dimensionality features and organizes it into a meaningful form. This type of unsupervised ML has been used to identify driver style [123]. Although this is not intuitively related to Intelligent mobility, the method is useful for studying driver policy [124]. K-means clustering is another form of Unsupervised ML and a form of vector quantization that is a popular choice in data mining. The process aims to drive a partition between n observations in which each observation belongs to the cluster with the nearest mean. This form of Unsupervised Learning is a popular choice for lane-detection [125], [126], however, with the onset of DL, it is beginning to be superseded [127], [128].

Unsupervised Learning is a much less structured problem than Supervised Learning and, therefore, more prone to making mistakes. The benefits of using such a system are that not all data needs to be labelled, so it closely represents the way humans learn critical skills. Unsupervised Learning has gained in popularity because of its ability to find previously unknown patterns in data. Also known as self-organized learning, its primary function models probability densities of given inputs to categorize its outputs [129], [130]. Unlike Supervised Learning, where the inference is a conditional probability distribution. Unsupervised Learning is used to find a prior probability distribution that is derived purely through deductive reasoning.

2.3.2 Regression & The Support Vector Machine

Regression is a statistical tool used to determine the relationship between two variables. Usually, one variable is dependent – denoted by a Y – and a series or single changing independent variable – denoted by an X . The most common form of regression is linear regression, where a vector, with the best fit to the dependent and independent variables, minimizes the sum of squared distances between the correct data and that of the line. Other types of regression, multiple linear regression and non-linear regression work using the same principle of dependent and independent variables. Although multiple linear regression more closely resembles linear regression, there is a distinct but subtle difference between the three forms.

The generalized models that describe the three forms of regression are relatively simple; Linear regression $Y = a + mX + u$, Multilinear regression $Y = a + m_1X_1 \dots + m_iX_i$ and a typical Polynomial Non-linear regression $Y = m + m_1X_1^1 \dots + m_iX_i^i$. Linear regression was the cause of one of the most significant scientific arguments of the 1800s. In 1805 Adrien-Marie Legendre published his seminal work ‘New methods for determining the orbits of comets’ where he described the basis of linear regression [131]. Four years later, in 1809, Carl Friedrich Gauss published work in the same field, where he described remarkably similar work on regression [132]. Gauss claimed to have invented the least-squares regression in 1795 and considered it so inconsequential and self-evident that he assumed someone must have discovered it before. Setting off one of the most renowned arguments in the history of mathematics, where Legendre disputed that Gauss deserved credit - it led to lifelong hostility between the two academics.

Some centuries later, and Linear regression has found a home in many different fields of study. One such field is Flock Traffic Navigation (FTN) – the study of traffic congestion using multi-agent technologies. In [133], researchers applied linear regression to FTN to study the response of agents. During the study, agents were instructed to cooperate with others to improve travel time by forming flocks. While not quite the field of Intelligent mobility, researchers in [134] studied the influence of agents interaction on neighbours by exchanging social beliefs. Although this research focused on social interaction, the same concepts are transferable to traffic management and driver policy.

Linear regression can be used for many different applications. However, in the real world, it is somewhat uncommon that a single dependent variable can have a relationship with a single independent variable. This is the case for multilinear regression that explains a relationship – both in a linear and a non-linear form – and works from the assumption that there is a relationship between every value of the dependent variable Y associated with the multiple independent variables X . The regression line for i explains variables $X_1, X_2 \dots X_i$ – when defined as $Y = a + m_1X_1 \dots + m_iX_i$ – and describes how the mean response changes. In [135], researchers described “descriptive, predictive and causal model,” describing how agents fight simulated forest fires. As with linear regression, this research is not case-specific but can be applied to driver policy and forecasting multi-agent response.

Traditional linear regression relates two variables with a straight line. In contrast, non-linear regression fits a non-linear line – usually a curve – linking a relationship to Y when Y is a random variable. The objective is to reduce the sum of the squared error so that the function best fits the data, where the line emulates Log, Trig, Exp functions, or other curve fitting methods. [136] reported on a non-linear task-oriented model for motion in the mobile environment of a robotic manipulator. While not quite driver policy or FTN, non-linear regression shows how interconnected and interrelated agents are modelled.

The simplest case of the SVM is when the data is linearly separable. Commonly referred to as binary classification, the goal is to find a decision boundary (Hyperplane) that linearly separates two different classes. Figure 5 depicts linearly separable data, a decision boundary, support vectors, and the margin. In this case, the decision boundary can be described as $w^T x + b = 0$. Where w^T is the transformed weight vector, x is the input vector, and b is the bias. Anything above the decision boundary falls into one class, and anything below falls into another class. These margins can be described as $w^T x + b = 1$ and $w^T x + b = -1$, respectively.

Labelling the margins in this way facilitates the description of the equation as a discrete function $f(x) = \text{sign}(w^T x + b)$. Subsequently, we can determine if a data point belongs to one class by checking that $y(w^T x + b) \geq 1$ or the other class $y(w^T x + b) \leq 1$. This accommodates for the margin on either side of the decision boundary and the nearest data points for both classes. If we scale the data such that anything on or above the boundary can be described as $w^T x + b = 1$ and anything on or below the boundary can be described as $w^T x + b = -1$.

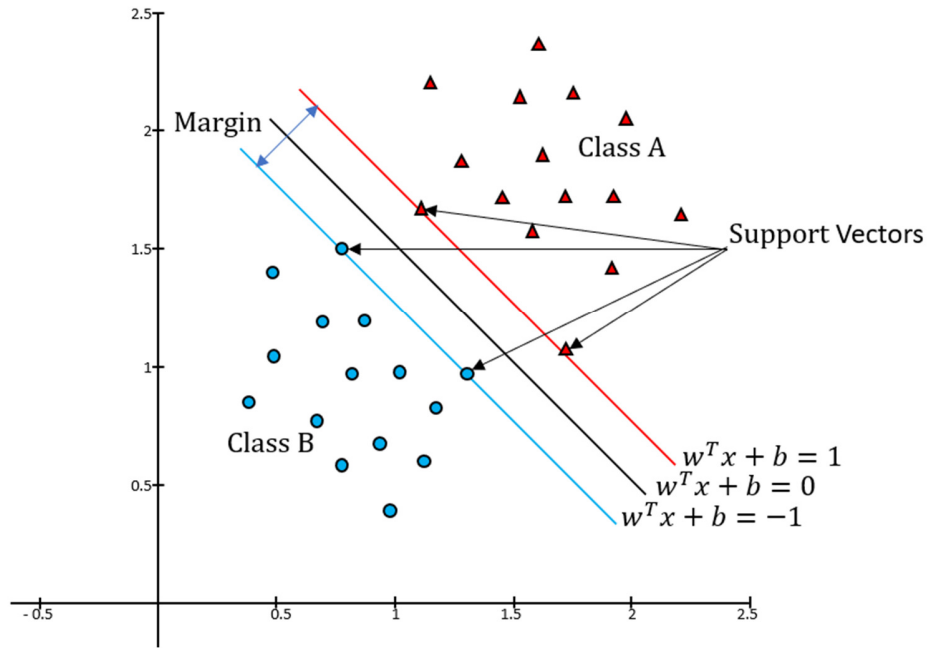


Figure 5: Shows linearly separable data, a decision boundary, Support Vectors, and the Margin. In this case, the data falls into one of two Classes.

If we wish to find the shortest margin (λw) width, we first acknowledge that the two vectors are parallel and therefore share the components w and b . Selecting a point x_1 on the lower margin $w^T x + b = -1$ we can say that the corresponding point on the opposing boundary $w^T x + b = 1$ can be described as $x_2 = x_1 + \lambda w$. Solving for λ we find:

$$w^T x_2 + b = 1 \tag{Equation 1}$$

From before we have $x_2 = x_1 + \lambda w$ making it possible to expand out equation 2 to become:

$$w^T (x_1 + \lambda w) + b = 1 \tag{Equation 2}$$

$$w^T x_1 + b + \lambda w^T w = 1 \tag{Equation 3}$$

If $w^T x + b = -1$ we can simplify equation 3 to become:

$$-1 + \lambda w^T w = 1 \quad (\text{Equation 4})$$

$$\lambda w^T w = 2 \quad (\text{Equation 5})$$

$$\lambda = \frac{2}{w^T w} = \frac{2}{\|w\|^2} \quad (\text{Equation 6})$$

If the shortest distance between the two margin boundaries is, λw , we can expand on equation 6 and say.

$$\lambda w = \frac{2}{\|w\|^2} w = \frac{2}{\sqrt{w^T w}}. \quad (\text{Equation 7})$$

A. Support Vector Kernel

The SVM is a supervised form of ML identified by Vladimir Vapnik in 1963 [110]. As the name describes, the vector is supported by the data. The vector describes an algorithmic approach to solving two-class or multi-class classification problems. An essential property of SVM is that any local solution simultaneously describes the global optimum since the determination of model parameters corresponds to a maximum margin (\subset convex optimization) [120]. Within SVM, the margin is defined as “*the perpendicular distance between the decision boundary and the closest of the data points*”[137][p327]. The maximization of the margin leads to the decision boundary that allows the classification of data points accordingly. Even though the SVM was designed as a two-class classifier, the different modifications, such as one-versus-the-rest or the one-versus-one approaches, allow classification of $K > 2$ classes.

Regression models are not too dissimilar to the SVM. In regression problems, the best fit line describes the relationship between an independent and dependent variable. The function that describes the best fit line is similar to the kernel function of the SVM [138]. The kernel function is a type of transform function applied to each data point. It maps the original non-linear observations into a higher-dimensional space so that the decision boundary can be easily found. Unlike SVM classification, regression models are predictive models that are applied to new data where we do not have the answer. When the kernel function of an SVM is linear, the model responds in a similar way to regression. However, it is quite rare in the real-world that the relationship between data is linear. This is the point of the kernel function; it uses a linear classifier to solve a non-linear problem. It transforms non-linear data into a linear plane so that the function can be applied.

Mathematically described as $K(X_i X_j) = \langle f(X_i) f(X_j) \rangle$, where K is the kernel function, $X_i X_j$ are the dimensional input variables, and f is the mechanism for mapping the dimensional input variables from the non-linear to the linear space using the dot product. This requires the calculation of $f(X_i)$ and $f(X_j)$ first, resulting in a two-step process that can be computationally expensive. These functions can have different types; Linear, Non-linear, Polynomial, Radial Basis Function (RBF), and Sigmoid.

B. Hyperplanes

Both regression and the SVM has been a favourite of the scientist working in the field of AI. For example, regression has been used for lane detection [139], whereas the SVM has been used for people detection the past [140]. However, to truly understand both we need to gain an understanding of hyperplanes. The Hyperplane is the resulting decision boundary that maximizes the margin between the two data types. Both margins and the Hyperplane are found using quadratic programming, where a function is maximized subject to one or more constraints [141]. Vladimir Vapniks genius comes into play here, as having used the kernel function to map non-linear observations into a higher-dimensional space and identified the optimal Hyperplane; resulting in data that is linearly separable. In ideal circumstances, the SVM should produce a hyperplane that linearly separates the data points into two distinct and separate classes. It is not always possible to do this, and more often than not, a hyperplane is derived to maximizes the margin and minimizes the misclassifications. In this case, we use a non-linear region to separate the groups more efficiently and then apply the kernel trick to map the non-linear vector into a linear plane.

2.3.3 The Artificial Neural Networks

ANN or Neural Networks are a form of supervised ML that perform classification tasks by learning patterns from previously labelled examples. Generally, computers are good with repetitive calculations and detailed instructions but are bad at recognizing patterns [142], [143]. Neural Networks solves this problem by breaking intricate patterns down into a series of simpler patterns [40]. For example, when a machine must decide whether an image contains a particular object, a Neural Network uses the edges to detect different features of the class in question. Only upon combining all the features to reconstruct the target class can the network estimate what the object is [84], [85]. In the most basic form, an ANN is a computation model used to solve problems by recognizing the pattern in a specific dataset. There are many different types of ANN; however, not all of them are relevant to this research.

A. Feedforward Neural Networks

The Feedforward Neural Network is the simplest form of an ANN [109], [144]. In these Neural Networks, data travels from start to end in one direction only. The architecture of a Feed-Forward Neural Network – depicted in Figure 6 – is relatively straightforward; there is an input layer, an output layer, and connected by one or more hidden layers [84], [85]. Each connection – like the synapse of a biological neuron – transmits information to the proceeding neurons – Feed-Forward propagation – before resulting in a score [145]. When implementing a Feed-Forward Neural Network, the input signal is a real number representing the data to be classified [146].

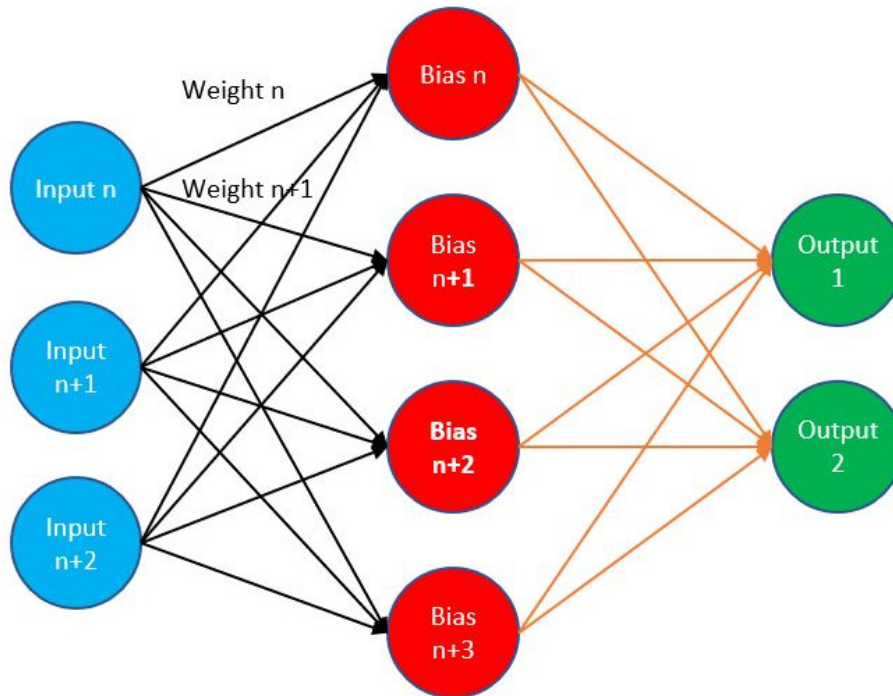


Figure 6: Shows a feedforward neural network is an ANN in which connections between the nodes do not form a cycle. The feedforward neural network was the first and most straightforward type of artificial neural network.

The Feedforward Neural Network was developed to correct the inaccuracy of the perceptron developed by Frank Rosenblatt [40]. Primarily used by supervised ML paradigms where the data to be learned is neither sequential nor time-dependent. These networks showed that a layered web of perceptrons outperformed a single perceptron when faced with complex multi-class problems [40]. The main shortcomings of Feedforward Neural Networks are their susceptibility to noise, making it easy for them to misclassify. Typically, when analysing simple problems, essential classification tools like perceptron's are good enough. However,

Consider the number four in full true colour splendour. If we take the red layer and separate it from the others, we still have the number four, but in the red colour only – as per Figure 7 (a). If we were to use an ANN to process this image, it would require flattening the image out into a column vector [149], lining up the data one column after another – as per Figure 7 (b). If the data is processed as a column vector and we do not know the image size or resolution, the neural network cannot easily understand the representation – it loses the spatial arrangement of the different pixels that compose the image.

	25	2	1	44					25.6	2.3	14.2
	223	7	6	60					225.1	8.8	24
	196	8	2	148					198.4	8.6	46.4
	249	1	3	40					249.3	1.9	15
	60	7	1	154					62.1	7.3	47.2
	59	1	7	213			Down Sampling Factor		59.3	3.1	70.9
	214	7	3	163		1	0.3		216.1	7.9	51.9
	89	182	219	13					143.6	247.7	222.9
	74	146	113	72					117.8	179.9	134.6
	89	18	244	85					94.4	91.2	269.5
	1	4	8	97					2.2	6.4	37.1
	3	4	2	121					4.2	4.6	38.3
	2	1	2	131					2.3	1.6	41.3
	7	6	8	47					8.8	8.4	22.1
	3	5	5	126					4.5	6.5	42.8
	7	6	8	121					8.8	8.4	44.3
	5	3	1	237					5.9	3.3	72.1

(A)

(B)

Figure 8: (a) Shows the original data of image four. (b) Shows the reduced horizontal dimensions of the image data still retains the spatial arrangement.

CNN's address this issue through a series of down sampling and pooling filters. They retain the relationship between data points when they convert the image to representation. If we extract the features from the original image, such that the spatial arrangement is preserved [150], it makes the classification process easier. This process is depicted in Figure 8 (a) and (b), where the down sampling factor modifies the original data. Once acted up, the data reduces the dimensions of the image, while still retaining the spatial arrangement.

It should be noted that the down sampling factor we use in Figure 8 take two consecutive horizontal pixels, and therefore only affect the horizontal dimensions of the data. Moreover, the leftmost and rightmost data points are only acted upon once by the down sampling factor. In consequence, data on the right and left edge of the image is influenced to a lesser degree than data towards the centre of the image.

Reducing the dimension of the image allows us to retain the relationship between image data and representation the network learns. On occasion, we might not want to reduce the dimensions of the image but rather increase them. In the situations where we do not want to reduce the dimensionality, we pack out the image with zeros, thus reducing the impact on

the peripheral data points [151]. Alternatively, in situations where we want to reduce the dimensionality and the influence the down sampling factor have on peripheral data points, we take multiple weight in a single turn and merge the two images [151]. Figure 9 (a) and (b) show the image data after having been acted on by the weight values (1 0.3) and (0.5 1), respectively. In both cases, the dimensionality has reduced the image data from a 17 by 4 matrix to a 17 by 3 matrix. Combining Figure 9 (a) and (b) produces an image with reduced dimensionality that retains more information about the original image than a simple column vector representation.

	25.6	2.3	14.2					22.5	6	220.5
	225.1	8.8	24					146.5	33.5	303
	198.4	8.6	46.4					138	14	741
	249.3	1.9	15					129.5	15.5	201.5
	62.1	7.3	47.2					65	8.5	770.5
	59.3	3.1	70.9					34.5	35.5	1068.5
	216.1	7.9	51.9					142	18.5	816.5
	143.6	247.7	222.9					954.5	1186	174.5
	117.8	179.9	134.6					767	638	416.5
	94.4	91.2	269.5					134.5	1229	547
	2.2	6.4	37.1					20.5	42	489
	4.2	4.6	38.3					21.5	12	606
	2.3	1.6	41.3					6	10.5	656
	8.8	8.4	22.1					33.5	43	239
	4.5	6.5	42.8					26.5	27.5	632.5
	8.8	8.4	44.3					33.5	43	609
	5.9	3.3	72.1					17.5	6.5	1185.5

(A)

(B)

Figure 9: (a) & (b) Show the image data after having been acted on by the down sampling factor (1 0.3) and (0.5 1), respectively. In both cases, the dimensionality has reduced the image data from a 17 by 4 matrix to a 17 by 3 matrix.

Up until this point, we have been using a down sampling factor that takes two consecutive horizontal pixels. In most cases, there is a requirement to maintain the spatial arrangement in both horizontal and vertical elements of the image [151]. Achieved using two rows of two consecutive horizontal pixels or a 2 by 2 matrix. It should be mentioned that the same dimensionality reduction that occurs on the horizontal plane also occurs on the vertical plane, thus further reducing the size to a 16 by 3 matrix – as per Figure 10 (a) and (b).

This process of extracting the features from the image while retaining the horizontal and vertical spatial relationship is vital so that the network can understand how the pixels are arranged. In its purest form, a CNN is a DL paradigm that can take a set of images, assign a down sampling factor to various aspects in the image, max pool the down sampled image data before passing it through the fully connected neural network to make a classification [152]. During training, a CNN learns to classify an image from previously labelled data [153] using this process. Much like the description of neural plasticity set out by Hebb, during training

within a specific window. A similar process occurs during GlobalMax and global average pooling layers. In all cases, the purpose is to downsample an input representation to reduce the image size. The process will inherently make assumptions about features contained in the window [156].

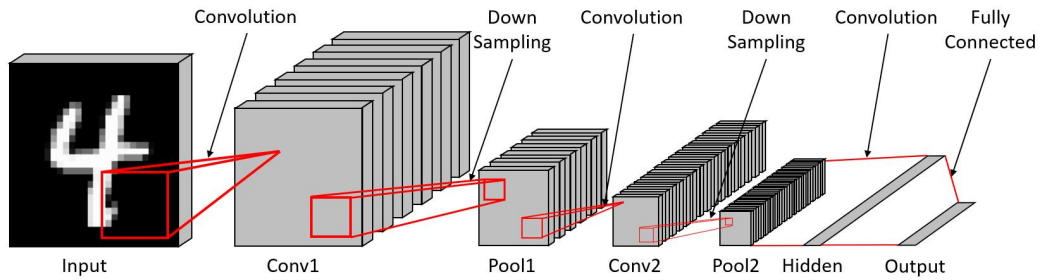


Figure 11: The generalized architecture of a CNN showing the three main elements of CNN; Convolutional Layers, Pooling Layers, and the Output Layer. Note the process that occurs between the different layers of the network [152].

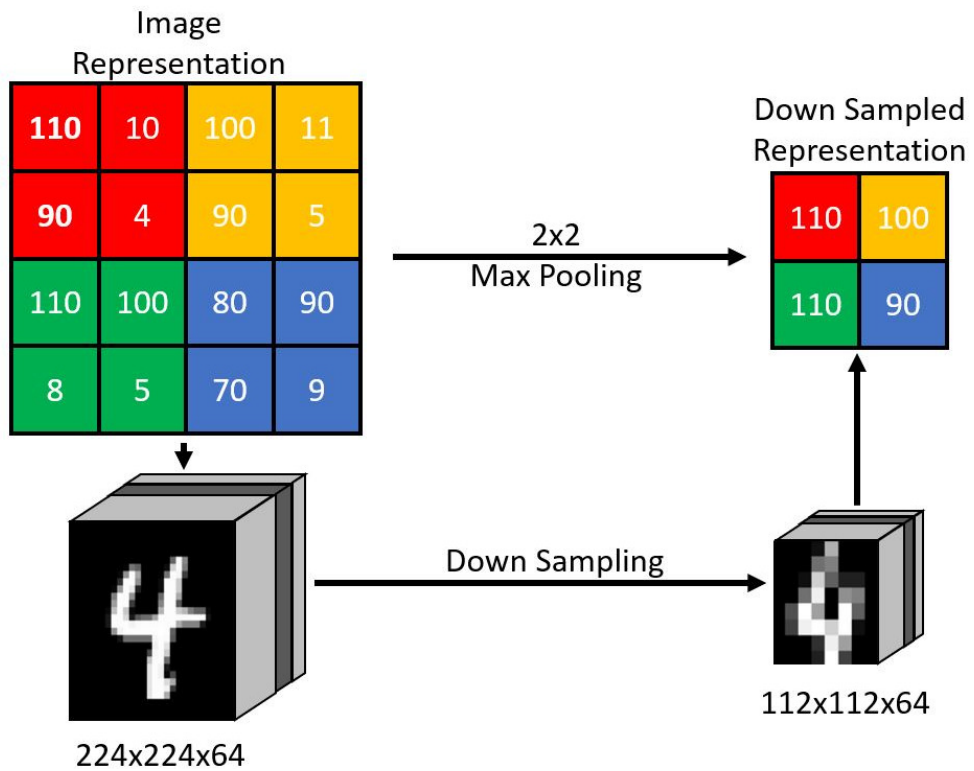


Figure 12: Shows the down sampling process when using a Max Pooling layer filter with dimensions of 2 by 2 acting on a 4 by 4 sample of data from an image representation.

Consider a 4 by 4 sample of data from an image representation. Passing a 2 by 2 filter over the input data at incremental steps of 2 pixels per stride with no overlapping, we can extract certain desired features. In the case of the Max pooling layer, the region that the

window passes over extracts the maximum value to the output matrix. The resulting output matrix contains the maximum value from each window for a region. This process is depicted in Figure 12.

The final layer of the generalized CNN is the output layer. It comes after multiple layers of convolution, data padding, and down sampling of the image representation. At the output layer, the different classes in the network are formed. We need the output layer since the convolution, and pooling layers are not capable of generating a class.

To generate the output equal to the number of classes, we use a fully connected Neural Network. In the output layer, the network uses an activation function like Rectifier Linear Unit (ReLU), a loss function like categorical cross-entropy, to compute the error in classification. During this process, the fully connected layer learns the representation features passed from the convolutional layers using forward and backpropagation, just like a Neural Network.

C. Activation Functions

The function that acts on the inputs of an ANN is called an activation function. Also known as a Transfer Function, it takes many different forms. The Activation Functions of neurons are the same for each element of the network, so it is the weights and bias that influence the score.

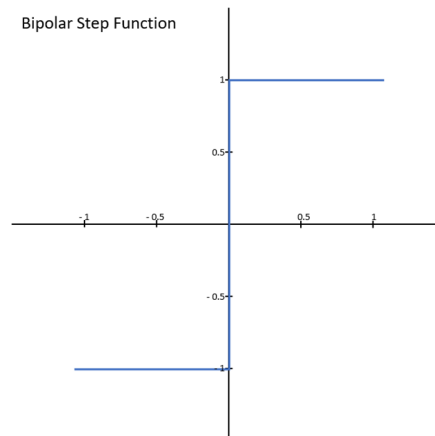
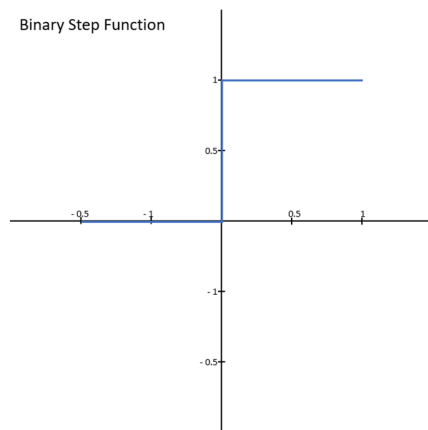


Figure 13: Shows the Binary Step activation function [157]. Figure 14: Shows the Bipolar Step activation function [157].

McCulloch and Pitts first proposed the basic structure for modern-day Neural Networks in 1943 [74]. Quite different to the Perceptron, there are three fundamental elements define a Neural Network; an information processing element, the organization of the connections (fully connected or convolutionally connected), and the training techniques used to update the weights and bias (loss function).

The biological neuron proposed by McCulloch and Pitts processes inputs using an integration function related to the input of a neuron. Contemporary Neural Networks, on the other hand, use a range of different non-linear activation functions. It should be noted that if we use a linear activation function, the obtained output would be the same as that of a single layer network. For this reason, the activation functions are almost always non-linear, some of which we describe below and shown in Figure 13, 14, 15 and 16.

The binary step activation function [157] is widely used in single layer network to map the input to output in binary form (1 or 0) – depicted in Figure 13 and defined as [158]:

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (\text{Equation 8})$$

Bipolar step function [157] where the threshold value is represented by a 0 is widely used in a single layer network to convert the input to an output that is either +1 or -1 – depicted in Figure 14 and defined as [159]:

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (\text{Equation 9})$$

The sigmoid function [160], depicted in Figure 15, focuses on the relationship between the function's value and the value of the derivative. This focus reduced the computational cost required by the network. Widely used by networks that apply backpropagation, sigmoid function falls into two different groups – Binary or Bipolar sigmoid function.

The Binary Sigmoid – also known as the unipolar sigmoid function – uses a steepness parameter, and outputs the values in the range of 0 to 1. The Bipolar sigmoid uses the same steepness parameter but outputs values in the range of -1 to 1. The Binary Sigmoid and Bipolar Sigmoid can be defined as [159]:

$$sig(x) = \frac{1}{1 + e^{-\beta x}} \quad (\text{Equation 10})$$

$$sig(x) = \frac{1 + 2}{1 + e^{-\beta x}} \quad (\text{Equation 11})$$

DL and CNNs have increased the popularity of the ReLU activation function [161]. If one processes images or performing research in the field of computer vision, ReLU is usually an excellent first choice [162]. The unique characteristics of ReLU is a linear identity for all positive values and a zero value for negative values.

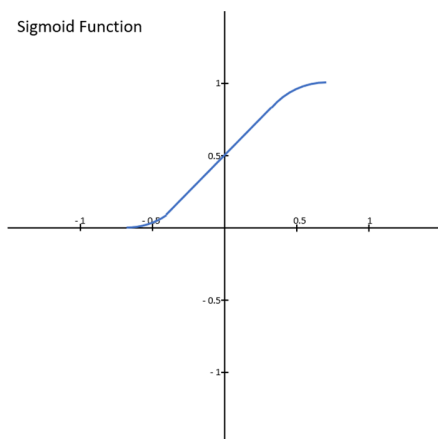


Figure 15: Shows the Sigmoid activation function [160].

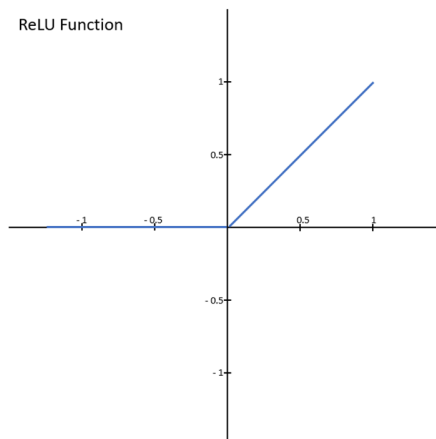


Figure 16: Shows the ReLU activation function [161].

These characteristics mean that ReLU is frugal on computational resources; takes less time to train and run; converges faster so that the slope does not plateau when the inputs get large, and – perhaps the most important – lacks the vanishing gradient problem suffered by some of its counterparts [146]. The vanishing gradient problem can be regarded as a major shortcoming of Feedforward Neural Network. Up until 2016, when researchers presented ResNet, the majority methods of dealing with the Vanishing Gradient problem was through hardware [148]. Before ResNet, when training a network, the error gradient would decent at a faster rate at the beginning. Towards the end of the training, the network is unable to propagate useful information from the output, and the error descends at a slower rate. ReLU overcame this issue through its unique characteristics when handling negative inputs – it fires for positive inputs only. The ReLU activation function depicted in Figure 16 can be described as [163]:

$$f(x) = \max(x, 0) \quad (\text{Equation 12})$$

Unlike the biological nervous activity that ANN is meant to model, most activation functions fire all the time. Contrary to this, nervous activity is sparse, and the different neural circuits within the human brain are activated at different times. For example, the Occipital Lobe in the cerebral cortex is primarily responsible for vision. Next to the occipital lobe is the Parietal lobe [164], whose primary function is understanding spatial relationships between objects. While both functions frequently fire together, there are occasions where they fire apart.

The ReLU function replicates this process when handling negative inputs that equate to zero. So, when ReLU fires, it is more likely that the artificial neuron is processing meaningful aspects of the problem rather than information that will not help at all. The effects of sphericity do not always complement ReLU, and on occasion, can detract from its use. Since

the slope of ReLU equates negative ranges to zero, once a neuron goes to zero, it is unlikely to recover. Although these neurons do not necessarily complement the problem in discriminating the input, over an extended period, the zeros add up. In the end, we find a large part of the network does nothing. This effect, commonly referred to as dying, usually occurs when the learning rate is too high, or there is a significant negative bias [165]; however, a lower learning rate often mitigates the problem.

D. Loss Function

The loss function probably has the most significant impact on the performance of the Neural Network. The primary purpose of the loss function is to measure how good the network performs with respect to the target and expected class [166]. At the start of training, the difference between the two is significant. In ideal circumstances, over time, the Loss will drop towards zero. While the Loss rate of change is dependent on the weights and bias, the actual function returns a scalar that describes how good the network performed as a whole. It is not a vector. In its purest form, the Loss function optimizes the parameters of the Neural Network by minimizing the loss.

In practice, we calculate the loss by matching the target class value to the predicted class value generated by the network (a probability). Then, the weights and biases alter using the gradient descent, so the loss is minimized. There are various loss functions available depending on both the objectives and the activation function used in the network. They fall into one of three groups: Regression Loss Functions, Binary Classification Loss Functions, and Multi-class Classification Loss Functions and are depicted in Figure 17.

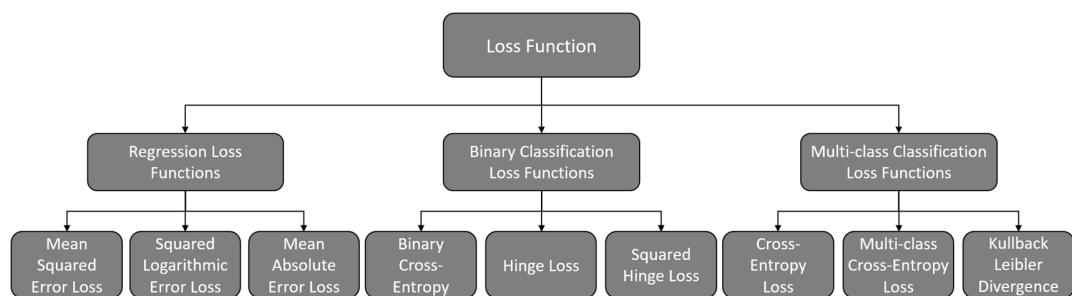


Figure 17: The Regression Loss, Binary Classification, Multi-class Classification loss function and their subdivisions [167].

Of specific relevance to this research are the mean squared error loss function, binary cross-entropy loss function, and the multi-class cross-entropy loss functions. The mean squared error loss function is used in most regression problems. Regarded as the favoured Loss function for the maximum likelihood estimate, where the distribution of the target class is

Gaussian [168]. In practice, we determine the mean squared error from the average of the squared difference between the predicted and actual class. Returning an ideal value of zero, the function always returns a positive result that influences the significant errors.

For binary classification problems, the Binary Cross-Entropy Loss Function is the ideal choice. It is mainly used when the target class can be zero or one. In practice, it is the favoured loss function under the inference framework of maximum likelihood [168]. When the binary cross-entropy loss function calculates a one, the average cost between the actual and predicted the average difference is summarized. Conversely, the score is minimized when a perfect cross-entropy value of zero.

Multi-class Cross-Entropy is the Loss function most used for multi-class classification problems. Somewhat like binary cross-entropy loss function, the intended application is where the target values are in the form {Class 1 Class 2, Class 3...Class n}. In this case, each class is assigned an integer value that indicates the likelihood, making it the preferred loss function under the inference framework of maximum likelihood. In practice, the score is summarized in the same way as the binary cross-entropy loss function. Whichever loss function is used, it needs to match the activation function since the loss benefit is continuously being evaluated, when comparing the predicted output to the actual output.

E. Backpropagation

Back Propagation is the mechanism in which loss is minimized. This is done by making small adjustments to weights and bias during the training process. It influences the gradient descent until the lowest acceptable loss is reached [146]. It assists these changes and enables a network to learn a perimeter by understanding how small changes in values for weights and bias affect the output. When a small change in value returns a small change in the output, only a small change in the network has occurred. Networks that only ever make small changes do not have the opportunity to learn and never make the giant network change that is required for autonomous decisions [169]. Moreover, the gradient of the network's output concerning the parameters in the first layers becomes extremely small; hence, the term the vanishing gradient problem [170].

The vanishing gradient problem is mainly dependent on how the activation function passes inputs into a small output range in a non-linear manner [170]. Sigmoid functions, for example, map real numbers onto a range between 0 and 1, resulting in large regions of the input mapped onto a small range. Even a significant change in the input results in small

changes in the output. For example, the first layer of a Deep Net passes a vast region of an image onto a small output region, which in turn is passed onto the next layer until it reaches the output. The net result is a small change in output, even though a significant change in input has occurred. Non-linearity's stack up, amplifying the problem, and forces the gradient ever-smaller [170], [171].

In 2006 Hinton, Osindero, and Yee-Whye Teh published breakthrough work on the vanishing gradient point problem [85], [171]. In real-world terms, the gradient can be thought of as a hill and the training process as a wheel rolling down the hill. The wheel rolls fast along a surface with a large gradient and slows along the low gradient. The same is true of a Deep Net – at the early stages of the net when there is a small learning curve, and the progress of the net is quite slow. However, towards the end, where there is a much larger learning curve, the net learns at a much quicker rate [171].

Giving way to a singularity, the layers at the start of the net are responsible for identifying simpler patterns and laying the building blocks of an image. If the layers at the start of the net misperceive things, then the next layers also get things wrong. When a net wants to learn, it starts looking at errors to identify the weights and bias that are affecting the output, before attempting to reduce the error by changing the weights [172]. This process is known as backpropagation and is used for training nets. It removes the issues created by the vanishing gradient problem [172].

2.4 Multimodal Machine Learning

Despite numerous developments in decision-making algorithms, the majority were demonstrated in tightly controlled settings with well-defined outcomes, performs singular tasks and use only one sensor stream [173]. For AI to perform as desired – that is to support humanity or even to supersede human intelligence – algorithmic developments should be adapted to perform in real-world environments. The decision-making process of an agent in such an environment is extremely complicated due to the interactions with other unpredictable agents, and the need to adhere to multiple constraints. Therefore, developing agents that make decisions using context information gathered from real-world environments, remains a mammoth task. The context of a scenario can be captured from various types of instruments, measurement techniques, and sensors. Also known as Multimodal sensing, it's where multiple sensor modalities capture context information about the environment there working in [174]. Multimodal sensing is the effective utilization of diversity captured by multiple heterogeneous

sensor streams [175]. Commonly referred to as sensor data fusion, the challenges can be categorized into two groups [116], [174]: challenges at acquisition level and challenges due to the uncertainty of data sources. Acquisition level challenges include differences in physical units of measurement (non-commensurability), differences in sampling resolutions, and differences in spatiotemporal alignment. [116] reported on a process of sensor data fusion using a Gaussian process.

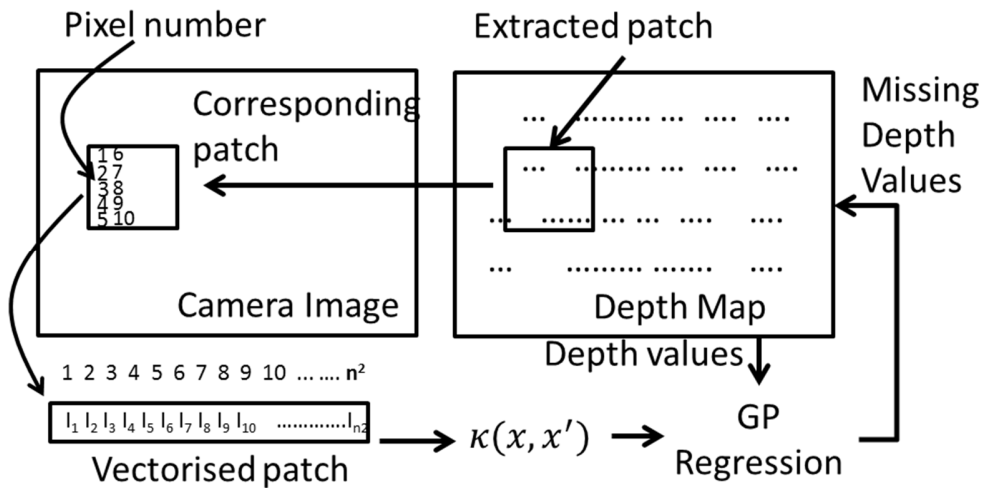


Figure 18: Shows the sequence of steps in the Gaussian process resolution matching algorithm. On the left side of the image is the high-resolution camera data, and on the right is the lower resolution LiDAR data [116].

The process, depicted in Figure 18, addressed some of these issues regarding fusion and a method of improving the resolution mismatch. The challenges due to uncertainty in data sources include noise such as calibration errors, quantization errors or precision losses, differences in the reliability of data sources, inconsistent data, and missing values. Overcoming heterogeneity of different sensors through effective utilization of redundancy across the sensors is the key to fusing different sensor streams.

2.5 Applications of Machine Learning in Intelligent Mobility

For automotive manufacturers, AV will shape the future of driving and their manufacturing strategy. In an industry that has been historically slow to react to change, when AV becomes ubiquitous, it is more likely than not that technology company will be at the forefront. Typically AV are composed of three technological pillars listed below [116], [176], [177]. Although the communication aspect of connected autonomous vehicle (CAV) can be considered a fourth pillar, it was regarded as beyond the scope of this research and, therefore, not reported.

1. Sensing and perception – the primary component – are responsible for understanding the surrounding environment the vehicle encounters. Logged data is used to make decisions about the direction of travel, obstacles the vehicle encounter, accelerating, and retardation [116], [176].
2. Localization and mapping is the second technological component driving AV [178]. While GPS can be used for this purpose outdoors, it ceases to function effectively in indoor environments. Therefore, localization performed by mapping the surrounding environment and comparing it against the historical data is regarded as a superior method [116], [176].
3. The final technological component is Driver Policy. Driver Policy is responsible for deciding the direction of travel when the vehicle interacts with other road users. When to stop, yield to other road users merging with traffic and over overtaking other vehicles are all dictated by driver policy [116], [176].

With the advances in autonomous technology, many autonomous driving competitions have been promoted by different bodies to accelerate the development of AV. VisLab Intercontinental Autonomous Challenge (VIAC) was one such competition that ran from July 20, 2010, to October 28, 2010. The challenge aimed to stress-test the current technology in a unique event - a 13,000 km trip starting in Italy and ending in China. Traversing remote areas in Russia, Kazakhstan, and China - for which no map was available. Part-funded by the European Research Council (ERC), the challenge showed that it was possible to transport goods between two continents with minimal human interaction [36], [177], [179], [180].

Further north of Italy, there was MadeInGermany and Spirit of Berlin. Both vehicles were the first cars licensed for autonomous driving on the streets and German roads [181]–[183]. Since initial conception, both projects have come a long way in developing technology for driver assistance as in [182], [184], and systems for fully AV as in [185]. Both vehicles were equipped with GPS, three laser scanners at the front and rear of the vehicle, several high-resolution cameras, and multiple radars. Perhaps the most renowned research group to come from Germany is the AnnieWAY autonomous platform at Karlsruhe Institute of Technology. The AnnieWAY autonomous platforms have been driven entirely autonomously for over 100 km without human interaction [186]–[188]. The vehicles use LiDAR, radars and stereo

cameras to develop the Karlsruhe Institute and Toyota Technological Institute (KITTI) dataset [189]–[191].

Outside of the northern hemisphere, research by Toyota has shown promising results using a technology called Intelligent Transport Systems (ITS). ITS connects vehicles with other vehicles, pedestrians, and the road. It relays information from LiDAR, Radar camera, and GPS. Toyota’s semi-autonomous platform has been engineered in such a way to prevent accidents [192].

Riddled with strict rules of operation and a severe lack of legislation, AV’s are restricted in what they can do. Outside of the field of AV, where the rules of operation are not quite severe, there have been significant advancements in the application of ML to Intelligent mobility. For example, assistive autonomous robots that help humans in day-to-day tasks are becoming increasingly popular in domestic and industrial applications. Indoor cleaning robots [193], [194], surveillance robots [195], lawn mowing maintenance robots [196], [197], and indoor personal assistant vehicles for the disabled [198] are a few applications of ML for Intelligent mobility. Soon, one of the most popular consumer applications of mobile robots will be self-driving cargo vehicles [199].

While several automobile manufacturers have set targets to launch commercially available fully autonomous driverless vehicles by 2020. Vehicles that are capable of roaming without human intervention are a distant reality that will require extensive research effort to make them a reality [200]. Although AV, for the most part, performs human-like control, they lack the data-driven decision-making ability needed for them to operate on their own [107].

2.6 Summary

Many cognitive pathways are employed to survey a visual scene before judgment, and associated action is made. During this period, objects in the sensory range are identified and classified. The sound, neck, and extra-ocular muscle contribute to this ability to geo-locate through a form of sensor data fusion, using biological signals. When the Occipital lobe becomes stimulated, dopamine and other neurotransmitters flood the nucleus accumbens and the central nervous system forcing the hippocampus to recall memories relating to the objects they see. For example, dopamine, a neurotransmitter that helps control the reward and pleasure centres of the brain, muscle innervation, and emotional response, enable humans not only see rewards but to take action to move towards them. Through reuse, the connections we form,

become robust and more relevant. In turn, we come to rely on these connections, adding gravitas or weight to the particular circuit.

Typically, an ANN – commonly referred to as Neural Networks – is a computer algorithm that imitates the biological function of the human brain. It contains virtual neurons that arrange in interconnected layers. Each artificial neuron passes on information by performing calculations in a similar way to the human brain. In the neural network, the input to each neuron is acted on by a function (activation function) that generates an output with typical values between 0 and 1.

The changing and shaping of connection in our brain is referred to as neuroplasticity. Neuroplasticity is the template that Neural Networks. mimic. Without this concept, we would not have modern-day CNN. It allows for a neural circuit to adapt. Conceived by Donald Hebb in his now-classic Hebb's postulate, it states that when two neurons fire at the same time, the connection between them is strengthened – becoming more likely to fire again in the future. This concept became the foundation for what we regard as the foundations of AI – neural networks.

AI is a broader umbrella under which ML, Representation Learning, and DL come. ML – the main subset of AI – can be broken down into two main groups – Supervised ML and Unsupervised ML. Even though new algorithms into ML are published almost every few months, it can be said that these advances fail to solve the real-world problems due to the high dimensionality of the working environment. This is not to say that the scientific community has not produced valuable contributions, but rather the reliability of the techniques used in Intelligent mobility seriously lacks the confidence to operate on their own [201]–[204]. The sobering truth is that most commercial applications today were agents that must act and perceive the world – for the most part, have no data-driven decision-making involved.

From the emergence of neural networks more than a decade ago, through to recent breakthroughs such as deep-neuroevolution, AI has made giant leaps of recent progress. Knowing that the fundamental goal of AI is to produce AGI that interacts with its environment to learn optimal behaviours for autonomous decision-making. Despite numerous developments in decision-making algorithms, the majority demonstrate in tightly controlled settings with well-defined outcomes. If AI is to live up to its science-fictional promises to support humanity or even to supersede human intelligence, algorithmic developments should be adapted to perform in real-world environments.

To address the shortcomings in decision-making algorithms, we focused on perception and cognition abilities of machines. Using supervised multimodal ML methods, we address higher-level context gathering cognitive skills. The way these networks process data is emulated the research performed by Donald Hebb. Not content with focusing solely on supervised multimodal ML methods, we examined the lower level perception skills using semi-supervised ML and sensor data fusion. These low-level perception skills emulate the way humans fuse sensory information to enhance their abilities. To develop algorithms that demonstrate low-level perception skills and high-level cognitive ability, we developed two datasets using an intelligent mobile robot designed and assembled in house. These contributions are reported on in Chapter 3, Chapter 4 and Chapter 5.

Chapter 3 Architecture for an Intelligent Mobile Robot

3.1 Introduction

Horse and carriage use dates back to 1900 BC [205]. While this is not entirely an AV – the humble horse and cart can be viewed as the first Intelligent vehicle. The horse and cart exhibit many of the autonomous qualities reported on in this research. They can adjust speed, direction, and detect objects for a variety of scenarios. They are so good at their job and are still in use today in many locations around the world.

In 1912 the Sperry Corporation developed an altitude and gyroscopic based orientation indicator to hydraulically operate the flaps and rudders of a plane [206]. Lawrence Sperry demonstrated this at an aviation safety contest in Paris in 1914. Some concepts, such as artificial horizon are still in use today [207]

It was not until 1930 that the first marine autopilot system came about in the form of a simple wind vane. Originally this crude system consisted of a counterweight that compensated for wind direction during gusts. Several iterations of this system evolved before a more sophisticated system called Braine Gear came into use [208]. Except for the horse and cart, the systems have evolved to become so complicated that they now work on other planets.

The first autonomous rover to work on another planet was Sojourner on the Mars Pathfinder Mission. From touchdown on the 4th July 1997 until Sojourner stopped transmitting

on the 27th of September 1997, the pathfinder mission captured over 16,500 images (550 images from Sojourner) during its extra-terrestrial mission [209], [210]. Sojourner was regarded as the pinnacle of rover technology. However, the quantity of data captured during its three-month mission is dwarfed by the data captured and the period of operation of Opportunity and its sister rover Spirit.

Opportunity and Spirit were part of NASA's long-term Mars Exploration Program. Both Opportunity and Spirit started their mission in the January and February of 2004. They finished at different times – Spirit in 2009 and Opportunity 2018 [211]. During its period of operation, Opportunity travelled 45.16km captured and transmitted 217,000 images back to Earth. Equipped with geological equipment – a Panoramic camera, a Miniature Thermal Emission Spectrometer, a Mössbauer Spectrometer, an Alpha Particle X-Ray Spectrometer, Magnets, a Microscopic Imager, and a Rock Abrasion Tool – designed to be the mechanical equivalent of a geologist unearthing a site [211]. The platform was fitted with a monopod – to which the camera was mounted – and a six-degree manipulator that can place the instruments in the correct position. Just like the geologist's rock hammer, the Rock Abrasion Tool was used to reveal the interior of samples, the robot found [209], [211]. While these sensors are quite different from the ones we expect to find on an AV, it is still a robot that performs the task.

Throughout history, these machines were used for different elements of intelligent mobility. The difference between them is that earlier developments reacted to the sensory inputs rather than comprehend their surroundings. Comprehending the surrounding of an AV is the area where most researcher involved in intelligent mobility are working. They are developing algorithms that can understand the context of what they see. To that end, the need to collect adequate quantities of data, so that the algorithms can either generalised to new surroundings or apply the benefits of multimodal ML becomes abundantly clear.

This chapter is organized into the following sections. Section 3.2 presents key Intelligent mobility developments before moving onto a review of Intelligent mobility datasets in Section 3.3. In Section 3.4, we define the problem before reporting on the platform and dataset requirements in Section 3.5. In Section 3.6, we discuss the Autonomous Mobile robotic platform before reporting on sensor data representation in Section 3.7. Finally, we present the results of the datasets developed during this research in Section 3.8, and provide a summary of the work in Section 3.9.

3.2 Key Intelligent Mobility Developments

AVs must be equipped with the ability to process dynamic sensor data so they can react adequately and seamlessly to the adapting environmental changes based on what they see. Although the technologies that facilitate autonomous robots navigate their surroundings in well-structured environments are established, doing so in dynamic, unstructured environments is a problem yet to be solved. For example, the Google driverless car company, Waymo, has been operating AVs without a safety driver and with a license in Phoenix, Arizona, since 2017 [212]. While they have reported much success, Waymo is using pre-loaded high definition maps with centimetre accuracy to assist the vehicles in making decisions [213]. Of course, one could argue that using centimetre accuracy maps to assist in making decisions is not precisely a data-driven solution.

Possibly the most critical events in the design and development of AV were the 2004, 2005 Defence Advanced Research Projects Agency (DARPA) Grand Challenge and the 2007 DARPA Urban Challenge. 2004, 2005 DARPA Grand Challenges were conducted in the Mojave desert, America. The objective of the challenge was to create a fully autonomous all-terrain vehicle within a specific time. Unfortunately, in 2004, there were no winners; however, in 2005, five vehicles completed the task based on the experience they gained from the previous year [214].

In the DARPA Urban Challenge vehicles were required to complete a task in urban settings. This Urban Challenge saw six teams successfully finished the course. Between the two different challenges – the Grand and The Urban – two groups stood out, Stanford University and Carnegie Mellon University. Both groups used a range of different sensors – a GPS, a Velodyne HDL-64 LiDAR, an Alasca LiDAR, a Radar, and multiple High Definition (HD) cameras [214], [215] – to perceive the surrounding information. Of the different sensors used, the most relied upon were the LiDAR, camera, and GPS [216], [217].

Since the success of these teams, the configuration of Radar, LiDAR, camera, and GPS has become commonplace in almost all AVs design. Authors in [218] developed an unmanned shuttle system equipped with 2-dimensional (2D) laser scanners, three cameras, an odometer, and a GPS locator mounted on a repurposed electric buggy. Composed of four modules: a perception module, a navigation module, a Graphical User Interface (GUI), and a system monitoring module. The autonomous platform performed different tasks such as

obstacle detection, road marking detection, localization, and mapping for behaviour and path planning. The platform can generate directional commands to follow a particular path [218].

In [219], the researchers used an estate station wagon retrofitted with four LiDAR, a millimetre-wavelength Radar, and two GPS units. The Radar and LiDAR were fitted to the front of the platform, and a further LiDAR fixed to the centre of the roof. The other two LiDAR and GPS units were fixed to the far side and near side of edges of the roof. The LiDAR was positioned in such a way to detect lane markings, road surfaces, and the position of almost all vehicles in its vicinity. The platform had three modules – a Perception module, a Path planning module, and the Controller module – that process information asynchronously. The primary focus of the different modules is to detect lane markings, mapping and localization, and object detection.

While the DARPA's Grand and Urban Challenges zeroed in on standard components used by AV's, research by Oxford Robotics Institute are using Low-carbon Urban Transport Zone (LUTZ) pods in unstructured environments to set the standard for the control algorithms that drive these systems [220]–[223]. Contrary to the structured environments where research groups like Waymo are testing their vehicles, LUTZ pods are being utilized in unstructured outdoor environments [224]. Although there is a reason to investigate AV use in both, unstructured environments are significantly more demanding as the lack of valuable cue that assists the AV to make decisions. This is most prevalent for robots that assist people in everyday tasks as their operating environment is not only unstructured but also dynamic.

In the future personal assistive robots are going to help the elderly to become independent [225], assist the blind [226], and support rehabilitation [227] of individuals with traumatic injuries. Robots will assist humans in many day-to-day tasks such as personal hygiene, mobility guidance [228], dressing support [229], feeding support [230], and rehabilitation support [227]. The opportunities for assistive robots are numerous, whether they are driving someone to work or collecting samples on Mars. For the most part, HAR is used to support assistive robots and AV in driving driver policy. Knowing what human subjects are doing is a crucial attribute for robots to plan and execute their duties. HAR forms an integral part of Human-Robotic Interaction (HRI) and involves posture analysis, gait analysis [228], and skeletal tracking [229] to determine what activities performed.

FSD is one of the most fundamental challenges for robots that assist humans. FSD involves the safe movement of the robot without colliding into obstacles. It assists in navigation-based decisions. An Intelligent mobile robot that drives safely on its own, without

having accidents is no easy task. Doing the same indoors is even harder. Indoor FSD is more complicated due to a cluttered, continually changing environment as the position of furniture and walls block line-of-sight. As a result, indoor FSD is enormously challenging, especially when training robots to make data-driven decisions [231].

Robotic navigation involves the movement of the autonomous platform while avoiding obstacles. Except for people, vehicles, and other road users, most obstacles are static in outdoor environments. Whereas, obstacles in an indoor environment, keep changing form, appearance, position. They are also frequently blocked by walls. For robust robotic navigation in indoor environments, the robotic platform must be able to see everywhere so that it can detect the relative location of events in another room. Localization is the process by which a robot understands its location relative to its surroundings. In outdoor environments, it is possible to utilize GPS. However, in indoor environments, the GPS signal attenuates. Therefore, it is better to use another method of localization; Radio Frequency (RF) based fingerprinting or Simultaneous Localization and Mapping (SLAM).

RF-based fingerprinting involves storing the RF characteristics of objects at different locations in a database. This database can then be compared to the characteristics of the unknown targets to find its approximate location. In [232], a Wi-Fi-based fingerprinting algorithm was combined with deterministic and probabilistic location estimation for the localization of a moving IoT target. A radio signal-based approach for localization is both low complexity and cost-effective solution [232]. Fingerprint-based positioning technology requires multiple wireless access points (APs) to improve its localization accuracy. To overcome the requirement for multiple wireless APs, the authors in [233] proposed an indoor localization system that uses a single Wi-Fi AP to locate terminals by utilizing Channel State Information (CSI) to compute the direct path length between a single AP and terminals.

SLAM is an alternative method of localization – some regard it as the optimal. SLAM constructs an Occupancy Grid Map (OGMap) and localizes the robot relative to the features on the map. Typically, an OGMap is used to describe occupied space in a discrete grid. Depth information is gathered from a range finder (LiDAR or ultrasound) to construct a discrete map of the environment. Initially introduced in [234], OGMap's have long been regarded as the standard for robotic environment representation. Possibly the most common range finder used for SLAM is LiDAR. Researchers in [235] proposed a method for self- parking AV using a Random Sample Consensus (RANSAC) algorithm and the Extended Kalman Filter (EKF) to evaluate the results. Ibisch *et al.* [235] showed their proposed method returned an OGMap

with an error between 6.4 and 8.3 cm. In [236], researchers presented a method of SLAM using OGMap's to address the issue of frequently changing environments, and demonstrated it with data collected over an extended period. For each data point collected, they measured the error distribution between different data collection periods. The error distribution used to determine the rate of change of the environment was accurate; however, the process is costly on resources.

3.3 Intelligent Mobility Datasets

Data sparsity becomes a significant problem where the information used to drive autonomous robots is particularly limited. Consider the application of DL techniques to Intelligent mobility. If few dominate data access, finding novel approaches to training a Neural Network becomes quite a difficult task [237]. Data sparsity is a critical issue because the information is at the core of any ML algorithm. The larger and more diverse the dataset, the better. When a dataset is inadequate, the performance of the ML algorithm suffers. In fact, data-related issues are the main reason why most ML projects cannot be accomplished [238].

3.3.1 Autonomous Vehicle Datasets

A review of publicly available datasets was undertaken to establish their suitability for training an agent for FSD. Table 1 details the datasets we reviewed during this research. None of these datasets were suitable for unstructured indoor and outdoor environments and in some cases, lacked the modality required to pursue the research objectives detailed in chapter 1. The algorithms that drive this technology are dependent on real-world data for development, testing, and validation. The CamVid Database was one of the first experimentally collected datasets with class labels for visual object analysis, testing, and validation [239]. Captured from the perspective of a driver, the images in the dataset address the need for experimental data to evaluate emerging algorithms quantitatively.

The work in [240] presents one of the most comprehensive datasets collected to date with a Multimodal sensor ensemble attached to an autonomous ground vehicle platform. In addition to optical and depth data capture, the platform was fitted with Inertia Measurement Units (IMU) for position and orientation. The data captured formed a base for SLAM of the respective vehicle. As with [240], the combination of the sensors was ideal for navigation, localization, or mapping. Currently, not that many multimodal autonomous driving datasets have been released into the public domain – some notable ones include [191], [240]–[242]. As of late, the Cityscapes dataset in [243] & [244] have proven to be amongst the most popular.

Typically, these AV datasets focus on the development of; stereo reconstruction as in [245], pedestrian and vehicle detection as in [244]–[246], semantic classification as in [239] and motion estimation as in [247] & [248].

TABLE 1: REVIEWED AV DATASETS

Name	Ref	Permission Environment	Year	Description
AEV Autonomous Driving Dataset	[249]	Licence Structured	2019	2.3 TB of camera, LiDAR sensor data, featuring Forty thousand frames with 2D semantic segmentation, 12000 frames with 3D bounding boxes, and unlabelled 3D Point Clouds. Also, 390000 frames of unlabelled sensor data.
Oxford Radar RobotCar Dataset	[250]	Opensource Structured	2019	The Oxford Radar RobotCar Dataset is a radar release to append the Oxford RobotCar Dataset from 2016. Sensors included a Navtech CTS350-X Millimetre-Wave Frequency Modulated Continuous Wave (FMCW) radar, dual Velodyne HDL-32E LiDAR with optimized ground truth radar odometry. Data was collected around Oxford over 280km.
Brno Urban Dataset	[251]	Licence Structured	2019	The Bruno urban dataset is a dataset recorded in the Czech Republic over 350km using four cameras, two LiDAR's, inertial measurement unit, IR camera, and a differential Global Navigation Satellite System (GNSS) receiver with centimetre accuracy. All data were timestamped with sub-millisecond precision.
A*3D	[252]	Opensource Structured	2019	A*3D dataset is a dataset recorded at different times of the day and night in sunny, cloudy, and rainy weather conditions. 230000 human laded 3D object annotations in 39,179 LiDAR Point Cloud frames with corresponding front-facing RGB images.
Waymo Open Dataset	[253]	Opensource Structured	2019	The Waymo open dataset is a dataset recorded in the USA using a LiDAR and camera sensors. The dataset contains LiDAR and camera data from 1,000 segments collected at 10Hz in diverse scenarios and environmental conditions. The dataset shows four object classes – Vehicles, Pedestrians, Cyclists, Signs, 12M 3D bounding box labels with tracking IDs on LiDAR data, and 1.2M 2D bounding box labels with tracking IDs on camera data.
Lyft Level 5	[254]	Licence Structured	2019	Lyft level 5 dataset is a large-scale dataset recorded by a fleet of multiple, high-end, AV, containing over 55000 humans laded 3D annotated frames. Data was captured using seven cameras and up to 3 LiDAR. A semantic map provides 4000 lane segments (2000 road segment lanes and about 2000 junction lanes), 197 pedestrian crosswalks, 60 stop signs, 54 parking zones, eight-speed bumps, and 11-speed humps.
Argoverse	[255]	Opensource Structured	2019	Argoverse dataset is a dataset recorded in Pittsburgh and Miami using LiDAR and camera sensors. Split into three releases; the first contains data from 113 scenes with 3D tracking annotations on all objects. The second release is a dataset of 300,000-plus scenarios. The third release is a set of HD maps of several neighbourhoods.
Berkeley Deep Drive	[256]	Opensource Structured	2018	Berkeley Deep Drive dataset is a dataset recorded in the USA using camera and GPS sensors. The dataset contains 100000 HD video – each running 40 seconds long at 30 fps – sequences over 1100-hour of driving across many different times of day, weather conditions, and driving scenarios.
ApolloScape	[257]	Opensource Structured	2019	ApolloScape dataset is a dataset recorded in China using camera and LiDAR sensors with pixel-by-pixel annotations, including 26 different recognizable objects – cars, bicycles, pedestrians, and buildings. The dataset offers numerous levels of complexity recorded in challenging environments, weather, and extreme lighting conditions.
nuScense	[258]	Licence Structured	2019	The nuScense dataset is a dataset recorded using LiDAR, Radar, camera, and GPS. nuScense consists of 1000 scenes containing 1.4 million camera images, 390000 LiDAR sweeps, 1.4 million Radar sweeps. Containing 23 different classes or 1.4 million objects annotated by hand and showed, nuScense was collected in Boston and Singapore.
Oxford RobotCar Dataset	[259]	Opensource Structured	2016	The Oxford RobotCar Dataset is a dataset recorded in oxford covering a fixed path, using LiDAR, camera and GPS sensor. Data was captured over one year, covering a variety of Weather and traffic conditions showing road users, along with longer-term changes to the environment.
KITTI	[190]	Opensource Structured	2013	The KITTI dataset is a dataset recorded in Germany using LiDAR, camera, Radar, and GPS sensors. The KITTI dataset is regarded as a benchmark dataset upon which a lot of the proceeding datasets were based on. The dataset contains 6 hours of diverse traffic scenarios recorded at 10-100 Hz covering autobahn, rural roads, and inner-city scenes.

One of the most recent developments in AV datasets is the nuScense dataset. Released in March 2019, nuScense consists of 1000 scenes containing 1.4 million camera images, 390000 LiDAR sweeps, 1.4 million Radar sweeps. It contains 23 different classes or 1.4 million objects annotated by hand. nuScense was developed using data captured Boston and Singapore. Primarily focusing on object detection, tracking, and segmentation of agents in outdoor environments, nuScense covers both right- and left-handed driving scenarios [258].

The majority of these autonomous driving datasets are primarily addressing the challenges of (a) scene understanding, (b) localisation/mapping, and (c) object detection. Mostly they rely on sensors such as LiDAR, Radar, and camera. However, some datasets – such as the BerkleyDeepDrive dataset – primarily focus on camera GPS and IMU data. While some elements of these datasets are useful, the lack of modality is a major shortcoming when pursuing Multimodal ML methods such as online active Learning. Furthermore, the fact that some datasets rely so heavily on GPS means they cannot be applied to indoor environments.

While some of the other datasets reviewed provide IMU data such as speed and direction of travel, the majority do not. Usually, the gathered data is optimized to detect objects and people in traffic – unfortunately, a lot of these datasets are designed for a single aspect of AV research. Consequently, most research projects do not tackle the many difficulties experienced during extended periods of autonomy: chiefly, localization under various environmental conditions as in [247], [248], mapping over time to see how scenes change as in [260], [261], and diverse object recognition using fused sensor data as in [262].

A benchmark dataset should cover many real-world scenarios both indoors and out, over an extended period, focus on all elements of the research – object detection, FSD, HAR – and annotation of the objects identified in the sensor data. Primarily a dataset should account for the three different pillars of which an AV is composed, and ill-disciplined road users such as pedestrians.

3.3.2 Human Activity Recognition Datasets

While there are many approaches to getting the AV's moving, few have researched the interaction with their surroundings. Works that are more closely related to Multimodal ML for HAR with applications to AV strive to decrease road accidents by recognizing pedestrian activities. Researchers in [263] and [264] provide an interesting application of HAR for a pedestrian recognition system that matches the predicted intention with that of a driver's direction. Action prediction is 'before the fact event' and supersedes recognition. Referred to

as the Human Action Prediction (HAP), where ML algorithms recognize a class from an incomplete or changing action [265]–[267]. pQuite different from activity recognition, where ML algorithms expect to see the entire set of action dynamics.

TABLE 2: REVIEWED HAR DATASETS

Name	Ref	Permission Environment	Year	Description
OA Dataset	[268]	Opensource Office	2015	The OA dataset covers the regular daily activities taken place in an office, and it is the largest activity dataset of RGB-D videos, which includes 20 classes performed by 10 subjects.
UTD-MHAD Dataset	[269]	Opensource Studio	2015	The UTD-MHAD dataset consists of four temporally synchronized data modalities. The modalities include RGB videos, depth videos, skeleton positions, and inertial signals from a Kinect camera and a wearable inertial sensor logging 27 classes performed by 8 subjects.
UWA3D Dataset	[270]	Opensource Studio	2016	The UWA3D dataset is a multiview activity dataset which contains 30 actions performed by 10 subjects.
NTU RGB+D	[271]	Opensource Office	2016	The NTU RGB+D dataset consists of 56 thousand video samples and 4 million frames, collected from 40 distinct subjects. The NTU RGB+D dataset contains 60 different action classes from typical daily, mutual, and health-related actions.
Wearable Computer Vision Systems dataset	[272]	Opensource Unstructured	2014	The wearable computer vision system dataset that includes trajectories of different users across two indoor environments performing a set of more than 20 different activities captured using wearable RGB and RGB-D sensors
Ajou University HAR dataset	[273]	Proprietary Office	2013	The Ajou University HAR dataset was acquired using a 3-axis accelerometer and a single camera worn on a body of subjects performing 8 activities. Quantity of data and subjects performing activities were not reported.
MIT Media Lab HAR dataset	[274]	Proprietary Unstructured	2015	The MIT Media Lab HAR dataset was acquired using the accelerometer, camera, gyroscope in google glasses to interpolate pulse and respiratory rate of 12 human subjects performing 6 activities.
MVPA dataset	[275]	Proprietary Unstructured	2013	The MVPA dataset was acquired using a hip-mounted accelerometer and a wearable camera. 49 Subjects engaged in 12 activities recorded over 3 days.
MultiTHUMOS	[276]	Proprietary Unstructured	2017	The MultiTHUMOS dataset was acquired using stereo video camera of an unknown number of subjects performing 65 different activities.
The Breakfast dataset	[277]	Opensource Unstructured	2014	The breakfast dataset includes a total of 52 participants, each performing a total of 10 cooking activities in multiple real-life kitchens, resulting in over 77 hours of video footage using stereo camera
The Okutama Action dataset	[278]	Opensource Unstructured	2017	The Okutama-Action dataset was captured using stereo camera from an aerial view. It consists of 43 minute-long fully annotated sequences with 12 action classes.
LboroLdn HAR dataset	[279]	Opensource Office	2019	The LboroLdn HAR dataset was captured using RGB-D, LiDAR, 360° camera sensors. 9 subjects performing 16 activities.

A review of publicly available datasets was undertaken to establish their suitability for training a Multimodal ML agent for HAR. Table 2 provides an overview of the reviewed datasets found during this research. Broadly speaking datasets can be classified into four different categories: (a) RGB stereographic image, (b) RGB-Depth image, (c) Biometric information recorded by wearable sensors (e.g., accelerometers) and (d) Multimodal data captured using LiDAR, RGB, RGB-Depth.

Except for the LboroLdn HAR dataset [279], all the HAR Datasets reviewed during this research either lacked multimodality or the use of LiDAR. Considering this and the limitations of reviewed HAR datasets, we broadened our field and reviewed Multimodal AV Datasets. In the context of Multimodal AV Datasets, the work in [240], [279] presents a

comprehensive Multimodal dataset collected using LiDAR and a range of cameras tagged to an AV. Datasets in [240], [242], [280], and [190] fit the criteria of Multimodality with focus on camera and LiDAR sensor data. Although the combination of the sensors was found to be ideal for navigation, localization, or mapping, the environment under which the data was captured limited its applications to an AV and rendered it difficult to use in HAR.

This research found that for HAR, AV datasets fit the Multimodal criteria but are restricted by the environment. Furthermore, when compared to the HAR datasets, AV datasets provide a lower density of information about pedestrian actions. This decreases the accuracy of human silhouette detection, which results in misclassification. Moreover, all the AV datasets reviewed were collected with vehicles in outdoor environments. During this research, we did not find a comprehensive dataset that covers both indoor and outdoor environments and fits the criteria of multimodality with camera and LiDAR sensor data streams. While some datasets utilised wearable sensors, such as the ones embedded in watches or phones, it was felt that using sensors ubiquitous to AVs would be most relevant.

3.4 Problem Definition

The autonomy of transport in real-life settings is a significant challenge that needs to be overcome [281]. Autonomous platforms used in transport, for the most part, utilize similar sensors – LiDAR, Radar, ultrasound, camera, and GPS [214], [215] [216], [217]. Typically, LiDAR is used to map the surrounding environment in 3-Dimensions (3D), identify free space, and measure distances in midfield ranges [116]. Radar is used for long-range sensing, while ultrasound sensors are useful at very short ranges. Imaging sensors are used to detect road surfaces, street furniture surrounding pedestrians, and vehicles [176].

Moreover, since these systems need to function seamlessly in both indoor and outdoor environments, complex social interactions are going to compound the problems further. AV and assisted living robots are a rapidly evolving element of Intelligent mobility. It was not so long ago that such technologies were considered science fiction. In such lively times, it is easy to jump the gun and fall favour to misconceptions about the technology, the potential, and the process used to develop these systems.

In the following sections of this Chapter, I propose an open-source experimental framework for data gathering, sharing, and experimental validation of driverless vehicle technology. The objective of the proposed platform is to demonstrate a solution to some of the core challenges discussed above, develop an open-source experimental framework for data

gathering, sharing, and experimental validation of driverless vehicle technology. I aim to enable researchers all over the world to utilize different test data, and to provide a unified interface to execute control algorithms on a prototype. Towards this end, I have developed a driverless platform equipped with several sensors and real-time control through a high-performance computer to derive data-driven driver policy.

3.5 Technical Parameters of an Autonomous Platform

A scalable, multi-layer context mapping and recognition system are depicted in Figure 19. The architecture has four layers: The Sensing Layer, the Data Analysis Layer, a Multi-layered Context Representation, and the Application Layer. The Sensing Layer is primarily concerned with gathering and presenting different types of information to the data analysis layer. The Data Analysis Layer consists of data pre-processing, data fusion, object detection, FSD, and HAR. Classifications made in the Data Analysis Layer are passed onto the Context Representation layer to be called by the different applications as needed. Finally, the Application Layer communicates with the Context Representation Layer to acquire location-dependent context information.

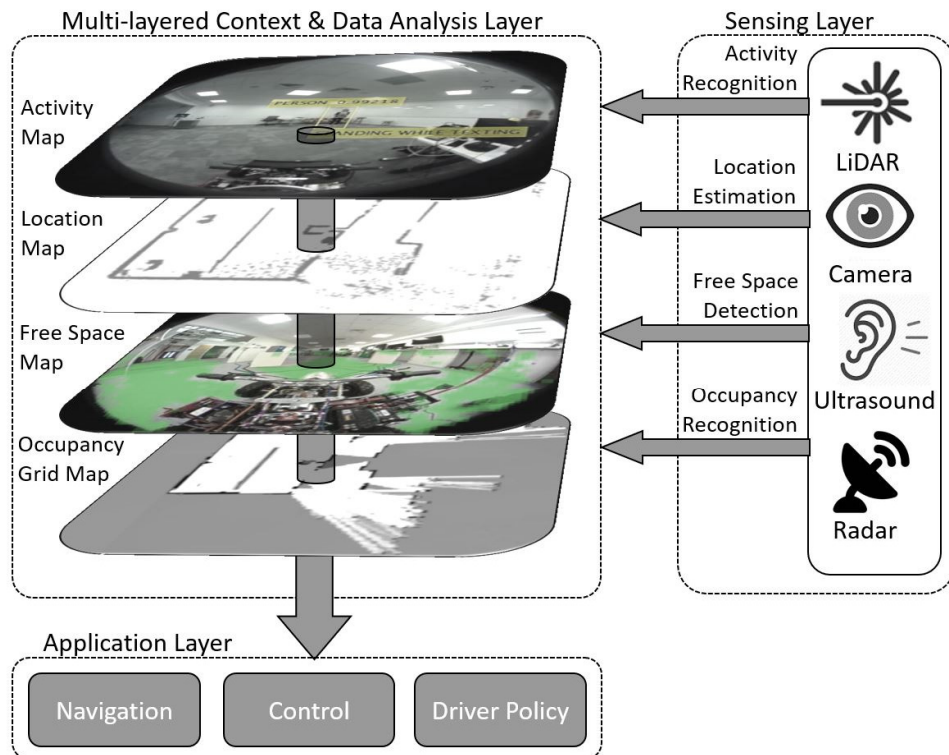


Figure 19: Shows the scalable, multi-layer context mapping and recognition system for the autonomous platform. They are depicting the sensing layer, a multi-layered context representation, the data analysis layer, and the application layer.

To cater for the requirement of scalability, the Context Representation Layer organizes the context information, such as a subject carrying a box, into different categories according to its complexity, location, resolution, and steadiness. A requirement for scalability is that different applications require different levels of context information. Furthermore, scalability can be realized as temporal and spatial resolution based on the uncertainty of observations. The purpose of the autonomous platform is to navigate at low speeds in unstructured environments, both indoor and outdoor. The autonomous platform moved at low speeds to adhere to the licence requirements and prevent an accident at speed from occurring. The autonomous platform has been developed as an AV for the collection of specific Multimodal data to facilitate the development of context-based algorithms such as FSD and HAR. The autonomous platform can roam without the need for human intervention. However, should it be required, an operator can take control.

3.5.1 Platform & Dataset Requirements

The inherent complexities of the built environment prevent AV from being hard programmed with a fixed set of rules that foresee all possible scenarios they might encounter [282]. Therefore, mobile robots need to learn to make decisions autonomously based on the events they encounter and the objects they perceive. Only in this manner can an adequate policy be derived – self-evolving over time depending on objects the agent's encounter.

Unfortunately, each type of sensor has its limitations. For example, LiDAR sensor readings are often affected by the weather, such as rain, fog, or snow [283]. Furthermore, the resolution of a typical LiDAR sensor is quite limited when compared to a camera. For example, at 300 RPM, the azimuth resolution of the VLP-16 LiDAR is 1°. By comparison, a typical high definition camera has a resolution of 1920 x 1080 pixel and is therefore much more densely packed on the horizontal axis. Similarly, stereo camera-based depth estimation is limited by its baseline distance [284]. For example, a shorter baseline would allow only a short-range depth estimation of objects. While a more extended baseline would increase the range of the system to measure more considerable distances, it is not always possible when considering the physical setup on a specific platform [285]. Therefore, given that each sensor has its limitations [283], the diversity offered by multimodality can only offer a positive contribution to the ability of any machine to perceive [286].

To realize a self-evolving AV, one sensor modality cannot capture all the context. Therefore, multiple sensors connected to a central processing system must be a prerequisite of any dataset. Secondly, the data captured by different sensors would be of little use if they are

not analysed to capture various parameters of context, such as FSD and HAR. Therefore, advanced Multimodal ML algorithms are necessary to make sense of the fused sensor data streams.

The most common sensors amongst all the prototypes that can be used in both indoor and outdoor environments are LiDAR and the camera. Both camera and LiDAR are used to map the surrounding environment, identify free space, and measure distances in the near and midfield ranges [116]. Radar is used for long-range sensing, while ultrasound sensors are reliable at very short ranges. Imaging sensors are used to detect road surfaces, street furniture surrounding pedestrians, and vehicles [176].

Restrictions concerning available datasets, and datasets that are suitable for our multimodal ML methods, are severely lacking. Coupled with this and the necessity to adhere to the three pillars described in section 2.5, we can define the platform requirements as

1. A small safe AV that operates in pedestrianized areas that can collect realistic Multimodal data from a set of sensors frequently utilized in an AV for both indoor and outdoor environments.
2. An ability to collect data from realistic test scenarios that are representative of changing environments that a vehicle operates in. One of the primary challenges faced by autonomous agents is safe operation in a changing environment. The autonomous platform needs to be able to collect data to address the issue of dynamically changing scenarios.
3. An application interface displaying telemetry data about the autonomous platform so that different parties can replicate the test scenarios and benchmark the performance of novel ML algorithms. This element will facilitate the dissemination of collected datasets and ML algorithms amongst the research community.

In terms of data variety, we can add an additional prerequisite to help us further identify requirements:

4. A wide variety of traversable surfaces, objects and activities being performed by people. Data should be collected in both indoor and outdoor environments accounting for weather conditions that influence the sensor data gathered by AV. The traversable surfaces, objects,

activities, and influencing factors should be recorded using multiple sensor types over an extended period.

It should be noted that while the second and fourth requirements are quite similar, they were defined to address separate elements of the data collection process. The second requirement was defined to address elements relating to the autonomous platform, whereas the fourth requirement was defined to address elements relating to the dataset we collected. The system architecture and datasets discussed in the following section were developed to address these requirements.

3.6 An Autonomous Mobile Robotic Platform

The proposed framework of the autonomous platform was composed of stackable layers: The Sensing Layer, The Data Analysis Layer, a Multi-layered Context Representation, and The Application Layer. Since the Data Analysis Layer and the Multi-layered Context Representation focuses on the data-driven algorithm development, they are reported on in chapters 4 and 5. The autonomous platform was developed in mind of the four requirements listed above.

The mobile robot platform utilizes seven different sensors. These sensors – part of the Sensing Layer – include cameras, RGB-D, ultrasound sensors, LiDAR, and Radar. The location-dependent context, processed on the Multi-layered Context & Data Analysis layer, provides information about the location of human subjects, human activities, free space, and obstacles. The sensor data, gathered by the Sensing Layer, is utilized by the robot and provides the system with information to make navigation decisions, on the Application Layer.

The platform chassis is a repurposed Rebo LT100E Electric Quad Bike. The LT100E Quad Bike features an adjustable three-speed 1 kW 36-volt brushless motor. The Quad Bike is powered by a rechargeable battery and can reach a top speed of 22 km/h. The height, length, and width of the platform are 0.992 meters, 1.02 meters, and 0.64 meters, respectively. The front overhang was 0.15 meters, wheelbase 0.71 meters, and the rear overhang 0.16 meters. The front and rear track width are approximate 0.5 meters, and the platform has a turning circle of 3.5 meters. Figure 20 shows the Loughborough University London Autonomous Platform indicating the location of the proximity and optical sensors.

Figure 21 shows the system framework for the proposed platform. Currently, the sensor data is logged on a Secure Digital (SD) card and a Micro-Star International (MSI)

Apache Pro, running windows 10 with an Intel Core i7, 16Gb Ram, a Nvidia GeForce graphics card, and a 1 TB solid-state hard drive.



Figure 20: Loughborough University London autonomous platform indicating the location of the range and optical sensors used to collect the LboroLdn AV and LboroLdnHAR datasets. The Kinect sensor, missing from this image, is positioned on the handlebar.

A cable connects the right and left front callipers to the right brake lever. When depressed, the callipers mounted over the disk close, thus stopping both the right and left front wheels from rotating. The rear brake, mounted under the right rear fairing, is controlled by a lever mounted on the left-hand side of the handlebar. Callipers connected to the rear suspension arm, mounted over the disc, are connected to the rear axle. When depressed, this stops the bike. As with the front brakes, a switch mounted in the lever housing cuts power to the motor adding a load to the rear axle before retarding the bike to a stationary position.

There is a three-speed lockable governor positioned below the left rear fairing, and a throttle mounted on the right-hand side of the handlebar. A chain drive turns a sprocket fixed to the rear axle and propels the bike at speed, depending on the position of the throttle. A switch mounted immediately to the left of the throttle determines the bike's direction of travel – forward/reverse. Switching direction while the bike is moving means power is cut from the motor. When this occurs, the motor acts as a load on the rear axle retarding the bike to a

stationary position. Once the throttle moves position, power returns to the motor, and the bike moves in the opposite direction.

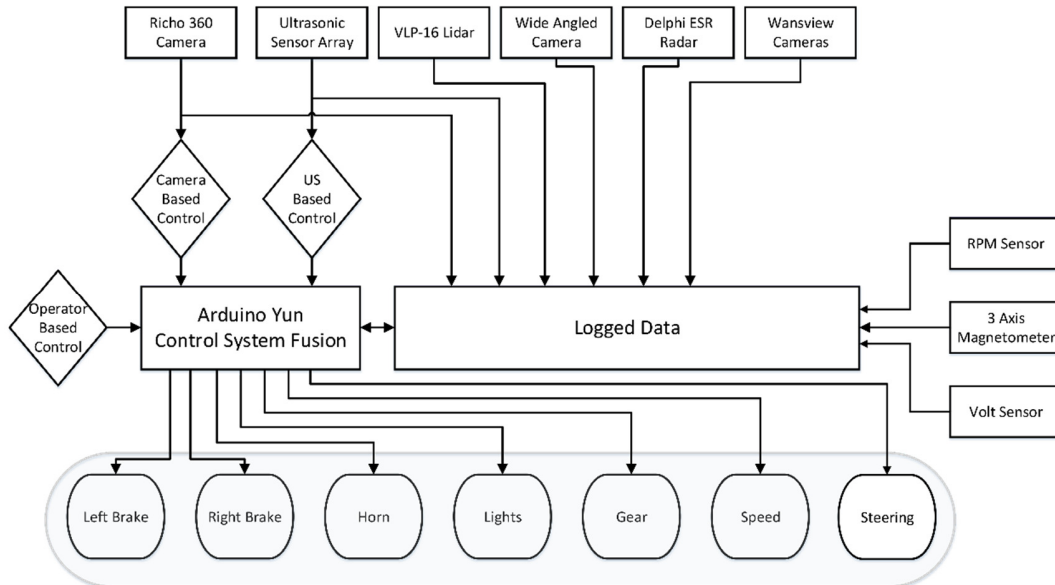


Figure 21: Proposed framework for the platform showing the controllers, actuators, depth, optical, and telemeter sensors. All-optical and proximity data was logged to the laptop. Telemetry data was timestamped and logged to an SD card.

3.6.1 The Sensing Layer

Ultrasound data is collected on a single plane (2D) with a Horizontal Field of View (HFoV) of 90° with a max range of 5 meters. The ultrasonic array supports multiple channel data streams and takes a total of 6 measurements per second utilizing six emitter/detector pairs and a spatial resolution of 20° . A monopod fixed to the Platform chassis at the centreline suspends the 360Fly Monocular camera, VLP-16 LiDAR, and the Ricoh Theta V 360° camera.

The VLP-16 is composed of 16 emitter detector pairs taking a total of 300,000 measurements per second. At a scanning rate of 5 Hz, the spatial resolution is 0.1° , captured over 360° on the horizontal axis. On the vertical, the spatial resolution is 2° captured over 30° . Conversely, the Delphi electronic scanning radar (ESR) has a max range of 175 meters, a 10° Vertical Field of View (VFoV), a 20° HFoV, and a spatial resolution or 0.5°

The 360Fly camera has a wide-angle fisheye lens with a Field of View (FoV) of 240° , and the Theta V 360° camera has two fisheye lenses, front and rear-facing, with a total field of view of 360° . Except for the two Wansview Internet Protocol (IP) cameras, the image data captured on the 360Fly and the Theta V 360° cameras, was done so using a wide-angle lens, with a spatial resolution of 1920×1080 and 3840×1920 , respectively. Cameras with a wide-angle lens were chosen to increase the depth of field, thus increasing the chances of getting the

entire scene into the frame. Because the autonomous platform was designed to operate both indoors and outdoors, space manipulation was crucial to gathering valuable context information. Therefore, using a camera with a standard prime, zoom, macro, or telephoto lens, would fail to capture context over a wide range – and were therefore not used.

Furthermore, to assure that data was gathered in areas deemed vital (i.e. towards the front of the platform), context gathering cameras were positioned to overlap their field of view. Contrary to this, the Wansview IP cameras had a standard prime lens. However, since these cameras are primarily concerned with gathering data over which the autonomous platform has previously travelled, they do not need the wide-angled context information, as they are concerned with traversable surfaces.

TABLE 3: PROXIMITY SENSORS SUMMARY

Sensor	Dimension	Range	HFoV	VFoV	Spatial Resolution	Scanning Rate
Ultrasonic Array	2D	0-5m	90°	30°	20°	6Hz
VLP-16 LiDAR	3D	3.5-100m	360°	30°	0.1°	5Hz
Delphi ESR	2D	0-175m	20°	10°	0.5°	5Hz

TABLE 4: OPTICAL SENSORS SUMMARY

Sensor	Resolution	FPS	FoV	Lens Baseline
360Fly Wide-angled camera	1504x1504	29.9	240°	30cm
Ricoh Theta V 360° camera	1920x1080	29.9	360°	10cm
Wansview IP camera (x 2)	480x360	29.9	95°	4cm

TABLE 5: TELEMETRY SENSORS SUMMARY

Sensor	Operation Voltage	Operational Current
Revolutions Per Minute (RPM) Optical Coupling Sensor	3.5-5V	15mA
HMC5883L 3 Axis Magnetometer Sensor	3.5-5V	15mA
STM32 Voltage Sensor	3.5-5V	15mA

Except for the two Wansview IP cameras, all perception sensors were mounted along the centreline of the platform. Speed, power, and orientation are determined using an Infra-Red (IR) break-beam rotary encoder, a voltage divider, and a 3 Axis Magnetometer, respectively. A break-beam sensor identified when a beam between the receiver and the emitter was broken. For example, when the disk brake on the rear axle passes in between the emitter and the transmitter, an impulse is counted. Knowing the number of gaps in the encoder disk and the time it takes to complete one full rotation, the platform speed can be determined.

The direction of travel is determined using the 3 Axis Magnetometer. The RPM sensor, 3 Axis Magnetometer, and Voltage divider are fixed inside the chassis structure.

Table 3 summaries the proximity perception sensors indicating the Dimensions, Range Scanning Rate, HFoV, VFoV. Table 4 summaries the optical sensors indicating the Resolution, Lens baseline, Frames Per Sec (FPS), and the FoV. Table 5 summaries the telemetry sensors, indicating the Operational Voltage and Current.

3.6.2 The Application Layer

Inherent complexities and the unpredictability of road users make programming the response of an AV a problematic, if not impossible task. Consequently, the AV needs to be able to make decisions autonomously, depending on the situation at hand. To arrive at this point, ML algorithms require diverse and sufficient quantities of training data to make a classification in varying environments. The importance of developing a dataset and then benchmarking it against Perception, Localization, and Driver Policy is abundantly necessary.

Open source and publicly evaluated datasets would allow researchers to compare algorithms objectively. This combination of obstacles is essential to benchmark systems for full 24/7 operation in all environments. Therefore, setting a well-defined challenge in conjunction with realistic well-balanced sensor data would be a valuable contribution to this field of research.

Towards the centre front of the test platform lies the steering assembly. The steering assembly ties the handlebars to the right and the left tie rod. The tie rods direct the platform using two servo motors. The right and left tie rod act as a lever pushing and pulling the right and left swing arm joint, thereby directing the wheels. Figure 22 shows an isometric view of the steering assembly.

An Arduino Uno controls two servo motors and receives commands through the Arduino Yun. The commands come in the form of Pulse Width Modulation (PWM) signals. Before being processed by the Uno, the signal passes through a low pass filter. The low pass filter smooths the discreet waveform into an analogue equivalent that represents the linear position of the Digital to Analog Conversion (DAC) dial on the user interface.

The Wi-Fi chipset on the Yun and the functions contained in the Bridge library allows the Yun's microprocessor to behave as a Linux server hosting a website. A web-enabled device displaying the user interface connects the quad bike to the gateway in a star configuration.

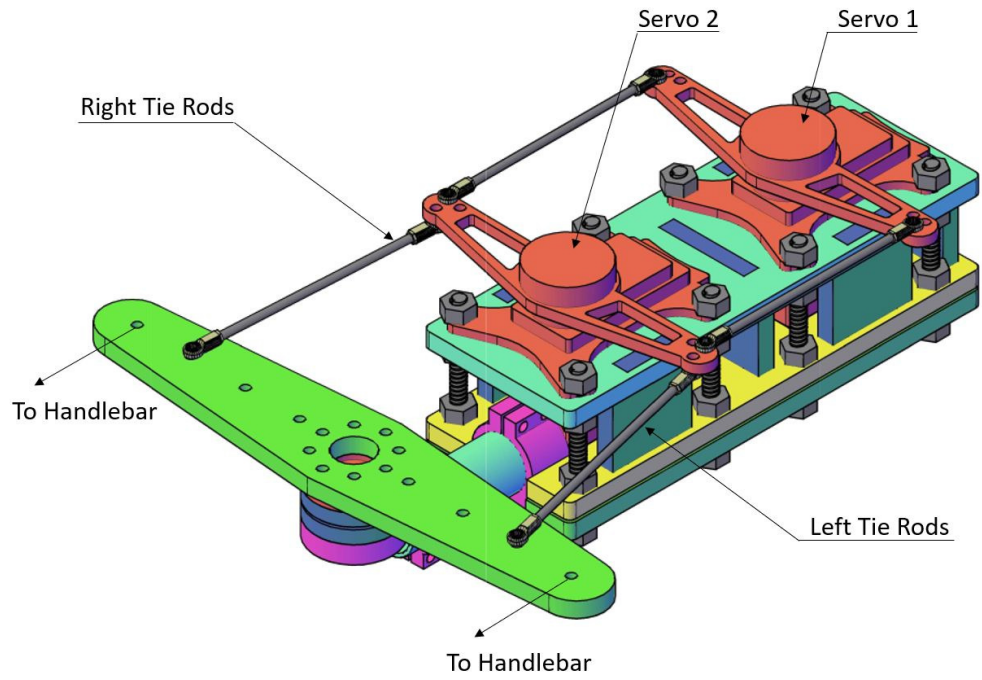


Figure 22: Steering assembly showing both servos motors the right and left tie rods linking to the handlebars. The link to the handlebars is through an additional two tie rods. Both servo motors are driven from the same signal – PWM – from the Arduino.

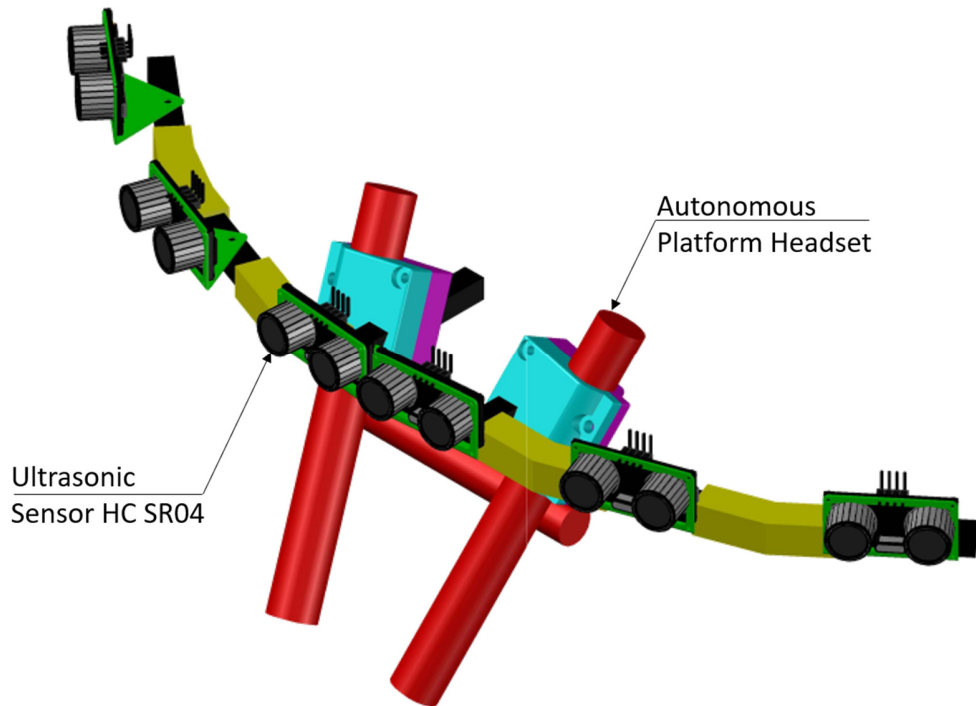


Figure 23: Shows the ultrasonic sensor array assembly, detailing the six HC-SR04 ultrasonic sensors and a section of the autonomous platform headset. The HC-SR04 were at an angel of 5°, 25°, and 45° either side of the longitudinal axis.

The autonomous platform can roam freely. It collects data autonomously with the possibility for human intervention should it be required. While human intervention is not desired, it was a prerequisite of the license and permit needed to perform the experiments. Furthermore, during the design stage of the autonomous platform, there was some discussion about the possibility of implementing reinforcement learning. While reinforcement learning is beyond the scope of this research, it is regarded by some as an essential contribution to Intelligent mobility. For this reason, the operator-based control was given the same priority as the camera and ultrasound-based control.

The current policy that drives the platform is rudimentary at best. Decisions are hard programmed based on proximity of objects relative to the ultrasonic sensor array, the colour that the 360° camera perceives, and the operator governing the movement of the platform. Figure 23 depicts the ultrasonic sensor array of six HC-SR04 ultrasonic sensors. The primary objective of the autonomous platform is to avoid colliding with obstacles. When the platform is overwhelmed, it does nothing until its path is cleared, all proximity and optical sensors indicate “Full Stop.”

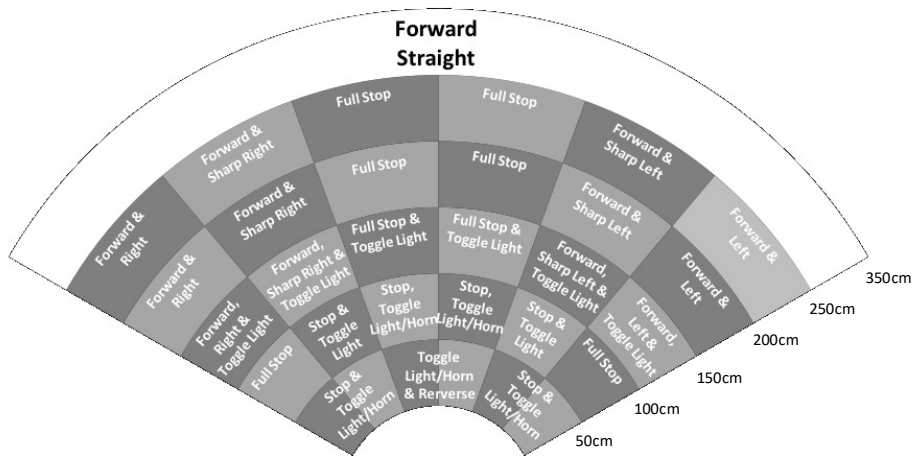


Figure 24: Ultrasound collision avoidance dictates navigation and driver policy for the autonomous platform for data collection. Decisions are hard programmed based on the proximity of objects relative to the ultrasonic sensor array.

A. Ultrasound-Based Control

A collision avoidance policy is implemented depending on the proximity of objects relative to the platform. Figure 24 shows the conditions and the reaction that the platform makes depending on the proximity of the obstacle. The ultrasonic sensor array consists of 6 HC-SR04 sensors positioned at 5°, 25°, and 45° either side of the longitudinal axis. Figure 25 depicts the ultrasound-based control architecture showing an insert of the collision avoidance

policy. Objects within the range are logged and acted upon, influencing the driver policy of the platform.

Upon initialization, the ultrasonic sensor array establishes a connection with the Arduino Yun. A timestamp and distance measurement from each HC-SR04 sensor is logged to an SD card. Depending on the range and the conditions in the collision avoidance policy, the response of the platform is determined – depicted in Figure 23. When two or more obstacles are detected in the path of the platform, the quad bike pauses until the path clears. This element of the control system can govern all elements of the platform – speed, gears (Forward/Reverse), brakes, lights, horn, and the direction it travels in.

B. Camera-Based Control

The second element governing driver policy is colour recognition. When the platform encounters the colour red, the quad bike stops and waits for further commands from the operator or until the colour changes to green. Upon initialization, the laptop establishes a connection with the Arduino Yun, loads a frame, and determines the Hue Saturation and Variance (HSV) value of the first-pixel patch – with dimensions 8×8 . Repeated for each pixel patch until a new image is loaded, all the pixels patches are checked for the corresponding colour. This element of the control system can govern the AV brakes and speed. Figure 26 shows the camera-based control architecture indicating the communication channels between the laptop and the Yun.

C. Operator-Based Control

An operator always accompanies the platform to oversee the safety of operation, override the collision avoidance policy and colour recognition governance. An in-house designed web app relays telemetry information and provides a control platform to the user. When the web page opens, a connection between the user and platform is established. The interface page can be accessed from any location with an internet connection. The current state of the platform is updated on the interface page, indicating the speed, direction, and logical state of the brakes, lights, horn, and gear (forward/reverse). As with the ultrasonic based control, this element of the control system can govern all elements of the platform. Figure 27 depicts the operator-based control architecture indicating the input from the operator and an insert showing the user interface. As a safety feature and the futureproof for reinforcement learning, the operator-based control algorithm can override the ultrasound and camera-based control algorithms.

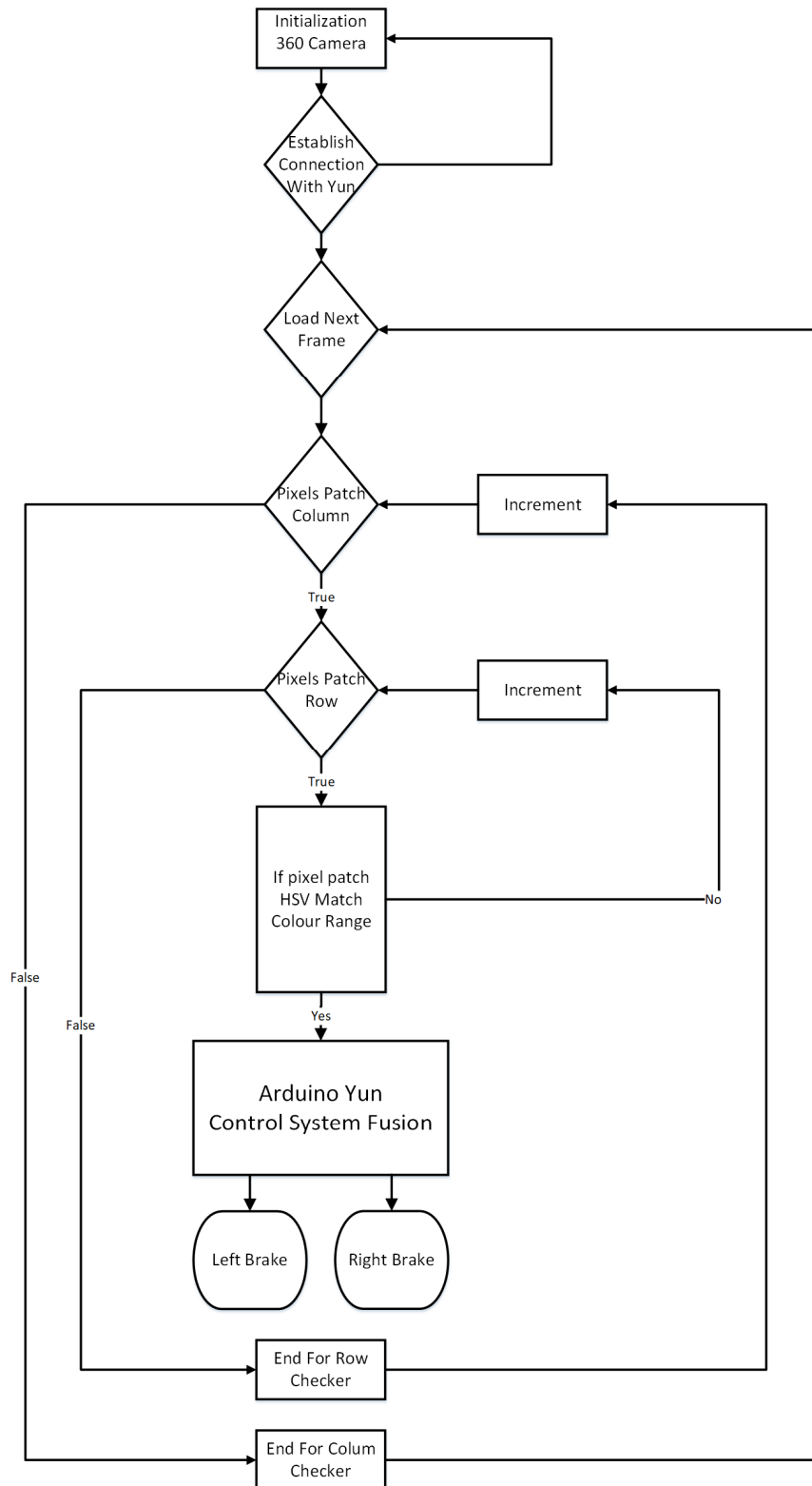


Figure 26: The camera-based control architecture for colour detection. Currently, the platform understands the colour of red green and amber.

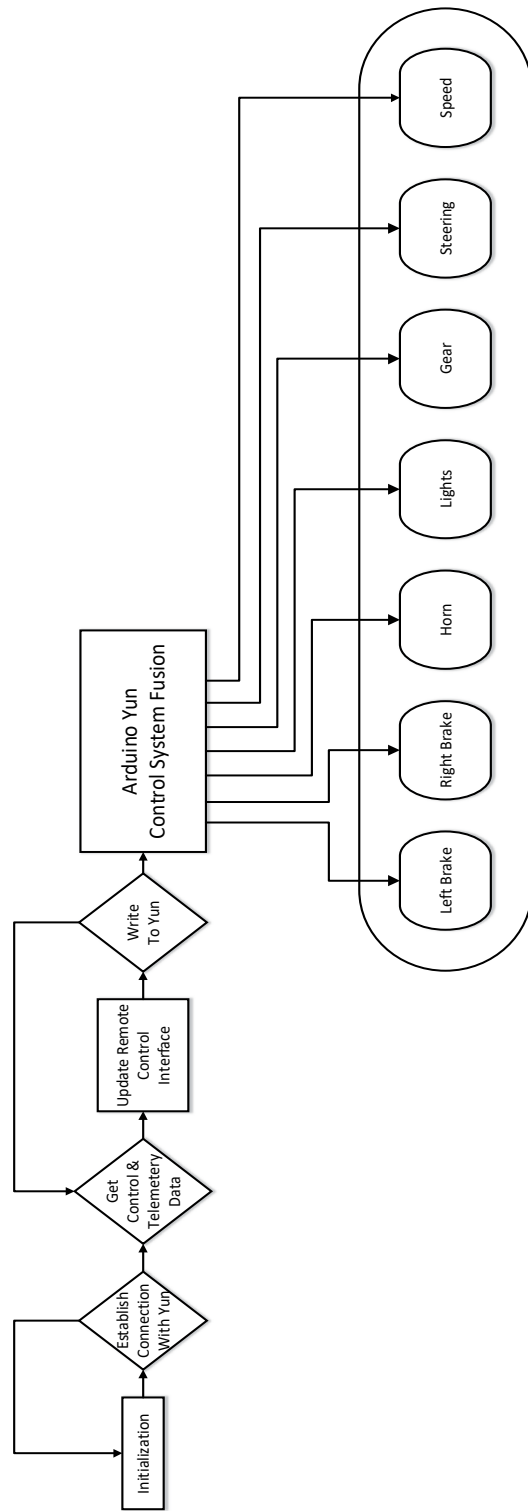


Figure 27: The operator-based control architecture for manual override. The user is the only element of driver policy that exerts control over the collision avoidance policy – in the ultrasound-based controller – and the colour governance in the camera-based controller.

3.6.3 Experimental Setup

The sensors discussed in this thesis are commonly used in the research for assisted or autonomous driving [191], [240]. To ensure reproducibility of our work, Figure 28 provides the metric dimensions of sensor positions relative to the front axle of the testbed. The location of each sensor is measured relative to the ground and the front axle of the platform. Except for the Wansview cameras, all sensors were positioned along the centreline of the platform.

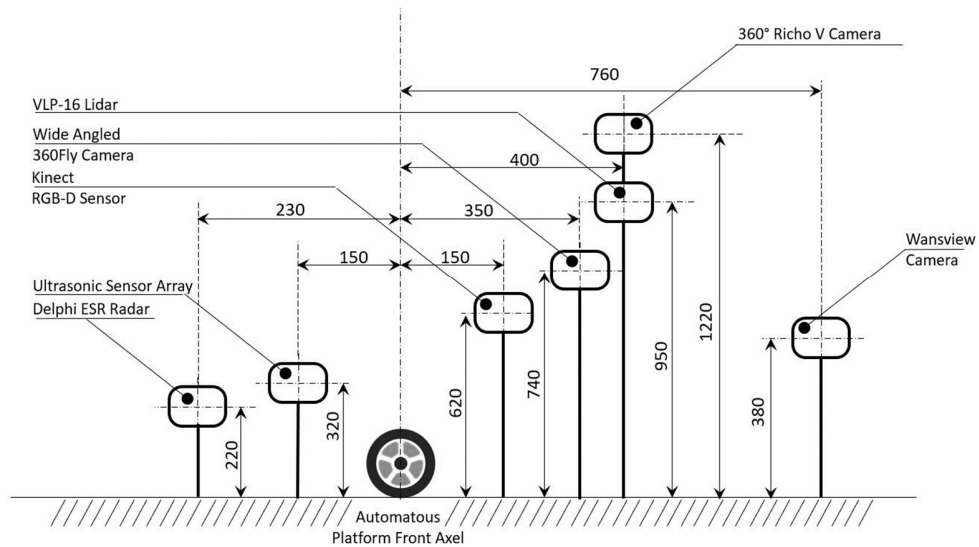


Figure 28: Platform setup indicating the location of the sensors relative to the front axle. The location of each sensor is measured relative to the ground and the front axle of the platform. Except for the Wansview cameras, all sensors were on the centreline.

The location of the Wansview cameras was 76 cm behind the front axle and 38 cm above the ground. Both Wansview cameras were offset from the centreline by 15 cm. The 360° Ricoh V and LiDAR were positioned 40 cm behind the front axle, 120 cm and 95 cm above the ground, respectively. The 360Fly Wide-Angled camera was positioned 35 cm behind the front axle and 90 cm above the ground. The ultrasonic sensor array was positioned 15 cm ahead of the front axle and 32 cm above the ground. Moreover, the Delphi ESR was positioned 23 cm forwards of the front axle and 22 cm above the ground.

To evaluate the applicability of ML algorithms for specific purposes, it is critical to guarantee an experimental setup that allows optimal parameter input for subsequent tests [287]. In other words, we need to provide conditions where each sensor reaches its optimal performance by reducing the impact of sensor-specific limitations. This premise led to the experimental setup FoV shown in Figure 29.

The insufficient detection area towards the rear of the testbed covers a range of 3.5 meters and 270°. Due to the vertical position of the LiDAR, the VFoV is limited to 15° either

side of the longitudinal axis. Consequently, the LiDAR sensor cannot perceive the traversable surface or small objects in the circular area immediately around the autonomous platform. The ultrasonic sensor array compensates for this towards the front of the autonomous platform.

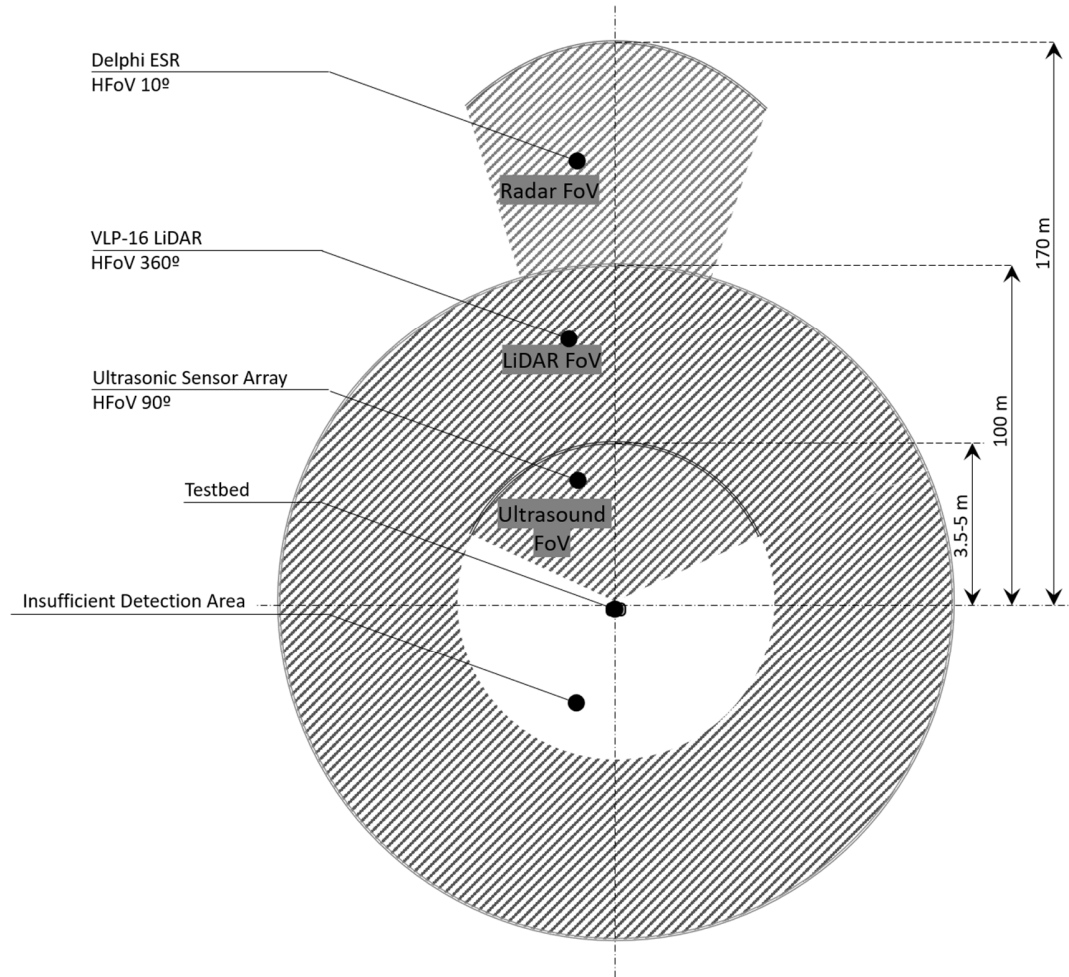


Figure 29: Platform setup indicating the horizontal FoV of the radar, the LiDAR, and the ultrasonic sensor array. The LiDAR provides mid-field depth data, the radar provides far-field depth data, and the ultrasonic sensor array provides near field depth data.

Put simply, LiDAR provides mid-field depth data, the Delphi ESR provides far-field depth data, and the ultrasonic sensor array provides near field depth data. Horizontal limitations of 360Fly camera is negligible in this setup, as image data captured by the 360° Ricoh V camera compensates for the restricted FoV at the rear of the testbed. The Wansview IP cameras have a somewhat restricted FoV due to their position. However, since they are primarily concerned with gathering data over which the autonomous platform has previously travelled – the limited resolution can be disregarded.

3.7 Sensor Data Representations

The term input device denotes that the sensor is only part of a bigger system that controls the response of a device like a microcontroller or a processor. Fundamentally, a sensor converts a signal from one domain to another that can be easily measured [288]. For example, a Light Dependent Resistor (LDR) converts the intensity of the light to resistance. The higher the light intensity, the higher the resistance. Generally, sensors can be broken down into two main groups – active and passive. Active sensors require power to operate. Passive sensors do not require any power. While both types can measure the same thing, they do so in a very different way. Further divisions can be made based on the medium there measuring; Electric, Biological, Chemical, Radioactive, etc. or on the conversion phenomenon they generate; Photoelectric, Thermoelectric, Electrochemical, Electromagnetic, Thermo-optic, etc. The output of the sensor can be either Digital or Analog. Sensors that produce an analogue output generate a continuous signal that changes with respect to the quantity being measured. Analogue signals are not prone to quantization errors and match the response more carefully. Digital sensors, on the other hand, work with discrete values that represent the analogue event there measuring.

The sensors used to capture the datasets can be grouped. The first group is IMU data gathered by telemetry sensors. The second group is the optical sensors, and the third group is the proximity sensors. While the IMU data gathered by the telemetry sensors, is valuable information, it was not used in this research. The sensor data used during this research was the wide-angle camera data, the LiDAR data, the Kinect data and the ultrasonic sensor array data. All the sensors reported on in this thesis can be classed as mechanical-electrical input devices. They measure a change in the event they are monitoring and provides an output. The input or event being monitored generates an output signal concerning a specific physical quantity being measured. For example, RGB and ultrasound data record information in quite different ways. cameras use reflected light, and ultrasound uses reflected sound. Furthermore, the data is structured very differently, RGB in a $m \times n \times 3$ matrix and ultrasound in a $x \times y$ array. Regardless of the difference in structure or the principle of operation, when pointing in the same direction, both sensors capture the same thing.

3.7.1 RGB Data

RGB camera images – often referred to as true colour images – are composed of an m -by- n -by-3 matrix. Each of the three layers of the matrix denotes a different component, red,

green, and blue. Conversely, the m element of the matrix denotes the row, and the n element denotes the column where the colour is stored. When an image is being captured, the light reflected off the object is passed onto a sensor at the back of the lens. Generally, there are two main types of camera sensors, Complementary Metal Oxide Semiconductor (CMOS) or Charge-Coupled Device (CCD). While CMOS is more common today, the principle of operation entails the capturing of photons that hit the sensor and converts them to an electrical signal – like an LDR.

The electrical signal that each element of the sensor produces is stitched together to form an m -by- n -by-3 matrix. The colour of each pixel, denoted by red, green, and blue intensities, is stored at those coordinates and represents the location and colour of specific points in the image. The number of rows and columns denotes the aspect ratio or size of the image. The values stored at the individual cells of the matrix range between 0 and 1. For example, a pixel at the (m_1, n_1) coordinates whose colour components are $(0,0,0)$ would display in the top left corner as black. Alternatively, a pixel at the (m_i, n_j) coordinates, whose colour components are $(1,1,1)$, would display white in the bottom right corner. Where i and j are the maximum value of the aspect ratio, and incidentally represent the max ranges of the FoV.

3.7.2 RGB-D Data

There are similarities between RGB and RGB-D images. With RGB images, the data is structured as a $m \times n \times 3$ matrix, where each layer of the matrix represents a Red, Green, and a Blue element of the colour. The location coordinate of any pixel can be identified by the row (m), and column (n) of the matrix and the colour of that pixel is denoted by different RGB layers. By adding a new layer to the structure, the image can be described as an RGB-D or $m \times n \times 3 + D$, where m is the row, n is the column, and D is the depth element of the pixel. In turn, the pixel represented by the row, column, and the depth has a colour assigned to it. It sounds complicated, so it is easier to view it as a 3D coordinate with an assigned RGB colour.

The RGB-D data is captured using a Microsoft Kinect V2 sensor. The Microsoft Kinect v2 consists of an RGB camera, an IR camera, an IR projector, and a multi-array microphone. With an angular FoV of 62° horizontally and 48.6° vertically, the RGB camera can provide the image with the resolution of 640×480 pixels at 30 Hz (optionally 1280×1024 pixels at 10 Hz) [289]. The depth sensor (IR camera and IR projector), provides depth images with nearly parallel configurations (640×480) pixels at 30 Hz; angular field: 58.5° horizontally, 46.6° vertically [290].

3.7.3 Point Cloud Data

Point Cloud data is a representation of a collection of 3D coordinates in space. It is generally captured by a 3D or 2D scanner, LiDAR, or specialist cameras. The data structure is different in many ways to the structure of the RGB camera sensor. The RGB data is structured and ordered, making it ideal for classification tasks using a CNN. Unlike RGB data, Point Cloud or 3D data is unstructured and without order [291]. For example, the CCD or CMOS sensor at the heart of the camera captures reflected light from a subject. The specific location on the sensor that the reflected light strikes is recorded along with the true colour value of the reflected light. This adds structure and order to the data that represents an image. Conversely, point cloud data captures the location of an object in 3D space. There is no sequence in which the ranges are recorded, and the number of points may vary from scan to scan. Furthermore, while the instruments used to measure the points in space vary in resolution, they all return either Spherical, Cylindrical or Cartesian coordinates of the objects they are measuring, hence the unstructured and unordered description.

The data captured by the VLP-16 LiDAR are x y and z values with millimetre accuracy. The VLP-16 is a low powered compact optical sensor with a useable range of up to 100m. The VLP-16 utilizes 16 emitter detector pairs measuring a total of 300,000 data points per second. Data is captured as coordinates over 360° on the horizontal axis and 15° either side of the origin on the vertical axis [292]. While the captured data points are not recorded in sequential order when plotted as a whole, they reproduce a workable representation of the object the sensor detected.

3.7.4 Ultrasonic Depth Data

While the ultrasonic sensor principle of operation differs significantly to the camera, the fundamentals behind both are primarily the same. Ultrasonic sensors measure distance by using the time it takes for a soundwave to reflect off an object inside the FoV. The soundwaves used to measure the distance are propagated from the sensor to the objects through an elastic medium such as air. The typical range of operation of an ultrasonic sensor is between 40 kHz to 50MHz. The speed at which the sound travels through the elastic constant is dependent on temperature and relative humidity. Ultrasonic sensors can also be used for relative density measurements, discontinuities in metals, composites, plastics, ceramics, and for water level detection.

To capture the range, the ultrasonic sensor transmits a short burst to a target. In turn, the target reflects the soundwave to the sensor. The time it takes for the echo to return to the sensor is used to calculate the distance using the speed of sound for the medium, air, as it travels through. Ultrasonic sensors are composed of two main components – a transmitter and receiver pair. The transmitter emits a short 40KHz burst before the receiver captures the reflected sound. In this case, it is fixed to hear only 40KHz frequencies. Onboard processing calculates the time of flight for the emitted signal and converts it to a range.

The ultrasonic sensor array used in this research was developed in-house and consists of 6 transmitter and receiver pairs. The transmitter and receiver pair fire sequentially from right to left at intervals 0.02 seconds and measures ranges in cm. They take a total of 36 data points or six full scans of the area towards the front of the platform every second. Figure 22 shows the assembled ultrasonic sensor array; sensors are positioned at 5° 15° and 25° of the longitudinal axes can accurately measure ranges with centimetre accuracy. Object ranges are captured by individual sensors. Distances are timestamped and logged as a string, indicating the proximity of objects within the 5-meter of the autonomous platform.

3.8 Data Collection

There are many decisions to be made before collecting a dataset – sensor modalities, event preparation, and the labelling protocol. The researcher's choice was guided by the requirements identified in Section 3.5.1. Data logging and the method for labelling the captured data was designed to record the high variability of indoor and outdoor scenes. For the LboroLdnAV dataset, frames were captured from the platform when it was moving, over three months. For the LboroLdnHAR dataset, we appended a Kinect sensor to the autonomous platform. Since the Kinect sensor was designed to maintain in a stationary position, frames were captured by a stationary platform. Except for a Kinect sensor appended to the autonomous platform, the sensor setup remained the same.

The sensor location and orientation described in the experimental setup was scrutinized before each data capture period. Sensor location and orientation are integral to the geometric alignment of data, so special attention was taken to make sure the sensors did not move during data collection. Frame rate sequencing for the individual sensors was explicitly chosen to facilitate the maximum resolution of individual sensors and to prevent the ghosting of captured objects. Ghosting occurs when the frame rate of a sensor falls too low, and a moving subject appears twice in a single frame of data captured.

3.8.1 Loughborough Autonomous Vehicle Dataset

As of Monday 16th December 2019, the LboroLdnAV dataset consists of 45.6 hours of Video, LiDAR, and ultrasound data collected over 1.2 km of indoor and outdoor environments under a variety of scenarios. Data collection is ongoing and expected to conclude during the summer of 2021. This will assist in the development of Multimodal ML algorithms for use by autonomous robots.

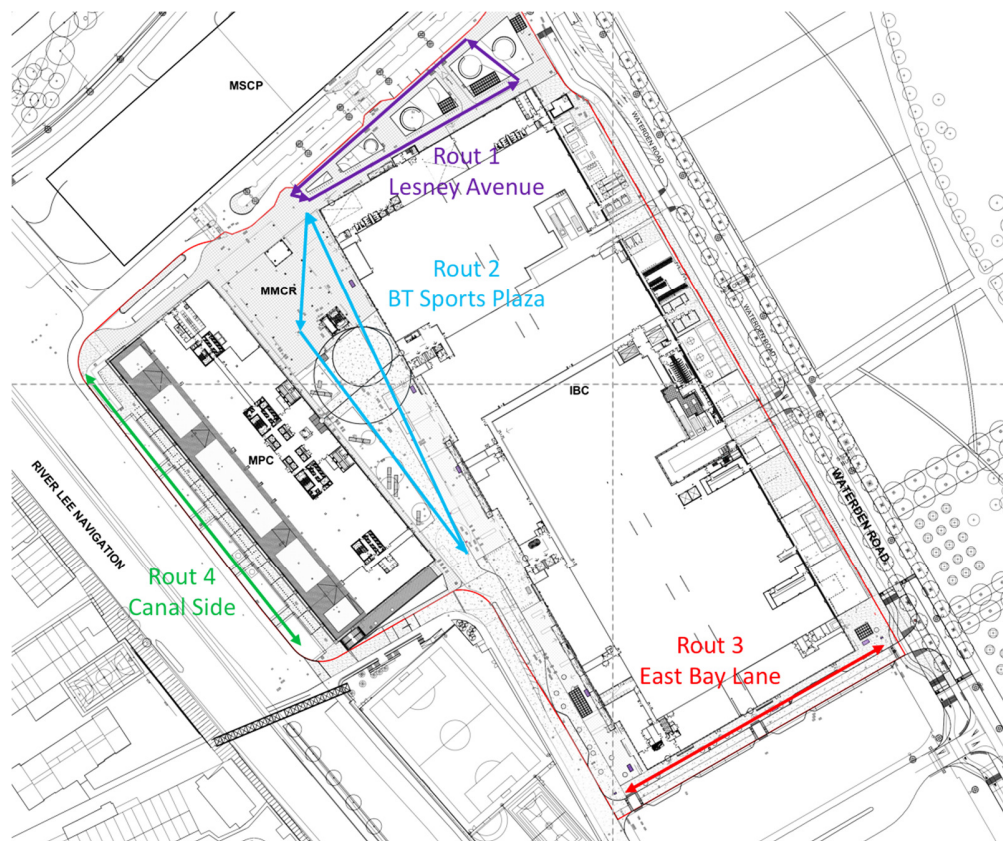


Figure 30: The primary data collection routes were traversed between 28th May 2018 and to 1st October 2018, and 1st November 2019 to 30th January 2020. The traversable distance was a total of 1.2 km over four different locations, Route 1 Lesney Avenue, Route 2 BT

The primary data collection route is shown in Figure 30. Not all sensors were in use during the data collection period. The data release reported in this Chapter is only the first part of a more extensive project that is currently (16th December 2019) in the process of expanding into Sri Lanka. As of the 16th December 2019, data was collected on the Here East campus in Queen Elizabeth Olympic Park, London. A permit and license were granted by Here East to collect data between 28th May 2018 to 18th August 2018 and 1st November 2019 to 30th January 2020. Although the data collection is over an extended period, it was not possible to collect data every day – due to restrictions imposed by the management company.

In total, there were 2.5 million frames captured by four cameras; 672k frames were captured by the 360Fly Wide-angled camera; 1.2 million frames were captured by the Ricoh Theta V 360° camera, and 624k frames were captured by the two Wansview IP cameras. Both the LiDAR and ultrasonic sensor array captured a total of 252k and 220k scans, respectively. Due in part to the resolution and the frequency of operation and the fact that some sensors were not in operation, there is a disparity between different quantities of data collected across sensors (i.e., Delphi ESR).

To help with everyday ML tasks, 7 object classes were annotated - People; Bus; Van; Car; Motorbike; Cyclist; and Traversable Surfaces. Each with accurate bounding boxes or polygons at 5Hz. While four cameras were used during the data collection period, only one of the data streams were used for the FSD process, the Ricoh Theta V 360° camera. Data was annotated by hand, and a ground truth label for the seven classes is appended to the dataset.

Different dataset requirements mean different quantities of data are required for the algorithms proposed in Chapters 4 and 5. All data captured and used during algorithm development was chosen based on the diversity of foreground and background objects, and the overall scene layout.

While output data from the RPM sensor, 3 Axis Magnetometer, and Voltage divider was not used in this research, it is equally valuable. The data is very low-level and predominantly used in applications such as SLAM. The LboroLdnAV dataset contains data of everyday objects encountered and captured by the seven sensors. This release of the dataset focuses on indoor and outdoor environments with a focus on traversable surfaces.

The acquisition period of the LboroLdnAV dataset spanned 28th May 2018 to 1st October 2018. The dataset covers 1.2 km of recorded driving in Queen Elizabeth Olympic Park, London. The autonomous platform was autonomously driven throughout the data collection period, with a minimal degree of human interaction. Sensors used during this research were the 360Fly wide-angled camera, the Ricoh 360° camera, two Wansview IP cameras, VLP-16 LiDAR scanner, and the ultrasonic sensor array.

Figure 31 presents a montage of images illustrating the diverse range of buildings and short-term lighting changes encountered by the testbed. Table 6 details the date, location, classes captured, and data streams of the LboroLdnAV dataset, while Table 7 lists summary statistics for the dataset so far.



Figure 31: Montage of six images taken on 28th May 2018, illustrating the diverse range of buildings and short-term lighting changes encountered by the platform.

Data collection periods were chosen to cover a wide range of classes, Pedestrian, Cyclist, and Vehicle Traffic, under many different environmental conditions. The dataset appended to this research is a partial release, and the weather was mostly sunny throughout its collection.

TABLE 6: DESCRIPTION OF THE LBORO L D N A V DATASET

Title	LBORO Dataset
Summary	Data captured by seven sensors
Date	28/05/2018 to /01/10/2018 and 01/11/2019 to 30 th January 2020t
Location	East Bay Lane, Lesney Avenue, BT Sports Plaza, Canal-Side
Total Size	23.75 hours over 1.2 km
Class	People, Bus, Van, Car, Motorbike, Cyclist, Traversable Surfaces

TABLE 7: SUMMARY STATISTICS FOR THE LBORO L D N A V DATASET

Sensor	Type	Size
360Fly Wide-angled camera	Image	2.34 GB
Ricoh Theta V 360° camera	Image	4.05 GB
Wansview IP camera (x 2)	Image	0.24 GB
HC-SR04 Ultrasonic Array	2D Scan	5.6 MB
VLP-16 LiDAR	3D Scan	12.8 GB
Delphi ESR	2D Scan	N/A

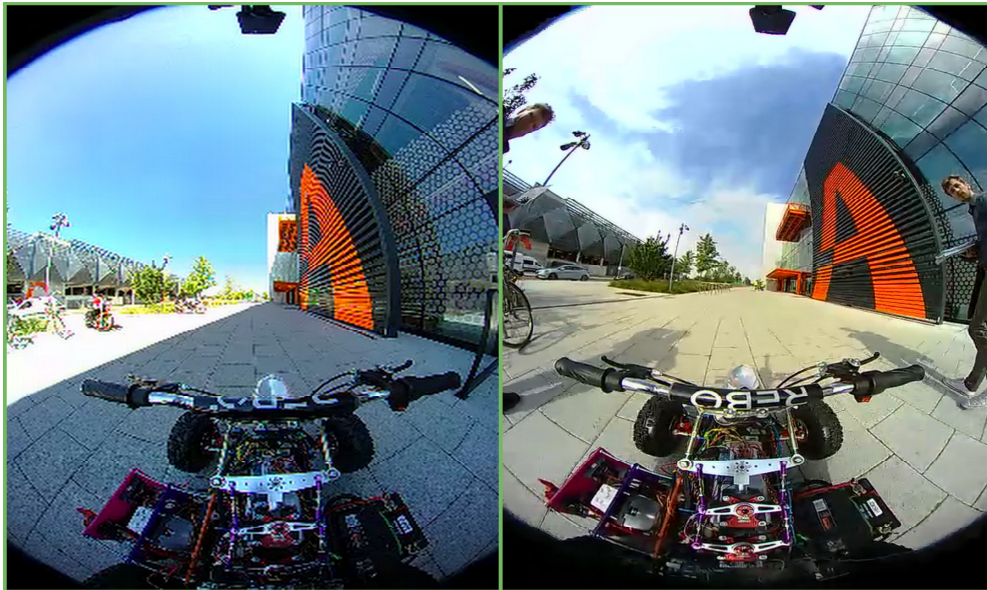


Figure 32: Illustrates two different views from a single location under different environmental condition. Note environmental conditions are limited for this release of the dataset as most of the data was logged over the summer months.

Figure 32 illustrates the two views from a single location under different environmental conditions. Figure 33 presents the environmental condition and the number of traversal days for different journeys. Due to events on the Here East campus and conditions dictated to the platform by the driver policy, it was not possible to retrace the exact route every time.

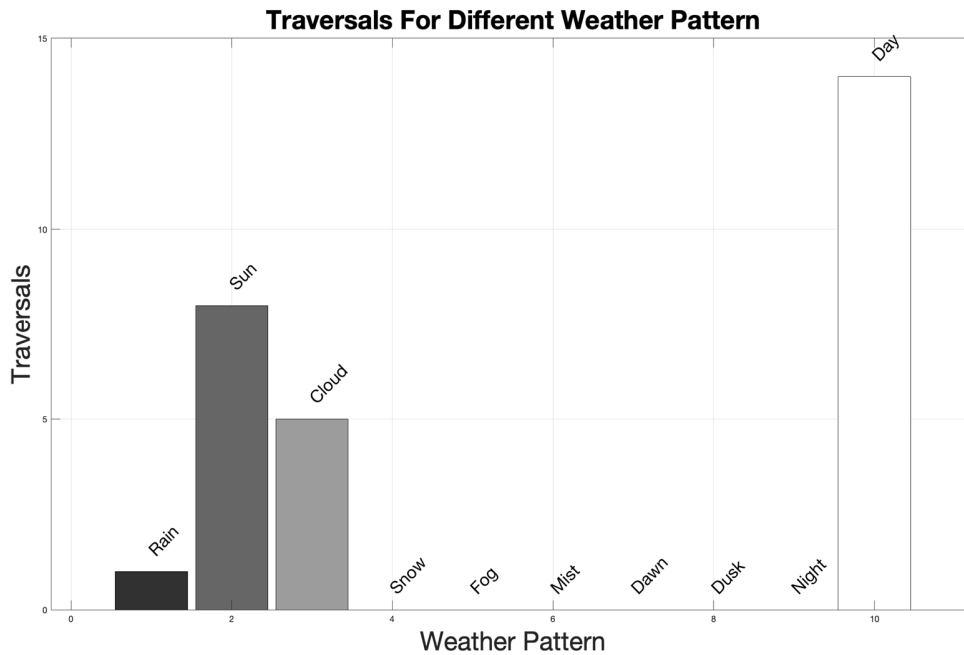


Figure 33: Traversals for different environmental conditions for the LboroLdnAV dataset. Factors influencing the route, include events on campus and driver policy. The license and permits were granted for three months between 28th May and 1st October 2018.

Sensors lenses and instruments were adequately maintained to prevent the build-up of contaminants. A label for each traversal was added to the captured data. Traversals were grouped by labels for an accessible collection of routes. Care was taken to ensure that all data was carefully archived, named commented, and stored securely as per data protection acts.

Conditions of the License and Permit granted by the management company restricted the speed at which the autonomous platform could operate – less than 4kph. While the autonomous platform can operate at speeds of up to 22kph, there would be little point in doing so since the frequency of operation of the LiDAR, and ultrasonic sensor array would return inaccurate measurements – ghosting and cross talk.

Ghosting is a replica of a recorded image, offset in position, that occurs when the subject being measured is detected twice in the data because of the frequency of operation. Of course, it is possible to increase the frequency of operation of the individual sensors. However, in the case of LiDAR, the resolution would significantly reduce, and in the case of the ultrasound, it would result in crosstalk. Cross talk between ultrasonic sensors occurs when the transmitted signal of one sensor is picked up by the detector of another. Somewhat like ghosting, it interferes with ranges measured by the sensor and produces a false distance measurement.

3.8.2 Loughborough Human Activity Recognition Dataset

The LboroLdnHAR dataset contains samples of typical human activities in indoor environments, captured by three sensors – camera, Kinect, and LiDAR. The LboroLdnHAR dataset is composed of 6712 RGB, RGB-D, and Point Cloud samples, annotated and classified depending on the activity.

Each sample contains one of the 16 subjects performing one of the nine activities. Activities were chosen based on a review of related work in Section 3.3.2. The focus was on indoor activities, where subjects had a limited attention span. The activities were sitting on a chair, standing and texting, sitting on a stool, lying on the couch, walking, walking and texting, carrying objects, pulling objects, and running. These activities were chosen because of the subject's lack of attention to their surroundings – it is highly likely that humans do not pay attention to an indoor AV. The activities selected helped facilitate the later development of driving policies for indoor AV in critical scenarios.

The LboroLdnHAR dataset was split into two subsets. The first subset contained RGB samples and consisted of images with annotated ground truth Region of Interest (ROI) labels,

indicating the location of people in the frame. This subset was used to train the object detector. The second subset was the activity annotated point data. This subset was used to train the point cloud classifier.

TABLE 8: DESCRIPTION OF THE LBOROLDNHAR DATASET

Title	LboroLdnHAR
Date	17/06/2018; 18/06/2018
Size	16 participants x 9 activities x 3 Sensors (RGB-D; LiDAR; 360° camera)
Activities / Classes	Carrying Boxes, Lying Down, Pushing A Board, Running, Sitting on a Chair, Sitting on a Stool, Standing While Texting, Walking, Walking While Texting
Contained Data Streams	360° camera Stream (each file contains a Scenario, ca. 2 min); LiDAR Stream (each file contains a Scenario, ca. 2 min); RGB Stream (captured by RGB-D sensor); Depth Map (captured by RGB-D sensor); Point Cloud of moving objects (captured by RGB-D sensor); body joint model (extracted from RGB-D sensor)
Joints Kinematics generated by iPi Mocap Studio [293]	coordinates (51 frames); velocity (50 frames); acceleration (49 frames)
Body Joints	Head; Neck; Chest; Middle Spine; Lower Spine; Hip; Centre of mass; Centre of mass projection to the ground; Left-Hand and; R Eye; Effector Head; R Clavicle; R Shoulder; R Forearm; Right-Hand; L Clavicle; L Shoulder; L Forehand; L Hand; R Thigh; R Shin; R Foot; R Toe; Effector R Toe; L Thigh; L Shin; L Foot; L Toe; Effector L Toe

TABLE 9: SUMMARY STATISTICS FOR THE LBOROLDNHAR DATASET

Sensor	Type	Size
Kinect V2	RGB-D	0.5 GB
Ricoh Theta V 360° camera	Image	0.9 GB
VLP-16 LiDAR	3D Scan	0.6 GB

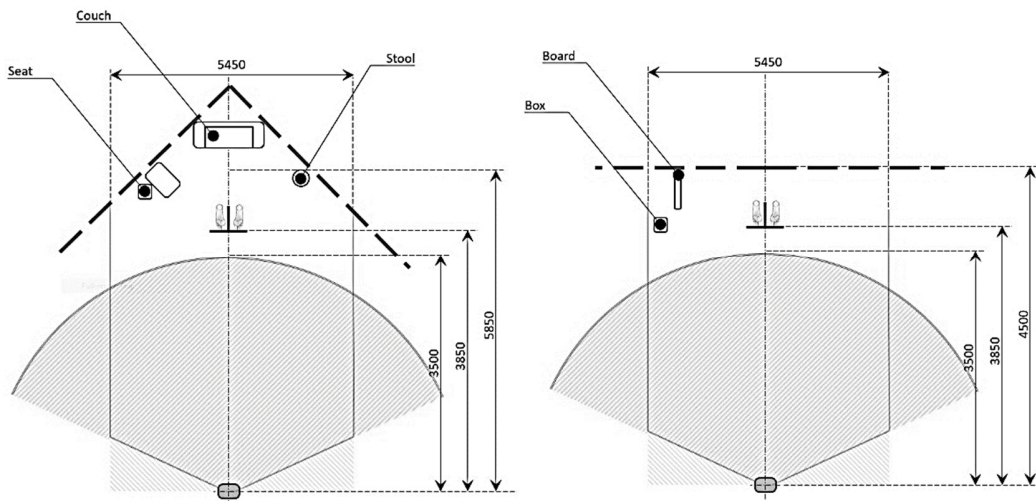


Figure 34: The layout of furniture and the position of the autonomous platform during the collection of the LboroLdnHAR dataset. The data was collected throughout the 17th and 18th August 2018. In both cases, the subjects start and finish the activity with a T pose.

The LboroLdnHAR dataset consists of 6712 LiDAR, RGB-D, and RGB aligned and transformed samples – 5,916 for training, 787 for validation, and 9 purely for visual presentation of this research. The dataset was divided in this manner to reduce the chances of

overfitting when training the network but still to retain sufficient quantity (~10%) for validation. Table 8 details the date, classes captured, and data streams of the LboroLdnHAR dataset, while Table 9 details the dataset statistics.

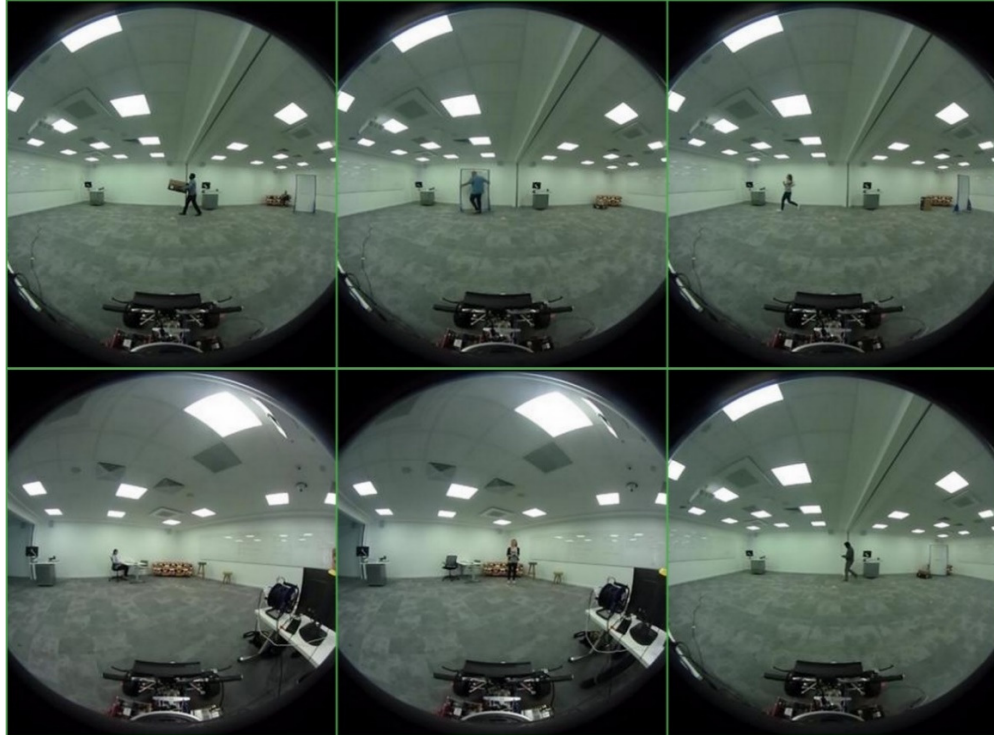


Figure 35: Montage of 6 images taken on 17th and 18th August 2018. The images illustrate six of the different activities performed during the data collection period. Carrying a box, pushing a board, running, sitting at a desk, standing, and walking while texting.

To evaluate the general applicability of ML algorithms, it is critical to guarantee an experimental setup that allows replication for subsequent tests [287]. Conditions, where each sensor reaches its optimal performance level, needs to be identified. Doing so will reduce the impact of sensor-specific limitations, such as minimum operative distance, to a negligible amount. This premise led to our experimental setup depicted in Figure 28 and 29.

Sensors used during this portion of the research were the Ricoh 360° camera, VLP-16 LiDAR scanner, and the Kinect RGB-D camera. All data was captured indoors under fixed lighting conditions. The activities were sitting on a chair, standing and texting, sitting on a stool, lying on the couch, walking, walking and texting, carrying objects, pulling objects, and running. The focus of this dataset was to capture humans performing typical everyday activities. Figure 34 shows the layout of the different scenarios when collecting the LboroLdnHAR dataset. Figure 35 shows a montage of six images collected over two days taken on 17th and 18th August 2018. While data was being collected, the participants were advised to pay little heed to the AV used for capturing the data.

3.9 Summary

Autonomous platforms used in transport, for the most part, utilize similar sensors – LiDAR, Radar, ultrasound, camera, and GPS. When robots that assist us in our daily tasks are expected to operate indoors, the combination LiDAR, Radar, ultrasound, camera, and GPS needs to be reconsidered. To further compound the problem, each type of sensor – used for perception – has its limitations. Therefore, given that each sensor has its limitations, the diversity offered by multimodality can only offer a positive contribution to the perceived ability of any machine.

Typically, AVs are composed of three technological pillars: Sensing and perception, Localization and mapping and Driver Policy. In this chapter, we present a scalable, multi-layer context mapping and recognition system based on the three pillars listed above. The architecture has four layers: The Sensing Layer, The Data Analysis Layer, a Multi-layered Context Representation, and The Application Layer. The Sensing Layer is primarily concerned with gathering and presenting different types of information to the data analysis layer. The Data Analysis Layer consists of data pre-processing, data fusion, object detection, FSD, and HAR. Classifications made in the Data Analysis Layer is passed onto the Context Representation layer to be called by the different applications as needed. Finally, The Application Layer communicates with the Context Representation Layer to acquire location-dependent context information. It should be noted that the Multi-layered Context Representation and the Data Analysis Layer are discussed in Chapter 4 and 5, respectively.

Currently, the autonomous platform roams and collects data with the possibility for human intervention should it be required. While human intervention is not desired, it was a prerequisite of the license and permit needed to perform the experiments. The policy that drives the platform is rudimentary at most. Decisions are hard programmed based on proximity of objects relative to the ultrasonic sensor array, a colour red that the 360° camera perceives, and the operator governing the movement of the platform.

As of the 16th December 2019, the LboroLdnAV dataset consists of 45.6 hours of Video, LiDAR, and ultrasound data collected over 1.2 km, displaying a variety of scenarios from both indoor and outdoor environments. The collection of data is an ongoing project to assist in the development of Multimodal ML algorithms for use by autonomous robots. In total, there were 2.5 million frames captured by four cameras – 672k frames captured by the 360Fly Wide-angled camera, 1.2 million frames captured by the Ricoh Theta V 360° camera, and

624k frames captured by the two Wansview IP cameras. Both the LiDAR and ultrasonic sensor array captured a total of 252k and 220k scans, respectively. The disparity between different quantities of data is due to resolution, frequency of operation and the fact that some sensors were not in operation.

The LboroLdnHAR dataset consists of 6712 LiDAR, RGB-D, and RGB aligned and transformed samples – 5916 for training, 787 for validation, and 9 for demonstration. The dataset was split into three subsets. The first subsets contained RGB samples and consisted of images with annotated ground truth ROI labels, indicating the location of people in the frame.

Both datasets contributed to the algorithms reported on in chapter 4 and 5, respectively. They were developed to address the shortcomings – unsuitable operating environments, lack or unsuitable modalities. While some datasets came close to the requirements, they all fell short of what we required to progress research in the areas of free space detection and human activity recognition.

Chapter 4 A Self-Evolving Free Space Detection Model

4.1 Introduction

Several major automobile manufacturers and technology companies have set ambitious targets to commercially launch fully AV's by the dawn of the next decade. Whether you believe them or not, these automobile giants, such as; Tesla Motors, BMW, Ford Motor Co., and Volvo, have promised to have fully autonomous cars on the road by 2030 [294]. Furthermore, the Chinese Government partnered with Chinese internet giant Baidu and set 2025 as the year by which 10-20% of vehicles will be highly autonomous, and by 2030 10% of cars will be fully self-driving [295].

Despite the promises from these multinational giants, vehicles that are good enough to roam extensively without human involvement are still a distant reality, and this warrants extensive research effort [200]. For example, when the California Department of Motor Vehicles (DMV) published its 2018 annual performance report on testing self-driving cars on public roads, there were numerous cases (39%) where the driver had to take control of the vehicle [296]. If numerous drivers are taking control of self-driving cars, the algorithm used for these Intelligent mobile robots, are falling short of their objectives. In this Chapter, a core-functionality of the proposed robotic platform, when applied to FSD, is illustrated. Considered part of the Multi-layered Context Representation and Data Analysis Layer, the proposed framework intends to demonstrate a solution for FSD that works on all kinds of surfaces.

This chapter is organized into the following sections: Section 4.2 presents the problem definition, motivation, and related work. Section 4.3 provides an overview of the different sensor data representations before moving onto the geometric alignment of the sensor data in Section 4.4. Section 4.5 provides an overview of Image and ultrasound-based FSD before discussing the proposed frameworks for online active FSD. Section 4.6 compares the results of the proposed framework to the DL approach for FSD. Section 4.7 summaries of the work presented in this Chapter.

4.2 Problem Definition

This chapter reports on FSD using sensor data fusion derived from data gathered by an ultrasonic sensor array, and the luminance data from a wide-angle imaging sensor. The data from the ultrasonic sensor array comes in the form of 2D ranges and can be used to improve FSD using a semi-supervised form of ML called online active Learning.

FSD research is predominately focused on camera sensors as in [297]–[301], radar-based FSD as in [302], [303], fusion-based FSD using camera LiDAR and/or Radar [116], [303]–[305]. Almost all the FSD algorithms reviewed, rely on a 3D reconstruction either by stereo vision, sensor data fusion of camera, radar and, or LiDAR before fitting a model to the data stream. Of course, these models can vary in complexity, and have increasing computation requirements. However, the cost can be significantly reduced by considering the profile of the road as horizontal and getting the machine to teach itself.

One school of thought is to view FSD as a classification problem where pixels are classed as free space, or not free space [116], [306]–[308]. Here feature information is used to identify what class pixels reside. Problems arise where data is limited. A classifier is only as good as the data used to train it. A solution for FSD needs to work in all kinds of environments, under all kinds of lighting conditions and on all surface types – both indoor and outdoor, carriageways, footpaths, grass, carpet, and tiles. It needs to learn by querying information from image sensors against information from a reliable sensor stream, such as ultrasound or LiDAR. Furthermore, because the framework uses two sensor modalities, the system needs to be able to self-calibrate. While there are arguments for and against, self-calibration was chosen over manual calibration. Although manual calibration generally proves to be more accurate, the practical application of placing a checkerboard in front of the platform every time it roams, had to be considered. With this in mind and the diverse environments the platform will operate in, the algorithm needs to be robust and function under different conditions. It cannot be

influenced by changes in lighting and should work reliably within the range of the different sensor streams and different environments.

4.2.1 State-of-the-Art Free Space Detection

Inherent complexities of the built environment prevent AV from being hard programmed with a fixed set of rules that foresee all possible scenarios they could encounter [282]. Therefore, AV need to learn to make decisions autonomously based on the events they encounter and the objects they perceive. Only in this manner can an adequate driver policy be derived – where the AV self-evolves over time, depending on objects the agent encounters.

FSD can be regarded as the most fundamental element of perception – crucial for researchers to understand, so we do not collide into other objects. In structured environments like carriageways, free space is mainly composed of a delineated road surface. Traditionally these areas are either detected based on colour [309] or texture segmentation [310], deduced from stereovision-based obstacle detection [311], or a combination of both [312]. Knowledge about free space is vital to understanding how to navigate one’s surroundings. Indirectly, FSD assists in the location estimation of an Intelligent autonomous robot. Since location estimation is used for indoor or outdoor navigation – FSD is vital to the safe operation of an autonomous platform.

The majority of FSD research has been focused on outdoor activity and the safe manoeuvring of a vehicle in traffic. As a result, lane-detection and FSD have often been clumped together, especially when considering Advanced Driver Assist Systems (ADAS) [313]. In the most basic form FSD and lane-detection algorithms usually solve the problem using three different steps; pre-processing images, filter noise, and classification [314]. Most research utilizes single or multiple cameras. The pre-processing stage is mostly as simple as colour space conversion, such as chroma-based analysis, that are typically used to reduce noise and, or to mitigate issues with shadow [315], [316]. In other cases, the images are transformed to produce a birds-eye view, in effect producing an OGMap [317]. The final step of most contemporary FSD pipelines is to extract features from the image and classify them. In most cases, the process of learning the features has been delegated to a CNN. While CNN’s are good at identifying the features that allow them to make a classification, they do not understand free space the way humans do. This makes it difficult for a machine to understand the difference between surface and traversable space.

Generally, when using Supervised Learning methods for image processing, a CNN obtains superior results with respect to traditional feature learning algorithms. However, in scenarios where data is sparse, a CNN can be outperformed. CNN for road boundaries [318], lane markings [128] and semantic segmentation of free space [319], get information about the geometric attributes of the lane from the free space they detect. Other methods use clustering and Unsupervised Learning to distinguish the difference between lanes [320]. For example, in [318], researchers used a CNN to segment the different lanes on a roadway.

Researchers in [321] presented a pooling module using a pyramid structure to aggregate background data. The module links the feature map developed by ResNet to the output of the unsampled layer. In addition to an unusual pooling module structure, [321] reported on a new loss function to solve difficulties with mismatched relationships, confusing categories, and inconspicuous classes. In [322], researchers reported on a Dense Up-sampling Convolution (DUC) network and a Hybrid Dilated Convolution (HDC) network. Both the DUC and HDC networks solved the up-sampling and dilated convolution problems by dividing the label map into a subsection with the same size as the input feature map. These techniques facilitated work directly on the feature map and the dilation rates of ResNet – the underlying network used in these cases to detect free space. There are many techniques used for FSD. While the majority of state-of-the-art use CNN to solve road detection tasks, they do so with large quantities of data. Although CNN's have demonstrated exceptional ability to generalize, there are difficulties in the classification of surfaces never encountered before when data is lacking.

4.2.2 Motivations and Requirements

While there are numerous attempts at solutions from academia, most of the experimentation and testing of an AV in different real-life conditions is being led by industry [323]. For example, the DMV in California issued 52 permits to test AV on the road. Of those permits issued, two were granted to academic institutes while the remainder being granted to industry [324]. Of course, having an accessible testbed, open-source data, and publicly evaluated datasets would allow researchers to compare algorithms objectively. Most developers use raw data void of the mechanisms used to train the agent, avoid challenging weather and lighting conditions, or proprietary data, which is rarely made available. This combination of obstacles is essential to overcome if we wish to benchmark a ML system for full 24/7 operation in all environments.

Although modern ML algorithms are an essential contribution to the field of AI, it is only one part of the grand challenge of constructing intelligent machines. While ML algorithms may understand the relationship between the inputs and the outputs, it cannot understand simple relationships – such as the relationship between surface and traversable space. This is one area where our understanding of the environment differs from that of the machine. With modern-day ML, machines understand the patterns that define the environment, whereas humans understand the environment from an egocentric point of view [325]. Consequently, most ML algorithms have difficulty understanding abstract ideas and has no obvious way of developing logical assumptions or integrating abstract knowledge [326]. For example, in contemporary methods of FSD – where traversable space is identified from features learned from data – the deep network holds little knowledge about what free space is. While the most advanced CNNs may recognize the features they learn, features that define free space – they do not understand the relationship like humans do.

Developing ML algorithms that form logical assumptions, integrate abstract knowledge, and understand abstract ideas is the primary objective of any AI research [327]. It may not be possible to construct the same relationship understanding that humans have. And it may also be possible to develop ML algorithms that can integrate abstract knowledge from different sensor streams to develop the ability to make logical assumptions about different sensor data. In this case, the machine may not understand free space in the same way humans do but will have a sort of wisdom learned from a different sensor stream. Under these circumstances, the machine stands a better chance of detecting free space when presented with more data. It will, in effect, self-evolve over time. The requirements for a free space algorithm can be defined as the ability to:

- (1) Learn new data as it is presented to the autonomous platform.
- (2) Develop an understanding of traversable space without large quantities of data at the start.
- (3) Understand the relationship between two different sensor streams and make logical assumptions about different sensor data.

The proposed framework presented in this chapter was developed to address these requirements.

4.3 Geometric Alignment of Sensor Data

The purpose of the geometric alignment is to find the corresponding pixel in the camera output for each data point output by the ultrasonic sensor array. In simple terms, we are taking the plan view of the OGMap, translating and aligning it to the elevated view of the camera data.

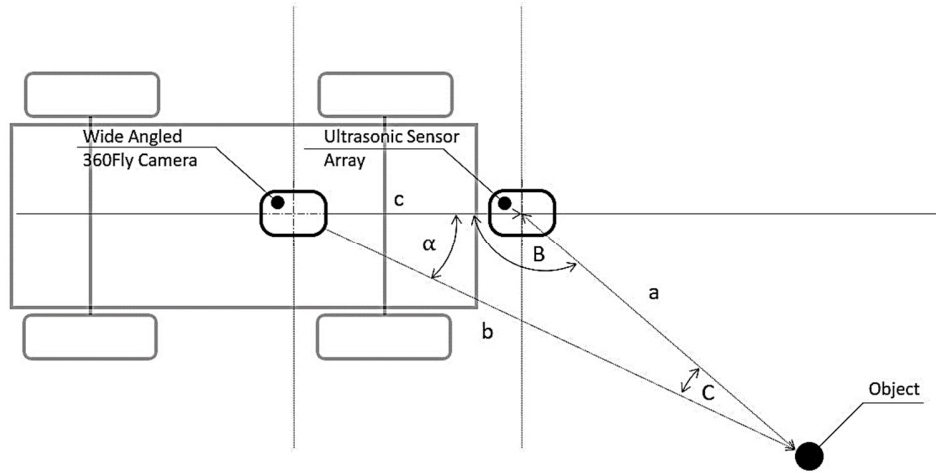


Figure 36: Illustrates the plan view of the sensor setup showing the location of the ultrasonic sensor Array and the wide-angled 360Fly camera. It should be noted that a denotes the range between the ultrasonic sensor array and the object, it is not the perpendicular distance.

To realize this, we need to know the relative location of different sensors – ultrasonic sensor array and the camera. A plan view of the autonomous platform and the sensor setup is graphically illustrated in Figure 36. Figure 37 shows a side elevation of the sensors.

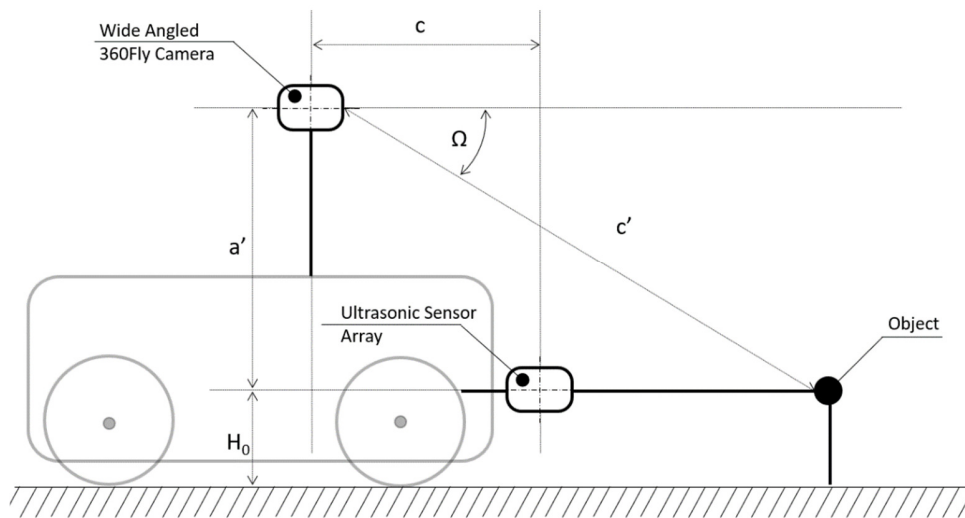


Figure 37: Illustrates the elevated view of the sensor setup showing the location of the ultrasonic sensor array and the wide-angled 360Fly camera.

For this derivation, consider an object O at a distance (a) of 140 cm from the focal point of the ultrasonic sensor array. The ultrasonic sensor array consists of 6 HC-SR04 sensors positioned at 5° , 25° , and 45° either side of the longitudinal axis. In this case, $B = 180^\circ - 45^\circ = 130^\circ$ for an object O identified to the right front of the ultrasonic sensor array. The longitudinal (c) and vertical (a') distance between the camera and ultrasound sensors array is 45 cm and 65 cm, respectively. The ultrasonic array is positioned (H_0) 33 cm above the ground. Considering the distance a between the object O and the ultrasonic array at an angle of B , we can describe b - the distance between the camera and the object O - as:

$$b = \sqrt{a^2 + c^2 - 2 \times a \times c \times \cos(B)} \quad (\text{Equation 13})$$

In turn, the angle α between vectors b and c - can best be described as:

$$\alpha = \cos^{-1} \left(\frac{b^2 + c^2 - a^2}{2 \times b \times c} \right) \quad (\text{Equation 14})$$

Knowing the azimuth angle α and the horizontal distance b , we can use the resultants from Equation 13 and 14 to solve for the range (c') between the object O and the camera:

$$c' = \sqrt{a'^2 + b^2} \quad (\text{Equation 15})$$

From Equation 15, we can calculate the corresponding elevation angle θ for the object O relative to the camera:

$$\Omega = \cos^{-1} \left(\frac{b^2 + c'^2 - a'^2}{2 \times b \times c} \right) \quad (\text{Equation 16})$$

The purpose of this alignment is to find the corresponding pixel in the camera output for each data point output by the ultrasonic sensor array. We assume that the longitudinal axis of the camera and the ultrasonic sensor array are aligned; however, an offset can be accounted for should it be required. Although this process does away with the need for calibration, this method of geometric alignment cannot be relied upon as an entirely robust mechanism. Even if it works quite well, unique imperfections in the sensor assembly and per-unit variations in the manufacturing processes can cause the sensor to deviate from the ideal geometry. Another problem that arises when fusing data from different sources is the difference in data resolution. It should be noted that the resolution of the ultrasonic sensor array is substantially lower than the camera. Although the resolution could be increased, it will never meet that of the camera data, and therefore is regarded as an intrinsic shortcoming of this process.

4.4 Self-Evolving Free Space Detection

One of the fundamental challenges to training most ML algorithms is access to annotated data. Online and active Learning provides a solution to this fundamental problem. Here a reliable sensor stream is used to label camera data as it becomes available before updating the SVM predictor for future events. Of course, we could use an alternative ML method, such as a neural network. However, the time to train the alternative method needs to be considered since this method is expected to update or re-train in as short a period as possible. For this reason, an SVM was chosen to serve as the classifier. Using this approach improves the classifier’s ability to recognize free space when it has little information to start with. Typically, a person is queried instead of a sensor. However, when this process is used in conjunction with sensor fusion – it does away with the need of a person – and returns a result that is in effect case-specific to space the machine has just encountered.

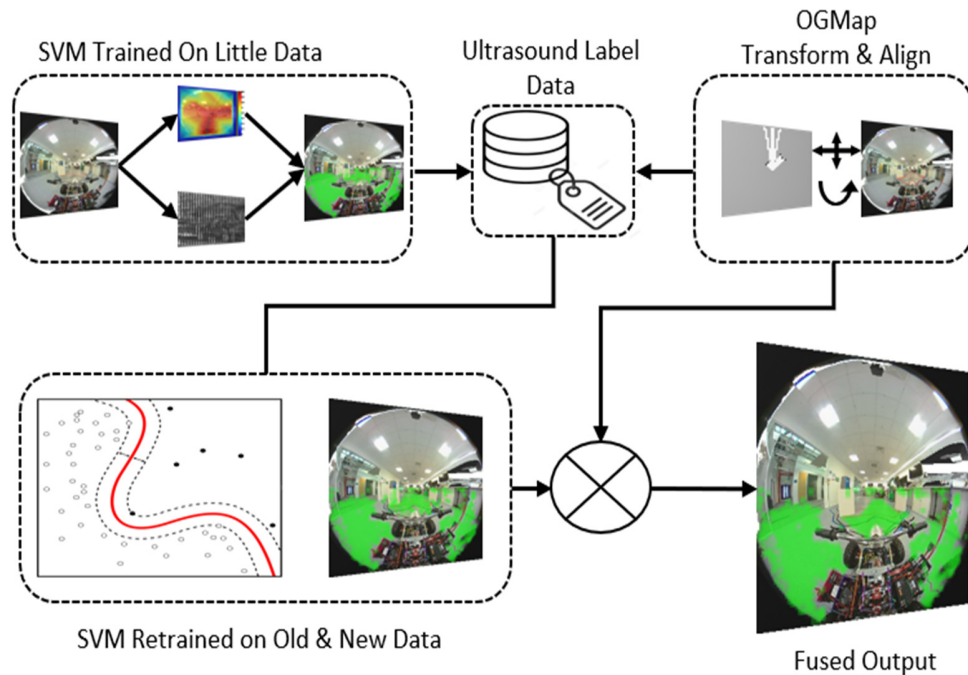


Figure 38: The proposed FSD pipeline. We first train an SVM on a small quantity of data. As new data becomes available, we quarry the robust sensor stream as to its class.

Figure 38 shows the proposed pipeline for a self-evolving free space detection model. Figure 39 shows the architecture of the proposed FSD framework. The proposed framework utilizes a robust 2D ultrasound sensor stream to self-learn. It improves the relative uncertainty of free space identified using monocular camera data alone. The framework is composed of three elements – Sensing and Perception, Localization and Mapping, and Driver Policy.

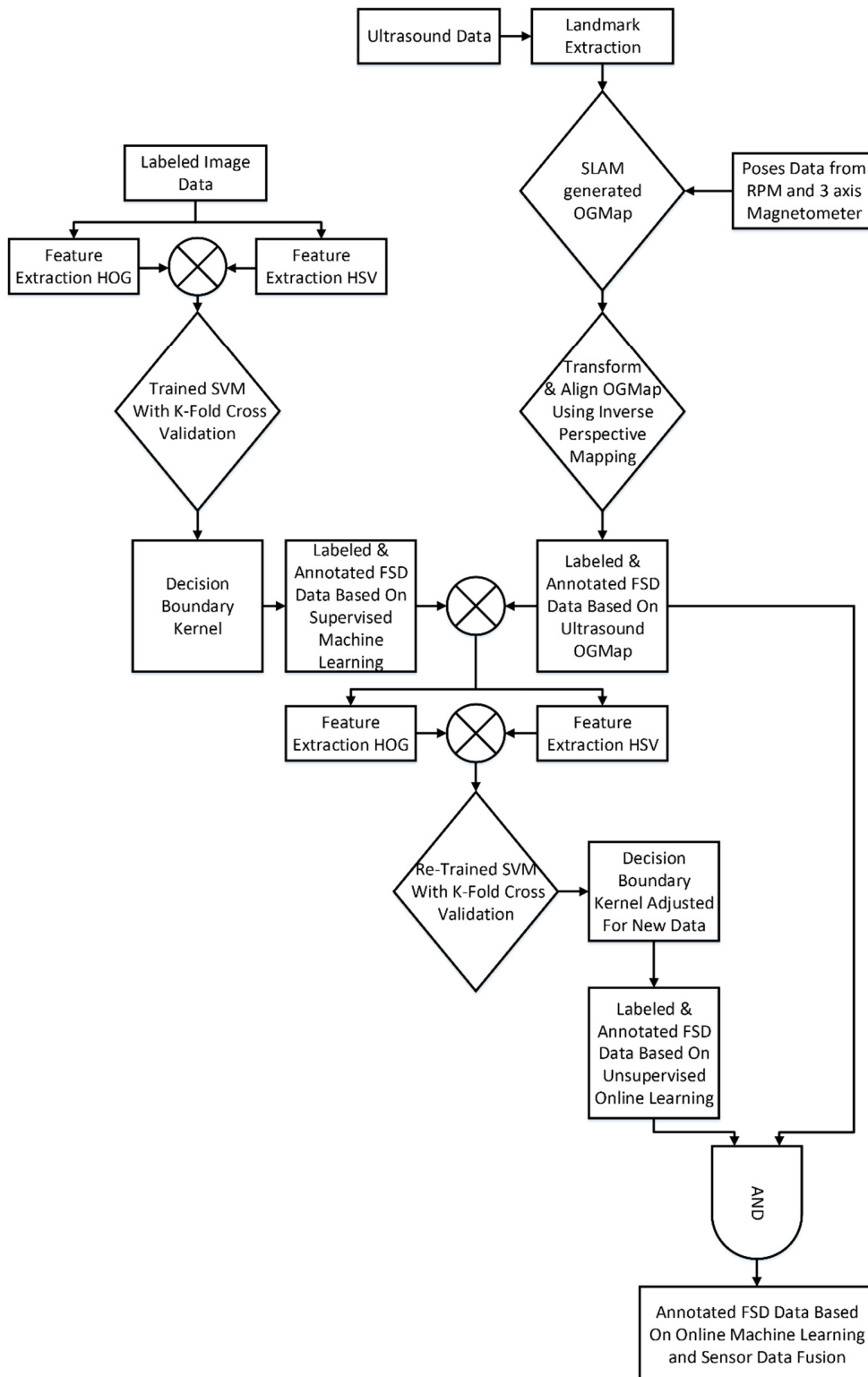


Figure 39: The proposed framework for FSD utilizing supervised and semi-supervised ML. This technique uses online active ML methods to self-learn free space and evolve as it encounters new data.

The first element is an SVM classifier trained using manually labelled image data. The features of the labelled data used to train the SVM were Histogram of Oriented Gradients (HOG) and HSV. Feature descriptors are tools used in computer vision to describe useful elements of an image or image patch on a pixel-by-pixel basis.

Both HSV and HOG values are feature descriptors that are commonly used in ML [328]–[331]. Usually, feature descriptors such as HOG turn a $m_i \times n_i \times 3$ matrix into a feature vector of a certain length. Of course, the patch size can change depending on the requirements, which in turn changes the length of the feature vector. While the feature vector makes little scenes when viewed as an image, it is the element that is learned by the classifier. It dramatically assists researchers in identifying free space. HSV can be used in the same way as HOG. Although a feature vector is not generated for the HSV value, it is a feature descriptor that is useful in classification.

In the first element of the proposed framework, the SVM classifier is trained on a small amount of both HSV and HOG values. In this case, the ML algorithm learns a basic understanding of free space. It stores the output as a kernel function that defines free space based on the small training set. The second components generate an OGMap using ultrasound and pose data. The ultrasound data is used to label the image data before adding it to the dataset and retraining the classifier. This is the online active Learning element, and it continuously updates as the autonomous platform traverses' new space.

The final element of the proposed framework fuses the OGMap generated by the ultrasound sensors with the prediction made in the second element of the proposed framework. Outside the FoV of the ultrasonic sensor array, there is no difference between the results generated by the second and third elements of the proposed framework. However, inside the FoV of the ultrasonic sensor array, the fused data makes a more conservative prediction about traversable space.

It is important to note that the longer the algorithm is in operation, the better it becomes at predicting free space. It should also be noted that at each point when the framework is presented with new labelled data - it retrains. As a result, any knowledge it has gained about free space is lost each time this procedure is undertaken. This makes for an efficient free space detector, but it is not cost-effective on resources – since it is performing a task, it has already performed in the past.

4.4.1 Image Based Free Space Detection

The image-based FSD algorithm at the start of the proposed framework is a Supervised ML algorithm. A training set was developed from patches of the camera images. Each patch set was assigned to a class as free space or not. 1,500 image patches of size 8×8 were collected from ten randomly selected video frames. All videos exhibited a variety of lighting conditions from indoors and outdoors scenarios and many different traversable surfaces. HOG features and HSV values were extracted from training image patches and used to train the SVM classifier.

The HOG features are calculated for every 4×4 blocks within the 8×8 patch, while HSV values were chosen for the entire patch. An RBF is used as the kernel for the SVM. The proposed framework utilizes pre-labelled data to train an SVM. K-fold cross-validation is utilized for model selection. In this case, the original sample is randomly partitioned into ten equal size subsamples. Of the ten samples, a single sample is retained for validation, with the remaining used for training. The inbuilt functions of MATLAB®R2018, an imaging processing toolbox were used for feature extraction and training of the SVM using the function 'fitcsvm'[332]. It should be noted that while image-based FSD utilizing an SVM is not state-of-the-art, it is a starting point for the online active ML element of the algorithm.

4.4.2 Ultrasound-Based Free Space Detection

The second element of the proposed framework is a SLAM generated OGMap. Typically, an OGMap is used to describe occupied space in a discrete grid. In this case, depth information is gathered from ultrasound data to construct a discrete map of the environment. Initially introduced in [234], OGMap has long been regarded as the standard for environment representation in robotics.

An OGMap is generated by extracting landmarks from the ultrasound data for individual scans before sequentially adding each scan using the RPM and three-axis Magnetometer. Using the pose data logged from the RPM and three-axis Magnetometer, it is possible to build an OGMap of the area over which the testbed has traversed. Here, all the ranges logged by the sensor array are mapped on to a 2D grid. In this case, the depth information is gathered from an ultrasound sensor to construct a discrete map of the environment.

Data is transformed and aligned to the camera view, as shown in Figures 36 & 37. A typical OGMap generated by the ultrasonic OGMap is depicted in Figure 40. In Figure 40, the

white rays emanating from the testbed's position at the zero coordinates correspond to free grid points, while the black grid points indicate occupied space.

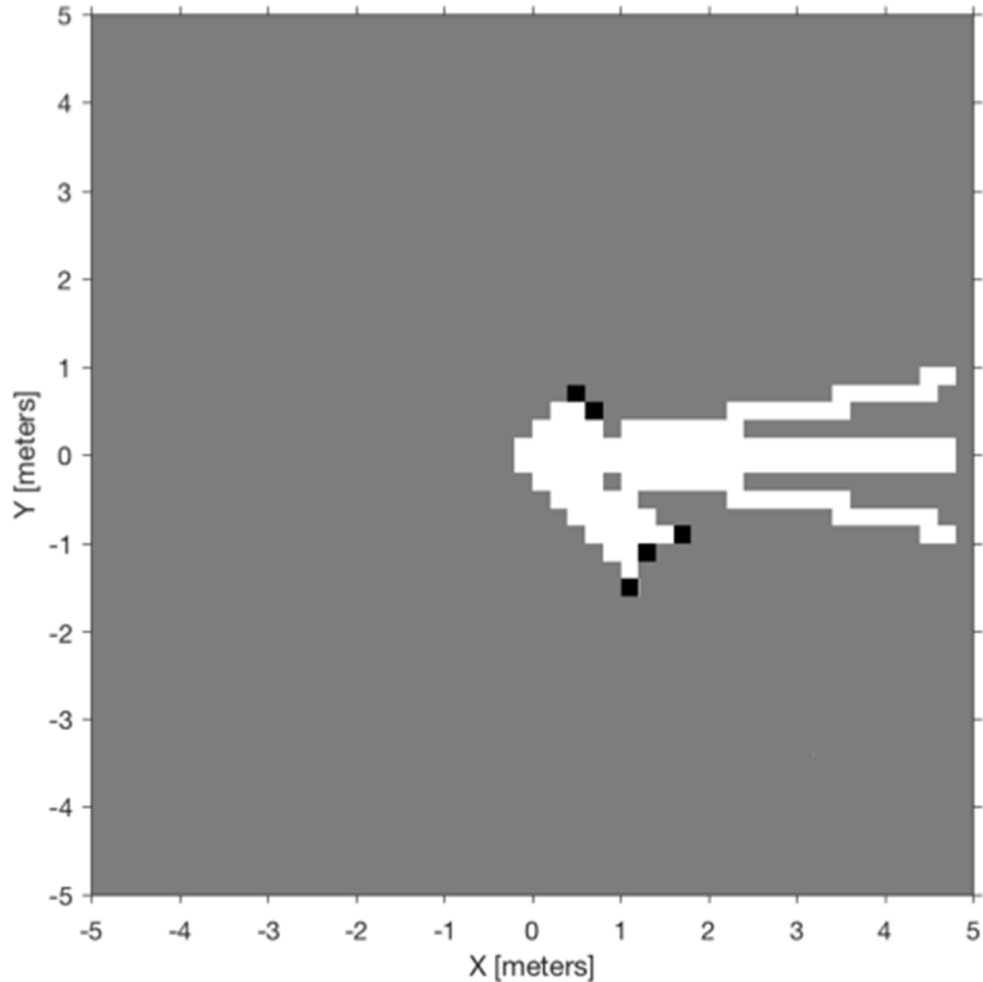


Figure 40: Shows the results of the ultrasound OGMap matching Scenario 1 in Figure 41. In this image, the white rays emanating from the testbed's position at the zero coordinates correspond to free grid points, while the black grid points indicate occupied space.

4.4.3 Online Active Learning for Free-Space Detection

The final element of the FSD algorithm of the proposed framework is a combination of unsupervised ML and sensor fusion. Free space is learned from examples that have been classified by the data from the ultrasonic sensor stream. Like the first element of the proposed framework, each patch is assigned a class. A total of 1,650 image patches of size 8×8 were used to retrain the SVM.

Image patches were a combination of data collected from the ten randomly selected frames used by the first component of the framework, and the new frames encountered by the testbed at that time. HSV and HOG features are extracted from the image patches. The fusion

element of this component occurs before the image is annotated. Since the OGMap has already been transformed and aligned to the camera data, fusion occurs with little difficulty using a logic function.

4.5 Results and Discussion

The purpose of the proposed algorithm is to assist the AV with perception tasks. This subsection presents the performance and results found during this research. Here the effectiveness of the proposed online active Learning Algorithm for Self-evolution of FSD (combined semi-Supervised ML and sensor fusion) is compared to the current state of the art, DeepLabV3+ [333]. For the most part, existing techniques for FSD are sensitive to lighting conditions and have difficulties generalising – because of the infinitely different traversable surfaces they encounter. Querying RGB data captured by a camera against a reliable sensor stream – ultrasonic ranges – removes the need for the classifier to generalize and reduces the influence lighting conditions might have.

4.5.1 Dataset

The proposed self-evolving FSD framework was trained and evaluated on the LboroLdnAV dataset [280]. The LboroLdnAV dataset consists of 45.6 hours of Video, LiDAR, and ultrasound data collected over 1.2 km. Since we were using both ultrasound and camera data it was not possible to do cross dataset validation. For the proposed self-evolving FSD framework, a small subset of the dataset was used to train the SVM classifier. This subset consisted of 3 frames extracted 10 videos displaying a variety of scenarios from both indoor and outdoor environments. All images used to train the classifier were annotated, indicating traversable space.

A small subset, 1500-pixel patches, were used to train the SVM classifier. Patches were extracted from the 3 frames randomly selected from 10 videos. All videos displayed a variety of scenarios from both indoor and outdoor environments. We reduced the number of classes in the testing dataset from 7 classes into 2 super classes. For example, Free Space is all traversable surfaces. Everything else was grouped into the superclass ‘Not Free Space. During the experiments, the testbed was moving at 4kph (1.1m/s), the camera frame rate was 29 fps, and the ultrasonic array capture rate was 6 fps. Figure 41 illustrates the screen capture of scenarios used for validation. Table 10 provide the scenario details depicted in Figure 41. The

subset of the dataset chosen to test the SVM indicates multiple different surface types – indoor, outdoor, lino, concrete paving slabs, tarmac, AstroTurf, tiles, and low pile carpet.



Figure 41: A subset of the data used during the comparison between the proposed framework and Deep LabV3+. From the top left corner: (a) Scenario 1, (b) Scenario 2, (c) Scenario 3, (d) Scenario 4, (e) Scenario 5, (f) Scenario 6, (g) Scenario 7, (h) Scenario 8, (i) Scenario 9 and (j) Scenario 10. Scenario details reported in Table 10.

TABLE 10: SCENARIO DETAILS DEPICTED IN FIGURE 41

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	Scenario 1: Indoor environment with stationary obstacle traversing lino.	Scenario 2: Outdoor environment traversing changing surface (Concrete to Tarmac).	Scenario 3: Outdoor environment traversing changing surface (Concrete to AstroTurf).	Scenario 4: Indoor environment with stationary obstacle traversing tiled surface.	Scenario 5: Outdoor environment with stationary obstacle traversing concrete.
Row 2	Scenario 6: Indoor environment with stationary obstacle traversing lino.	Scenario 7: Indoor environment traversing carpet into a corner.	Scenario 8: Indoor environment stationary obstacle traversing lino.	Scenario 9: Outdoor environment with Stationary and moving obstacle traversing concrete.	Scenario 10: Outdoor environment moving obstacle traversing changing surface (Astroturf to Concrete).

For evaluation purposes and to test the ability of a deep network to generalize, we trained the current state of the art semantic segmentation network DeepLabv3+ [333] using the CamVid dataset [239]. The CamVid database was collected using a 3CCD Panasonic HVX200 digital camera mounted to the dashboard on the passenger side of a vehicle driven for two hours around Cambridge. The footage used to construct the dataset came from 22 min and 14 sec of video footage. Data was collected at 30 fps with a resolution of 960 x 720 pixel. The dataset provides ground truth labels that associate each pixel with one of 32 semantic classes. From the 22 min and 14 sec of video footage, 10 min of high-quality 30Hz footage was chosen and semantically labelled images at 1Hz and in part, 15Hz [239].

We used a pre-trained ResNet-18 [148] to initialize the weights of Deeplab v3+. ResNet is a reliable CNN commonly used for image recognition tasks. Furthermore, we reduced the number of classes in the CamVid dataset from 32 classes into two super classes. For example, Free Space is a combination of Sidewalk, Road, Road Shoulder, Drivable Lane

Markings and Non-Drivable Lane Markings. The remaining classes of the CamVid dataset were grouped into the superclass ‘Not Free Space.

Testing of the proposed self-evolving FSD framework and DeepLabV3+ framework was done using a subset of the LboroLdn AV dataset. The video and ultrasound data used for evaluating the different frameworks were not part of the video data used to train the proposed framework. Testing data was chosen in this manner to demonstrate the generalizability of both frameworks to new and unseen scenarios. This subset of the dataset indicates multiple different surface types – indoor, outdoor, lino, concrete paving slabs, tarmac, AstroTurf, tiles, and low pile carpet.

4.5.2 Performance

There are many networks suitable for semantic segmentation of free space [309]–[312], [318]–[320]. While all these networks function quite well when tested on data, they are familiar with, their ability to generalize semantic scene representations for FSD – that they have never encountered before – is the real litmus test of their ability. Although there are several approaches to FSD – OGMap’s [302], [334], fused LiDAR and Image data [60], [335] stereo/monovision [336], [337] – few if any have used fused ultrasound data and image data to facilitate this process.

Typically, in FSD tasks, the accuracy, the bfScore, and the weighted intersect over union (IoU) suffice as metrics used to measure performance. However, when scrutinizing FSD frameworks, two body of opinion regarding metrics should be kept in mind; how well the classifier works on the test data, or dataset metrics; and how well the classifier works on the individual class, or class metrics. To that end, we report on the performance of the proposed self-evolving FSD framework and the current state of the art – DeepLabV3+.

The dataset metrics describe metrics that rank the response of the proposed framework to the test data. They aggregate the response of the algorithm and provide detail as to how well the framework performs over different scenarios. The class metrics indicate the response of the framework to specific classes. Thus, the class metrics tell how well specific class are identified by the framework. While dataset and class metrics tell different things, they both utilize similar techniques.

For example, the accuracy indicates the percentage of correctly identified pixels for each class. Defined as the ratio of correctly classified pixels to the total number of pixels in that class, according to the ground truth. For the aggregate dataset, the mean accuracy is the

average accuracy of all classes in all images. Consequently, class accuracy is typically used in conjunction with IoU for a complete evaluation of segmentation results.

The IoU is the most commonly used metric in semantic segmentation and object detection. For each class, IoU is the ratio of correctly classified pixels to the total number of ground truth and predicted pixels in that class. For the entire data set, the mean IoU is the average IoU score of all classes in all images. Concurrently we can weight the IoU by the number of pixels in that class if we want a statistical method that penalizes false positives. This metric is used if images have disproportionally sized classes, to reduce the impact of errors in the small classes on the aggregate quality score.

TABLE 11: SEMANTIC SEGMENTATION NETWORKS MEAN IOU

Method	Ref	Year	mean IoU
Deep Layer Cascade	[338]	2017	82.7
ResNet-DUC-HDC (TuSimple)	[322]	2018	83.1
GCN (Large Kernel Matters)	[339]	2017	83.6
RefineNet	[340]	2016	84.2
ResNet-38	[341]	2019	84.9
Pyramid Scene Parsing Network	[321]	2017	85.4
IDW-CNN	[342]	2017	86.3
Stacked Deconvolutional Network	[343]	2019	86.6
Deep Dual Learning	[344]	2017	86.8
DeepLabv3	[345]	2017	85.9
DeepLabv3+	[333]	2018	89.0

Like accuracy, the bfScore or boundary F1 Score considers both the precision and the recall of the classifier to determine the advantage of one system over another. Typically, the BF score is a metric that tends to correlate better with human qualitative assessment than the IoU. For each class, the mean bfScore is the average BF score of that class overall images. For the aggregate data set, the mean bfScore is the average BF score of all classes in all images.

To evaluate the performance of the proposed framework, we compare metrics and visually appraise the results of the proposed framework against those of DeepLabV3+ [333]. DeepLabV3+ was chosen because of the high mean IoU. While the bfScore would have been a better correlation to human qualitative assessment, it is not as readily available as the mean IoU. Table 11 lists alternative networks capable of classifying free space that was deemed incongruous because the mean IoU was less than that of DeepLabv3+.

A. Online Active ML Performance

Table 12 reports on the global average, mean accuracy, mean IoU, weighted IoU and mean bfScore for the proposed Online Active ML framework. These metrics report on the response of the Online Active ML Framework to all the test data. In this case, the proposed framework performs quite well for most of the metrics and reasonably well for the mean bfScore.

TABLE 12: DATASET METRICS FOR THE ONLINE ACTIVE ML FRAMEWORK

	global average	mean accuracy	mean IoU	weighted IoU	mean bfScore
Test Data	0.9097	0.8517	0.7682	0.8369	0.5858

Table 13 reports on the accuracy, IoU, and mean bfScore for the proposed self-evolving FSD framework. These metrics report on the response of the individual classes in the dataset. Interestingly, when considering the individual class metrics, the proposed self-evolving FSD framework performs better on the “not free space” class when compared to the “free space” class.

TABLE 13: CLASS METRICS FOR THE PROPOSED ONLINE ACTIVE ML FRAMEWORK

	accuracy	IoU	mean bfScore
Free Space	0.7461	0.6446	0.4935
Not Free Space	0.9571	0.8917	0.6779

Figure 42 shows the confusion matrix for the proposed self-evolving FSD framework. On the Y-axis are the Output Class, and on the X-axis are the Target Class. The diagonal cells, dividing either side of the matrix, indicate true positives that are correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified.

As the metrics reported in Table 12 and 13 show, the confusion matrix indicates something similar – the proposed self-evolving FSD framework performs better on the “not free space” class when compared to the “free space” class. Overall, the metrics in Table 12, Table 13 and the confusion matrix in Figure 42 shows that online active ML generalizes exceptionally well to environments never before encountered.

Furthermore, the quantity of data required to get the network to the point where it can classify traversable surfaces with a high degree of accuracy is relatively little. When compared to a Neural Network and the time it takes to learn new data, the practicality of using time consuming ML methods become abundently clear.

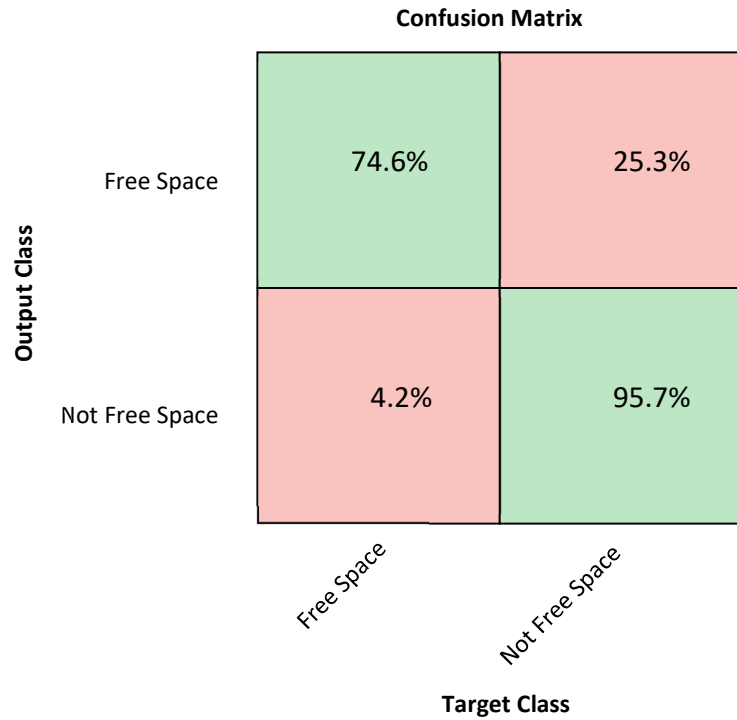


Figure 42: Confusion Matrix for the online active ML framework. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified

B. DeepLabV3+ Performance

Table 13 reports on the global average, mean accuracy, mean IoU, weighted IoU and mean bfScore for the DeepLabV3+ framework. These metrics report on the response of DeepLabV3+ to all the test data. It should be noted that in all cases, the dataset metrics for DeepLabV3+ were lagging behind those of the proposed self-evolving FSD framework.

TABLE 14: DATASET METRICS FOR DEEPLABV3+

	global average	mean accuracy	mean IoU	weighted IoU	mean bfScore
Test Data	0.8450	0.6997	0.6101	0.7294	0.4421

Table 14 reports on the accuracy, IoU, and mean bfScore for the DeepLabV3+ framework. These metrics report on the response of the DeepLabV3+ to the individual classes in the dataset. Similar to the proposed self-evolving FSD framework, DeepLabV3+ reports a higher error on the “free space” class than the “not free space” class.

TABLE 15: CLASS METRICS DEEPLABV3+

	accuracy	IoU	mean bfScore
Free Space	0.4309	0.3905	0.2973
Not Free Space	0.9686	0.8296	0.5869

Figure 43 shows the confusion matrix for the DeepLabV3+. As before the Y-axis are the Output Class, and on the X-axis are the Target Class. The diagonal cells, dividing either side of the matrix, indicate true positives that are correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified. When comparing Table 12 to Table 14, Table 13 to Table 15 and Figure 42 to Figure 43, it is clear that the proposed self-evolving FSD model outperforms DeepLabV3+. While DeepLabV3+ performs relatively well, it lags behind the online active ML method for generalizing. Furthermore, the quantity required to train DeepLabV3+ is many times greater than the data to get the proposed self-evolving FSD framework up and running.

Confusion Matrix

Output Class	Free Space	43.0%	56.9%
	Not Free Space	3.1%	96.6%
		<i>Free Space</i>	<i>Not Free Space</i>
		Target Class	

Figure 43: Confusion Matrix for DeepLabV3+ framework. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified.

4.5.3 Visual Results

The purpose of the proposed algorithm is to assist an AV with FSD. We benchmarked the proposed self-evolving framework against DeepLabV3+ and presented some of the findings in Figure 44. Table 16 reports on the scenario's depicted in Figure 44. The results presented in Figure 44 Scenario 1 (a), Scenario 2 (a), Scenario 3 (a) and Scenario 4 (a), indicate

that the proposed Online Active ML framework outperformed its counterpart DeepLabV3+ when presented with scenarios never encountered before.

In consonance with the results in Figure 44 Scenario 1 (a), Figure 44 Scenario 2 (a), Figure 44 Scenario 3 (a) and Figure 44 Scenario 4 (a), the proposed online active ML algorithm for Self-evolution of FSD returns a superior result to detecting free space to DeepLabv3+.

While Deeplab v3+ is still capable of identifying free space, there are several misclassifications – as can be seen in Figure 44 Scenario 1 (b), Figure 44 Scenario 2 (b), Figure 44 Scenario 3 (b) and Figure 44 Scenario 4 (b). For example, Figure 44 Scenario 1 (a) the area immediately to the front of the autonomous platform is correctly classified, whereas in Figure 44 Scenario 1 (b) DeepLabv3+ misclassifies the area as occupied space. This corresponds to a situation where DeepLabv3+ has generalized relatively well from the data it was trained on to the unfamiliar data used to test both frameworks.

Yet again, in Figure 44 Scenario 2 (b), DeepLabv3+ fails to detect a large portion of free space to the front of the platform. Whereas in Figure 44 Scenario 2 (a), the proposed framework outperforms and accurately classifies the area to the front of the platform. The proposed framework performs poorly on the concrete paving stones either side of the autonomous platform handlebars, as opposed to DeepLabv3+, which classifies the paving stones correctly. Moreover, DeepLabv3+ mistakenly classifies the sky and part of the autonomous platform as free space.

Almost a repeat of the results found in Figure 44 Scenario 2 (b), Figure 44 Scenario 3 (b) misclassifies the area to the front and correctly classifies the area either side of the platform correctly. Oppositely Figure 44 Scenario 3 (a) outperforms DeepLabv3+ in almost all the free space in the image except for a small area to the right front of the autonomous platform on the AstroTurf. In Figure 44 Scenario 4 (a), we have a similar situation where the proposed framework classifies almost all the free space in the image correctly. Conversely, Figure 44 Scenario 4 (b) classifies the area to the front and side of the platform, incorrectly.

TABLE 16: SCENARIO DETAILS DEPICTED IN FIGURE 44

	Images
Row 1	Scenario 1: Indoor environment with stationary obstacle traversing lino.
Row 2	Scenario 2: Outdoor environment traversing changing surface (Concrete to Tarmac).
Row 3	Scenario 3: Outdoor environment traversing changing surface (Concrete/AstroTurf).
Row 4	Scenario 4: Indoor environment with stationary obstacle traversing tiled surface.

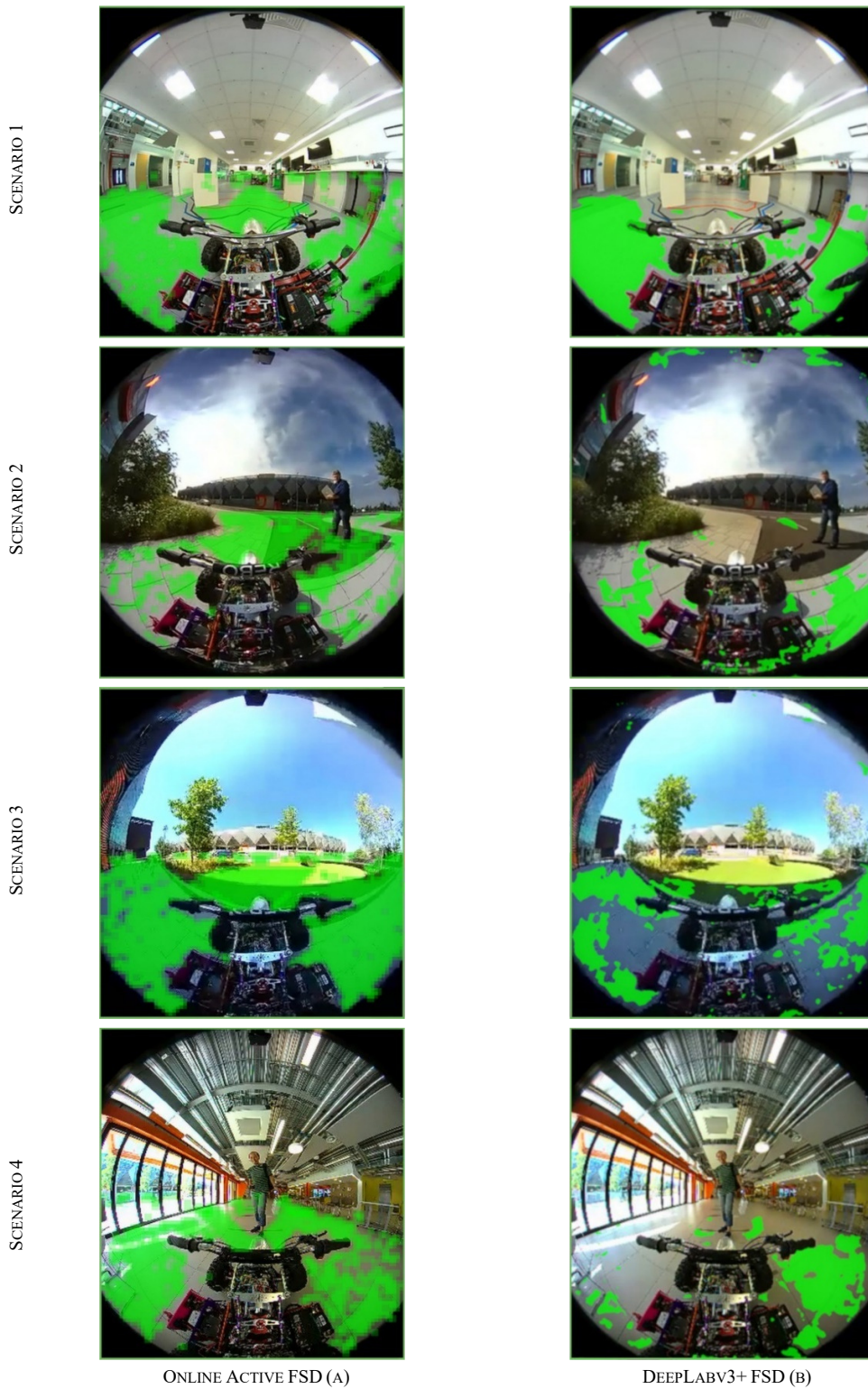


Figure 44: Visual results of the Semi-supervised and Fusion approaches to FSD and DeepLabv3+ FSD. From the top-down. Scenario 1, Scenario 2, Scenario 3 and Scenario 4. Table 16 details the scenarios depicted in Figure 44. Images in the first column indicate the output of the proposed self-evolving FSD framework. Images in the second column indicate the output of DeepLabV3+.

Visually the proposed online active Learning Algorithm for Self-evolution of FSD returns a superior result to DeepLabv3+. Although it could be argued that DeepLabv3+ was not trained on the same dataset as the proposed framework (1500-pixel patches), it would be difficult to see how such a small data would impact the overall performance [117]. Moreover, one of the desired outcomes of DL is to produce a network that can generalize from scene to scene. This clearly is not happening here as DeepLabv3+ misclassifies so much of the test data.

Further arguments regarding the underperformance by DeepLabv3+ rotate around the wide-angled lenses used to capture the test data. Although a valid argument, it can be disregarded since DeepLabv3+ performs poorly on all the test at the centre of the image. As is the case with all images captured using a wide-angle lens, the area at the centre of the image is least distorted and is almost identical to data captured using a standard prime lens. Since the CamVid dataset was captured using a 3CCD Panasonic HVX200 digital camera fitted with a standard prime lens, DeepLabv3+ should have been able to generalize from its training data and classify pixels in the centre of our test data with relative ease.

It should be noted that the proposed framework uses online active ML to self-learn free space. The fundamental principle behind the proposed framework is the querying of optical data as it comes available against the robust sensor stream – ultrasound. In effect, the proposed framework will classify on a case by case basis and improve its accuracy over time.

4.6 Summary

In this chapter, we report on the method of FSD using sensor data fusion derived from data gathered by an ultrasonic sensor array and luminance data from a wide-angle imaging sensor. The data from the ultrasound comes in the form of 2D ranges and can be used to improve FSD using a semi-supervised form of ML called online active learning.

The purpose of the proposed algorithm is to assist the AV with perception tasks. This subsection presents the results found during this research. In this section, we compare the effectiveness of image-based FSD to the ultrasound-based FSD against the proposed online active Learning Algorithm for the Self-evolution of FSD. Furthermore, we compare one of the most prominent methods of FSD, DeepLabv3+ against the proposed framework. Existing techniques for FSD are sensitive to lighting conditions and have difficulties generalising – because of the infinitely different road surfaces they encounter. The difficulties in generalization is a consequence of limited data and diversity.

This research compares four different methods for identifying free space. In consonance with the results, the proposed framework takes a more conservative approach in detecting free space as it utilizes both online, active ML & sensor data fusion. For example, in Figure 44 Scenario 1 (a) and (b), the same obstacles are detected as occupied space. However, in Figure 44 Scenario 1 (b), DeepLabv3+ incorrectly classifies the space between the autonomous platform and the boxes as occupied space. This corresponds to a situation where the image-based FSD is performing poorly due to high saturation, and DeepLabv3+ cannot generalize from the data used to train it. In Scenario 1 of both Figures 40 and 41, it is only the proposed framework that can identify the boxes as obstacles and the area in front of the autonomous platform as free space.

This research addresses the problem of FSD utilizing online and active ML, and fusion of ultrasound and monocular image data. The proposed framework can be broken into three components. The first component is a supervised ML classifier. The second component is a semi-supervised ML classifier that utilizes a robust sensor stream – ultrasound data – to query information from image data to classify free space. The final component fuses the results of the semi-supervised ML classifier with the ultrasound sensor data to improve FSD.

Chapter 5 A Multimodal Fisher Vector Network for Human Activity Recognition

5.1 Introduction

Long-range optical depth scanners such as LiDAR sensors are becoming increasingly commonplace in industrial systems. These optical sensors often complement traditional cameras (RGB-imaging sensors) but perceive the environment more robustly. Detecting and recognizing objects within an image is an essential element of many emerging industrial control systems. CNN is a type of ML. They are mostly used in processing RGB image data for tasks, such as object detection and recognition. Most recently, researchers have been exploring ways of applying CNNs to 3D optical data. Although these methods are encouraging, they are typically based on a single modality and cannot draw on information from other complementing sensor streams, like a camera. Multimodal sensing merges data from different sensor streams to improve the accuracy of recognition tasks.

This chapter investigates a novel CNN architecture to leverage the benefits of sensing redundancy for HAR. A Multimodal RGB and-3D modified Fisher Vector Network (3D-MfV Net) is presented to process RGB image data and 3D LiDAR data collectively. It is demonstrated for a use-case in HAR on LiDAR streams. Evaluation of a custom captured multimodal dataset demonstrates that the model outputs are remarkably accurate in object detection of RGB images, 3D segmentation, and human activity classification. Furthermore, the proposed method provides results that compare favourably with the state-of-the-art ML

algorithms such as PointNet and its variations [1], [2], [346]. HAR is a combination of research into computer vision, ML and human-computer interaction [269]. Achieved using optical-based sensors [347], radio-based sensors [348], or fuse data captured from different sensor types [269], HAR is applied to different scenarios ranging from assisted living (AL) [279] to driver policy for AV [116], [349].

HAR is realized by classifying the activity of a person, before getting a machine to anticipate the activity the subject is performing. For successful advancements in HAR, machines need to understand and make use of multiple signals [350]. Naturally, where more diverse sensor types are used, the focus of research shifts from traditional single modality into multidimensional feature learning networks. The success of CNN's ability to classify images can, in part, be attributed to their architecture [351]. Coincidentally, it is the architecture and the way they process image data that also prevents their application to 3-Dimensional feature learning.

The purpose of this current research is to segment and classify Point Cloud data using a ROI extracted from an RGB image. In this chapter, we propose a MfV Net. The proposed framework uses a pre-trained CNN (ResNet-50) [148], [351] and a region proposal network (RPN) as an Object Detector [352]. The purpose of the Object Detector is to identify an ROI in the RGB image, such as a person performing an activity. The corresponding ROI is translated and aligned to Point Cloud data before being segmented and classified using a 3D Fisher Vector representation, which is derived from a Gaussian Mixture Model (GMM) [353].

This approach extends research on the 3DmFV Net algorithm proposed in [353]. Specifically, we developed a technique where a detected ROI is projected onto LiDAR data, followed by a Point Cloud classification of an activity performed by a human. Figure 45 depicts the MfV- Net pipeline. In line with [26], there are five core technical challenges for this research. They can be described as; representation which is concerned with the difference between data types and how they relate to each other; translation which is the mechanism of moving data from one plane to another; alignment which indicates the relationship between two different sensor streams and is typically denoted by the proximity of each sensor to each other; Fusion which denotes the joining of sensor data to improve the accuracy of a predictive algorithm; and co-learning which indicates how knowledge is recognized from one modality for use with another.

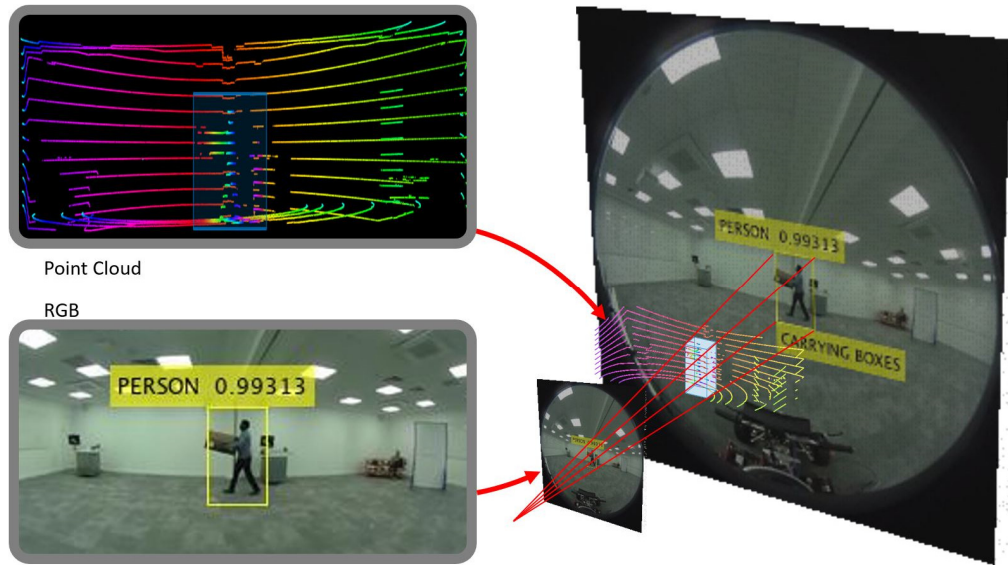


Figure 45: Depicts the MfV Net HAR pipeline. Given RGB data, we first detect a subject and generate an ROI. Each RGB ROI is translated and aligned onto the 3D point cloud. The corresponding 3D ROI is then passed onto the classifier before deciding what activity is being performed.

The main contribution of this work is a new, accurate, and reliable method of HAR, which leverages multiple sensing modalities. Additional contributions that are a result of our methodology include Detection of an ROI, the translation and geometric alignment of multiple modalities such as RGB, RGB-D and Point Cloud data, and the Segmenting of 3D objects. It should be noted that 3D data and Point Cloud data are used interchangeably through the course of this chapter.

The rest of this chapter is organized as follows: Section 5.2 details some of the problems with HAR and the motivation and context; Section 5.3 reports on sensor data representations used in this chapter, followed by the proposed framework in Section 5.4. Section 5.5 details the results and discussion before moving onto the summary in Section 5.6.

5.2 Problem Definition

HAR is the study of extracting human silhouettes, tracking them in the temporal domain, and classifying activities based on the patterns analysed [354]. This is a very advanced area of research with some significant performance, albeit on RGB images/videos or wearable sensors. HAR is an essential element in many industrial applications, such as assistive robots [355], driverless cars [356], and sports analytics [357].

HAR is proposed in this research to decrease road accidents by recognizing pedestrian activities. In [263] and [264], authors provide a new application of HAR for a pedestrian

recognition system that matches the predicted intention with that of a driver's direction. While most of the HAR activities have been focused on RGB images, most of the emerging assistive systems are equipped with depth sensors such as LiDAR. Such sensors often produce 3D Point Clouds. Most of the driverless vehicle prototypes are attached with LiDAR sensors. The advantage of LiDAR is that it can complement other sensors. For example, during the night-time, while cameras would not produce accurate results, LiDAR would be able to sense with a higher degree of accuracy.

There is very little research to develop detection and recognition tasks on 3D Point Cloud data. Annotating and labelling 3D Point Cloud data is a challenging and time-consuming task when compared to labelling RGB images. This makes it difficult to utilize state-of-the-art techniques such as CNNs for Supervised Learning tasks. To exacerbate the issue, there are a limited number of multimodal datasets available, which can be used to develop recognition algorithms. For example, among the available datasets for HAR, only a handful contain both LiDAR data and RGB images/videos [279] [240] [242].

5.2.1 3D Point Cloud Machine Learning

Point Cloud data is not a natural input to CNN. CNN's were designed to process RGB images or suitable data format, therefore adapting them to work on 3D data is not a straightforward extension. Depending on the sensor FoV Point Clouds are unstructured, unordered, prone to missing data, and are affected by noise and rotations[353].

Early attempts at training a Network using Point Cloud data, required transforming the 3D data to a series of RGB images at multiple views. These networks learned the depth map of the scene, rather than the 3D objects [358]. Several adaptations of this process convert Point Cloud data to a bird's eye view before using CNN to make a classification [359]. Although some success was attained, information can be lost in the transformation, and the accuracy of the network suffered [360], [361].

A DL architecture called PointNet was proposed in [1], [2] to process Point Cloud data. During training, PointNet takes unordered data points as a set of functions and maps a point set onto a vector. After training PointNet has learned the vector representation of the different classes. When making a classification, the vector representation is checked against the patterns it identified in the dataset during training, While retaining all the data during classification and being suitable for object classification, the network lacks an understanding of the relationships between points. The advantage of PointNet is an end-to-end network that

can process 3D data without translating it to a complex representation other than the simple vector [1]. Depicted in Figure 46, PointNet is good at making a classification, it is void of a mechanism to facilitate detection and localization. Moreover, the process is limited by available memory, considering the high computational cost of large Point Clouds.

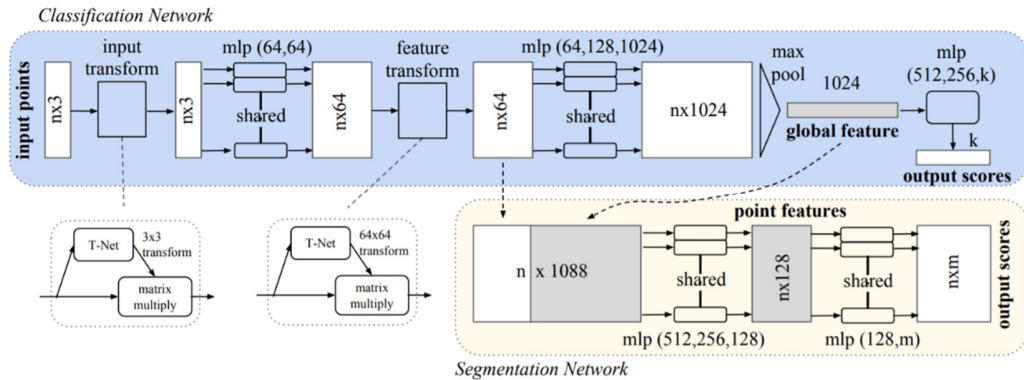


Figure 46: Shows the PointNet Architecture. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net to facilitate the segmentation the components that construct the object its classifying [1].

Recently, research on rasterizing Point Cloud data into a 3D voxel grid [362], where authors proposed the voxelization of a Point Cloud data for region proposal object detection. A voxel – synonymous with a pixel in RGB images – is a digital representation of a unit of volume. VoxelNet used a feature learning network to partition a volume occupied by 3D data into voxels. Points within each voxel are then translated into vector representation of the 3D object. The convolutional middle layers of VoxelNet, aggregate the Point Cloud representation before an RPN generates a 3D bounding box around the object. While VoxelNet returns encouraging results, this approach suffers from a trade-off between its computational cost and approximation accuracy, as the size of voxels dictates accuracy making it useless for tasks that require granular analysis [362].

A different approach proposed in [353] uses a hybrid that combines a discrete grid structure with a modified Fisher Vector (FV). Typically, the FV represents the rate of curvature of the normal distribution. If a probability density function is used to model the rate of change, the gradient of the likelihood with respect to the elements that represent the image can be computed. [353] proposed a classification pipeline constructed of two main components. The first component converts the 3D data to a modified FV representation, and the second process it using a CNN. Counter to conventional wisdom – where a feature learning network uses raw data to derive a pattern – a vector that represents the image is learned.

5.2.2 Motivation and Context

Having a system that can recognize human activity without the need for individual sensors dramatically improves reliability and increase the chances of this research becoming ubiquitous in AV infrastructure. For this to occur the sensors used in identifying HAR need to be the same as the ones found in AV's. Furthermore, if one wishes to develop an algorithm for what can be regarded as a critical cognitive skill, we need to do so with reliability as one of our primary objectives. To that end, the proposed framework is only made possible through the availability of the multimodal LboroLdn HAR dataset [279].

Multimodal ML has found a home in many disciplines: computational linguistics to assisted living and AV. Possibly one of the earliest applications of Multimodal ML was AVSR in [363]. Motivated by the association between vision and sound, researchers in [363] studied the McGurk effect. The McGurk effect demonstrates the relationship between hearing, vision and speech perception. For example, if a person is getting inadequate sound data but is receiving good quality visual data, they can, on occasion, decipher a third sound or a combination of the visual and audio signals. McGurk [363] demonstrated this by playing the sound “ba–ba” while showing a person saying “ma–ma”. The subject being tested than perceived a third “da–da”. Originally AVSR was intended to improve the speech recognition ability of Hidden Markov Models [364]. However, the current popularity in DL has shown real advantages with low signal to noise ratio, where visual information is used to identify noisy signals correctly [365].

One of the most prolific applications of Multimodal ML is in the area of ambient assisted living centres, hospitals and rehabilitation centres – especially elderly care units [366]. In this area of research, the ML algorithm is designed to determine what situations require urgent medical assistance using multiple signals to determine a change in health. The intention is to differentiate between typical day-to-day activities and a drift away from homeostasis [367]. In this case, the event is identified using Multimodal ML by combining multiple biometrics to deduce what has just occurred.

Research work that is closely related to Multimodal ML for HAR is recognizing pedestrian activities. Applying this to AV could decrease road accidents. [263] and [264] provide an interesting application of HAR for a pedestrian recognition system that matches the pedestrians predicted intention with the driver's direction. Depending on the actions, the vehicle brake is initiated to avoid a collision. Action prediction is ‘before the fact event’. It supersedes recognition. Referred to as the HAP, where ML algorithms recognize a class from

an incomplete or changing action [265]–[267]. This is quite different from action recognition, where ML algorithms expect to see a set of action dynamics.

5.3 Point Cloud Representation Learning for Activity Recognition

This section presents the proposed method for Multimodal HAR, which is based upon an advanced CNN architecture. As illustrated in Figure 47, the proposed framework utilizes a robust Object Detector, Faster Region Convolutional Neural Network (Faster-R-CNN), to identify an ROI and a 3D classifier – 3DmFV Net – to identify the activity. It is understood that the framework can perform multi subjects activity recognition. However, it has not been tested for this purpose. The corresponding ROI is translated and aligned to the Point Cloud before the ROI on the 3D-LiDAR data is classified into the activity being performed.

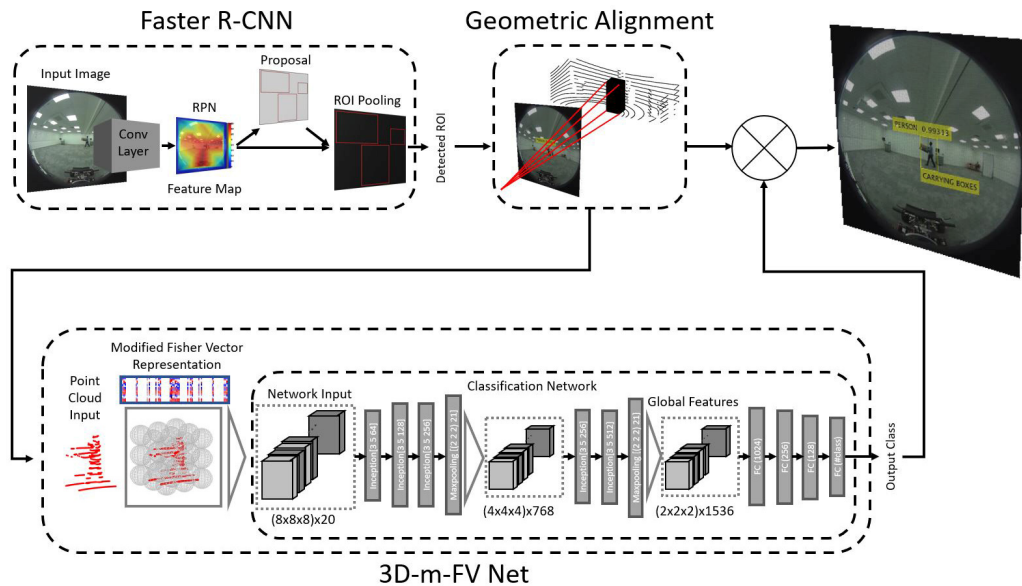


Figure 47: The proposed HAR architecture. They are depicting all the elements in the Multimodal Fisher Vector Network. We first leverage the power of a CNN to detector objects and propose an ROI. The geometrically aligned and translated Point Cloud data allows us to identify the corresponding 3D ROI in the 3D sample. This segmented ROI is converted into a modified FV representation before being passed onto the classification network. The resultant class is overlaid on the object detector image proposed by the object detector network.

5.3.1 Object Detection Network

The principal component of the proposed architecture is an Object Detector that extracts an ROI from an input RGB image. An Object Detector is a classifier that also locates the region where the target class resides. The ResNet [148] was modified by training an RPN to extract ROI's from images before classifying the data contained in that region. This process is commonly referred to as a Faster-R-CNN [352]. The Faster-R-CNN gets its name by

performing the convolution operation once per region before generating the feature map for that data. The Faster-R-CNN Object Detector is an extension of the R-CNN and the Fast-R-CNN. The main difference between the three is the use of an RPN by a Faster-R-CNN, versus the use of a selective search for generating ROI by the Fast-R-CNN and the R-CNN. And the use of a dynamic selective search by the Fast-R-CNN versus a fixed selective search by the R-CNN [352], [368]. The base classification network modified for Object Detector was ResNet-50 trained on the ImageNet dataset [369]. As is the case with most supervised ML methods are sensitive to lighting conditions and have difficulties generalising – because of the infinitely different classes they encounter. To address this issue, we finetuned ResNet-50 using a subset of RGB images only from the LboroLdnHAR [279] dataset.

The detector was validated using an IoU between the ground truth bounding boxes and the predicted ROI. IoU detects the difference between the annotated bounding boxes of the LboroLdnHAR dataset RGB images and the proposed ROI. Using pairs of anchors with sizes {30,19;60,38;120,76}, the ROI was labelled positive (object of interest present) when accuracy was higher than 0.65, and negative (object of interest, not present) when less than 0.35. In this case, the accuracy that the classifier returns for that region behave as a threshold for the detector. It determines whether the features being observed in that region can be classified as an object of interest or not. During training, the shortest side of all images scaled to 246 pixels. Trained using stochastic gradient descent with a learning rate of 0.0001 and a momentum of 0.9, the R-CNN network used a patch size of 16×16 pixels.

5.3.2 Geometric Alignment of Sensor Data

To facilitate the classification of Point Cloud data, the ROI identified by the object detector was translated and aligned to the corresponding area in the Point Cloud sample. This process of geometric alignment requires knowledge about the proximity, orientation, and modality of the sensor types. Attained, through the alignment of Kinect sensor data to LiDAR sensor data, before transforming the resulting point cloud sample (Kinect and LiDAR Data combined) to the camera data. This two-step process facilitates the segmentation of sample data and the labelling of class activities. A most useful tool when building a dataset, it should be noted that translation does not occur between the Kinect and LiDAR, but rather the point cloud sample and the data captured by the camera. While possible to use the inbuilt camera in the Kinect sensor, it could not capture 360° images and therefore the valuable context information we desire.

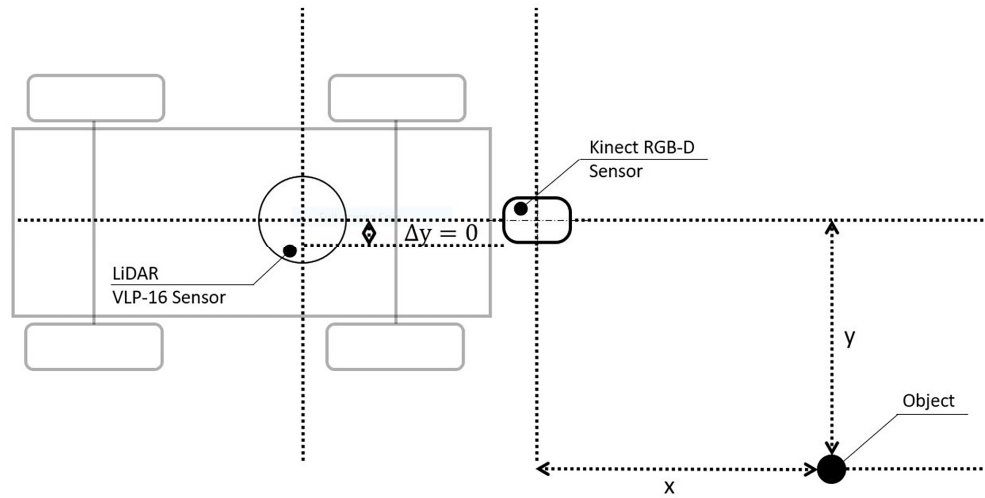


Figure 48: Illustrates the plan view of the setup showing the position of the Kinect and LiDAR sensors relative to the autonomous platform.

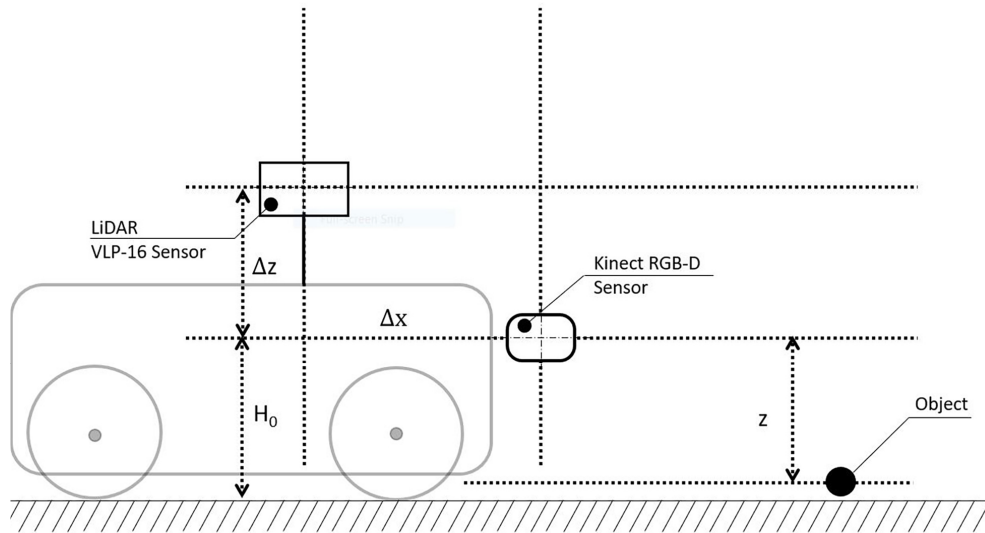


Figure 49: Illustrates the elevation view of the setup showing the position of the Kinect and LiDAR sensors relative to the autonomous platform.

This process can be explained if we consider a point in space represented by 3D coordinate. In the first stage, we shift the point's origin from one location to another using a scalar (Kinect data and LiDAR data alignment). Note, at this point; we are only concerned with the RGB-D and Point Cloud data. Therefore, the plan and elevated view in Figure 48 and Figure 49 only show the location of the Kinect and LiDAR sensor. For this example, consider an object O 4.76 m (x) in front of the AV, 2.75 m (y) to the right, and 0.9 m (z) below the horizontal axis of the Kinect sensor. If we know that the Kinect sensor is offset -0.45 m on the (Δx) axis, 0 m on the (Δy) axis, and 0.5 on the (Δz) axis – it is merely a matter of adding the scalar to corresponding coordinates for alignment.

$$X_T = \Delta x + x \tag{Equation 17}$$

$$Y_T = \Delta y + y \tag{Equation 18}$$

$$Z_T = \Delta z + z \tag{Equation 19}$$

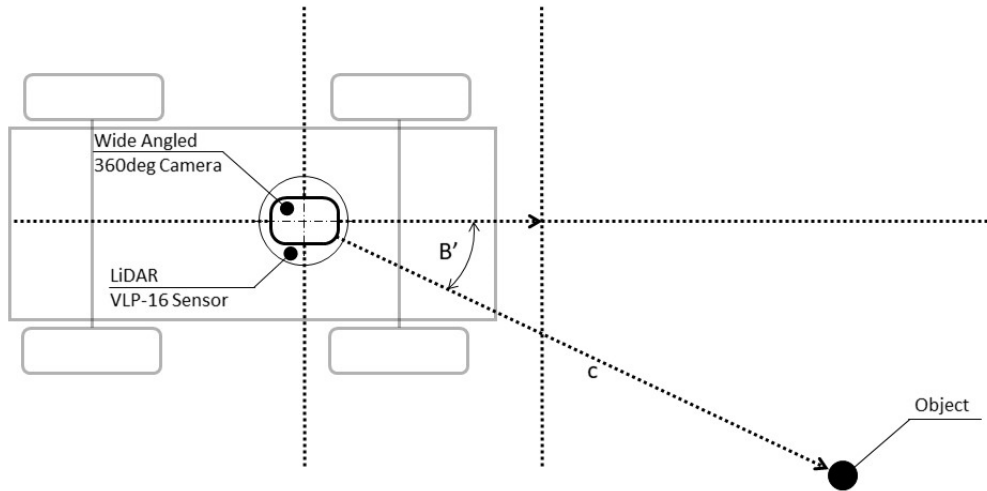


Figure 50: Illustrates the plan view of the setup showing the position of the RGB and LiDAR sensors relative to the LboroLdn autonomous testbed.

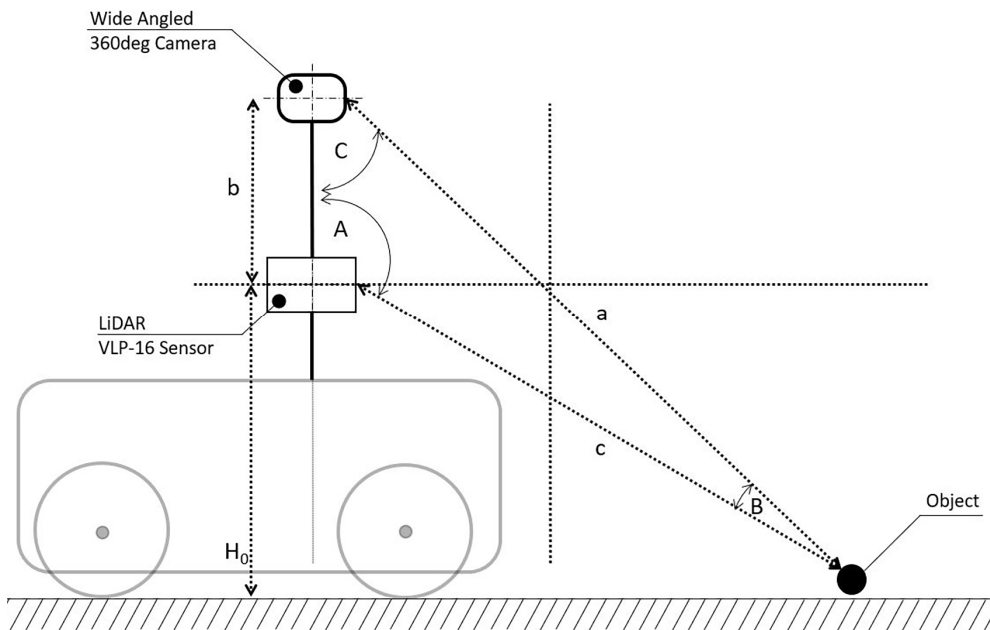


Figure 51: Illustrates the elevated view of the setup showing the position of the RGB and LiDAR sensors relative to the LboroLdn autonomous testbed.

In the second stage of the process, each corresponding point cloud sample of aligned LiDAR and Kinect data is transformed to match each pixel output by the RGB sensor. Figure 50 illustrates the plan view of the sensor setup, while Figure 51 illustrates the elevated view of the sensor setup. At this stage, we are only concerned with the translation of the point cloud sample to the camera data. This process is similar to the first stage but somewhat more complex, as the data types are not comparable.

For this derivation, consider an object O at a distance (c) of 5.6 m from the LiDAR sensor. In this case, the object (O) is identified at an azimuth angle (B') of 3° and a zenith angle of 99° . The Vertical (b) distance between the LiDAR and the RGB sensor is 27 cm, and the RGB sensor is positioned (H_0) 122 cm above the ground. Considering the distance c between the object O and the RGB sensor, we can describe B – the RGB azimuth angle.

$$B = \sin^{-1}\left(\frac{\sin(A) \times b}{a}\right) \Rightarrow \sin^{-1}\left(\frac{\sin(99^\circ) \times 0.27}{5.62}\right) = 2.7^\circ \quad (\text{Equation 20})$$

In this case, because the RGB and LiDAR sensors are on the same longitudinal axis. The RGB zenith angle C between the object O and the RGB sensor can be described as:

$$C = 180 - A - B \Rightarrow 180 - 99 - 2.72 = 78.2^\circ \quad (\text{Equation 21})$$

Furthermore, the distance c between object O and the RGB sensor described as:

$$c = \frac{\sin(C) \times a}{\sin(A)} \Rightarrow \frac{\sin(78.28) \times 5.62}{\sin(99^\circ)} = 5.5m \quad (\text{Equation 22})$$

The purpose of this process is to translate and align one data stream to another. It is assumed that the longitudinal axis of the three sensors are aligned; however, an offset can be accounted for, if necessary. The effectiveness of translation and alignment procedures were visually compared. Figure 52 Scenario 1 (a) shows a subject is carrying a box overlaid with the translated and aligned LiDAR data and the ROI identified by the 2D image detector. Figure 52 Scenario 1 (b) shows the Point Cloud data and the ROI identified by the 2D image detector – denoted by a blue cuboid. Inevitably, the 3D ROI includes some unwanted artefacts. For example, in the foreground of Figure 52 Scenario 1 (b), there are some unwanted surface features included in the sample. Towards the rear of the 3D sample, an unwanted portion of the wall is included in the segmented point cloud data. With the proposed framework, it is expected that there is going to be some unwanted artefacts processed by the network. While it

is possible to filter the data using a moving average, some artefacts are going to remain. However, through accurate geometric alignment, the artefacts can be further minimised.

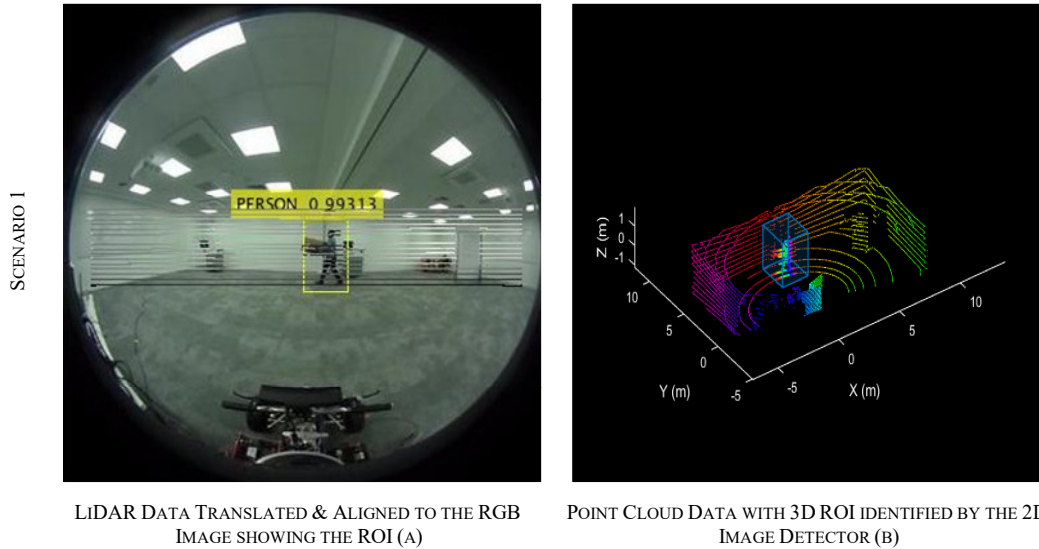


Figure 52: (a): A Scenario showing the subject carrying a box. Overlay with the 2D ROI, the translated and aligned Point Cloud data. (b) Shows the Point Cloud data with corresponding 3D ROI overlay.

5.3.3 3DmFV Classification Network

There are distinct differences between 2D RGB data and 3D data. 2D RGB data is structured and organized, whereas 3D data is unstructured, unorganized, prone to noise, and missing points. Unlike 2D RGB data, 3D data does not contain valuable descriptors, like HOG or HSV, that are frequently used by feature learning networks. While the positional relationship between coordinates is a descriptor present in 3D data, it is different from the positional relationship in 2D images. For example, a CNN will assume some sense of locality, where pixels in the image are related, and therefore have similar colour, lighting and texture. In point cloud data, they are near to each other and do not share any relationship in colour, lighting and texture. By its nature, 3D data cannot provide the structure and organization that CNN requires to make a prediction. Void of valuable descriptors, structure, and order; 3D data needs to be modified before a CNN can classify it. The proposed method for classifying Point Cloud data consists of two main modules. The first module converts 3D data to the modified FV representation. The second module processes the modified FV representation in CNN before making a classification. An in-depth description of the 3DmFV network can be found in [353].

A. Fisher Vectors

Intraclass variability describes variations in appearance between two views of the same class or variations in appearance between two instances of the same class. In HAR, for example, when viewing a person from the front, it is difficult to determine if they are walking with a phone in hand or no phone at all. Moreover, variations in appearance between two instances can occur, for example, during different phases of an individual’s gait.

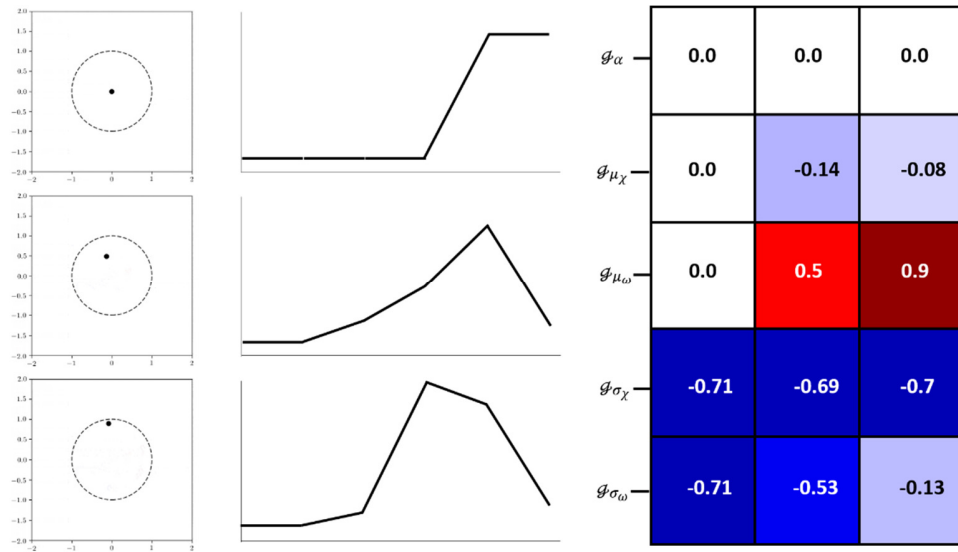


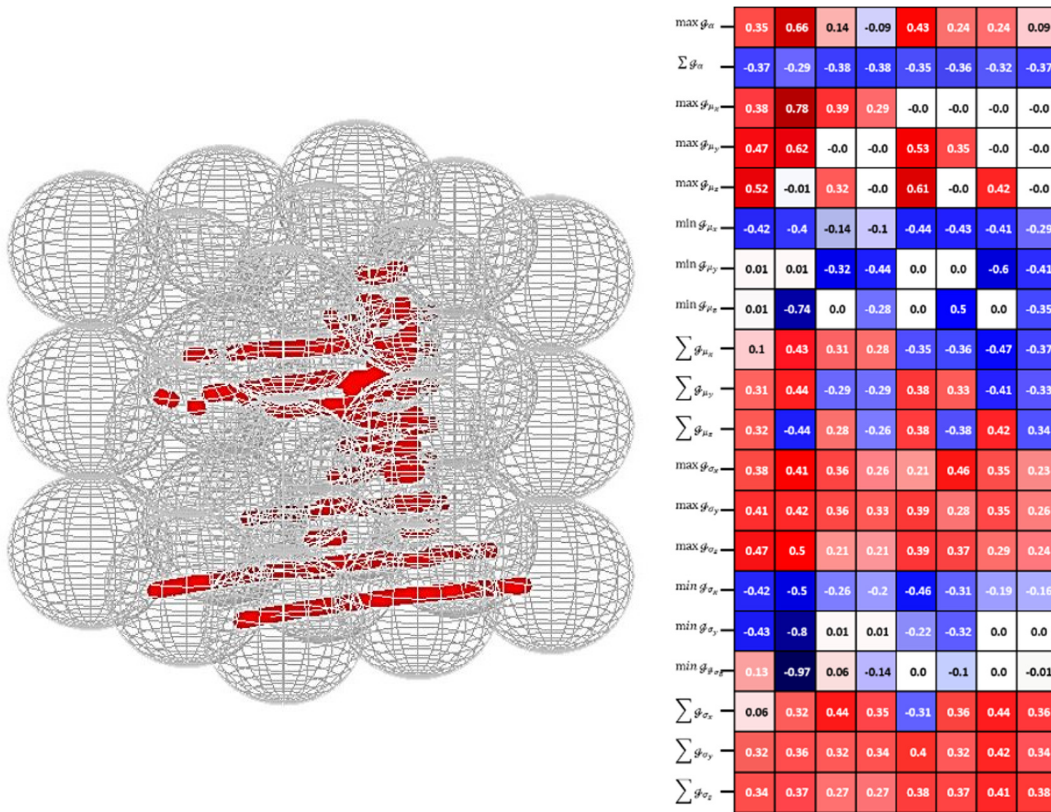
Figure 53: Shows three 2D points superimposed on a 2D Gaussian on the left of the image. In the centre of the image is the FV, and on the right is the FV representation for the three points. The vector in the centre of the image indicates the rate of curvature for the data.

FVs are a way of dealing with intraclass variability and determine the likelihood of getting specific data given an underlying theory. A 2D representation of the FV is shown in Figure 53. Formally, it is the expected value of the observed information. Under ideal circumstances, this can best be described as having a probability distribution with a very sharp peak. Conversely, when the distribution curve is very broad, there is a high likelihood over a large range of points.

By using a GMM, the rate of curvature can be quantified by looking at the probable distribution of the different points. Often referred to as the curvature vector, it describes how curved the function is around the maximum. In simpler terms, the more significant the FV, the more curved the distribution is, meaning, the more constrained the data is for that scenario.

Overlaying spherical Gaussians on a coarse grid provides structure, size, and the foundation for a representation of the image. By computing each point derivative, with respect to the Gaussian parameters, the results can be aggregated using three symmetric functions –

Maximum, Minimum, and Summation. The FV representation is the aggregated values. It has a constant size making it invariant to permutations for a different number of points. For multiple points, we can superimpose several Gaussian onto the grid resulting in a statistically unique fingerprint of the 3D data [370]. Figure 54 (a) depicts the spherical Gaussians superimposed on the Point Cloud data, and Figure 54 (b) shows the modified FV representation. It should be noted that the 3DmFV representation in Figure 54 (b) is for visualization purposes only, concretely it is a 4-Dimensional array that is fed into the CNN.



(A) SPHERICAL GAUSSIAN SUPERIMPOSED ON POINT CLOUD DATA

(B) FISHER VECTOR REPRESENTATION

Figure 54: (a) Shows the spherical Gaussians superimposed on the Point Cloud data. (b) Shows the modified FV representation of the GMM for the Point Cloud data

B. Classification of Fisher Vector Representation

The main parts of the 3DmFV network used to classify the representations comprise of several inception modules, max-pooling layers, and four fully connected convolutional layers [371], [372]. Training is done using backpropagation, a standard softmax, and cross-entropy loss with batch normalization. Dropout is placed on the fully connected layers to prevent excessive co-adapting and overfitting during training. Dropout refers to a technique of ignoring specific neurons response during a forward or backward pass after a time. Between

the last max-pooling layer and the first fully connected layer, the network has approximately 4.6 million trained parameters.

Through the network, several inception modules were used. The overall objective of the inception module is to overcome the dimensionality of the multiplier effect. An Inception module allows the use of several filters in a single module [372]. While this will make the network architecture broader and more complex, the process works remarkably well because of what it is the researchers are trying to achieve.

The penultimate element of the 3dmFV network is a max-pooling layer. This layer reduces the dimensionality of the Inception module further. They work bypassing all output elements of the Inception module through an $n \times n$ filter. The $n \times n$ filter passes over the data with a fixed stride. Taking the first $n \times n$ region, the max value for that region is calculated before passing it on to the next. This process is repeated using the stride value to shift the max-pooling filter over in increments, passing the filter over all elements of the data.

The final component in the 3dmFV network is four fully connected layers. The output of the previous layers is flattened into a single vector of values. These represent a probability that a feature belongs to a specific class. For example, in HAR, when a person is running, the fully connected layers should identify features representing the action in a single frame with a high probability.

5.4 Results and Discussion

A modified FV representation network classifies the probable distribution of points around a spherical Gaussian. Concurrently the relationship between 3D points is accounted for in the probable distribution, and similarly in the modified FV representation. Other Point Cloud classifiers do not account for the relationship between points, but rather the location of points in space or as a volume. This is the reason a modified FV representation network for HAR was chosen. This chapter reports on the performance of the proposed framework and an alternative Point Cloud classifier – PointNet. Performance of the object detector network, 3D classification network, and PointNet classification networks were scrutinized using the LboroLdnHAR dataset. Using metrics such as average precision, recall, and F-score, the researcher evaluated whether the networks perform as desired and, therefore, suitable for HAR. At the end of this chapter, the visible results of the proposed framework were presented alongside discussions on some of the limitations and benefits of the network.

5.4.1 Dataset

Existing HAR techniques using image data are sensitive to viewpoint variations. This is a consequence of extracting features from viewpoint dependent images. In contrast, the proposed framework mines 2D images to identify a 3D region in a Point Cloud scan. Since the proposed framework classifies viewpoint independent 3D data, this technique is completely insensitive to viewpoint variations, as long as the subject can be identified in the image.

The proposed MFV Network was evaluated on the LboroLdnHAR dataset [279]. The LboroLdnHAR dataset consists of 6712-Point Cloud, RGB-D, and RGB annotated samples. The dataset was split into three subsets: RGB data for detection, RGB-D, and Point Cloud data for classification and one sample from each class for demonstration. The first subset contained RGB samples and consisted of images with annotated ground truth ROI labels. This subset was used to train and validate the 2D image detection network. The second subset contained annotated classes of aligned Kinect and LiDAR data. It was used to train and validate the Point Cloud classification network. The final subset was a small number of aligned and translated samples not used during the training or validating of the proposed network. This was utilized in the results section of this paper to demonstrate the ability of the proposed network.

The LboroLdnHAR dataset contains data of typical human activities in indoor environments. Data was captured by three sensors – LiDAR, RGB-D, and RGB. The 9 activities performed by 16 participants were chosen based on practical activities performed in an office environment. The activities were: carrying boxes, lying down, pushing a board, running, sitting on a chair, sitting on a stool, standing while texting, walking, and walking while texting.

The focus was indoor activities where humans have a limited attention span, resulting in scenarios where it is highly likely that humans would not pay attention to an AV. The mean period for each data capture was approximately 35 seconds. Subjects started and ended the data capture periods with a T pose – standing upright with arms outstretched. During the experiments, the RGB 360° camera, RGB-D Kinect V2, and VLP-16 LiDAR sensor logged data at 30 fps, 30 fps, and 6 fps, respectively.

5.4.2 MFV Net Performance

Comparative evaluation describes a mechanism where the performance of the proposed process is evaluated in a comparative framework. It is not always clear how to do this, as what works for one system will not necessarily work for another. Typically in object

detection and classification tasks, the recall, mean average precision, and F-score are frequently used metrics to evaluate the benefits of one framework over another [352], [373], [374].

The average precision uses Precision and Recall for ranked retrieval results. The concept of recall and precision stems from the rate of change in the true positives and false-positive results. Like the average precision, the F-score considers both the precision and the recall of the detector to determine the advantage of one system over another. These metrics were used to identify the benefits of the detection network and the optimal Point Cloud classification network for HAR.

A. 2D Detection Network Performance

The detector was trained and evaluated on the first subset of the LboroLdnHAR dataset [279]. To measure the detector's performance, the research focused on the average precision, shown in Figure 55, the log-average miss rates, shown in Figure 56, and the F-score detailed in Table 17.

TABLE 17: 2D DETECTION NETWORK PERFORMANCE

Evaluation Method	Person
Precision	0.95
Recall	0.96
Log-Average Miss Rate	0.1
F-Score	0.95

The average precision is a method of evaluation that incorporates the ability of the detector to make correct classifications, i.e., precision, and the ability of the detector to find all relevant objects, i.e., recall. The log-average miss rate was attained by varying the thresholds on the detector confidence prediction and by measuring the rate of change in the true and false positives. The log-average miss rate returns the results of the Object Detector compared to the ground truth table. This metric is another method of measuring the performance of the Object Detector [375], [376]. In this case, the research found that the average precision and log-average miss rate for the detector was 0.95 and 0.1, respectively. The F-score considers both the precision and the recall of the detector. Using binary classification, which separates two elements of a dataset into distinct groups, in conjunction with the F-score, allows the accuracy of the framework to be determined [377]. Defined as two times the precision times the recall over the precision plus the recall. The F-score reaches its best value at 1 and worst at 0. For the object detector portion of the framework, the F-score, reported in Table 10, was determined as 0.95.

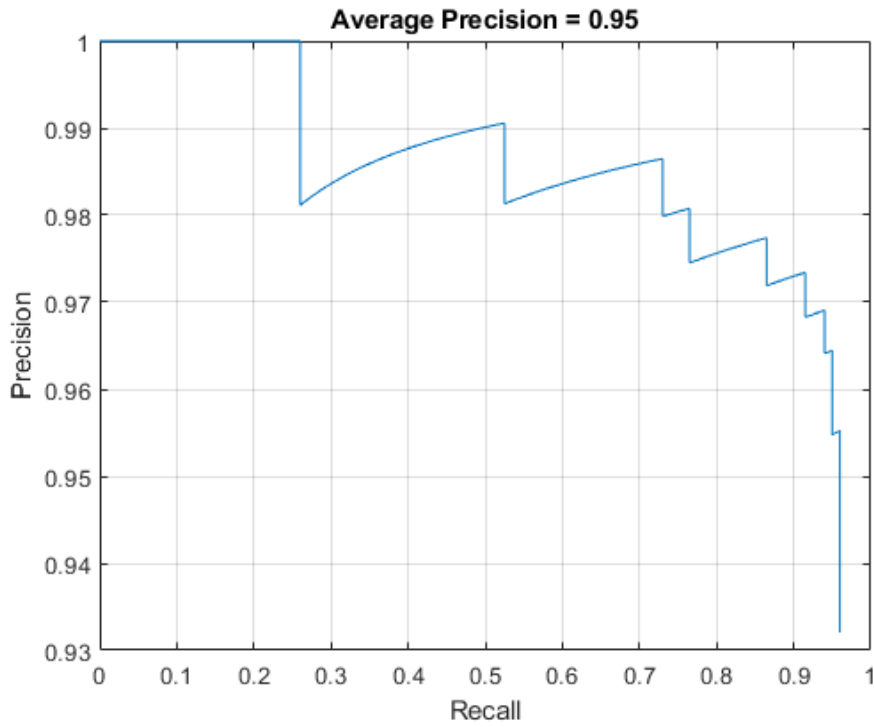


Figure 55: The average precision of the object detector portion of the proposed framework showing the trade-off between precision and recall. A high area under the curve represents both high recall and high precision – as we can see here

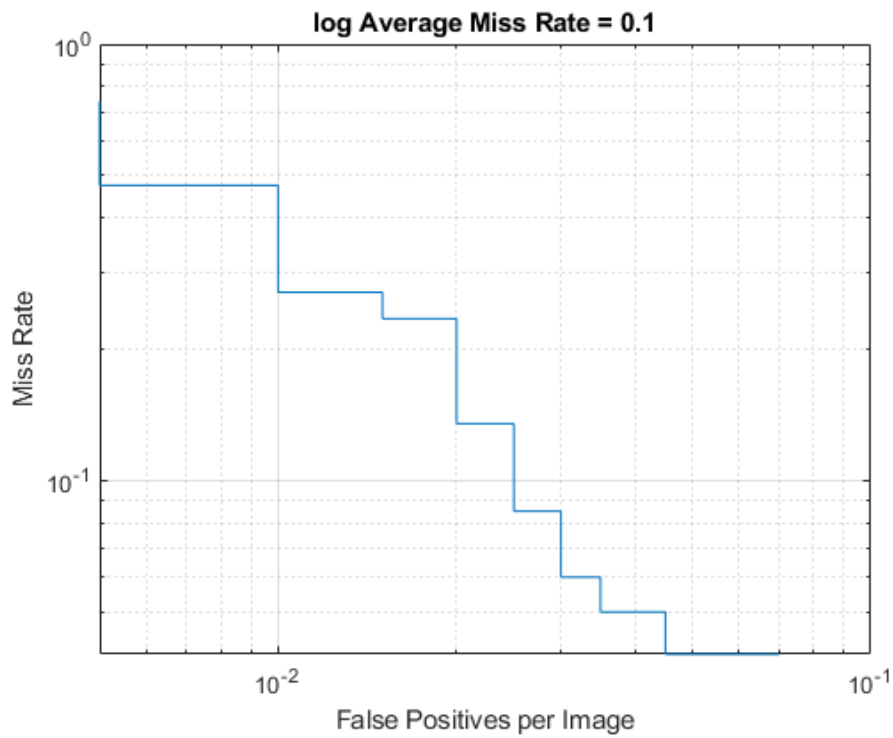


Figure 56: The log-average miss rate of the object detector portion of the proposed framework indicating the quality of detection of the object detector.

B. 3D Classification Network Performance

Table 11 reports on the average score for each class of 3DmFV-Net when tested with the LboroLdnHAR dataset [279]. To measure the classifier's performance, we focused on the precision, recall, and the F-score of each class. When scrutinizing the performance, for the most part, it was found that the higher the instances of each class, the more significant the F-score per class.

TABLE 18: 3D CLASSIFICATION NETWORK PERFORMANCE

	Carrying Boxes	Lying Down	Pushing A Board	Running	Sitting On A Chair	Sitting On A Stool	Standing While Texting	Walking	Walking While Texting
Precision	1	1	0.988	0.726	0.881	0.968	0.9134	0.722	0.864
Recall	0.959	1	1	0.688	0.890	0.957	0.989	0.776	0.795
F-score	0.979	1	0.994	0.706	0.885	0.962	0.95	0.748	0.828
Sample	74	103	86	77	100	95	96	67	88

Figure 57 shows the confusion matrix for the classifier. On the Y-axis are the Output Class, and on the X-axis are the Target Class. The diagonal cells, dividing either side of the matrix, indicate true positives that are correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified. The overall accuracy for the classifier was 0.903. The Precision and Recall were 0.895 and 0.894, respectively.

Of all the activities, the 3D classification network had the greatest difficulty in identifying the class titled Running, and the greatest success in identifying the classes titled Sitting On A Chair and Laying Down, respectively. Unsurprisingly, Running was misclassified as the class titled Walking 14 times, thus reducing the Walking class accuracy and that of the overall network. Similarly, the class titled Walking While Texting was misclassified as the classes titled Walking and Running a total of 6 and 10 times, respectively. It can be assumed that the reason for the miss classification is because of the similarities between the human silhouette. It should be noted, that while the network had difficulties in identifying some of the classes that shared similarities between silhouette, it still performed exceptionally. In fact, 5 of the activities (Carrying Boxes, Laying Down, Pushing A Board, Sitting On A Stool, Standing While Texting) were classified with an accuracy over 90%, 2 activities (Sitting On A Chair, Walking While Texting) were classified with an accuracy over 80% and the remaining 2 activities (Running, Walking) were classified with an accuracy over 70%.

Confusion Matrix

Output Class	CARRYING BOXES	71 9.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	LYING DOWN	0 0.0%	103 13.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	PUSHING A BOARD	0 0.0%	0 0.0%	86 10.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	98.9% 1.1%
	RUNNING	0 0.0%	0 0.0%	0 0.0%	53 6.7%	2 0.3%	0 0.0%	0 0.0%	8 1.0%	10 1.3%	72.6% 27.4%
	SITTING ON A CHAIR	3 0.4%	0 0.0%	0 0.0%	5 0.6%	89 11.3%	3 0.4%	0 0.0%	0 0.0%	1 0.1%	88.1% 11.9%
	SITTING ON A STOOL	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	91 11.6%	1 0.1%	0 0.0%	1 0.1%	96.8% 3.2%
	STANDING WHILE TEXTING	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 1.0%	1 0.1%	95 12.1%	0 0.0%	0 0.0%	91.3% 8.7%
	WALKING	0 0.0%	0 0.0%	0 0.0%	14 1.8%	0 0.0%	0 0.0%	0 0.0%	52 6.6%	6 0.8%	72.2% 27.8%
	WALKING WHILE TEXTING	0 0.0%	0 0.0%	0 0.0%	5 0.6%	0 0.0%	0 0.0%	0 0.0%	6 0.8%	70 8.9%	86.4% 13.6%
			95.9% 4.1%	100% 0.0%	100% 0.0%	68.8% 31.2%	89.0% 11.0%	95.8% 4.2%	99.0% 1.0%	77.6% 22.4%	79.5% 20.5%
		Target Class									
		CARRYING BOXES	LYING DOWN	PUSHING A BOARD	RUNNING	SITTING ON A CHAIR	SITTING ON A STOOL	STANDING WHILE TEXTING	WALKING	WALKING WHILE TEXTING	

Figure 57: Confusion matrix for the 3D classifier portion of the proposed framework. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified.

5.4.3 PointNet Classification Network Performance

To benchmark, the performance of the proposed method, PointNet was trained and tested with the LboraLdnHAR dataset [279]. Table 19 reports on the average score for each class from PointNet. As with the performance of the 3D Classification Network, we focused on the precision, recall, and F-score of each class. It should be noted that the same process for determining the ROI was used to identify the corresponding 3D ROI. This data was fed into PointNet, and a prediction was made. The same split of the data was used for both classification methods.

Figure 58 shows the confusion matrix for PointNet. On the Y-axis are the Output Class, and on the X-axis are the Target Class. The overall accuracy for the classifier was 0.098. The Precision and Recall were 0.111 and 0.147, respectively. Of interest was the network's inability to classify subjects lying down, sitting on a chair, and sitting on a stool. While the network was able to make a prediction, it performed extremely poorly when compared to the 3D

classification network discussed in the previous section, so much so that it was deemed unusable for HAR.

TABLE 19: POINTNET CLASSIFICATION NETWORK PERFORMANCE

	Carrying Boxes	Lying Down	Pushing A Board	Running	Sitting On A Chair	Sitting On A Stool	Standing While Texting	Walking	Walking While Texting
Precision	0.013	0.057	0.87	0	0.059	0	0	0	0
Recall	0.019	0.714	0.11	0	0.038	0	NaN	NaN	NaN
F-score	0.016	0.106	0.20	0	0.046	0	0	0	0
Sample	74	103	86	77	100	95	96	67	88

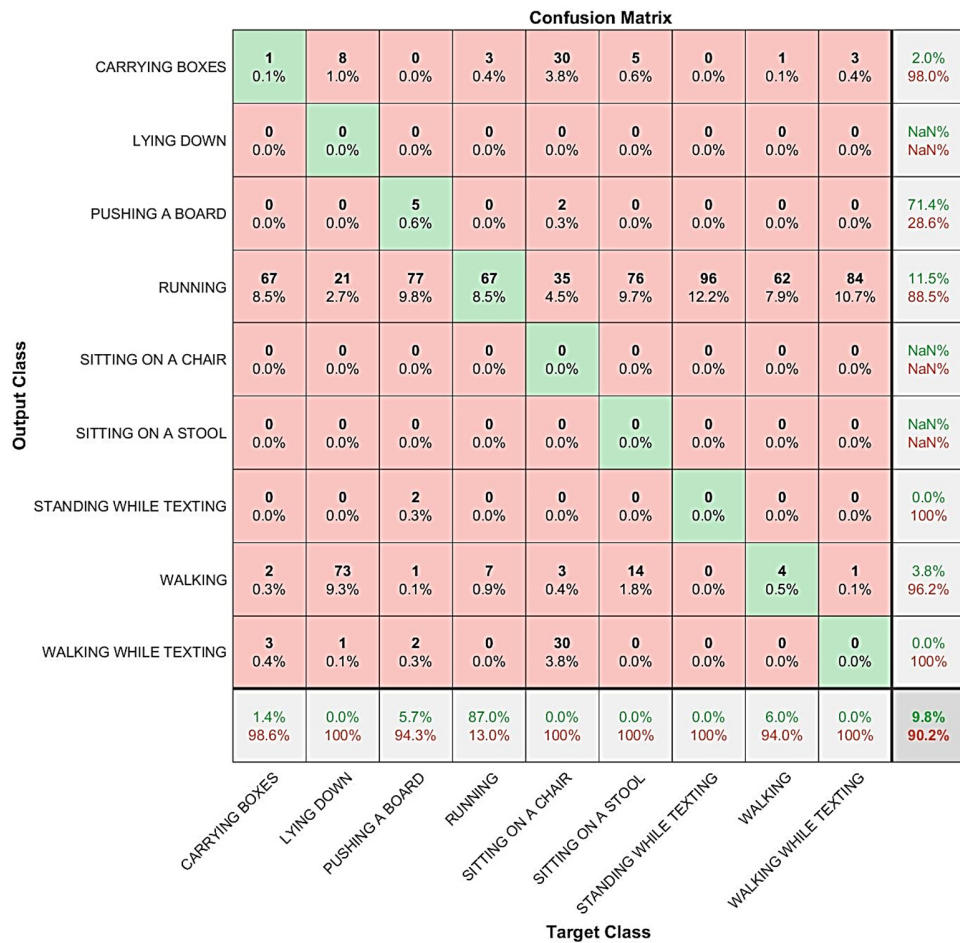


Figure 58: Confusion matrix for PointNet. The diagonal cells indicate true positives correctly classified. The off-diagonal cells indicate false positives that are incorrectly classified.

5.4.4 Visual Results

The purpose of the proposed algorithm is to assist an AV with HAR. The results displayed here were a subset taken from the LboroLdnHAR dataset but not used during training or validation of the network [279]. Samples from the LboroLdnHAR dataset [279]

were captured while the AV was stationary. All samples in the LboroLdnHAR dataset [279] were performed indoor under controlled lighting. The results presented in Figure 59 Scenario 1 (c), Scenario 2 (c), Scenario 3 (c), and Scenario 4 (c) indicate that the proposed MfV Net performed as desired. In all cases, the subject performing the activity was detected, segmented, and classified correctly. The output image in Figure 59 Scenario 1 (c), Scenario 2 (c), Scenario 3 (c), and Scenario 4 (c) depicts the fused results of the different feature learning networks. In line with the five core technical challenges – Representation, Translation, Alignment, Fusion, and Co-Learning – this research shows how Multimodal ML can be successfully applied to HAR.

In Figure 59 Scenario 1 (a), the image shows a subject carrying a box. The first part of the proposed framework, the Object Detector, identifies, with a high degree of confidence (99%), the location and region the person occupies. Figure 59 Scenario 1 (b) shows the 3D ROI identified by the Object Detector. Although there are unwanted artefacts located in the 3D ROI, the activity being performed by the subject is correctly classified, as shown in Figure 59 Scenario 1 (c).

In Figure 59 Scenario 1 (a), there is a person in the background. In this case, the Object Detector did not function as desired. This can be attributed to the design of the Faster-R-CNN network, where input images with an aspect ratio of 360×360 are reduced in size before processed. When processing images, the Faster-R-CNN extracts features within the first few layers. When dealing with small objects in small images, the feature can, in effect, disappear in the middle of the network. In this case, the person in the background is not detected, and therefore, their activity is never classified in the latter part of the network. Table 20 details the scenarios depicted in Figure 59.

Similar but somewhat different issues occur in Figure 59, Scenario 2 (a). In this image, the subject is standing while texting. The Object Detector correctly identifies the ROI for the subject but misses the person behind the computer due to occlusion. Contrary to this, the subject of interest was detected correctly, and the corresponding 3D ROI was extracted from the 3D data. Of interest is Figure 59, Scenario 2 (b). In this image, the subject is standing upright with their hands in front clasping a phone. This ROI is passed onto the classifier component of the proposed network before the correct result is overlaid onto the image shown in Figure 59 Scenario 2 (c).

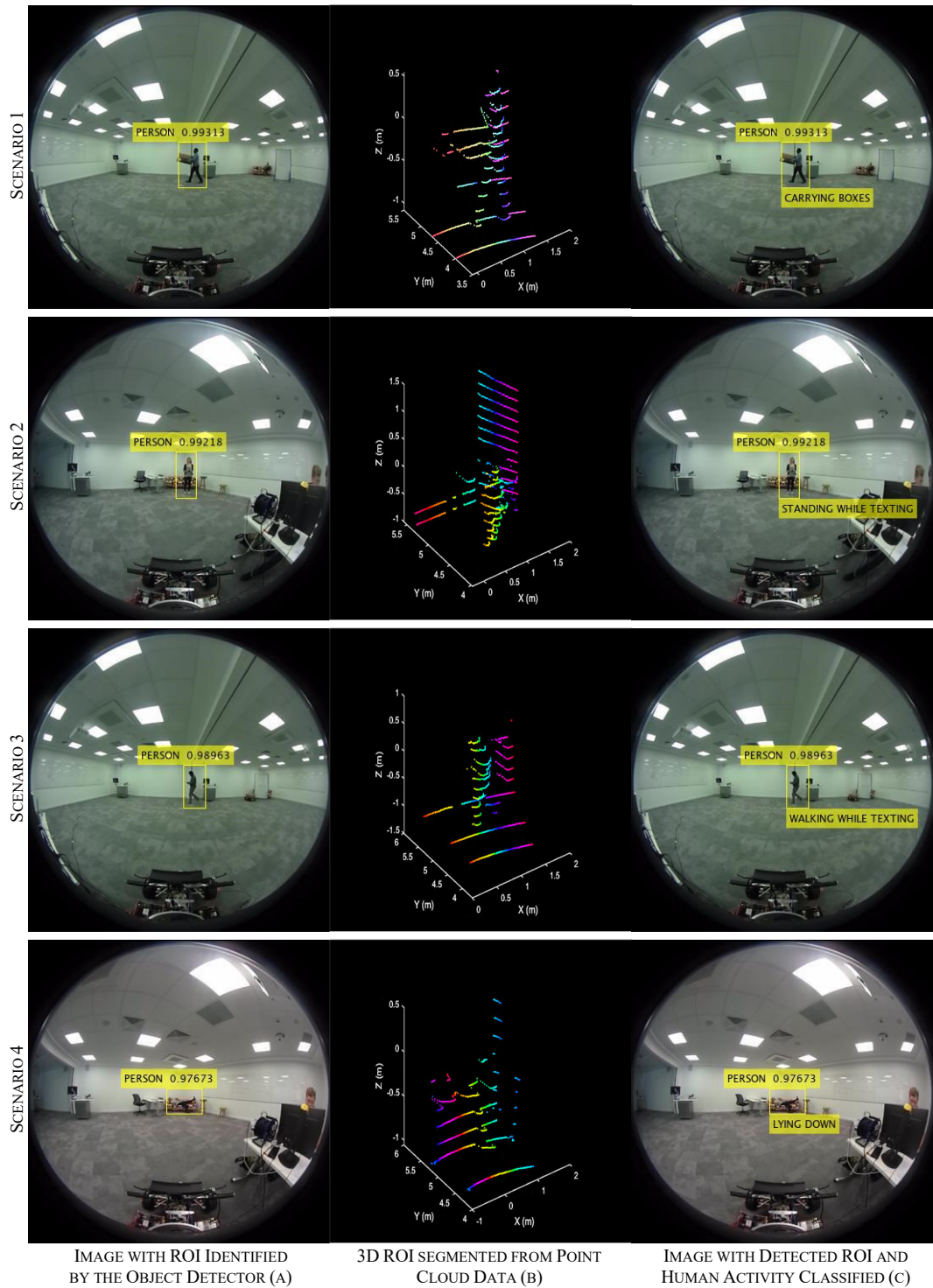


Figure 59: A subset of the data used to assess the visual performance of the proposed MfV Net. From the top: Scenario 1, Scenario 2, Scenario 3 and Scenario 4. Table 19 details the scenarios depicted in Figure 59. Images in the first column indicate the output of the Object Detector. The centre column shows the segmented Point Cloud data. The final column shows the output of the network and the associated class identified for the activity being performed.

The phone in Figure 59, Scenario 2 (b), occupies 19 3D points. Conversely, the person sitting in the background in Figure 59 Scenario 1 (a) occupies 79 pixels. Given that the object detection failed to identify the person occupying 79 pixels infers that the Point Cloud classifier is better at identifying features regardless of proximity to the sensor.

TABLE 20: SCENARIO DETAILS DEPICTED IN FIGURE 59

	Images
Row 1	Scenario 1: Showing the subject carrying a box.
Row 2	Scenario 2: Showing the subject standing while texting.
Row 3	Scenario 3: Showing the subject of walking while texting
Row 4	Scenario 4: Showing the subject lying down.

Figure 59 Scenario 3 (a) shows a subject walking while texting. In this image, the subject was identified with 98% confidence. The corresponding 3D ROI was segmented using the ROI identified by the Object Detector is shown in Figure 59 Scenario 3 (b). As in Figure 59 Scenario 2, (b), the subject in this image has their hands out front clasping a phone. This ROI was passed onto the classifier component before the correct result overlay onto the image shown in Figure 59 Scenario 3 (c).

Figure 59, Scenario 4 (a) shows a subject lying down. In this image, the area occupied by the subject, and some of the couch, was identified with 97% confidence. Figure 59 Scenario 4 (b) shows the 3D ROI identified by the Object Detector. In this image, the ROI contains a partial Point Cloud of the subject and a portion of the couch the subject was lying on. When passed onto the classifier, the correct result was overlaid onto the image. Much like Figure 59 Scenario 1, where the subject was carrying a box, the object the subject is performing the activity with, is incorporated into the classification processes.

5.4.5 Limitations & Comparison

Figure 59 presented four scenarios displaying the output of the different sections of the MfV Net for HAR. These images saw the case of non-occluded subjects performing certain activities. In each case, the model outputs remarkably accurate results. Also, this research found that the MfV Net can make a correct prediction from partial data with few points.

For example, in Figure 59, Scenario 4 (b), the subject is in the supine position on a couch. The lack of Point Cloud data cannot be attributed to the instrument, but a principle of operation of the sensor. Note the horizontal planes contacted by a laser – the further away, the more significant the gap between planes. In this case, points on the subject contacted by the laser are reduced to a single plane, making it difficult to classify. Moreover, given the obscurity

and the absence of data points, humans trying to annotate the same data would find this a difficult task.

Notwithstanding the successes, this research observes a few failure patterns of the proposed framework. The primary and most pronounced issue occurs with the frame rate of the LiDAR (5 HZ). While it is possible to increase the frame rate, doing so reduces the number of points captured during a single frame. Although the proposed MfV Net has shown resilience to frugal data captures, there is a point where the network will misclassify. Conversely, it is possible to reduce the frame rate below 5Hz and thus increase the data captured in a single frame. Doing so results in temporal ghosting of subjects moving at speed. In this case, the aligned RGB, Point Cloud, and RGB-D data would fall out of alignment, causing issues in the proposed pipeline. Although this issue is partially mitigated using redundant sensors, the issue is best managed by carefully selecting the frame rate of the sensors.

Another issue encountered was the false positives and negatives identified by the network. As commented in the results section, Figure 59 Scenario 1 (a) showed a subject performing the task and a person sitting on the couch. In this Scenario, the Object Detector misses the person on the couch but identified the subject of interest. This issue was attributed to the distance the subject was from the RGB sensor. It is believed that this issue can be overcome by using sensors with a high-resolution image patch for far-away objects.

In terms of comparison to other networks and their suitability for Point Cloud learning, we looked at [2], [346], [362]. Researchers in [362] converted the Point Cloud data to a voxel grid array. How the CNN acts on a voxel is dependent on the size of the grid – course or a fine. Applying HAR choosing a coarse grid leads to quantization and a substantial loss of information, whereas a fine grid increases the computational cost. Both result in trade-offs depending on the approach and, therefore, are not suitable for classifying human activity.

Point Net, another approach to classification, directly consumes unordered Point Clouds [2], [346]. The Point Net approach applies a symmetric function to n variables whose value given n arguments is the same. This refers to the fact that the model does not assume any spatial relationships between features. This is not the case for HAR, which assumes neighbourhood relations between the location of different points. Consequently, PointNet performs poorly when classifying HAR, regardless of how well it performs on classifying objects.

5.5 Summary

The objectives of this research work are to segment and classify human activity from Point Cloud data using an ROI extracted from an RGB image. Traditional methods of HAR utilize RGB images and wearable sensors attached to the subjects. In this paper, we proposed a MfV Net for HAR. The network is composed of an Object Detector and a 3D classification network. The Object Detector utilizes a Faster–R-CNN to identify an ROI containing a person performing an activity. Translation and alignment of data types allow for the segmentation of the Point Cloud data. The corresponding 3D ROI is converted to a modified Fisher Vector representation before a CNN classifies the activity.

The proposed framework utilizes RGB images, and Point Cloud data acquired from sensors that are commonly used in Intelligent mobility. The benefit of such a system over others is that it removes the need for wearable sensors and provides a reliable and accurate method of HAR. The main challenges to the proposed framework lie with accurate alignment and translation of the different sensor modalities. A further challenge is the sensor frequency of operation, and this plays an integral role in perceiving the correct presence of subjects.

The object detection portion of the proposed framework accurately identified the presence of the subject in an image with an average precision of 95% and a log–average miss rate for 0.1, respectively. The classification portion of the proposed framework accurately classified the activities performed by the subject with 90.3%. The Precision and Recall were 89.5% and 89.4%, respectively.

The implication of this research is a machine that can better recognize human activities being performed in both indoor and outdoor environments. Indirectly this research assist machines in deriving data-driven driver policy, recognize when the elderly suffer a fall and holds the potential to advance human-computer interaction to a new level of understanding.

Performing HAR using sensors that are common to AVs removes the need for wearable sensors and provides valuable context information to machines so they can make better decisions. Furthermore, classifying point cloud data in this manner allows a machine to understand the crucial relationships between a system of rigid segments, connected by articulated joints, the way humans do. This is not the case for other Point Cloud classifiers, where spatial relationships between features are disregarded, thus allowing the proposed framework to achieve state-of-the-art results.

Chapter 6 Conclusion

6.1 Introduction

The objective of this thesis is to contribute to the literature in the field of AV technology and to aid the future of intelligent mobility. While most research for driverless vehicular technology has focused on structured environments such as highways. Visions of intelligent mobility aspire beyond structured environments and seek to assist humans in indoor as well as pedestrianized areas such as pavements. When developing intelligent algorithms for AVs, indoor and pedestrianized areas pose different challenges to structured environments like highways. This is because unstructured environments are diverse, constantly changing, and are normally populated by humans. This thesis aimed to address this gap in AVs research. Motivated by recent developments in deep learning, this thesis focused on investigating data-driven algorithms for intelligent mobility. Within this context, the overall hypothesis of this research was that “it is possible to make autonomous systems safer and more Intelligent with algorithms that are capable of adapting to new environments by leveraging multiple heterogeneous data streams to make robust decisions”.

Towards investigating this hypothesis, this thesis set out the following research objectives:

- To develop a data collection mechanism to investigate a wide spectrum of environments that are to be catered by Intelligent mobility applications, such as indoor spaces and pedestrianized areas.

- To explore ML algorithms that are capable of adapting to new environments and data streams with little or no training.
- To investigate methods of leveraging multiple heterogeneous data streams (Multimodal data streams) to make robust decisions in safety-critical autonomous systems.

To achieve the objectives, multiple contributions, specifically focused on the AV technology with applications to intelligent mobility, were developed. The contributions of this thesis are as follows

1. Throughout the course of this research and in the quest to prove the research hypothesis, many data sets were reviewed – they were deemed unsuitable for the project’s requirements. Therefore, a means of collecting specific data was needed. The first contribution of this thesis is to develop an autonomous platform as an open-source experimental framework for data gathering, sharing, and experimental validation for driverless vehicle technology.
2. Using the platform developed two novel Multimodal data sets are collected for data-driven algorithm development and experimental validation. Firstly, the LboroLdnAV Dataset is a dataset gathered from unstructured indoor and pedestrianized outdoor environments, annotated with 7 object classes collected using seven different perception sensors. Secondly, the Loughborough London HAR Dataset is a multimodal open-source dataset collected indoors, using three different sensors and annotated with 9 classes of human activity.
3. A self-evolving FSD algorithm is developed, which leverages the relative uncertainty of different sensors as a utility to automatically label new data (active learning) and re-learn the data-driven model whenever new data streams are encountered (online learning).
4. Knowing what human agents are doing in their environment is crucial for safe decision-making in AVs. A Multimodal Fisher Vector Network, which is a type of deep convolutional neural network, is proposed as a new methodology for the classification of different human activities

leveraging both RGB camera data and the Point Cloud data that are gathered from LiDAR sensor.

The work developed throughout this research has shown that a self-evolving semi-supervised ML algorithm can detect free space. It has also shown that point cloud data can be used to recognize the human activity. They were developed using an open-source experimental autonomous platform that enabled the collection of specific multimodal sensor data.

6.2 Contributions

6.2.1 Architecture for an Intelligent Mobile Robot

A scalable, multi-layer context mapping and recognition system were presented earlier in this thesis. The architecture has four layers: The Sensing Layer, The Data Analysis Layer, a Multi-layered Context Representation, and The Application Layer. The autonomous platform roams and collects data with the possibility of human intervention.

The experimental autonomous platform produced a tangible output of two datasets – LboroLdnAV & LboroLdnHAR – for AV and HAR, respectively. The development of the datasets is an ongoing project in Loughborough University, London, and additional releases will follow.

Data collected by the autonomous platform was used to construct two different datasets. The LboroLdnAV dataset consists of 45.6 hours of Video, LiDAR, and ultrasound data collected over 1.2 km, displaying a variety of scenarios from both indoor and outdoor environments. In total, there were approximately 2.5 million frames captured by four cameras – 672k frames captured by the 360Fly Wide-angled camera, 1.2 million frames captured by the Ricoh Theta V 360° camera, and 624k frames captured by the two Wansview IP cameras. Both the LiDAR and ultrasonic sensor array captured a total of 252k and 220k scans, respectively.

The LboroLdnHAR dataset consists of 6712 annotated, aligned, and transformed LiDAR, RGB-D, and RGB samples – 5916 for training, 787 for validation, and 9 for demonstration. The dataset was split into three subsets. The first subsets contained RGB samples and consisted of images with annotated ground truth ROI labels, indicating the location of people in the frame. The LiDAR and RGB-D samples were aligned transformed and annotated, indicating the activity being performed by the subjects.

Two algorithms we reported on in this thesis – online active Learning for FSD and an MfV Net for HAR. An autonomous platform used to collect two datasets facilitated their development.

6.2.2 A Self Evolving Free Space Detection Model

Chapter 4 reports on a framework using online active ML for FSD. The proposed framework queries new image data against ranges before fusing the results with the ultrasonic sensor data to make more robust and reliable decisions. Experiments compared four different methods for identifying free space. As the results demonstrate, the proposed framework returns a superior result to the alternative methods. Performing Free Space Detection in this manner allows for the algorithm to self-learn, evolve and adapt to new environments never encountered before. While it is possible to use alternative techniques such as DeepLabv3+ to identify free space, they require large datasets and have difficulty generalising to unfamiliar surroundings. This was demonstrated where all alternative cases of examined underperformed when compared against the proposed framework of online active Learning with sensor data Fusion.

6.2.3 A Multimodal Fisher Vector Network for Human Activity Recognition

Chapter 5 reports on the MfV network for HAR. The proposed framework utilizes RGB images, and Point Cloud Data acquired from sensors that are commonly used in intelligent mobility. The MfV network segments and classifies human activity from the point cloud data using an ROI obtained from an RGB image. The ROI is identified using a Faster R-CNN object detector. Translation and alignment of data types allow for the segmentation of the point cloud data. The corresponding 3D ROI is converted to a modified FV representation before a CNN classifies the activity. The object detection portion of the proposed framework identifies the presence of the subject in an image with an average precision of 95%. The classification portion of the proposed framework accurately classified the activities performed by the subject with an average precision of 90.3%.

6.3 Benefits and Implications of this Research

In this research, we have sought to replicate some of the skills humans frequently rely. Our main aim of this research was to address issues relating to intelligent mobility. Specifically, we focused on the most basic perception skills, FSD, and higher cognitive skill HAR.

The FSD framework proposed in this research is self-learning and self-evolving algorithm. The framework utilizes online active learning and sensor data Fusion to adapt to scenarios never encountered before. Because of the structure of the proposed framework, it is better able to generalize than the current state of the art in FSD and requires little data to start the process. Moreover, as the algorithm encounters new terrains, it becomes better at traversable space.

The HAR framework proposed in this research merges two ML algorithms to perform a task that is largely elusive when using optical data alone or requires the use of ubiquitous wearable sensors. Optical data is susceptible to varying lighting conditions, and ubiquitous wearable sensors are impractical in real-world scenarios. Since LiDAR sensors are commonplace in most AV, having a system that can classify human activity using point cloud data is of the utmost importance.

The implications of this research are autonomous systems that can function safely in both structured and unstructured, dynamic environments. Increasingly complex automated driving functions require more accurate environment models ever. If the relevant information is missing, mobile agent control can contribute to safety risks. Robust and reliable methods of FSD and HAR ensure that relevant information is not missing when a mobile agent needs to make data decision. They will enrich the context information on the Data Analysis Layer, and Multi-layered Context Representation all while contributing towards systems that more closely resemble human ability

6.4 Future Work

Jean Piaget was an educator and psychologist during the early to mid-1900s. A lot of his work centred on how humans developed an understanding of objects [359]. He thought how humans develop over time – through the construction of a schema, the assimilation and accommodation of new information – and during the different stages of our cognitive development, can have a profound and fulfilling understanding of the world. If understanding the world is a consequence of these components, then relationship reasoning is integral to human cognitive development. In fact, learning to detect the relationship between objects is something humans have been doing since birth. Understanding relationships is so omnipresent to our learning that we hardly recognize it happening at all. Currently, humans understand very little about the methods that allow us to distinguish spatial and temporal relations between objects and how they evolve. From a machine learning perspective, relational reasoning is an

elusive goal. While there is some research into relationship reasoning, this field of study is mostly unknown.

Relationship reasoning enables machines to understand the implicit connections between different things. For example, consider the following statement: “All employees finish work at five. John is an employee.” In this case, the relational is that John finishes work at 5, but this was never explicitly stated. We can understand from the statement that there is a relationship between John and employees. By and large, this understanding does not come so easy to machines. While it is possible to encode an understanding between information types or the individual data points in data capture, understanding this relationship is a different thing.

Never more prevalent than in detection and classification tasks. While there are many schools of thought as to how humans achieve this, there is one that accounts for the construction of a schema, the assimilation and accommodation of new information. Titled Recognition by Component, this theory of perception was conceived in 1987 by Irving Biederman. Geometric based Recognition uses pre-defined metrics and some knowledge about the subject before deciding on the objects perceived. To function, effectively Recognition by Component requires an image to be segmented at regions of deep concavity. This allows an image to be broken into an arrangement of simple geometric components - cubes, cylinders, prisms, etc. The theory, first proposed in 1987 by Irving Biederman, makes the fundamental assumption that humans segment objects of any form into 36 generalized components, called primitives [378].

For true identification, the position of the primitive is the key relationship between perceptual order and object recognition. This enables humans to reliably perceive an image at an obscure angle and still understand what is being observed [378]. If the image can be viewed from any orientation, the projection at that time can be regarded as two-dimensional. Objects; therefore, do not need to be presented as a whole but can be represented as a series of simplified shapes, even if some parts are occluded [379], [380].

In addition to filling in the blanks for occluded sections of an object, humans are excellent at trying to make sense of the unknown. For example, when presented with unfamiliar objects, humans easily recognize the primitives of which the image is composed, even if the overall image is not recognized [379], [380]. Biederman and others believed that humans perform this process regularly [378], [380], [381]. Therefore, humans rely on what the image is composed of rather than the familiarity of the image as a whole. This is a representational system that identifies elements of complex images to assist in human

understanding and development [378]. The phenomenon of recognition by component allows humans to rapidly identify objects from obscure scenes, at peculiar angles and under noisy conditions. Deep concavities between primitives are identified using the surface characteristics of the overlapping parts. Non-accidental properties - shapes that look alike from certain angles - are distinguished by co-linearity and symmetry of the primitive being observed [382]. Co-linearity and symmetry play a vital role in identifying components, as does the orientation of the components. For example, a triangle on top of a square bears a striking resemblance to a house, whereas a square on a triangle makes little sense – the components need to match the representation of the memory both in shape and orientation. With the exception of the relationship between adjoining components, identifying the primitives that construct an image, is not a particularly challenging task. If possible, to encode relationship reasoning into an algorithm so that it can make sense of the order and proximity of components, classification using Biederman's recognition by component should return perception skills akin to that of a human.

This research has shown how a relationship between data types or individual data points can work to the benefit of a classification task. For example, the relationship between ultrasound and camera data can be used to classify free space. While a relationship between the proximity of data points in point cloud data facilitates the recognition of an activity being performed by the subject. While the frameworks reported are not precisely relationship reasoning, they show how connections allow ML algorithms to understand abstract links between different things. With further research, a relation network should easily attain relational reasoning capabilities and bring us a step closer to AGI.

Appendix A: LboroLdnAV Dataset Sample



Figure 60: Montage of 48 traversals of the same location on 22nd May 2018, illustrating the diverse range of images, short-term lighting and weather changes encountered by the testbed when collecting the LboroLdnAV dataset.

Appendix B: LboroLdnHAR Dataset Sample

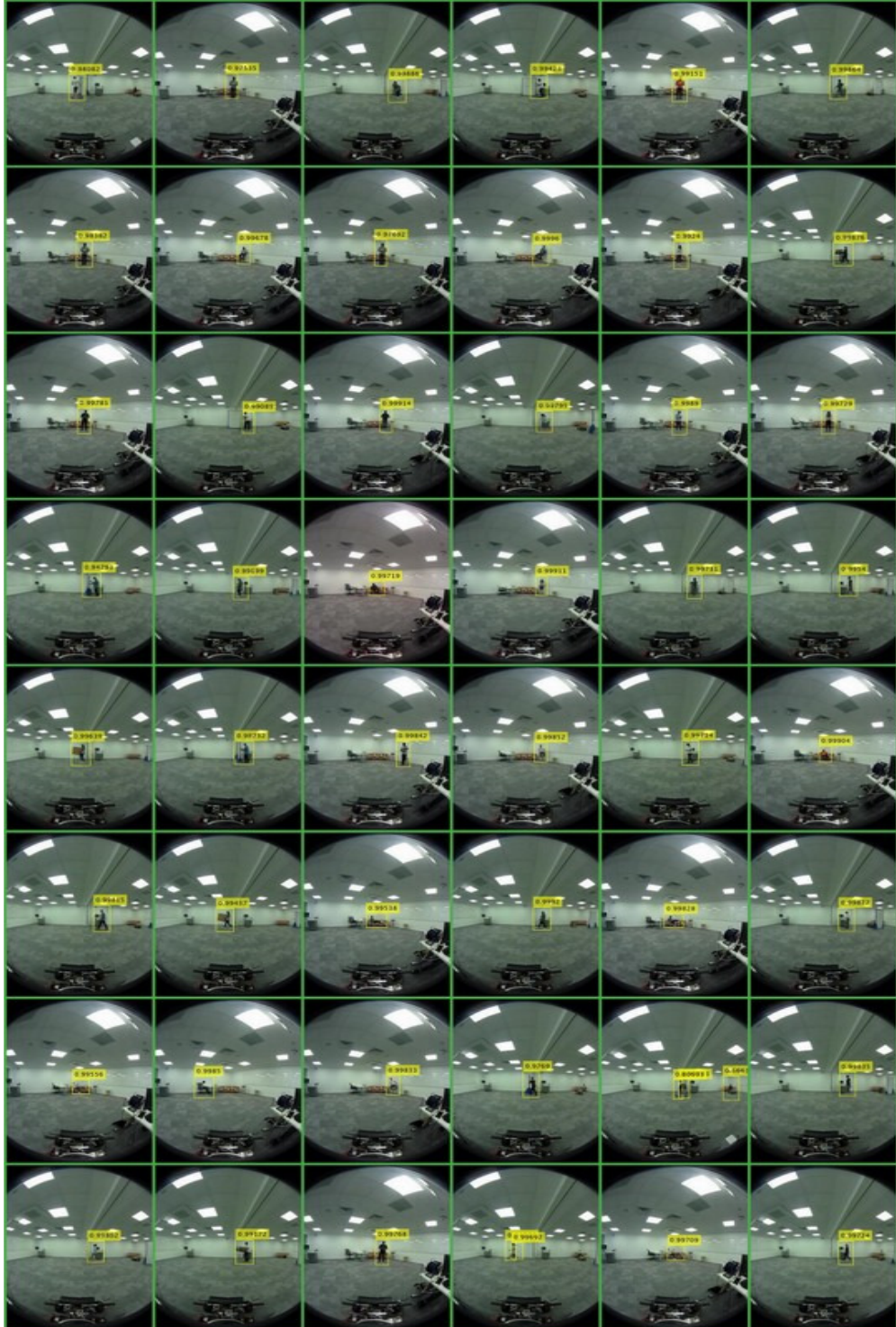


Figure 61: Montage of 48 scenarios performed by different subjects captured on 17th June 2018 and 18th June 2018, illustrating the diverse range of activities in the LboroLdn HAR dataset.

Appendix C: Technical Drawings

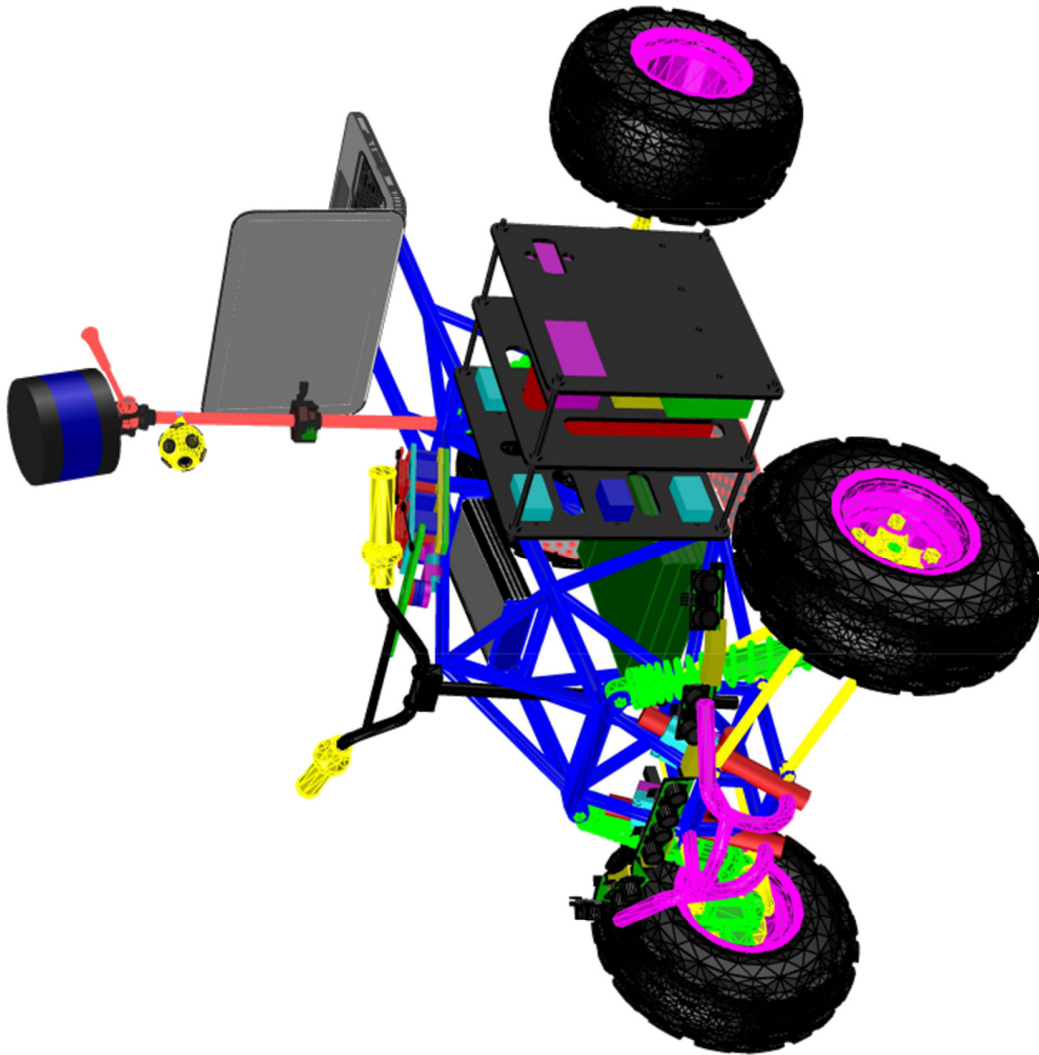


Figure 62: Isometric View of the Autonomous Platform showing the quad bike chassis, electronic rack, ultrasonic array, steering assembly and MSI laptop.

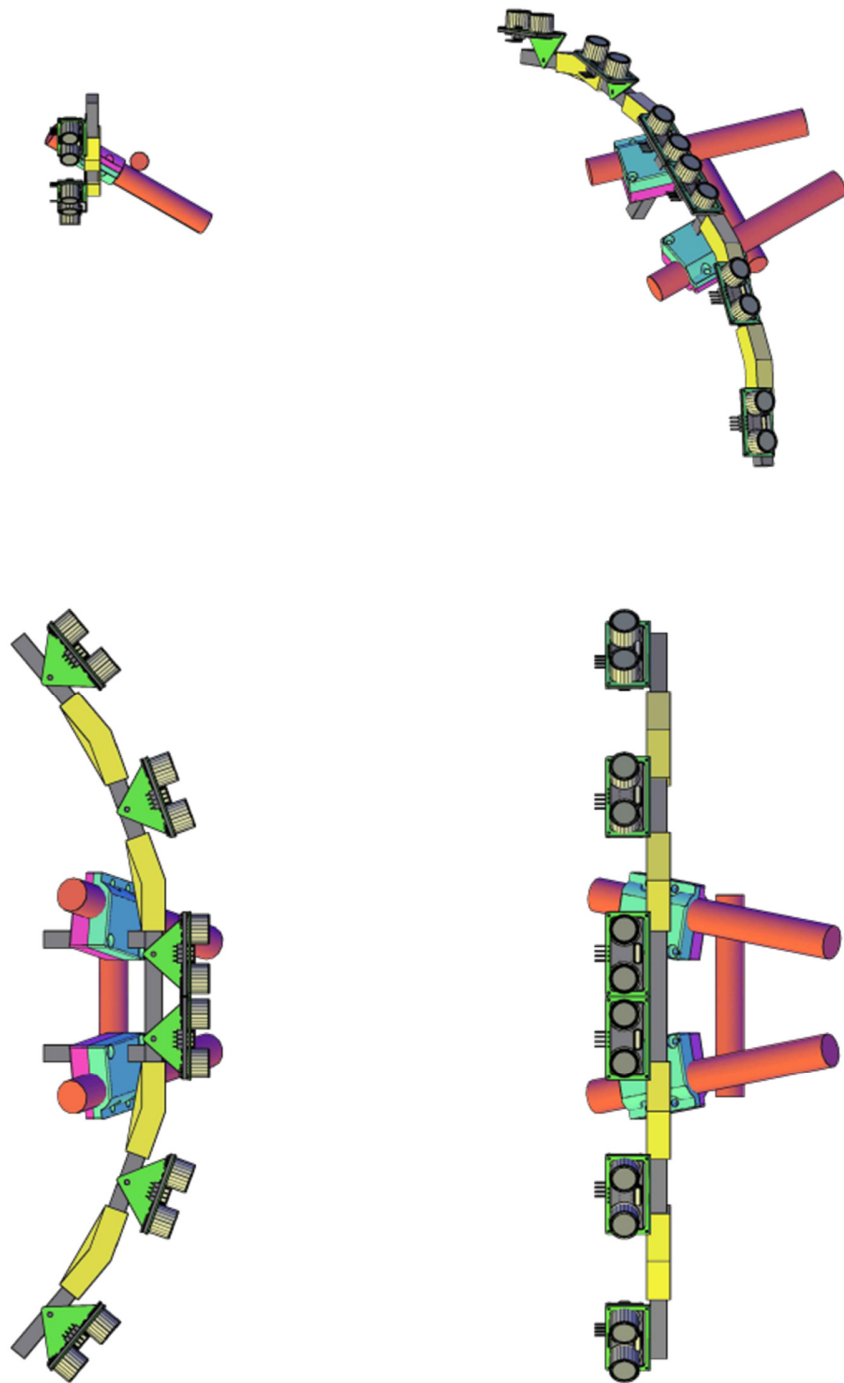


Figure 63: Shows the ultrasonic sensor array. Top left showing side elevation. Top right showing the isometric view. Bottom Left showing plan view. Bottom right front elevation.

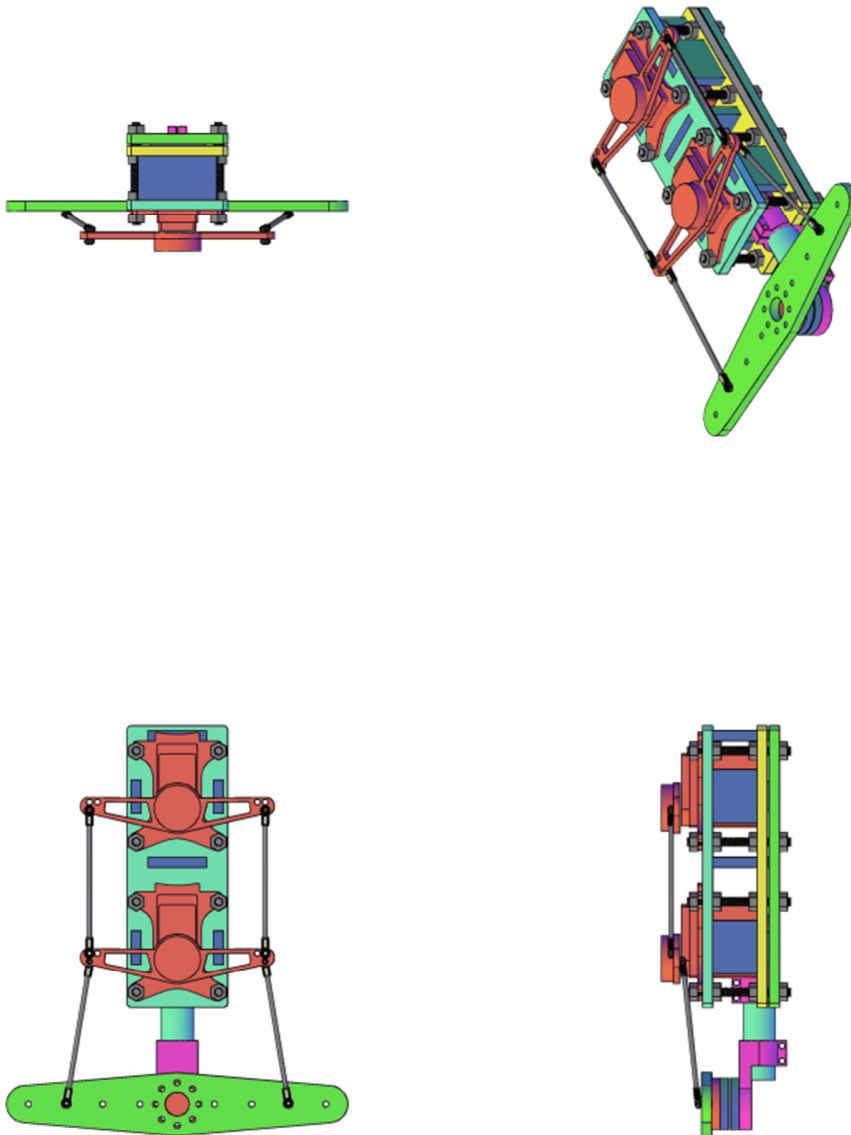


Figure 64: Shows the Steering assembly. Top left showing side elevation. Top right showing the isometric view. Bottom Left showing plan view. Bottom right front elevation.

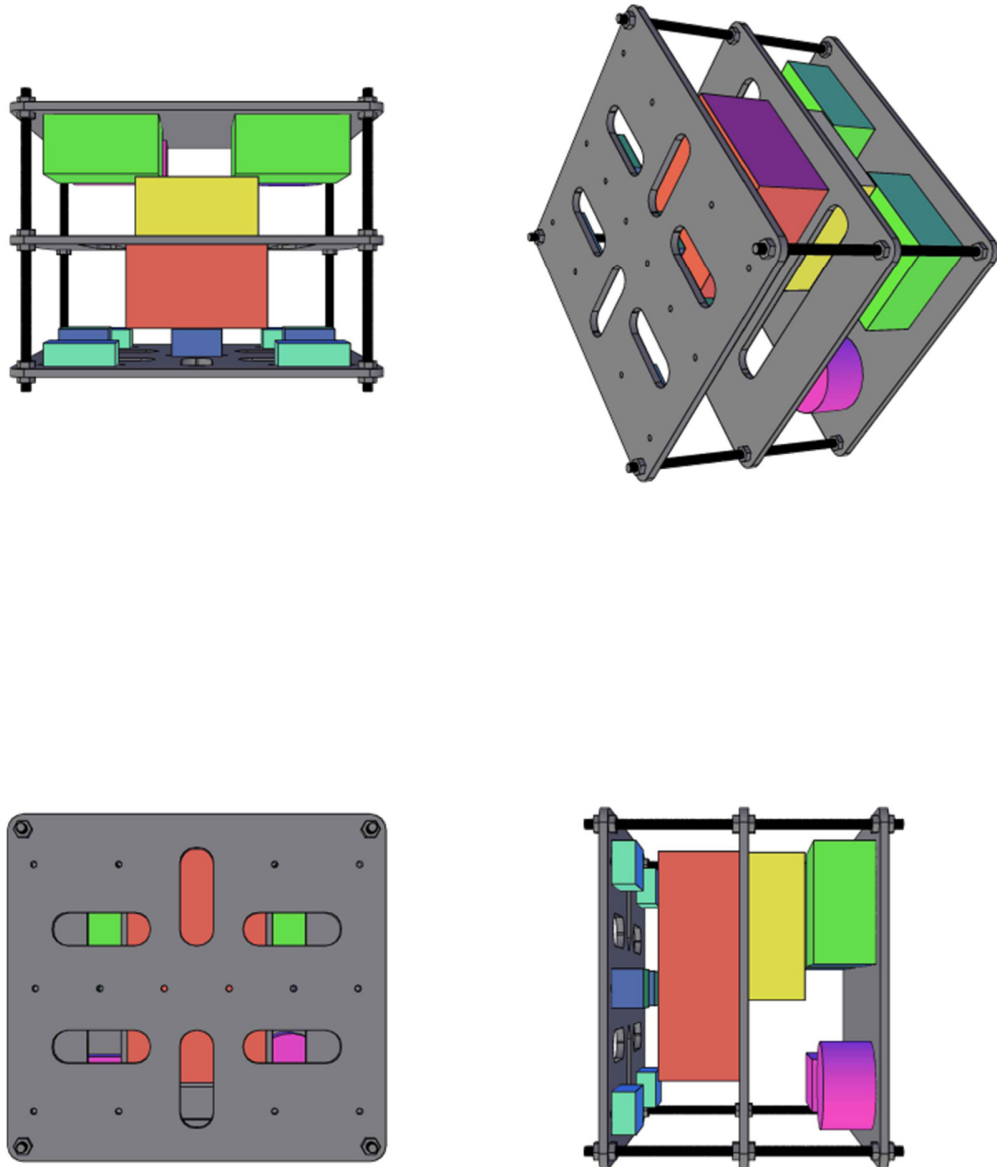


Figure 65: Shows the Power Electric Rack. Top left showing side elevation. Top right showing the isometric view. Bottom Left showing plan view. Bottom right front elevation.

Bibliography

- [1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017.
- [3] N. Brown and T. Sandholm, "Superhuman AI for multiplayer poker," *Science (80-.)*, 2019.
- [4] A. M. Zador, "A critique of pure learning and what artificial neural networks can learn from animal brains," *Nat. Commun.*, 2019.
- [5] H. Shah, "Turing's misunderstood imitation game and IBM's watson success," in *AISB 2011: Towards a Comprehensive Intelligence Test*, 2011.
- [6] J. McCracken, *Oxford dictionary of English*. 2003.
- [7] B. Goertzel, "Artificial General Intelligence: Concept, State of the Art, and Future Prospects," *J. Artif. Gen. Intell.*, 2014.
- [8] F. Truck and H. Moravec, "Mind Children: The Future of Robot and Human Intelligence," *Leonardo*, 1991.
- [9] Kyle Wiggers, "MIT's AI makes autonomous cars drive more like humans," *Venture Beat*, 2019. [Online]. Available: <https://venturebeat.com/2019/05/23/mits-ai-makes-autonomous-cars-drive-more-like-humans/>. [Accessed: 27-Nov-2019].
- [10] R. P. Hall and D. F. Kibler, "Differing Methodological Perspectives in Artificial Intelligence Research," *AI Mag.*, 1985.
- [11] F. Attneave, M. B., and D. O. Hebb, "The Organization of Behavior; A Neuropsychological Theory," *Am. J. Psychol.*, 1950.
- [12] S. Das, W. K. Wong, T. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017.
- [13] N. Rubens, M. Elahi, M. Sugiyama, and D. Kaplan, "Active learning in recommender

- systems,” in *Recommender Systems Handbook, Second Edition*, 2015.
- [14] S. Doherty, “Narrow vs general ai is Moravec’s paradox still relevant,” *Graphcore*, 2016.
- [15] K. Agrawal, “To study the phenomenon of the Moravec’s Paradox,” *arXiv:1012.3148*, 2010.
- [16] W. Halal, J. Kolber, and O. Davies, “Forecasts of AI and future jobs in 2030: Muddling through likely, with two alternative scenarios,” *J. Futur. Stud.*, 2016.
- [17] Neurons.co.uk, “The Human Brain: From Neurone to Nervous System,” 2016. [Online]. Available: <http://neurons.co.uk/Neurosciences/Tutorials/M4/M.4.2b Visual Pathway.html>.
- [18] A. Ligeza, “Artificial Intelligence: A Modern Approach,” *Neurocomputing*, 1995.
- [19] Z. Wang, Z. Liu, and C. Zheng, “Introduction to neural networks,” *Studies in Systems, Decision and Control*. 2016.
- [20] C. Fernando *et al.*, “PathNet: Evolution Channels Gradient Descent in Super Neural Networks,” *arXiv:1701.08734v1*, 2017.
- [21] R. Millsap and A. Maydeu-Olivares, *The SAGE Handbook of Quantitative Methods in Psychology*. 2012.
- [22] *Hybrid Intelligence for Image Analysis and Understanding*. 2017.
- [23] N. Y. Masse, G. D. Grant, and D. J. Freedman, “Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 115, no. 44, p. E10467—E10475, Oct. 2018.
- [24] T. Flesch, J. Balaguer, R. Dekker, H. Nili, and C. Summerfield, “Comparing continual task learning in minds and machines,” *Proc. Natl. Acad. Sci. U. S. A.*, 2018.
- [25] J. Lee, “A survey of robot learning from demonstrations for Human-Robot Collaboration,” 2017.
- [26] T. Baltrusaitis, C. Ahuja, and L. P. Morency, “Multimodal Machine Learning: A Survey and Taxonomy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [27] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski, “Integration of Acoustic and Visual Speech Signals Using Neural Networks,” *IEEE Commun. Mag.*, 1989.

- [28] L. P. Morency and T. Baltrušaitis, “Multimodal machine learning: Integrating language, vision and speech,” in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference Tutorial Abstracts*, 2017.
- [29] K. Sennaar, “AI in Agriculture - Present Applications and Impact,” *techemergence*, 2018.
- [30] M. Coccia, “Artificial Intelligence Technology In Oncology: A New Technological Paradigm,” *J. Chem. Inf. Model.*, 2013.
- [31] K. Arun, G. Ishan, and K. Sanmeet, “Loan Approval Prediction based on Machine Learning Approach,” *IOSR J. Comput. Eng.*, 2016.
- [32] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, “Applications of artificial intelligence in transport: An overview,” *Sustainability (Switzerland)*. 2019.
- [33] BBC, “Every death on every road in Great Britain 1999-2010,” 2011. [Online]. Available: <http://www.bbc.co.uk/news/uk-15975564>.
- [34] S. Copsey, “A review of accidents and injuries to road transport drivers .” European Union, Luxembourg, 2012.
- [35] TfL, “Casualties in Greater London during 2017,” Transport for London, 2015.
- [36] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, “Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 873–878.
- [37] S. Sivaraman and M. M. M. Trivedi, “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, 2013.
- [38] Volkswagen, “Park assist,” *Helps you park your car quickly and easily*, 2017. [Online]. Available: <http://www.volkswagen.co.uk/technology/parking-and-manoeuving/park-assist>.
- [39] M. Benz, “Parktronic,” *Parktronic Including Parking Guidance*, 2017. [Online]. Available: <https://www.mercedesbenzofboston.com/new-features-mercedes-benz-parktronic.htm>. [Accessed: 12-Dec-2019].
- [40] M. Nielsen, *Neural Nets and Deep Learning*. Determination Press, 2015.

- [41] Q. J. Zhang, S. J. Stanley, and D. W. Smith, “Internal workings of feed-forward neural networks,” *J. Environ. Eng. Sci.*, 2004.
- [42] V. N. Vapnik, *The Nature of Statistical Learning Theory*. 2000.
- [43] R. Tempo, G. Calafiore, and F. Dabbene, “Statistical Learning Theory,” in *Communications and Control Engineering*, 2013.
- [44] R. M. Kretchmar *et al.*, “Robust reinforcement learning control,” *Proc. Am. Control Conf.*, 2001.
- [45] J. Roche, V. De Silva, and A. Kondozi, “A Cognitive Framework for Object Recognition with Application to Autonomous Vehicles,” in *Proceedings of the 2018 Computing Conference*, 2018, p. 9.
- [46] N. Name, “How AI Learns,” *The Science Thinkers*, 2018. [Online]. Available: <https://www.thesciencethinkers.com/2018/05/ai-how-ai-learn.html>. [Accessed: 18-Nov-2019].
- [47] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition,” 2015.
- [48] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognit. Lett.*, 2019.
- [49] R. Szegedy, C. Zaremba, W. Sutskever, I. Bruna, J. Erhan, D. Goodfellow, I.J., Fergus, “Intriguing properties of neural networks,” in *International Conference on Learning Representations*, 2013.
- [50] Douglas Heaven, “Why deep-learning AIs are so easy to fool,” *Nature*, vol. 574, p. 4, 2019.
- [51] C. Adami, “Robots with instincts,” *Nature*, vol. 521, no. 7553, pp. 426–427, 2015.
- [52] K. Battaglia, P.W., Pascanu, R., Lai, M., Rezende, D.J., & Kavukcuoglu, “Interaction Networks for Learning about Objects Relations and Physics,” in *NIPS*, 2016.
- [53] A. Santoro *et al.*, “A simple neural network module for relational reasoning,” in *Advances in Neural Information Processing Systems*, 2017.
- [54] R. Fisher, “Dictionary of Computer Vision and Image Processing,” *J. Electron. Imaging*, 2006.
- [55] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. 2016.

- [56] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, “Building machines that learn and think like people,” *Behav. Brain Sci.*, 2017.
- [57] G. K. Kostopoulos, “Computers cannot learn the way humans do – Partly, because they do not sleep,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017.
- [58] M. Hutson, “How researchers are teaching AI to learn like a child,” *Science (80-.)*, 2018.
- [59] L. Neumann, B. Vanholme, M. Gressmann, A. Bachmann, L. Kahlke, and F. Schule, “Free Space Detection: A Corner Stone of Automated Driving,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015.
- [60] S. Patra, P. Maheshwari, S. Yadav, S. Banerjee, and C. Arora, “A joint 3D-2D based method for free space detection on roads,” in *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018.
- [61] Ó. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE Commun. Surv. Tutorials*, 2013.
- [62] D. G. Pearson *et al.*, “Assessing mental imagery in clinical psychology: a review of imagery measures and a guiding framework,” *Clin Psychol Rev*, vol. 33, no. 1, pp. 1–23, 2013.
- [63] J. C. Badcock, “The Cognitive Neuropsychology of Auditory Hallucinations: A Parallel Auditory Pathways Framework,” *Schizophr. Bull.*, vol. 36, no. 3, pp. 576–584, 2010.
- [64] N. J. Wade, “Pioneers of eye movement research,” *Iperception.*, vol. 1, no. 2, pp. 33–68, 2010.
- [65] G. N. Dutton, “Cognitive vision, its disorders and differential diagnosis in adults and children: knowing where and what things are,” *Eye London*, vol. 17, no. 3, pp. 289–304, 2003.
- [66] B. W. Tatler and M. F. Land, “Vision and the representation of the surroundings in spatial memory,” *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 366, no. 1564, pp. 596–610, 2011.
- [67] M. N. Hebart, G. Hesselmann, and H. Guido, “What Visual Information Is Processed in the Human Dorsal Stream?,” *J. Neurosci.*, vol. 32, no. 24, pp. 8107–8109, 2012.

- [68] M. Mishkin, L. G. Ungerleider, and K. A. Macko, "Object vision and spatial vision: two cortical pathways," *Trends Neurosci.*, vol. 6, pp. 414–417, 1983.
- [69] W. Winlow and R. Markstein, *The Neurobiology of Dopamine Systems*. Manchester University Press, 1986.
- [70] J. S. Ross, K. J. W. Wilson, and J. Wallace, *Anatomy and physiology in health and illness*, 7th ed. Churchill Livingstone, 1990.
- [71] Pankaj Sah, "What are neurotransmitters?," *Brain Function*, 2017. [Online]. Available: <https://qbi.uq.edu.au/brain/brain-physiology/what-are-neurotransmitters>. [Accessed: 01-Dec-2019].
- [72] D. Robertson and I. Biaggioni, *Primer on the Autonomic Nervous System*. Elsevier Academic Press, 2012.
- [73] H. Lodish, *Molecular Cell Biology*. W. H. Freeman, 2008.
- [74] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
- [75] G. A. Tagliarini, J. F. Christ, and E. W. Page, "Optimization Using Neural Networks," *IEEE Trans. Comput.*, 1991.
- [76] J. College, "Brain Map and Phineas Gage." [Online]. Available: <http://www.jonescollegeprep.org/ourpages/auto/2015/9/11/44795072/Brain Map and Phineas Gage.pdf>. [Accessed: 10-Dec-2019].
- [77] S. Twomey, "Phineas Gage: Neuroscience's most famous patient," 2010. [Online]. Available: <http://www.smithsonianmag.com/history/phineas-gage-neurosciences-most-famous-patient-11390067/?no-ist>. [Accessed: 10-Dec-2019].
- [78] C. L. Hammond D., "Phineas Gage: The man with a hole in his head," 2011. [Online]. Available: <http://www.bbc.co.uk/news/health-12649555>. [Accessed: 10-Dec-2019].
- [79] BigPicture, "Brain case study: Phineas Gage." [Online]. Available: <https://bigpictureeducation.com/brain-case-study-phineas-gage>. [Accessed: 10-Dec-2019].
- [80] M. Macmillan, *An Odd Kind of Fame: Stories of Phineas Gage*. MIT Press, 2002.
- [81] R. A. Ryerson and A. J. Lewis, *Manual of remote sensing. Vol.2, Principles and applications of imaging radar*, 3rd ed. /. New York ; Chichester: J. Wiley, 1998.

- [82] C. Fyfe, *Hebbian Learning and Negative Feedback Networks*. Springer London, 2007.
- [83] M. Pfeiffer, B. Nessler, R. J. Douglas, and W. Maass, “Reward-Modulated Hebbian Learning of Decision Making,” *Neural Comput.*, vol. 22, no. 6, pp. 1399–1444, 2010.
- [84] I. Arel, D. C. Rose, and T. P. Karnowski, “Deep machine learning-a new frontier in artificial intelligence research [research frontier],” *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, 2010.
- [85] J. Heaton, *Artificial Intelligence for Humans: Deep learning and neural networks*. Heaton Research, Incorporated, 2015.
- [86] J. Davis, M. Balda, D. Rock, P. McGinniss, and L. Davachi, “The science of making learning stick: An update to the AGES model.,” *NeuroLeadership J.*, 2014.
- [87] H. A. Simon, “Why Should Machines Learn?,” in *Machine Learning*, 1983.
- [88] V. D. Sánchez A., “Neural network design and the complexity of learning,” *Neurocomputing*, 1991.
- [89] A. L. Samuel, “Some Studies in Machine Learning,” *IBM J. Res. Dev.*, 1959.
- [90] R. Kurzweil, *The Age of Intelligent Machines*. Viking, 1992.
- [91] L. Muganda and E. Standley, *Automated Cars Prophesied by William Branham*. Xulon Press, Incorporated, 2009.
- [92] R. Nath, *Philosophy of Artificial Intelligence: A Critique of the Mechanistic Theory of Mind*. Universal Publishers, 2009.
- [93] S. Wermter, E. Riloff, and G. Scheler, “Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing,” *Connect. Stat. Symb. Approaches to Learn. Nat. Lang. Process.*, 1996.
- [94] C. M. Bishop, *Pattern Recognition and Machine Learning*. 2013.
- [95] S. Doyle-Lindrud, “Watson will see you now: A supercomputer to help clinicians make informed treatment decisions,” *Clin. J. Oncol. Nurs.*, 2015.
- [96] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [97] H. Liu and L. Wang, “Gesture recognition for human-robot collaboration: A review,” *International Journal of Industrial Ergonomics*, 2016.

- [98] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.
- [99] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A Brief Survey of Deep Reinforcement Learning," *Spec. ISSUE Deep Learn. IMAGE UNDERSTANDING.*, 2017.
- [100] S. Sajad Mousavi, M. Schukat, and E. Howley, "Deep Reinforcement Learning: An Overview," 2018.
- [101] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and Autonomous Control Using Reinforcement Learning: A Survey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, 2018.
- [102] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," *Transp. Res. Part A Policy Pract.*, 2016.
- [103] C. Hewitt, T. Amanatidis, I. Politis, and A. Sarkar, "Assessing public perception of self-driving cars: The autonomous vehicle acceptance model," in *International Conference on Intelligent User Interfaces, Proceedings IUI*, 2019.
- [104] R. Nader, "Unsafe at any speed: the designed-in dangers of the American automobile. 1965.," *Am. J. Public Health*, 2011.
- [105] SAE international, "Automated driving - levels of driving automation are defined in new sae international standard j3016," *SAE Int.*, 2016.
- [106] S. Standard, "J3016," *Taxon. Defin. Terms Relat. to On-Road Mot. Veh. Autom. Driv. Syst.*, 2014.
- [107] Lex Fridman, "Human-Centered Autonomous Vehicle Systems: Principles of Effective Shared Autonomy," *arXiv:1810*, 2018.
- [108] F. Rosenblatt, "A probabilistic model for visual perception," *Acta Psychol. (Amst).*, 1959.
- [109] F. Rosenblatt, "The Design of an Intelligent Automaton," *U.S. Off. Nav. Res.*, vol. 6, no. 2, p. 7, 1958.
- [110] V. VAPNIK, "Pattern recognition using generalized portrait method," *Autom. Remote*

- Control*, 1963.
- [111] J. Brownlee, "Supervised and Unsupervised Machine Learning Algorithms," *Understand Mach. Learn. Algorithms*, 2016.
- [112] V. Roman, "Unsupervised Machine Learning: Clustering Analysis -- Towards Data Science," *Towards Data Science*. 2019.
- [113] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2015.
- [114] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *Advances in Neural Information Processing Systems*, 2005.
- [115] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 2005.
- [116] De Silva V. Roche J. Kondo A, V. De-Silva, J. Roche, and A. Kondo, "Robust Fusion of LiDAR and Wide-Angle Camera Data for Autonomous Mobile Robots," *Sensors (Switzerland)*, vol. 18(8), no. 2730, 2018.
- [117] C. Zhang *et al.*, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations*, 2019.
- [118] A. Ghatak, *Machine Learning with R*. Springer, 2017.
- [119] U. Kamath and K. Choppella, "Mastering Java Machine Learning." p. 557, 2017.
- [120] D. Marr, "A theory for cerebral neocortex.," *Proc. R. Soc. London. Ser. B. Biol. Sci.*, 1970.
- [121] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann Machines.," *Cogn. Sci.*, 1985.
- [122] E. R. Ziegel, "The Elements of Statistical Learning," *Technometrics*, 2003.
- [123] Y. Feng, S. Pickering, E. Chappell, P. Iravani, and C. Brace, "A support vector clustering based approach for driving style classification," *Int. J. Mach. Learn. Comput.*, 2019.
- [124] S. J. Roberts, "Parametric and non-parametric unsupervised cluster analysis," *Pattern*

Recognit., 1997.

- [125] W. Liu and S. Li, “An effective lane detection algorithm for structured road in urban,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013.
- [126] R. Ajaykumar, A. Gupta, and S. N. Merchant, “Automated lane detection by K-means clustering: A machine learning approach,” in *IS and T International Symposium on Electronic Imaging Science and Technology*, 2016.
- [127] H. Jia, Z. Wei, X. He, and M. Li, “A Research on Lane Marking Detection Algorithm Based on Neural Network and Least Squares Method,” *Qiche Gongcheng/Automotive Eng.*, 2018.
- [128] S. Lee *et al.*, “VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [129] G. Hinton. and T. A. Poggio., “Unsupervised learning: Foundations of neural computation,” *Comput. Math. with Appl.*, 1999.
- [130] A. B. Tucker, *Computer science handbook, second edition*. 2004.
- [131] A. M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes (New methods for determining the orbits of comets)*. 1985.
- [132] C. F. Gauss, “Theorie der Bewegung der Himmelskörper, die die Sonne in Kegelschnitten umkreisen (Theory of the Motion of Heavenly Bodies Moving about the Sun in Conic Sections),” in *Werke*, 2012.
- [133] L. Martinez-Elizalde and C. Astengo-Noguez, “An improvement to flock traffic navigation algorithm using linear regression techniques,” in *7th Mexican International Conference on Artificial Intelligence - Proceedings of the Special Session, MICAI 2008*, 2008.
- [134] W. K. V. Chan, “Agent-based and regression models of social influence,” in *Proceedings - Winter Simulation Conference*, 2018.
- [135] P. R. Cohen, D. M. Hart, R. St. Amant, L. A. Ballesteros, and A. Carlson, “Path Analysis Models of an Autonomous Agent in a Complex Environment,” 1994.
- [136] I. V. Miroshnik and X. L. Huang, “Nonlinear control of robot spatial motion in dynamic

- environments,” in *Proceedings of 2002 International Conference on Machine Learning and Cybernetics*, 2002.
- [137] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern Classification (2nd ed .),” *Comput. Complex.*, 1998.
- [138] R. Berwick, “An Idiot’s guide to Support vector machines,” *MIT Press*, 2003. [Online]. Available: <http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf>. [Accessed: 10-Dec-2019].
- [139] R. Hota, S. Syed, S. Bandyopadhyay, and P. Krishna, “A Simple and Efficient Lane Detection using Clustering and Weighted Regression.,” *COMAD*, 2009.
- [140] P. Tribaldos, J. Serrano-Cuerda, M. T. López, A. Fernández-Caballero, and R. J. López-Sastre, “People detection in color and infrared video using HOG and linear SVM,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013.
- [141] P. Cichosz, *Data Mining Algorithms: Explained Using R*. 2015.
- [142] J. Pavlus, “Computers Now Recognize Patterns Better Than Humans Can,” *Scientific American*, New York, Dec-2016.
- [143] S. C. Kothari and H. Oh, “Neural Networks for Pattern Recognition,” *Adv. Comput.*, 1993.
- [144] C. M. Bishop, *Pattern Recognition and Machine Learning*. 2006.
- [145] D. Liu, H. Zhang, M. Polycarpou, C. Alippi, and H. He, *Advances in Neural Networks -- ISNN 2011: 8th International Symposium on Neural Networks, ISNN 2011, Guilin, China, May 29--June 1, 2011, Proceedings*. Springer, 2011.
- [146] L. Iliadis, H. Papadopoulos, and C. Jayne, *Engineering Applications of Neural Networks*, no. pt. 1. Springer, 2013.
- [147] G. Anthony, H. Greg, and M. Tshilidzi, “Classification of Images Using Support Vector Machines,” *arXiv:0709.3967*, 2007.
- [148] K. He, X. Zhang, S. Ren, and J. Sun, “ResNet,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2016.
- [149] B. Planche and E. Andres, *Hands-On Computer Vision with TensorFlow 2: Leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and*

- Keras*. Packt Publishing, 2019.
- [150] E. Kakaletsis *et al.*, “Semantic map annotation through UAV video analysis using deep learning models in ROS,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019.
- [151] D. J. Hemanth and V. V. Estrela, *Deep learning for image processing applications*. IOS Press, 2017.
- [152] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, 1998.
- [153] N. Weiss, H. Kost, and A. Homeyer, “Towards Interactive Breast Tumor Classification Using Transfer Learning,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.
- [154] I. F. for the P. of M. and M. S. W. Congress and T. Uhl, *Advances in Mechanism and Machine Science*. Springer International Publishing, 2019.
- [155] V. V. Romanuke, “Appropriate Number of Standard 2×2 Max Pooling Layers and Their Allocation in Convolutional Neural Networks for Diverse and Heterogeneous Datasets,” *Inf. Technol. Manag. Sci.*, 2018.
- [156] D. Scherer, A. Müller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.
- [157] E. J. Berg, *Heaviside’s Operational Calculus as Applied to Engineering and Physics*. University of California: McGraw-Hill book Company, 1929.
- [158] C. K. Gately, L. R. Hutyra, I. S. Wing, and M. N. Brondfield, “Introduction to Neural Networks Using Matlab 6.0,” *Environ. Sci. Technol.*, 2013.
- [159] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.
- [160] H. R. Wilson and J. D. Cowan, “Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons,” *Biophys. J.*, 1972.
- [161] R. H. R. Hahnioser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, “Digital selection and analogue amplification coexist in a cortex- inspired silicon

- circuit,” *Nature*, 2000.
- [162] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*. 2015.
- [163] J. Patterson and A. Gibson, *Deep Learning A practitioner’s approach*. O’Reilly Media, 2008.
- [164] G. J. Tortora and R. L. Evans, *Principles of human physiology*. Harper & Row Limited, 1986.
- [165] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2017.
- [166] I. Zafar, G. Tzanidou, R. Burton, N. Patel, and L. Araujo, *Hands-On Convolutional Neural Networks with TensorFlow*. Packt Publishing Ltd, 2018.
- [167] W. Wei, J. A. Gulla, and Z. Fu, “Advanced Intelligent Computing Theories and Applications,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2010.
- [168] J. Brownlee, *Better Deep Learning. Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery, 2018.
- [169] B. Nemire, “CUDA Spotlight: GPU-Accelerated Deep Neural Networks,” 2014. [Online]. Available: <https://devblogs.nvidia.com/parallelforall/cuda-spotlight-gpu-accelerated-deep-neural-networks/>. [Accessed: 10-Dec-2019].
- [170] N. Garg, “What is the vanishing gradient problem?,” 2015. [Online]. Available: <https://www.quora.com/What-is-the-vanishing-gradient-problem>.
- [171] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Berlin Heidelberg, 2012.
- [172] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [173] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, “Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning,” *arXiv:1712.06567*, 2017.
- [174] D. V. S. X. De Silva, W. A. C. Fernando, and S. T. Worrall, “Intra mode selection method for depth maps of 3D video based on rendering distortion modeling,” *IEEE Trans. Consum. Electron.*, 2010.

- [175] H. M. Le, C. Peter, and Y. Yue, "Data-Driven Ghosting using Deep Imitation Learning," *MIT Sloan Sport. Anal. Conf.*, 2017.
- [176] A. Broggi *et al.*, "PROUD-Public Road Urban Driverless-Car Test," *IEEE Trans. Intell. Transp. Syst.*, 2015.
- [177] A. Broggi, P. Medici, E. Cardarelli, P. Cerri, A. Giacomazzo, and N. Finardi, "Development of the control system for the VisLab intercontinental autonomous challenge," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2010.
- [178] S. Ionita, "Autonomous vehicles: From paradigms to technology," in *IOP Conference Series: Materials Science and Engineering*, 2017.
- [179] M. Bertozzi *et al.*, "The VisLab intercontinental autonomous challenge: 13,000 km, 3 months, no driver," in *17th World Congress on ITS*, 2010.
- [180] U. N. P. Martinet, C. Laugier, "Perception and Navigation for Autonomous Vehicles," *IEEE Robotics and Automation Magazine (RAM)*, 2014.
- [181] K. Bimbraw, "Autonomous Cars: Past, Present and Future - A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology," in *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics*, 2015.
- [182] A. Reuschenbach, M. Wang, T. Ganjineh, and D. Göhring, "IDriver - Human machine interface for autonomous cars," in *Proceedings - 2011 8th International Conference on Information Technology: New Generations, ITNG 2011*, 2010.
- [183] S. Thrun, M. Montemerlo, and H. Dahlkamp, "Stanley: the robot that won the DARPA grand challenge," *J. F. Robot.*, vol. 23, no. 9, p. 31, 2006.
- [184] D. Göhring, D. Latotzky, M. Wang, and R. Rojas, "Semi-autonomous car control using brain computer interfaces," *Intell. Auton. Syst.* 12, 2013.
- [185] T. Langner, D. Seifert, B. Fischer, D. Goehring, T. Ganjineh, and R. Rojas, "Traffic awareness driver assistance based on stereovision, eye-tracking, and head-up display," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2016.
- [186] U. Franke *et al.*, "Making bertha see," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

- [187] J. Dickmann *et al.*, “Making bertha see even more: Radar contribution,” *IEEE Access*, 2015.
- [188] J. Ziegler, P. Bender, M. Schreiber, and others, “Making Bertha Drive - An Autonomous Journey on a Historic Route,” *Intell. Transp. Syst. Mag.*, 2014.
- [189] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.
- [190] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “The KITTI Vision Benchmark Suite,” *KITTI Vis. Benchmark Suite*, 2013.
- [191] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. J. Rob. Res.*, 2013.
- [192] IEEE, “Intelligent transportation systems,” in *2000 IEEE Intelligent transportation systems Proceedings*, 2000, pp. vi, 509 p.
- [193] S. Park, W. Choi, D. K. Han, and H. Ko, “Acoustic event filterbank for enabling robust event recognition by cleaning robot,” *IEEE Trans. Consum. Electron.*, 2015.
- [194] W. Lumpkins, “Driverless cars and driverless vacuums: Will the madness never end?,” *IEEE Consum. Electron. Mag.*, 2014.
- [195] J. Zhang, G. Song, G. Qiao, T. Meng, and H. Sun, “An indoor security system with a jumping robot as the surveillance terminal,” *IEEE Trans. Consum. Electron.*, 2011.
- [196] L. S. R. Mechsy, M. U. B. Dias, W. Pragithmukar, and A. L. Kulasekera, “A mobile robot based watering system for smart lawn maintenance,” in *International Conference on Control, Automation and Systems*, 2017.
- [197] H. Şahin and L. GuÜvenç, “Household Robotics Autonomous Devices for Vacuuming and Lawn Mowing,” *IEEE Control Syst.*, 2007.
- [198] M. Webster *et al.*, “Toward Reliable Autonomous Robotic Assistants Through Formal Verification: A Case Study,” *IEEE Trans. Human-Machine Syst.*, 2016.
- [199] B. Markwalter, “The Path to Driverless Cars,” *IEEE Consum. Electron. Mag.*, 2017.
- [200] L. Jones, “Driverless when and cars: where?,” *IEEE Eng. Technol.*, vol. 12, no. 2, p. 4, Mar. 2017.
- [201] M. Rowthorn, “How should autonomous vehicles make moral decisions? Machine

- ethics, artificial driving intelligence, and crash algorithms,” *Contemp. Readings Law Soc. Justice*, 2019.
- [202] E. Awad, J.-F. Bonnefon, A. Shariff, and I. Rahwan, “The Thorny Challenge of Making Moral Machines: Ethical Dilemmas with Self-Driving Cars,” *NIM Mark. Intell. Rev.*, 2019.
- [203] P. Lin, “Is Tesla Responsible for the Deadly Crash On Auto-Pilot? Maybe.,” *Forbes*, 2016.
- [204] I. Kankam, “Design of an Immersive Virtual Environment to Investigate How Different Drivers Crash in Trolley-problem Scenarios,” 2019.
- [205] L. Tarr, *The history of the carriage*. Arco Publishing, 1969.
- [206] H. Chao, Y. Cao, and Y. Chen, “Autopilots for small unmanned aerial vehicles: A survey,” *Int. J. Control. Autom. Syst.*, 2010.
- [207] B. L. Stevens and F. L. Lewis, “Aircraft Control and Simulation,” *Aircr. Eng. Aerosp. Technol.*, 2004.
- [208] W. J. Daniels and H. B. Tucker, *Model Sailing Craft*. Chapman & Hall, 1932.
- [209] M. Bajracharya, M. W. Maimone, and D. Helmick, “Autonomy for Mars Rovers: Past, present, and future,” *Computer (Long. Beach. Calif.)*, 2008.
- [210] S. Loff and B. Dunbar, “Mars Pathfinder and Sojourner,” *National Aeronautics and Space Administration*, 2019. [Online]. Available: https://www.nasa.gov/mission_pages/mars-pathfinder. [Accessed: 14-Nov-2019].
- [211] S. Loff and B. Dunbar, “Mars Exploration Rovers Overview,” *National Aeronautics and Space Administration*, 2019. [Online]. Available: <https://mars.nasa.gov/mer/mission/overview/>. [Accessed: 14-Nov-2019].
- [212] P. Bigelow, “Waymo increases ‘rider only’ operations in Phoenix,” *Automotive News*, 2019.
- [213] H. Vardhan, “HD Maps: New age maps powering autonomous vehicles,” *Geospatial World*, 2017. .
- [214] C. D. Crane, “The 2005 DARPA Grand Challenge,” in *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA 2007*, 2007.

- [215] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge*. Springer, 2009.
- [216] M. Montemerlo *et al.*, “Junior: The stanford entry in the urban challenge,” in *Springer Tracts in Advanced Robotics*, 2009.
- [217] C. Urmson *et al.*, “Autonomous driving in Urban environments: Boss and the Urban Challenge,” in *Springer Tracts in Advanced Robotics*, 2009.
- [218] J. Byun, K.-I. Na, M. Noh, and S. Kim, “ESTRO: Design and development of intelligent autonomous vehicle for shuttle service in the ETRI,” in *Proceedings of the Workshop Planning Perception Navigation Intelligent Vehicles*, 2012.
- [219] N. Suganuma and T. Uozumi, “Development of an autonomous vehicle - System overview of test ride vehicle in the Tokyo Motor Show 2011,” in *Proceedings of the SICE Annual Conference*, 2012.
- [220] M. Brandão, M. Fallon, and I. Havoutis, “Multi-controller multi-objective locomotion planning for legged robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [221] T. Suleymanov, L. M. Paz, P. Piniés, G. Hester, and P. Newman, “The path less taken: A fast variational approach for scene segmentation used for closed loop control,” in *IEEE International Conference on Intelligent Robots and Systems*, 2016.
- [222] B. Lacerda, F. Faruq, D. Parker, and N. Hawes, “Probabilistic planning with formal performance guarantees for mobile service robots,” *Int. J. Rob. Res.*, 2019.
- [223] S. Chadwick, W. Maddetn, and P. Newman, “Distant vehicle detection using radar and vision,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2019.
- [224] M. Gadd and P. Newman, “A framework for infrastructure-free warehouse navigation,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2015.
- [225] S. H. Chen *et al.*, “Assistive control system for upper limb rehabilitation robot,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2016.
- [226] J. Paulo, P. Peixoto, and U. J. Nunes, “ISR-AIWALKER: Robotic Walker for Intuitive and Safe Mobility Assistance and Gait Analysis,” *IEEE Trans. Human-Machine Syst.*, 2017.
- [227] A. Jevtic *et al.*, “Personalized Robot Assistant for Support in Dressing,” *IEEE Trans.*

Cogn. Dev. Syst., 2018.

- [228] D. Park, Y. Hoshi, and C. C. Kemp, “A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder,” *IEEE Robot. Autom. Lett.*, 2018.
- [229] E. Martinez-Martin and A. P. Del Pobil, “Object detection and recognition for assistive robots: Experimentation and implementation,” *IEEE Robot. Autom. Mag.*, 2017.
- [230] M. Ficocelli, J. Terao, and G. Nejat, “Promoting Interactions between Humans and Robots Using Robotic Emotional Behavior,” *IEEE Trans. Cybern.*, 2016.
- [231] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, “Context-Aware Computing, Learning, and Big Data in Internet of Things: A Survey,” *IEEE Internet of Things Journal*. 2018.
- [232] T. Wenge, M. T. Chew, F. Alam, and G. Sen Gupta, “Implementation of a visible light based indoor localization system,” in *2018 IEEE Sensors Applications Symposium, SAS 2018 - Proceedings*, 2018.
- [233] H. Zheng, Z. Xu, C. Yu, and M. Gurusamy, “A 3-D high accuracy positioning system based on visible light communication with novel positioning algorithm,” *Opt. Commun.*, 2017.
- [234] A. Elfes, “Sonar-Based Real-World Mapping and Navigation,” *IEEE J. Robot. Autom.*, 1987.
- [235] A. Ibisch *et al.*, “Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded LIDAR sensors,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2013.
- [236] D. Withers and P. Newman, “Modelling scene change for large-scale long term laser localisation,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2017.
- [237] L. Lu, Y. Zheng, G. Carneiro, and L. Yang, *Deep Learning and Convolutional Neural Networks for Medical Image Computing: Precision Medicine, High Performance and Large-Scale Datasets*. Springer International Publishing, 2017.
- [238] D. D. Gutierrez, “Machine Learning and Data Science,” *Igarss 2014*, no. 1. pp. 1–5, 2014.
- [239] G. J. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-

- definition ground truth database,” *Pattern Recognit. Lett.*, 2009.
- [240] G. Pandey, J. R. McBride, and R. M. Eustice, “Ford Campus vision and lidar data set,” *Int. J. Rob. Res.*, 2011.
- [241] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: A benchmark,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009.
- [242] J. L. Blanco-Claraco, F. Á. Moreno-Dueñas, and J. González-Jiménez, “The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario,” *Int. J. Rob. Res.*, 2014.
- [243] M. Cordts *et al.*, “The Cityscapes Dataset,” in *Ieee Conference on Computer Vision and Pattern Recognition*, 2015.
- [244] M. Cordts, “Understanding Cityscapes: Efficient Urban Semantic Scene Understanding,” *PhD Thesis TU Darmstadt*, 2017.
- [245] Q. Du, J. Zhao, L. Shi, and L. Wang, “Efficient improved marching cubes algorithm,” in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, 2012, pp. 416–419.
- [246] J. Prinsloo and R. Malekian, “Accurate Vehicle Location System Using RFID, an Internet of Things Approach,” *Sensors*, vol. 16, no. 6, p. 825, 2016.
- [247] A. Geiger, J. Ziegler, and C. Stiller, “StereoScan: Dense 3d reconstruction in real-time,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2011.
- [248] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications,” *J. F. Robot.*, 2006.
- [249] Audi Electronics Ventures, “AEV Autonomous Driving Dataset,” *Audi*, 2019. [Online]. Available: <https://www.audi-electronics-venture.de/aev/web/en/driving-dataset.html>. [Accessed: 16-Nov-2019].
- [250] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, “The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset,” *arXiv:1909.01300*, p. 5, 2019.
- [251] A. Ligocki, A. Jelinek, and L. Zalud, “Brno Urban Dataset -- The New Data for Self-Driving Agents and Mapping Tasks,” *arXiv:1909.06897*, p. 7, 2019.

- [252] Q.-H. Pham *et al.*, “A*3D Dataset: Towards Autonomous Driving in Challenging Environments,” *arXiv:1909.07541*, p. 7, 2019.
- [253] Waymo, “The Waymo Open Dataset,” *Waymo*, 2019. [Online]. Available: <https://waymo.com/open/>. [Accessed: 16-Nov-2019].
- [254] Lyft, “Lyft Level 5,” *Lyft*, 2019. [Online]. Available: <https://level5.lyft.com/dataset/>. [Accessed: 16-Nov-2019].
- [255] M.-F. Chang *et al.*, “Argoverse: 3D Tracking and Forecasting With Rich Maps,” in *CVPR*, 2019.
- [256] F. Yu *et al.*, “BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling,” *arXiv:1805.04687*, p. 16, 2018.
- [257] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, “The ApolloScape Open Dataset for Autonomous Driving and its Application,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.
- [258] H. Caesar *et al.*, “nuScenes: A multimodal dataset for autonomous driving,” *arXiv:1903.11027*, 2019.
- [259] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 Year, 1000km: The Oxford RobotCar Dataset,” *Int. J. Robot. Res.*, 2016.
- [260] C. McManus, B. Uproft, and P. Newman, “Learning place-dependant features for long-term vision-based localisation,” *Auton. Robots*, 2015.
- [261] C. Linegar, W. Churchill, and P. Newman, “Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2016.
- [262] A. Angelova *et al.*, “Real-Time Pedestrian Detection With Deep Network Cascades,” *Bmvc2015*, 2015.
- [263] F. Diederichs, T. Schuttke, and D. Spath, “Driver Intention Algorithm for Pedestrian Protection and Automated Emergency Braking Systems,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015.
- [264] E. Rehder, H. Kloeden, and C. Stiller, “Head detection and orientation estimation for pedestrian safety,” in *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 2014.

- [265] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.
- [266] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [267] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.
- [268] K. Wang, X. Wang, L. Lin, M. Wang, and W. Zuo, "3D Human Activity Recognition with Reconfigurable Convolutional Neural Networks," *Comput. Vis. Pattern Recognit.*, 2015.
- [269] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *Proceedings - International Conference on Image Processing, ICIP*, 2015.
- [270] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, "Histogram of Oriented Principal Components for Cross-View Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2430–2443, 2016.
- [271] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1010–1019.
- [272] M. Moghimi, P. Azagra, L. Montesano, A. C. Murillo, and S. Belongie, "Experiments on an RGB-D wearable vision system for egocentric activity recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 611–617.
- [273] Y. Nam, S. Rho, and C. Lee, "Physical activity recognition using multiple sensors embedded in a wearable device," *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2, pp. 1–14, 2013.
- [274] J. Hernandez, Y. Li, J. M. Rehg, and R. W. Picard, "BioGlass: Physiological parameter estimation using a head - mounted wearable device," in *EAI 4th International Conference on Wireless Mobile Communication and Healthcare (Mobihealth)*, 2014,

pp. 55–58.

- [275] A. R. Doherty *et al.*, “Using wearable cameras to categorise type and context of accelerometer-identified episodes of physical activity,” *Int. J. Behav. Nutr. Phys. Act.*, vol. 10, 2013.
- [276] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, “Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos,” *Int. J. Comput. Vis.*, vol. 126, no. 2–4, pp. 375–389, 2017.
- [277] H. Kuehne, A. Arslan, and T. Serre, “The language of actions: Recovering the syntax and semantics of goal-directed human activities,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 780–787.
- [278] M. Barekatin *et al.*, “Okutama-Action: An Aerial View Video Dataset for Concurrent Human Action Detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, vol. 2017-July, pp. 2153–2160.
- [279] M. Monks, V. De-Silva, J. Roche, and A. Kondoz, “Adaptive Feature Processing for Robust Human Activity Recognition on a Novel Multi-Modal Dataset,” *arXiv:1901.02858*, p. 13, 2019.
- [280] J. Roche, V. De-Silva, and A. Kondoz, “A Multimodal Perception Driven Self Evolving Autonomous Vehicle,” *Except. to IEEE Trans. Cybern.*, p. 12, 2020.
- [281] P. T. Jardine and S. N. Givigi, “A Robust Model-Predictive Guidance System for Autonomous Vehicles in Cluttered Environments,” *IEEE Syst. J.*, 2019.
- [282] De Silva V. Kondoz A, “Agent-based modelling for driving policy learning in connected and autonomous vehicles,” in *Intelligent Systems Conference*, 2018, pp. 113–125.
- [283] R. H. Rasshofer, M. Spies, and H. Spies, “Influences of weather phenomena on automotive laser radar systems,” *Adv. Radio Sci.*, vol. 9, p. 21, 2011.
- [284] M. Kytö, M. Nuutinen, and P. Oittinen, “Method for measuring stereo camera depth accuracy based on stereoscopic vision,” in *Three-Dimensional Imaging, Interaction, and Measurement*, 2011.
- [285] S. Praveen, “Efficient Depth Estimation Using Sparse Stereo-Vision with Other Perception Techniques,” in *Coding Theory*, 2020.

- [286] R. C. R. C. Luo *et al.*, “Multisensor fusion and integration: Approaches, applications, and future research directions,” *IEEE Sens. J.*, vol. 2, no. 2, p. 12, 2002.
- [287] A. Dean, Ma. Morris, J. Stufken, and D. Bingham, *Handbook of Design and Analysis of Experiments*. 2015.
- [288] S. Nihtianov and A. Luque, *Smart sensors and MEMS: Intelligent sensing devices and microsystems for industrial applications: Second edition*. 2018.
- [289] Microsoft, “Microsoft Kinect Version 2,” 2019. [Online]. Available: <http://download.microsoft.com/download/f/6/6/f6636beb-a352-48ee-86a3-abd9c0d4492a/kinectmanual.pdf>.
- [290] Z. Cai, J. Han, L. Liu, and L. Shao, “RGB-D datasets using microsoft kinect or similar sensors: a survey,” *Multimed. Tools Appl.*, 2017.
- [291] H. Ben, C. Cruz, F. Boochs, and C. Nicolle, “From Unstructured 3D Point Clouds to Structured Knowledge - A Semantics Approach,” in *Semantics - Advances in Theories and Mathematical Models*, 2012.
- [292] Velodyne, “VLP-16 Velodyne LiDAR Puck: User Manual,” 2017. [Online]. Available: <https://velodynelidar.com/vlp-16.html>.
- [293] iPi Soft LLC, “iPi Mocap Studio.” Ottobrunn, Germany, 2019.
- [294] Keith Naughton and Mark Bergen, “It’s Aye, Robot, as Driverless Cars Finally Steer Near Showrooms,” *Bloomberg Technology*, Jan-2017.
- [295] A. Mavropoulos, “China aims to 10% - 20% autonomous cars by 2025!,” *Wasteless Future*, 2017.
- [296] State of California, “Testing of Autonomous Vehicles,” *Department of Motor Vehicles*, 2018. [Online]. Available: <https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/testing>. [Accessed: 10-Dec-2019].
- [297] V. Haltakov, H. Belzner, and S. Ilic, “Scene understanding from a moving camera for object detection and free space estimation,” in *2012 IEEE Intelligent Vehicles Symposium*, 2012.
- [298] R. Grewe, A. Hohm, S. Hegemann, S. Lueke, and H. Winner, “Towards a generic and efficient environment model for ADAS,” in *IEEE Intelligent Vehicles Symposium*,

Proceedings, 2012.

- [299] A. Wedel, H. Badino, C. Rabe, H. Loose, U. Franke, and D. Cremers, “B-spline modeling of road surfaces with an application to free-space estimation,” *IEEE Trans. Intell. Transp. Syst.*, 2009.
- [300] C. Vestri, D. Tsishkou, F. Abad, S. Wybo, S. Bougnoux, and R. Bendahan, “Vision-based safe maneuvers with detection of 10cm height obstacles,” in *17th ITS World Congress*, 2010.
- [301] F. Oniga and S. Nedeveschi, “Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection,” *IEEE Trans. Veh. Technol.*, 2010.
- [302] M. Schreier and V. Willert, “Robust free space detection in occupancy grid maps by methods of image analysis and dynamic B-spline contour tracking,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2012.
- [303] C. Lundquist, T. B. Schön, and U. Orguner, “Estimation of the Free Space in Front of a Moving Vehicle,” Linköping University Electronic Press, Automatic Control, Department of Electrical Engineering, Linköping University, 2009.
- [304] C. Fernández *et al.*, “Free space and speed humps detection using lidar and vision for urban autonomous navigation,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2012.
- [305] Q. Li, L. Chen, M. Li, S. L. Shaw, and A. Nüchter, “A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios,” *IEEE Trans. Veh. Technol.*, 2014.
- [306] M. Wu, S. K. Lam, and T. Srikanthan, “Nonparametric Technique Based High-Speed Road Surface Detection,” *IEEE Trans. Intell. Transp. Syst.*, 2015.
- [307] F. Dornaika, J. M. Álvarez, A. D. Sappa, and A. M. López, “A new framework for stereo sensor pose through road segmentation and registration,” *IEEE Trans. Intell. Transp. Syst.*, 2011.
- [308] J. Pazhayampallil *et al.*, “Free Space Detection with Deep Nets for Autonomous Driving,” *arXiv:1604*, 2014.
- [309] J. D. Crisman and C. E. Thorpe, “UNSCARF-a color vision system for the detection of unstructured roads,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2496–2501 vol.3.

- [310] J. Zhang and H.-. Nagel, "Texture-based segmentation of road images," in *Proceedings of the Intelligent Vehicles '94 Symposium*, 1994, pp. 260–265.
- [311] A. Broggi, C. Caraffi, R. I. Fedriga, and P. Grisleri, "Obstacle Detection with Stereo Vision for Off-Road Vehicle Navigation," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, 2005, p. 65.
- [312] N. Soquet, D. Aubert, and N. Hautiere, "Road Segmentation Supervised by an Extended V-Disparity Algorithm for Autonomous Navigation," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 160–165.
- [313] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognit.*, 2018.
- [314] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Machine Vision and Applications*. 2014.
- [315] I. Katramados, S. Crumpler, and T. P. Breckon, "Real-time traversable surface detection by colour space fusion and temporal analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009.
- [316] J. M. Alvarez, A. M. López, and R. Baldrich, "Shadow resistant road segmentation from a mobile monocular system," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007.
- [317] M. Felisa and P. Zani, "Robust monocular lane detection in urban environments," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2010.
- [318] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards End-to-End Lane Detection: An Instance Segmentation Approach," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018.
- [319] X. Zhao, Q. Zhang, D. Zhao, and Z. Pang, "Overview of image segmentation and its application on free space detection," in *Proceedings of 2018 IEEE 7th Data Driven Control and Learning Systems Conference, DDCLS 2018*, 2018.
- [320] J. Liu, L. Lou, D. Huang, Y. Zheng, and W. Xia, "Lane detection based on straight line model and K-means clustering," in *Proceedings of 2018 IEEE 7th Data Driven Control and Learning Systems Conference, DDCLS 2018*, 2018.

- [321] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [322] P. Wang *et al.*, "Understanding Convolution for Semantic Segmentation," in *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018.
- [323] E. B. David Welch, "Who's Winning the Self-Driving Car Race?," *Bloomberg*, 2018. .
- [324] D. of M. V. Californina, "Autonomous Vehicle Testing Permits (with a driver)," 2018. [Online]. Available: <https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/permit>. [Accessed: 10-Dec-2019].
- [325] M. J. Proulx, O. S. Todorov, A. T. Aiken, and A. A. de Sousa, "Where am I? Who am I? The relation between spatial cognition, social cognition and individual differences in the built environment," *Front. Psychol.*, 2016.
- [326] P.-W. Wang, P. L. Donti, B. Wilder, and Z. Kolter, "SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver," in *International Conference on Machine Learning*, 2019, p. 14.
- [327] D. G. T. Barrett, F. Hill, A. Santoro, A. S. Morcos, and T. Lillicrap, "Measuring abstract reasoning in neural networks," in *35th International Conference on Machine Learning, ICML 2018*, 2018.
- [328] J. Konečný and M. Hagara, "One-shot-learning gesture recognition using HOG-HOF features," *J. Mach. Learn. Res.*, 2014.
- [329] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," in *Proceedings - International Conference on Image Processing, ICIP*, 2011.
- [330] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOGgles: Visualizing object detection features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.
- [331] T. Yamasaki, "Histogram of Oriented Gradients (HoG)," *J. Inst. Image Inf. Telev. Eng.*, 2010.
- [332] I. The MathWorks, "Matlab." Natick, Massachusetts, 2020.

- [333] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.
- [334] A. A. S. Souza and L. M. G. Gonçalves, “2.5-Dimensional grid mapping from stereo vision for robotic navigation,” in *Proceedings - 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium, SBR-LARS 2012*, 2012.
- [335] V. De-Silva, J. Roche, and A. Kondozi, “Fusion of LiDAR and Camera Sensor Data for Environment Sensing in Driverless Vehicles,” *IEEE Sens. J.*, 2017.
- [336] H. Wang, Y. Cai, Y. Jia, L. Chen, and H. Jiang, “Scene adaptive road segmentation algorithm based on Deep Convolutional Neural Network,” *Dianzi Yu Xinxi Xuebao/Journal Electron. Inf. Technol.*, 2017.
- [337] T. Kühnl, F. Kummert, J. Fritsch, T. Kuhl, F. Kummert, and J. Fritsch, “Monocular road segmentation using slow feature analysis,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2011, pp. 800–806.
- [338] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang, “Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [339] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, “Large kernel matters - Improve semantic segmentation by global convolutional network,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [340] G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement networks for high-resolution semantic segmentation,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [341] Z. Wu, C. Shen, and A. van den Hengel, “Wider or Deeper: Revisiting the ResNet Model for Visual Recognition,” *Pattern Recognit.*, 2019.
- [342] G. Wang, P. Luo, L. Lin, and X. Wang, “Learning object interactions and descriptions for semantic image segmentation,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [343] J. Fu, J. Liu, Y. Wang, J. Zhou, C. Wang, and H. Lu, “Stacked Deconvolutional Network for Semantic Segmentation,” *IEEE Trans. Image Process.*, 2019.

- [344] P. Luo, G. Wang, L. Lin, and X. Wang, "Deep Dual Learning for Semantic Image Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [345] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation Liang-Chieh," *arXiv.org*, 2018.
- [346] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.
- [347] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, 2011.
- [348] S. Wang and G. Zhou, "A review on radio based activity recognition," *Digital Communications and Networks*. 2015.
- [349] R. Q. Mínguez, I. P. Alonso, D. Fernández-Llorca, and M. Á. Sotelo, "Pedestrian Path, Pose, and Intention Prediction Through Gaussian Process Dynamical Models and Pedestrian Activity Recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1803–1814, 2019.
- [350] H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-garadi, "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions," *Inf. Fusion*, 2019.
- [351] S. Das, "CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more," *Medium*, 2017.
- [352] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks.," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.
- [353] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3DmFV: Three-dimensional point cloud classification in real-time using convolutional neural networks," *IEEE Robot. Autom. Lett.*, 2018.
- [354] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, "Human activity recognition using multidimensional indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002.
- [355] S. Sangavi and B. A. M. Hashim, "Human Activity Recognition for Ambient Assisted Living," in *2019 International Conference on Vision Towards Emerging Trends in*

- Communication and Networking (ViTECoN)*, 2019, pp. 1–4.
- [356] C. Braunagel, E. Kasneci, W. Stolzmann, and W. Rosenstiel, “Driver-Activity Recognition in the Context of Conditionally Autonomous Driving,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015.
- [357] Z. Zhuang and Y. Xue, “Sport-related human activity detection and recognition using a smartwatch,” *Sensors (Switzerland)*, 2019.
- [358] N. Pittaras, F. Markatopoulou, V. Mezaris, and I. Patras, “Volumetric and Multi-View CNNs for Object Classification on 3D Data Supplementary Material,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2017.
- [359] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [360] Z. Wang, W. Zhan, and M. Tomizuka, “Fusing Bird’s Eye View LIDAR Point Cloud and Front View Camera Image for 3D Object Detection,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018.
- [361] Y. Cai, T. Zhang, H. Wang, Y. Li, Q. Liu, and X. Chen, “3D Vehicle Detection Based on LiDAR and Camera Fusion,” *Automot. Innov.*, 2019.
- [362] Y. Zhou and O. Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.
- [363] H. Mcgurk and J. Macdonald, “Hearing lips and seeing voices,” *Nature*, 1976.
- [364] H. Bourlard and S. Dupont, “New ASR approach based on independent processing and recombination of partial frequency bands,” in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, 1996.
- [365] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, “Multimodal deep learning,” in *International Conference on International Conference on Machine Learning*, 2011, pp. 689–696.
- [366] V. Ghasemi and A. A. Pouyan, “Human activity recognition in ambient assisted living environments using a convex optimization problem,” in *Proceedings - 2016 2nd International Conference of Signal Processing and Intelligent Systems, ICSPIS 2016*,

- 2017.
- [367] H. Tabatabaee Malazi and M. Davari, "Combining emerging patterns with random forest for complex activity recognition in smart homes," *Appl. Intell.*, 2018.
- [368] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [369] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "Imagenet," *Adv. Neural Inf. Process. Syst.* 25, 2012.
- [370] R. A. Fisher, "Inverse Probability," *Math. Proc. Cambridge Philos. Soc.*, 1930.
- [371] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, "InceptionNet/GoogLeNet - Going Deeper with Convolutions," *Cypr*, 2016.
- [372] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
- [373] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.
- [374] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [375] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, 2010.
- [376] S. Ceri *et al.*, *An Introduction to Information Retrieval*. Cambridge University Press, 2013.
- [377] P. Bhatia, *Data Mining and Data Warehousing: Principles and Practical Techniques*. Cambridge University Press, 2019.
- [378] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychol Rev*, vol. 94, no. 2, pp. 115–147, 1987.
- [379] E. Ronald, "Patterns of identity : hand block printed and resist-dyed textiles of rural Rajasthan," De Montfort University, 2012.
- [380] B. Tversky and K. Hemenway, "Objects, parts, and categories," *J Exp Psychol Gen*, vol. 113, no. 2, pp. 169–197, 1984.

- [381] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artif. Intell.*, vol. 17, no. 1, pp. 285–348, 1981.
- [382] D. Marr and H. Nishihara, *Representation and recognition of the spatial organization of three dimensional shapes*. Cambridge: Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1977.