

Tartu University

Faculty of Science and Technology

Institute of Technology

Martin Tammvee

**Human Activity Recognition Based Path Planning For Autonomous  
Vehicles**

Master's thesis (30 ECTS)

Robotics and Computer Engineering

Supervisor:

Prof. Gholamreza Anbarjafari

Tartu 2020

# Resümee/Abstract

## **Human Activity Recognition Based Path Planning For Autonomous Vehicles**

Human activity recognition (HAR) is wide research topic in a field of computer science. Improving HAR can lead to massive breakthrough in humanoid robotics, robots used in medicine and in the field of autonomous vehicles. The system that is able to recognise human and its activity without any errors and anomalies, would lead to safer and more empathetic autonomous systems. During this thesis multiple neural networks models, with different complexity, are being investigated. Each model is re-trained on the proposed unique data set, gathered on automated guided vehicle (AGV) with the latest and the modest sensors used commonly on autonomous vehicles. The best model is picked out based on the final accuracy for action recognition. Best models pipeline is fused with YOLOv3, to enhance the human detection. In addition to pipeline improvement, multiple action direction estimation methods are proposed. The action estimation of the human is very important aspect for self-driving car collision free path planning.

—.

**CERCS:** P170 Computer science, numerical analysis, systems, control; T125 Automation, robotics, control engineering; T111 Imaging, image processing

**Keywords:** — Neural Networks, self-driving car, object detection, human detection, human action detection, path planning

## **Isesõitvate Autode Tee Planeerimine Baseerudes Inimese Tegevuse Tuvastamisele**

Inimtegevuse tuvastamine video kaadrite pealt on populaarne ning laialdaselt uuritud valdkond teadlaste hulgas. Antud tehnoloogia areng võib tuua esile suure läbimurde humanoid robotite, meditsiinrobotite ja isesõitvate autode valdkonnas. Süsteem, mis oleks suuteline tuvastama inimesi ning nende tegevusi ilma tõrgete ja anomaaliateta, viiks isesõitvate autode turvalisuse järgmisele tasandile. Antud lõputöö käigus uuritakse mitmeid tehisnärvivõrgu mudeleid, mis on suutelised tuvastama inimete tegevust. Uuritavad närvivõrgud on erineva keerukuse astmega. Iga uurimise all olev tehivõrk treenitakse ümber, kasutades esitletud unikaalset andmekogumikku. Antud andmed on kogutud automatiseeritud juhtsõiduki peal, kus on kasutusel tänapäeva tipp-tehnoloogilised andurid. Antud andurid on tavapäraselt kasutatud teiste isesõitvate sõidukite peal. Uuritud närvivõrkude seast valitakse välja parima täpsusega võrk inimese tegevuse tuvastamisel. Parimat mudelit täistatakse YOLOv3 närvivõrguga, tehes efektiivsemaks inimese tuvastamist video pealt. Lisaks mudeli täiustamisele lisa närvivõrguga, lisatakse parimale mudelile uus iseärasus, mis on võimeline ennustama inimese tegevuse suunda. Inimese tegevuse suunda saab kasutada isesõitvate autode sõidu ohutuks trajektoori planeerimiseks.

—.

**CERCS:** P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria) ; T125 Automatiseerimine, robotika, control engineering; T111 Pilditehnika

**Märksõnad:** — Tehisnärvivõrk, isesõitev auto, objekti tuvastus, inimese tuvastus, inimese tegevuse tuvastus, trajektoori planeerimine

—

—.

# Contents

<b>Resümee/Abstract</b>	<b>2</b>
<b>List of Figures</b>	<b>6</b>
<b>List of Tables</b>	<b>7</b>
<b>Abbreviations. Constants. Generic Terms</b>	<b>8</b>
<b>1 Introduction</b>	<b>10</b>
1.1 Problem Definition . . . . .	10
1.2 Thesis Objectives and Aims . . . . .	11
<b>2 Machine learning and path planning</b>	<b>12</b>
2.1 Levels of Autonomy . . . . .	12
2.2 Motion Planning / Path Planning . . . . .	14
2.2.1 Terminology . . . . .	14
2.3 Neural Network . . . . .	15
2.3.1 Neural Network Structure . . . . .	16
2.3.2 Convolutional Neural Networks . . . . .	18
2.3.3 You Look Only Once (YOLO) . . . . .	19
<b>3 LboroHAR Data Set</b>	<b>21</b>
3.1 Data Acquisition . . . . .	21
3.2 Data Pre-Processing . . . . .	22
<b>4 Related Work</b>	<b>25</b>
4.1 Human Activity Recognition in Human-Robot Interaction . . . . .	25
4.2 Classification of Human Activity Recognition Approaches . . . . .	26

4.2.1	Sensor Based HAR . . . . .	26
4.2.2	Vision Based HAR . . . . .	26
4.3	Applications of Human Activity Recognition . . . . .	27
<b>5</b>	<b>Proposed Approaches</b>	<b>29</b>
5.1	Uni-Modal Approach . . . . .	29
5.2	Uni-Modal Approach with Skeleton Detection . . . . .	30
5.3	Multi-Modal Approach . . . . .	30
5.3.1	Getting The Pose of The Human . . . . .	31
5.3.2	Feature Verification . . . . .	33
5.3.3	Tracking Each Person . . . . .	33
5.3.4	Feature Extraction For Action Classification . . . . .	34
5.3.5	Action Classification . . . . .	35
<b>6</b>	<b>Result and further development</b>	<b>37</b>
6.1	Results . . . . .	37
6.2	Further Development . . . . .	39
6.2.1	Human Action Direction Estimation . . . . .	39
6.2.2	Anomalies . . . . .	43
<b>7</b>	<b>Conclusions</b>	<b>46</b>
7.1	Conclusion . . . . .	46
7.2	Future Work . . . . .	46
	<b>Bibliography</b>	<b>49</b>
	<b>Non-exclusive license</b>	<b>57</b>

# List of Figures

2.1	Basic structure of Neural network. . . . .	16
2.2	Simple architecture of CNN [12] . . . . .	19
2.3	YOLO network architecture [13]. . . . .	20
2.4	YOLOv3 compared to other object detection methods [15]. . . . .	20
3.1	Loughborough University London Autonomous Driving Sensor Test Bed. . . . .	21
3.2	Basic structure of neural network. . . . .	22
3.3	Output of conversion from 360 degree to 2D. . . . .	23
3.4	Five classes from LboroHAR data set. . . . .	23
5.1	OpenPose algorithm architecture [57]. . . . .	32
5.2	OpenPose predicted skeleton parts [58]. . . . .	32
5.3	Feature extraction steps. . . . .	35
6.1	Uni-Modal Approach Plot of Training and Validation Loss. . . . .	38
6.2	Uni-Modal Approach with Skeleton Detection Plot of Training and Validation Loss. . . . .	38
6.3	Multi-Modal Approach Confusion Matrix. . . . .	39
6.4	First method output. . . . .	40
6.5	Optical flow tracking neck. . . . .	41
6.6	Result of mean optical flow. . . . .	42
6.7	Anomalies on video frames. . . . .	43
6.8	Applying OpenPose only on YOLOv3 human bounding area. . . . .	44
6.9	New pipeline to detect human and their pose. . . . .	45
6.10	New pipeline outcome. . . . .	45

# List of Tables

2.1	SAE classification table. . . . .	13
2.2	Sheridan's scale of degree of automation [2]. . . . .	13
3.1	LboroHAR data set classes. . . . .	24
5.1	Frames per action. . . . .	31
6.1	Presented models average accuracy. . . . .	37

# Abbreviations, constants, definitions

**AGV** - Automated guided vehicle

**AI** - Artificial intelligence

**AU** - Action Unit

**AV** - Autonomous Vehicles

**CNN** - Convolutional Neural Network

**HAR** - Human Activity Recognition

**IoT** - Internet of Things

**LoA** - Level of Autonomy

**LSTM** - Long Short Time Memory

**mAP** - Mean average precision

**ML** - Machine learning

**MLP** - Multi layer perceptron

**MSE** - Mean Squared Error

**NN** - Neural Network

**PAF** - Part Affinity Fields

**PCA** - Principal Component Analysis

**ResNet** - Residual Neural Network

**SAE** - Society of Automobile Engineers



**SGD** - Stochastic Gradient Descent

**VAE** - Variational Autoencoder

# 1 Introduction

Human activity recognition ( HAR ) is wide field of study dedicated on identifying the specific movement or action of a person based on acquired data. Data can be gathered by multiple different sensors, depending on field of usage for HAR. Most common activities that are tracked are walking, standing and sitting. Actions can be more specific if model needs to be used in more narrow field, for example medicine.

This thesis contributes on finding the best model of HAR for self-driving cars and improving them with state-of-an-art techniques. Second part of the thesis is validating a new data set gathered by the most modern sensors in the field of self-driving car. Third part of the thesis is adding new features to the researched models as well as proposing various methods estimating humans movement direction in videos.

Next part of the introduction will introduce the main aspects of self driving car autonomy, motion planning of robotics and neural networks. Second paragraph will talk about the data acquisition and will introduce new data set suitable for HAR, that was gathered recently in Loughborough Univesity London. Third paragraph is going to introduce the related work done in the field of HAR. Moving on the forth paragraph will describe different models used in order to find out the best suitable model for self-driving cars. In fifth paragraph the results of models from section 4 is brought out and new methods how human can be tracked are analysed. The last section will suggest options how new features are implemented in order to improve the best model found from section four. In the end future work is discussed and conclusion is made.

## 1.1 Problem Definition

Autonomous cars can be allowed into public areas only when they are completely safe to humans. As the resources on the self-driving cars are limited, the procedure can not be computationally expensive, while at the same time it has to run fast and maintain the high accuracy.

Human action recognition has a huge impact on the safety of autonomous vehicles. In order to make a good model, the suitable data is needed. There is not much labelled data for human action classification in a field of self-driving car. Moreover for the autonomy not only the human and their action is relevant but also the direction of the action. There are not many proposed methods and researches what would be the best model of HAR for self-driving cars and only few them consider the direction of the human action.

## **1.2 Thesis Objectives and Aims**

Main objective of this thesis is to find the fastest and most accurate open source HAR model for self-driving cars. Each model under view is re-trained on the proposed data set and hyper parameters fine-tuned accordingly to achieve the best performance. For better accuracy of human prediction the best models pipeline is fused with YOLOv3 network. The best models is also supplemented with direction estimation of the humans action.

## 2 Machine learning and path planning

### 2.1 Levels of Autonomy

Autonomous ( also called self-driving, driverless or robotic) vehicle is a subject undergoing intense study. There has been studies showing how quickly self-driving vehicles are likely to be developed based on previous knowledge on robotic vehicles. It is estimated that autonomous vehicles will reduce heavily the traffic congestion, help to solve parking spot problems for cars and reduce air pollution while making driving more secure.

Robots are becoming more popular in our every day life and there are many reasons for that. Robotic machines can reduce and avoid various risk to human life. They may be customised based on the environment, for example we can build robots that can help us navigate under water or in space or in other hazardous environments. Another factor of robots rising popularity is their wide field of usage in scenarios where human life can be in danger, for example there are robots meant for bomb-disposal or rescue missions. Moreover, robots can have super-human capabilities - they are much more precise, stronger and have more durability than human beings. Robots have taken the task productivity to whole another level in industries compared to humans. Today's robots have already surpassed human in so many aspects like :

- They are able to respond faster to control signals and apply exactly the calculated force smoothly and precisely
- They are irreplaceable in performing repetitive and routine tasks
- Their ability to store, erase, making computational tasks is not comparable to humans and it is evolving every day
- They are capable of doing multiply parallel operation

The robot development is a huge motif for scientist and businessman. They are working daily to make new robots more advanced, that could replace work done by humans. The key to human

comfort and towards ideal word is having machines that are able to do every task human asks them, with minimal oversight. That is why machine autonomy is a large research topic at the moment in robotics.

The autonomous levels are being mainly regulated by two classifications : one from The Society of Automobile Engineers (SAE)(Table 2.1) and Sheridan’s classification (Table 2.2). SAE is meant for autonomous cars and Sheridan’s classification for machines/robots [1].

SAE classification	
Level 0	The driver has to do everything manually
Level 1	Machine has driver-assistance level. Only few actions can be controlled by machine (like steering or accelerating). All other functions are controlled by the driver.
Level 2	At least one of the driver assistance system is automated. Driver can take over the system immediately, by turning it off.
Level 3	In well known roads / areas the driver can activate the autonomous system, which can let him to turn away his/hers attention. NB! Driver must be prepared to take over the car when needed.
Level 4	Autonomous system is able to drive the car in most of the situation. While system is enabled, driver attention is not required.
Level 5	Driver need to set the destination and start the system. Cars’ autonomous system can drive to any location, where it is legal to drive.

Table 2.1: SAE classification table.

Sheridan’s classification table for robotics.	
Level 1	Human has to do all the tasks. There is no assistance from computer.
Level 2	The computer offers a complete set of actions alternatives, and
Level 3	narrows the selection down to a few, or
Level 4	suggest one, and
Level 5	executes that suggestion if the human approves, or
Level 6	allows the human a restricted time to veto before automatic execution, or
Level 7	executes automatically, then necessarily informs the human, or
Level 8	informs him/her after execution only if he asks, or
Level 9	informs him/her after execution if it, the computer, decides to.
Level 10	The computer decides everything and acts autonomously, ignoring the human.

Table 2.2: Sheridan’s scale of degree of automation [2].

Usually level of autonomy (LoA) is defined how much does human need to interact with the robot or machines. Therefore level of autonomy is mostly dependant on the sensor technology and how robot can use it in order to perceive its work space. Combing those two factors it can be concluded that robot task performance can be evaluated by the level of autonomy of the robot and the human-robot interaction. It consists of many difficult technological challenges to make it complete. Looking at a self-driving car as a robot, the sensor technology is considered

as a key challenge, where engineers and scientist have to develop new algorithms for fast and reliable interpretation of the environment and human intent. Scientists / engineers have not reached level 4 nor 5 (SAE classification) yet. So far the main reason of failure in autonomous vehicles has been the usage of single sensor stream, which can not be relied upon for HAR. This issue is trying to be solved by multi modal sensor usage and their data fusion. Scientist believe that this can make HAR much reliable in safety critical application. In order to research that field multi modal data set is required. So far there are not many multi modal data sets that can used for self-driving vehicles. This thesis introduces a data set that consist of data taken with most commonly used sensors on driverless vehicles in section two.

## **2.2 Motion Planning / Path Planning**

Every robot's task is defined with a goal where it needs to reach. To reach its goal it has to complete collision free motions for its complex bodies in different environments. Environment has some kind of static or dynamic obstacles that robots need to avoid during its way. This problem is called motion planning. Motion planning on self-driving cars is huge research area as well. The car needs to perform such manoeuvres and choose such paths that it can not harm humans for example. This relates path motion planning of a car and HAR.

Path planning consists of two main concepts that need to be defined : robot work space and whole work space where robot can manipulate. Robot work space holds an area where it is able to operate without moving its base joint and without breaking any of his joints. Whole work space is a room or area where robot is allowed to move. Robot overall work space can be defined as  $\mathbf{W} = \mathbf{R}^N$ , where  $N$  defines the dimensions of a room. The work space is a static environment populated with different obstacles. Objective is to find robot work space to move from initial pose to final goal pose with out any collisions [3–5].

### **2.2.1 Terminology**

Path planning and decisions that robot or autonomous car needs to make in urban environments, enable autonomous cars to find the safest, most convenient and most economical route from point A to point B. A path is considered as sequence of configurations with a certain start and ending point. So path planning is finding geometric path from from initial point to final destination, without violating any configurations. In order to follow the path, the autonomous car needs

to make manoeuvres. Manoeuvres are a high level characteristics of a vehicle's motion, which can be described with a mathematical model. For manoeuvre calculation the position and the speed of the car is required. To make the path planning with manoeuvres as efficient as possible manoeuvre planning is done. Considering the path validated by the path planning algorithm, manoeuvre planning takes the best high-level decision for a vehicle. For example where should the car turn or where it should go straight. To make a real-time planning of autonomous vehicle, reaching from one state to another trajectory planning is needed. Trajectory planning depends on the car's kinematic limits based on its dynamics, where the trajectory is calculated based on the car's velocity and time taken. To complete above tasks safely the car needs to have different safety avoidance features, like HAR and estimation the direction of the human action [3–5].

## 2.3 Neural Network

Within the last decade machine learning and artificial intelligence has become a part of a everyday life, whether using Google Translate or ordering a service on smartphone. The technologies are implemented also in various machines that make our everyday life easier by supporting or doing physically demanding tasks, in example robotic lawn mowers. Involving various technology near humans comes always with large safety criteria. One important benchmark is making sure that technology will not harm living beings. Latest mode shows that the robots should make the decision by themselves, by receiving data from miscellaneous sensors. Data is being analysed by the robot with various algorithms leading to final decision how robot should act.

Neural network is a part of artificial intelligent (AI), which creates a model or network of neurons similar to human brain. Network allows the computer to learn by analysing the data, which is given as a input to the system. The more accurate data is and the more samples of data is provided, the more precise prediction can the system make. These days there are many different artificial intelligent algorithms, that perform deep learning. The essential building block of an artificial intelligence is a perceptron, which is able to do signal processing. Multiple perceptrons are connected into a large mesh that forms a network. Unlike other programming task neural network is not so straight forward. In order to perform, they first need to learn the information. Machine learning (ML) is a subset of AI, that is meant to train a machine how to learn, while AI is a broad science mimicking human abilities. There are four learning strategies:

- Supervised learning - System is provided with labelled data set, that algorithm goes through to find patterns. Based on the data and the patterns the system is able to make

decision or classifications on a new input data.

- Unsupervised learning - Method is used in a situations where labelled data set can not be provided. Network is able to analyse the data set with algorithms that classify similar data into one class. When new data is fed to the system the cost function shows how far is it from network generated classes and to which class it is most similar to.
- Reinforcement learning - This approach is based on the network weights. The network weights are made higher if the prediction is positive or correct and when the prediction is false or negative its weights are decreased. That method forces the network to learn over time.
- Transfer learning - This approach uses pre-trained models with its weights on a new data. The problem that is trying to be solved can be slightly different than one used for the initial model. To satisfy the results one or more layer weights are modified from a pre-trained model, the layers that are untouched are frozen for learning.

### 2.3.1 Neural Network Structure

The basic neural network (NN) consists of artificial neurons that are grouped into layers. The most common NN structure consist of input layer, one or more hidden layer and with an output layer. Neurons send signals to each other by using complex algorithms. Every connection has a weight, which can weather activate the node or deactivate it. Figure 2.1 shows a network structure with inputs  $x_1$  to  $x_n$  being connected to neurons with weights  $\omega_1$  to  $\omega_n$  on each connection. Each signal is multiplied with its corresponding weight and summed together with other nodes. The output of the summation is passed through activation function to receive the final output,  $Y$ , of the network. Most frequently used activation functions are ReLu and sigmoid.

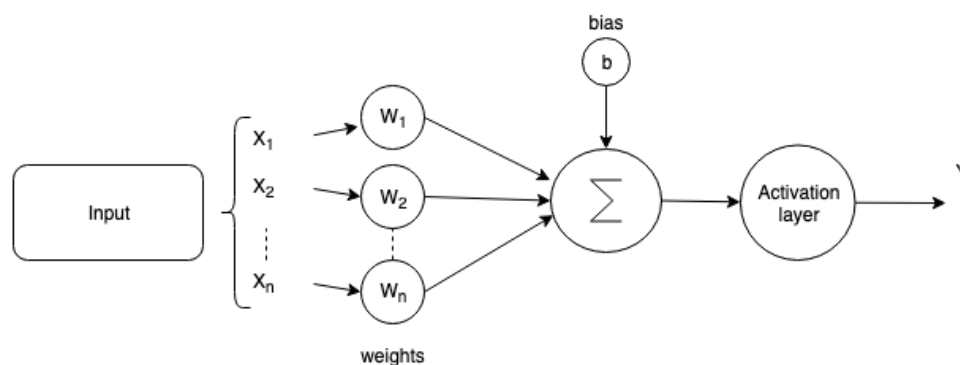


Figure 2.1: Basic structure of Neural network.



Based on the basic NN model there are many different models that vary based on their usage. For example some network types perform better on images while other better on alphabet characters. As this research is focused on the images and data points, described network types below are being used:

- Feed Forward (Deep Feed Forward) - This network type is most similar to basic neural network structure. Network follows three main rules where all the nodes are fully connected; activation flows from the input to output; there is one ( or if deep feed forward more ) hidden layers between input and output. Most common way to train this particular kind of NN is back propagation, where it is used to calculate the contribution of each node based on the error and correct them in order to reduce error in next iteration. [6]
- Recurrent Neural Networks (RNN) - This network type has instead of regular nodes recurrent nodes. Network is capable of using previous information to the present task. It means that each hidden node receives its own output as an input on the next iteration. RNN is used when decisions from the past have an impact on the current node state. [7]
- Long / Short Term memory (LSTM) - This network type is special type of RNN, where each hidden node has its own memory, which can process data when data has time gaps. This gives the network opportunity to learn long-term dependencies. LSTM has the ability to process a video frame, taking into account an output that was predicted many frames ago. The memory of each node or if to be more accurate each node cell state is able to decide whether to activate the cell or not. Cells' state decides what shall be stored in the cell and finally what will be output from the cell.

As complicated hardware systems have latency lags caused by the data transfer or reaction time of a motor etc. , this network type is very useful to fill in the gaps, that can be missed due to latency caused by outside factors. [8]

- Residual neural network (ResNet)- Residual neural network is a best solution for vanishing gradient problem. Vanishing gradient problem occurs in a deep neural networks where after multiply iterations of a chain rule, the loss function calculated becomes zero. This results on the weights never updating and no learning is performed. To overcome ResNet has skip connection in its model, where several layers are skipped during learning procedure, while the gradient will get passed through the skip connection backwards from previous steps to initial filters. [9]

### 2.3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are most popular while working with images and are widely used for object detection. For example they are used to detect humans, face expressions, roads from satellite images, text from hand written letters and much more. CNN-s consists mainly of three types of layers : convolutional layer, pooling layer and fully connected layer.

First layer to extract features from the input is done by convolutional layer. For that the image matrix is multiplied with the filter matrix. Convolution with a different filter will extract out different features. For example there are filters for edge detection, sharpening, blurring and much more. In order to move the filter around the input image matrix, strides are used. When the stride is set to one, it means the filter will move one pixel at the time and so on if the stride is higher.

As different filters are with different size, then not always does the filter fit perfectly onto the input matrix. In order to overcome this problem padding is used. There are two type of padding that are used.

First method is to pad the input matrix with zeros, so that it would fit with the filter. Second method is to drop the part of the input matrix, where the filter matrix does not fit. Second method will keep only the valid part of the input image.

Purpose of the pooling layer is to reduce the number of the parameters. Spatial pooling is used to reduce the dimensionality of each map, while keeping all the important features. There are three different spatial pooling types : max pooling, average pooling and summation pooling. Max pooling method keeps the highest value from the enhanced feature map. Average pooling keeps the average value over the rectified region and sum of all values are called summation pooling.

Fully connected layer is used for final classification of the complex features gained from the convolutional and pooling layers. Features are flattened and merged into a vector. The vector is being passed forward to the fully connected layer. The network where the feature vector is passed has a structure where, each node is connected to all of the nodes on the next layer. Each node weights are initialised by a small random value and are trained using back-propagation. The final layer is activation layer, usually softmax, which computes the probability of each class from the given features ( [10–12])

Figure 2.2 shows the simple architecture of the convolutional neural network.

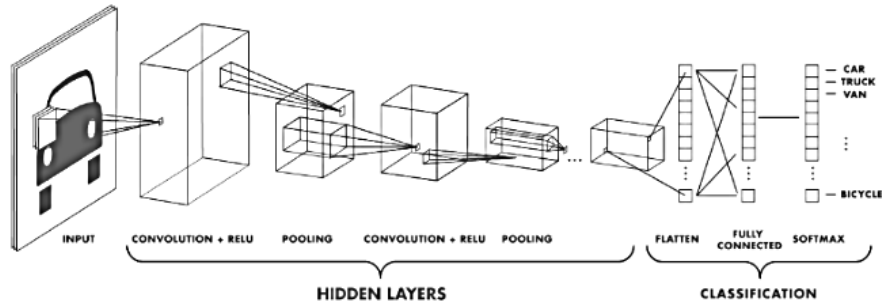


Figure 2.2: Simple architecture of CNN [12]

### 2.3.3 You Look Only Once (YOLO)

YOLO (You Look Only Once) is a network for object detection. Network is determining and classifying the location of a certain objects on the image. Other methods for object detection use a pipeline to perform the task in multiple steps. It slows down the network and is hard to optimise, because each individual object needs to be trained individually. YOLO applies a single neural network to a whole image. Image is divided into multiple regions. In each region the network predicts bonding boxes and probabilities of an objects. Bounding boxes are weighted by the predicted probabilities [13].

YOLO network architecture consist of 24 convolutional layers followed by two fully connected layers. Some convolutional layers use  $1 \times 1$  reduction layers alternatively to deuce the depth of the feature maps. Between the predictions and ground truth the sum-squared error is used. The loss function is integrated of classification loss, localisation loss (error between predicted boundary box and ground truth) and confidence loss(fact of being an object in a box).The final output of the networks is  $7 \times 7 \times 30$  tensor of predictions [13]. The architecture of YOLO can be seen from figure 2.3.

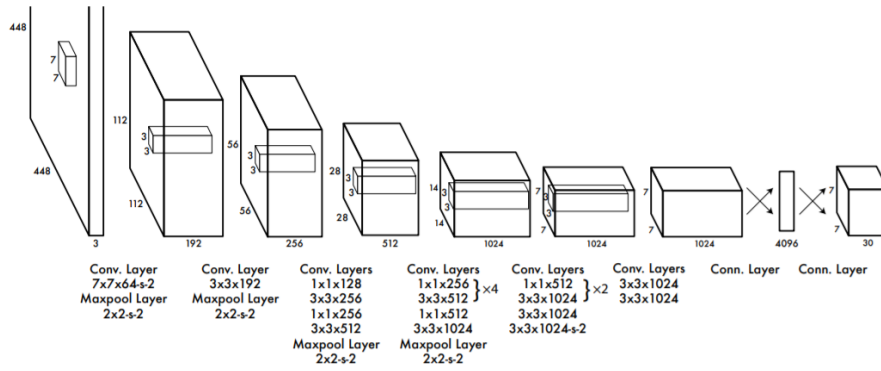


Figure 2.3: YOLO network architecture [13].

The latest version of YOLO network is called YOLOv3 Darknet. This state-of-the-art method is the fastest and most accurate object detection algorithms at the time being. From the figure 2.4 it can be seen that YOLOv3 runs significantly faster than any other object detection methods. YOLOv3 allows a trade of between speed and accuracy, simply by changing the size of the model.

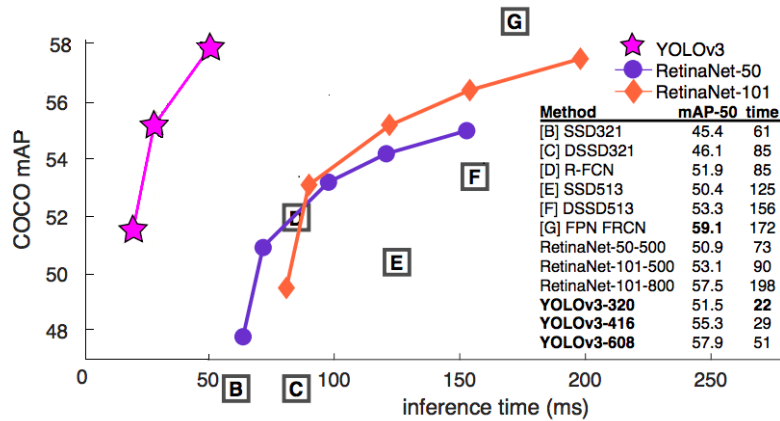


Figure 2.4: YOLOv3 compared to other object detection methods [15].

In the latest model of YOLO the multi-scale prediction and better backbone classifier are introduced. The changes compared to the other version are made in bounding box prediction, class prediction, predictions across the scale and in feature extractions [14].

## 3 LboroHAR Data Set

### 3.1 Data Acquisition

During this work, an unique data set, LboroHAR [16], is used for training, validating and testing on different models of neural networks. The data was gathered in Loughborough University London. Fact that makes this particular data special is that it was recorded on the autonomous ground vehicle test bed (shown on the figure 3.1 ) with three different sensors : RGB-D camera, LiDAR and 360 degree camera. The data represents the actual footage that today's automated guided vehicles (AVG) are capable of recording. Most of today's AGV-s have weather a LiDAR or RGB camera to detect humans. So this data set can give us opportunity for further research. In order to enhance proposed neural network models accuracy, different data types can be fused together, representing the same outer condition.

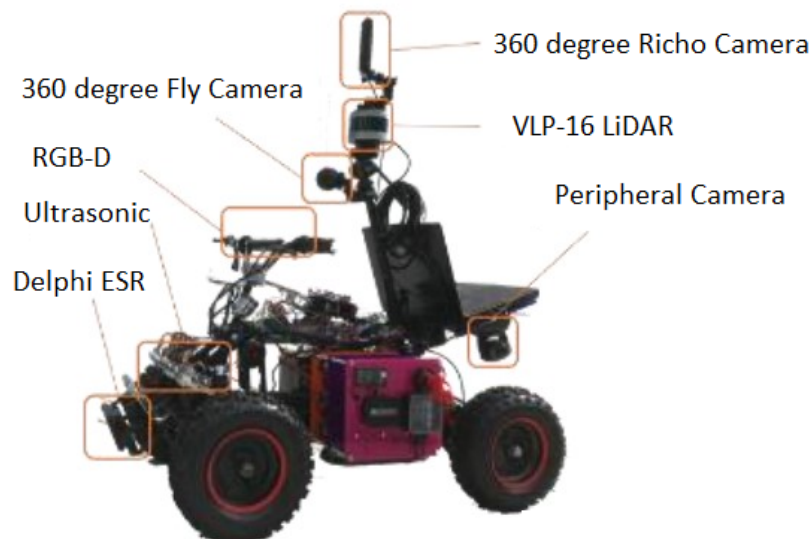


Figure 3.1: Loughborough University London Autonomous Driving Sensor Test Bed.

The proposed LboroHAR data set has sixteen different participants, doing nine different activities. The data was collected during 17.06.2018 and 18.06.2018. All of the activities take

place indoors. Each person carries out at least four different activities, with the maximum video length of three minutes. Each scene starts with a person coming to the centre of the frame and raising his or her hands. Then the person who is being recorded performs one of the nine activities and finally every scene ends with a person coming back to the centre of the frame and raising his or her hands. The initial nine activities gathered are : sitting on the office chair, standing and texting on the mobile phone, sitting on the stool, lying on the couch, walking, walking and texting on the mobile phone, carrying different objects, pulling different objects, running.

## 3.2 Data Pre-Processing

This research is focused on the 360 degree camera footage. The raw data consisted of dual fisheye video files (shown on the figure 3.2). In total there are 133 videos recorded over the two days.



Figure 3.2: Basic structure of neural network.

Main objective of the research is to detect person and classify their action, therefore only the front view of the data is taken under consideration. In order to transform 360 degree dual fisheye camera video footage to 2D video, the rear view has to be cut out. For that FFmpeg [17] software is being used. Setback of doing the process was that the resolution dropped after the conversion. The result of the conversion can be seen from image 3.3

For applying the conversion for all of the 133 initial videos, a special script was conducted, that looped for a folder containing raw-videos and outputting to result folder. *Dfisheye* command was used to decompose the dual-fisheye frame with padding of 1% . To enhance the output frame quality the chroma and luminance values were fused. To make the output more sharp and smooth cubic interpolation was used.



Figure 3.3: Output of conversion from 360 degree to 2D.

Image interpolation is a method in image processing where different algorithms are used in order to guess the missing pixels values after re-scaling or remapping the image. Bicubic interpolation is one of the image interpolation algorithms that considers the closest 4x4 neighbourhood of known pixels, total of 16 pixels, to assign unknown pixel a value. As the area taken into account is rather big, closer pixels are given higher weight in the calculation.

In order to retrain machine learning models the data needed to be labelled. The nine initial classes would be too narrow for action classification. Instead new more general classes were created :

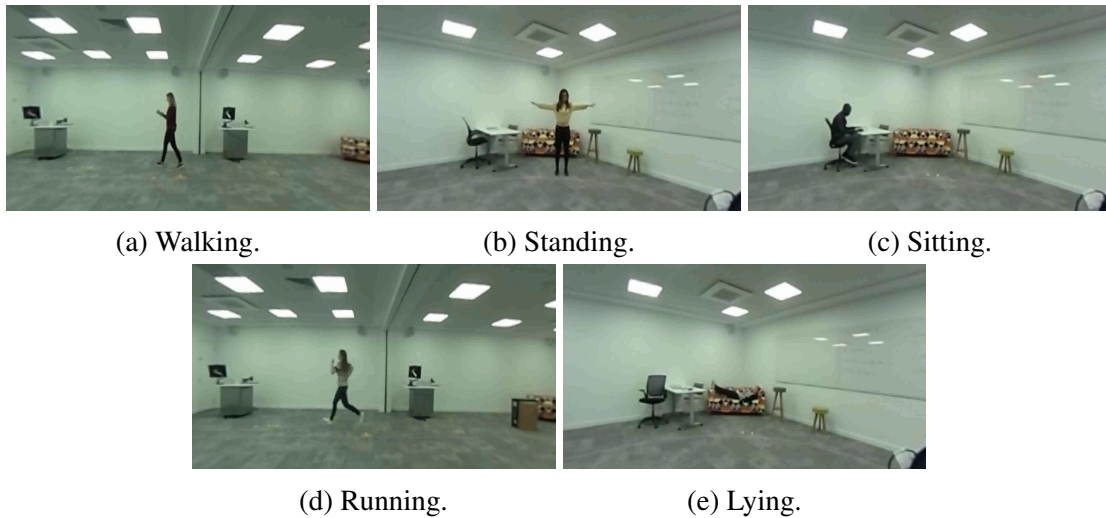


Figure 3.4: Five classes from LboroHAR data set.

All the data needed to be labelled manually. The labelling process for this research meant giving a label to each frame. In order to do that a special script was done that would separate all the frames from 133 raw 2D videos. In the end total of 136 710 frames were gained. Table 3.1 shows how many samples each class has and each class distribution over the total number

of samples. Output image size was 1080 x 1920 pixels.

Class	Total of frames	Distribution
Walk	39 558	29 %
Stand	68 356	50 %
Sit	20 447	15 %
Run	1177	1 %
Lie	7172	5 %
<b>Total</b>	<b>136 710</b>	<b>100 %</b>

Table 3.1: LboroHAR data set classes.

From the last column of the table 3.1 it can be seen that the classes are not distributed equally. This fact can make a big impact on the accuracy of correct action classification. Neural network models can perform poorly on classes that are under represented because it will not have enough data to learn specific features of that particular class. For research purposes, all those classes are maintained to see the actual outcome on different machine learning models with given data set and classes.



## 4 Related Work

HAR research is a part of computer vision science where images and videos are processed. Machines analyse sensor data streams, to detect, recognise and estimate human activities. HAR has a huge role in many emerging applications such as autonomous vehicles, business analytics, personal assistant robots.

### 4.1 Human Activity Recognition in Human-Robot Interaction

Robots are becoming more common in human prosaic life as for example domestic robots are starting to get irreplaceable in our rushy lifestyle. To make robots compatible with new situations without causing any damage to environment, human activity recognition is inevitable. Coming to self-driving car, HAR is unavoidable. Only when the car understands where are the humans and what is their activity, it is able to safely navigate through populated areas without having any external guidance.

Research done in Chinese University of Hong Kong [18] introduces convolutional neural network to classify 3D human activities for mobile robots. Developed model was trained on Vicon Physical Action data set [19]. The model was tested on new data to validate its performance in new circumstances and environment.

In hazardous environments or tasks where high precision is required, robots outperform humans. HAR is used to make the robot move based on the human action or for mimicking humans. Research in Galatasaray University [20] proposed a method how to control a robot based on a HAR. During the experiment the human action was tracked by wearable sensors. Special neural network was conducted to classify the action from the sensor data. Based on the network output and task based function, robot movement was performed.

## **4.2 Classification of Human Activity Recognition Approaches**

Human activity recognition can be classified into multiple research branches. Most popular branches are sensor and vision based. HAR based on sensor data can be separated into three sub-branches regarding sensor's deployment: based on wearable sensors, object tagged and dense sensing [21].

### **4.2.1 Sensor Based HAR**

HAR based on wearable sensors is very attractive research topic mainly because of its application areas. Named method is widely tested in healthcare and smart environments. Main sensors that are being used to gather data are accelerometer, gyroscope and magnetometer. Sensors are being attached on the test object or person to log the data while certain activities are being done. Further on, different neural network approaches are used to classify the action [22].

Wearing a sensor can be impractical and there is not much data available. Therefore the method is not used very often for autonomous vehicle HAR. Researchers are putting more effort on device-less (dense sensing) methods. The device-free method is more practical, because it does not require the human to wear any sensor device while performing the activity. The sensors are placed around the test area where they able to monitor the environment where the action is being performed. Many different sensor can be used like motion sensor, pressure sensor, temperature sensor etc. For example there is study [23] to monitor elderly people at home. The focus of this study is to analyse elderly people daily routine and the detect the changes of the routine that can be a an early sign of some kind of disease. For capturing the activities infrared sensors are set up around the flat which data are later used for HAR analyses.

### **4.2.2 Vision Based HAR**

The most popular HAR method is based on the camera footage. The reason for that lies behind the amount of data available and small cost of the sensors needed. There is a lot of open source data that can be used for human action classification [24–26]. Although those named data sets have to be pre-processed in order to label the action for example.

LiDAR is widely used on autonomous cars to detect the surroundings of a vehicle. Research is made detecting pedestrians from raw LiDAR point cloud data [27]. Article presents an algorithm using single template matching with kernel density estimation clustering for hu-

man recognition. Proposed algorithm is able to segment out individual person from crowd and distinguish them from other object from real world. For further development there are brought out methods, how to apply HAR on LiDAR point cloud data [16].

Vision based HAR can be applied using different methods on different input data. Research has been done [28] on applying HAR with uni-modal and multi-modal methods. Uni-modal methods use data from single modality, where human activities are represented as a set of visual features extracted from a video. Multi-modal approaches use input from different sources. Event of an action can be described by different type of features or even fusion of multiple features.

### **4.3 Applications of Human Activity Recognition**

Depending on the hardware used for human activity recognition, this methodology is being used in various fields for different purposes.

HAR is often used for healthcare purposes. In addition for monitoring elderly people activity (paragraph 3.2.1) there are medical facilities using HAR for detecting their patients activity. Information is provided to medical staff, which will save their time and energy by knowing where the help is needed the most [29]. In hospitals and health care facilities HAR is also used for patient health monitoring, which helps to assist and reduce the risk of many diseases like diabetes and cardiovascular [30].

HAR is also used for security purposes. Abnormal activities can be recognised effectively and used for prevent criminal acts. Research [31–33] has shown how effectively the HAR can be used in airports, shopping malls, banks, railway station to prevent and detect crimes.

Action recognition is also being used in field of entertainment. Microsoft Kinect for example uses HAR to accurately interpret human action for the game and the player interaction [34]. Facebook founder Mark Zuckerberg has stated that Virtual Reality (VR) will be next technological boom over the upcoming decade. HAR plays a huge role in VR [35–37], where human body and action are being constantly tracked.

Robots are being widely used in different type of industries. Each robot has its strict working boundaries and are often restricted with safety fences. However safety fences are expensive and do not often let humans to cooperate with robots. There as [38] introduces a method for safe human-robot interaction. Algorithm lets human and the robot work together while following strict safety rules. HAR is used for human tracking and to guarantee safe working region for

the robot.

Human activity recognition is a key factor in a field of autonomous driving. There are studies ([39–41]) showing how car driver actions are being analysed in different situations. The result of the analyses are adjusted and implemented on a self driving car. Hot-topic in research is how HAR can be applied more in autonomous cars. One possibility is to use it for autonomous vehicle trajectory planning [42–44]. Having premonition of surrounding people activities will take self-driving car path planning onto another level considering safety features.

A lot of autonomous robots are meant to be driven only on the sidewalk or indoors and not on the roadway. As for humans, this task is not so trivial for robots. A lot of different aspects needs to be taken into account in order for robot to drive safely in these complicated situations. It does not only need to acknowledge the surroundings of itself but consider the human and their movements as well.

For robot to be able to navigate safely it needs to understand its surroundings. Based on the information it receives it needs to find the best path without any collisions. Research [45] gives an overview of path planning algorithms that are used for autonomous robot navigation while finding the shortest path suitable. It is brought out that classical methods, that require a large memory and computational power, do not perform best in real world. The mobile robot path planning based on the predefined map fails with dynamic environment. In real world heuristic approaches perform much better where multiple algorithms are fused.

This research [46] contributes on human movement tracking and analysing it. By the wearable sensors humans were tracked while walking around the buildings. A model was created based on the data gathered that estimates the natural movements of the person. The model could be used in many different application as well as for autonomous vehicles for understand where the human is heading. Knowing the direction of an action of a person, help autonomous car to calculate a safe route for itself, while not harming anyone.

Every human needs personal space and has a right for it. Autonomous vehicles need to obey the rule as well. In researches [47] and [48] autonomous robots take into account human private space while planning their path. Algorithms presented are dynamic in a sense that the robots path can change in real time as the humans move. The algorithm is able to detect multiple humans and track their activities via HAR. The robots path is being calculated considering the action of the human and so that the robot will leave the needed private space for each individual.

# 5 Proposed Approaches

This section of the thesis will validate different algorithms and approaches for HAR implementation on proposed data set. For that three different models, with different computational complexities, are being viewed and tested. Models were retrained on proposed data set (introduced in paragraph 2). Data was divided into three sets: training, validation and test set. Training set formed 60 % , validation set 30 % and testing set 10 % of the whole proposed data set. Goal is to test different methods and approaches to find the model with the best performance for HAR in autonomous vehicles.

## 5.1 Uni-Modal Approach

First model under consideration would be rather simple neural network [49]. This model was tested because the author reported that he got the prediction accuracy of 91.27 % on the UFC-101 data set [50]. Named data set is gathered especially for action detection, where it has 101 action categories. Data is collected from the YouTube and has 13320 videos. The action categories can be divided into five types : human-object interaction, body-motion, human to human interaction, playing musical instruments and doing sports. In order to evaluate HAR, video frames are feed into LSTM model as an input. The model was re-trained with proposed data set, where data was split 60 %, 30% and 10%, respectively for training, validation and testing. Network input image size needed to be resized to 512 pixels. LSTM model had one hidden layer of size 1024 followed by batch normalisation with appropriate linear transformations. For activation Relu function followed by Softmax activation function was used. The output of the model was probability of each action class.

## 5.2 Uni-Modal Approach with Skeleton Detection

For the second model [51] [52] more complicated pipeline is used. First part of the model is human pose estimation. The human pose is based on rather simple pose estimation algorithm [52]. The model is ideal for application where low latency is required, especially on self-driving cars where every millisecond of latency can be fatal for a living being. Model extracts eighteen features from human pose: left eye, nose, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, right ankle and neck. Network extracting the features is based on ResNet [53], which is one of the most common backbone residual neural networks for image feature extraction. Transfer learning techniques are used for deconvolutional layers and are added to the last convolutional layers in the ResNet. All in all three deconvolutional layers with batch normalisation and ReLU activation are used. Each layer has 256 filters with 4x4 kernel with stride two. For loss Mean Squared Error (MSE) is used. For optimisation Adam function with learning rate of  $1e^{-6}$ . The model was initially trained on the COCO data set [54]. Coco data set contains a large-scale images for object detection. It has over 330 000 samples, while around 200 000 of them are labelled. The model outputs points of human body. Each point is described with position. Points that can be connected together in order to form a skeleton are predefined. For example left elbow and left shoulder will be connected together while for example left shoulder and right eye will not be connected. Output of the first pipeline part would be a human skeleton.

Next part of the pipeline uses the output of the previous step as an input to a classifier, where the action is estimated. The author used [51] used logistic regression classifier. After retraining the model with proposed data set the initial classifier performed poorly. Two other classifiers were tested, multi-layer perceptron classifier and stochastic gradient descent (SGD) classifier, where SGD outcome proved to suit the best for proposed data set.

## 5.3 Multi-Modal Approach

Third model that was researched [55] was trained on videos taken inside a room with a camera. The recorded material was with resolution of 640x480 pixels and with a frame rate of 10 (fps). Action classification was made between 9 different actions: waving, standing, punching, kicking, squatting, sitting, walking, running, jumping. The videos were from 0.8 seconds to 2

minutes long. All in all there were more than 11 000 frames to train the model. The table 5.1 shows how many samples did each action have.

Action	Wave	Stand	Punch	Kick	Sqaut	Sit	Walk	Run	Jump
Number of Frames	1239	1703	799	1162	964	1908	1220	1033	1174

Table 5.1: Frames per action.

The data that third model was trained on, was closest to the proposed data. Both data sets were gathered for HAR purposes inside a room. Although the classes varied, both data sets had more or the less same number of samples for each class.

### 5.3.1 Getting The Pose of The Human

Algorithm to classify human action consists of multiply sub-algorithms.

For the first step algorithm detects human skeleton with OpenPose algorithm. Skeleton of the person is visualised by coordinates, where a right order of coordinates form a specific joint. Thus not all combinations of coordinates form a joint. The combination, that will form a specific joint, is defined by the user. For example coordinates that show the position of the elbow and the wrist will form a joint called hand. In order to get more accuracy on prediction the specific joint, more coordinates can be taken into consideration. In example coordinates showing the position of left shoulder, left elbow, left wrist and left hand palm will form a left hand of a human being detected by the system. In order to fasten up the algorithm for multi-person pose detection, mainly two methods are used : bottom-up and top-down.

Bottom up approach detects first all human parts on the image and then groups joints belonging to individual person and estimates the pose. Top-down approach detects first all the humans on the image, followed by finding joints on each separated human and then estimating the pose for each skeleton. Top-down approach is normally easier to use, because adding person detection takes less effort than adding grouping algorithm. Performance wise these two methods are equal. [56]

As this work is focusing mostly on self-driving cars, therefor multi-person algorithms are taken under consideration.

OpenPose is one of the most popular bottom-up methods. Algorithm first detects the joints associated to all persons on the image, subsequently joining joints to unique person.

The video file or camera feed is given as an input into the system. Camera feed means that this algorithm can be used in real time, for example on a self driving car. OpenPose first

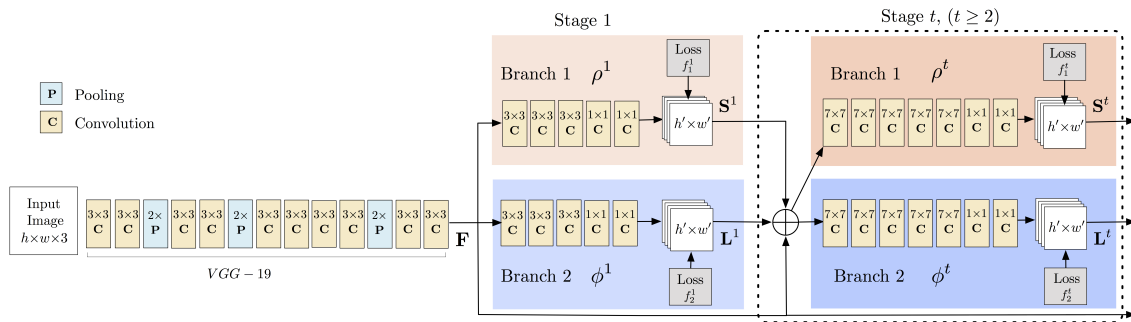


Figure 5.1: OpenPose algorithm architecture [57].

extracts features from an image using the first layers (VGG-19, shown on figure 4.1). Extracted features are then fed into the two parallel branches of convolutional layers. Upper branch from the figure 5.1 predicts set of eighteen confidence maps. Each map representing coordinates of a particular part of the human pose skeleton. The branch below predicts set of 38 Part Affinity Fields(PAFs) which represents the degree of association between parts. In other words OpenPose is using CNNs to predict two heat maps. The first heatmap or the upper part of the branch shown on the figure 4.1, is predicting the positions of skeleton parts and the bottom branch is joining the predicted parts into one human skeleton. Repeating the procedure will lead to more coordinate points and will improve the accuracy of the predictions made by each branch. Using the part confidence maps, bipartite graphs are formed among pairs of parts. Using the PAF values, weaker links in the bipartite graphs are neglected. From all the above steps, human pose skeletons can be estimated and assigned to every person in the image. In OpenPose algorithm each human skeleton has 18 joints, shown on image 5.2

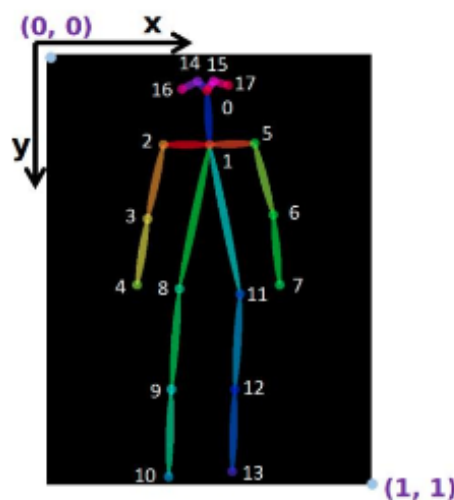


Figure 5.2: OpenPose predicted skeleton parts [58].



### **5.3.2 Feature Verification**

OpenPose algorithm detects in total 18 features ( shown on the figure 4.2 ), where five of the features represent the human head area. Head movement does not affect the action classification for self driving cars and therefore some of the points are being dropped. As a matter of fact this simplifies and makes the model faster for the next steps. In total 13 features are left: neck, left shoulder, left elbow, left hand palm (last three form a left hand), right shoulder, right elbow, right hand palm ( last three form a right hand), left knee, left ankle ( last two form a left leg), right knee, right ankle (last two form a right leg), left thigh and a right thigh.

From the human skeleton - body velocity, joint velocity and normalised joint position are extracted. Every point has x and y value, but OpenPose outputs them with a different unit. In order to work with them coordinates are scaled to be the same unit. After the previous steps have been completed, the algorithm now verifies weather the detected skeleton has a neck and at least one thigh. If one of these components are missing, given frame becomes incompetent and no prediction is made on particular frame.

In addition for the action classification, other joints must be checked as well. If OpenPose could not predict all the joints for the current frame, missing joints have to be filled. This is necessary for the feature classification, where the input has to be fixed-size feature vector. To find the missing joints the previous frame is taken under consideration. Algorithm compares the skeleton on the current frame to a skeleton on a previous frame. The comparison is made based on the coordinates of the neck. The missing joints can be carried forward from the previous frame, if the position difference of the neck in the consecutive frames is less than a set threshold. For example if the left ankle is missing on the current frame and the skeleton match is found, where on previous frame left ankle exists, given feature is transferred to a current frame with a previous frames coordinates respect to the neck. No skeleton is being predicted, if any of the thirteen features are not predicted on the current frame or the skeleton match between two consecutive frames cannot be found.

### **5.3.3 Tracking Each Person**

With self-driving cars we are interested in videos or consecutive images, where the human pose must be detected and the action tracked. In order to track a human pose in consecutive frames euclidean distance is used. The distance is calculated between the coordinates of two skeletons from a previous and current frame. If the distance between two skeleton is lower than the thresh-

hold defined, human identifier from the previous frame is transferred to current frame. A new identifier is being set to a human skeleton when there is no match found between the skeleton coordinates in the two consecutive frames. This mean that a new person has entered a frame. Numeration of the skeleton is initially given based on the human position on the image. The lowest number is acquired to the skeleton which is closest to the centre. The centre is defined by midpoint of the frame. This is import in order to be able to track the most dangerous situation for the car. The most dangerous situations are when human are just in front of the car.

### 5.3.4 Feature Extraction For Action Classification

Previous body part positions are used in order to extract custom features that are used for action classification. The algorithm takes in consecutive five frames and concatenates them. This means for the first 4 frames of the video, no action is detected. Thus if the video frame rate is ten frames per second, it takes around half a second before the action can be estimated.

Skeleton joint positions is the first feature for the action classification. Let  $X_s$  be a direct concatenation of skeleton joints positions, then it has dimensions of:

$$X_s(Dimension) = 13 \text{ joints} \times 2 \text{ position / joint} \times N = 130 \quad (5.1)$$

*, where  $N = 5$  frames*

Very important feature for this model is human height. Human height (H) is calculated from the skeleton neck position to skeleton thigh position. Height is used in order to normalise the extracted features. This feature has  $Dimension = 1$ .

The velocity of the body, as the next feature, is derived based on the skeleton neck. Skeleton neck velocity is normalised by diving the values by the height of the human (equation 5.2).

$$Velocity \text{ of body} = \frac{Velocity \text{ of neck}}{H} \quad (5.2)$$

$$Dimension = N - 1 = 4$$

Third feature that is extracted for the action classification is joint velocity. During testing, this feature showed to have the biggest impact. Each point velocity is calculated by the normalised coordinate values (extracted from OpenPose - section 4.3.2). Velocity is derived between

two consecutive frames.

$$\text{Velocity of joint } joint = X_{t_k} - X_{t_{k-1}} \quad (5.3)$$

$$\text{Joint velocity dimension} = 13 \text{ joints} \times 2 \text{ velocity / joints} \times (N - 1) = 104$$

There were more custom feature extracted like joint angles and length of each body part. These featured did not affect the final precision of action classification, so they were left out. Moreover as the body velocity appeared as a very good feature for action classification, its' weight was increased ten times.

These three features are converted to one feature vector. After concatenation of all three features used, the feature vector has dimensions of 238. In order to reduce the feature vector size principal component analysis (PCA) is used. After the feature reduction, the vector has dimension of 50. The procedure is shown on figure 5.3.

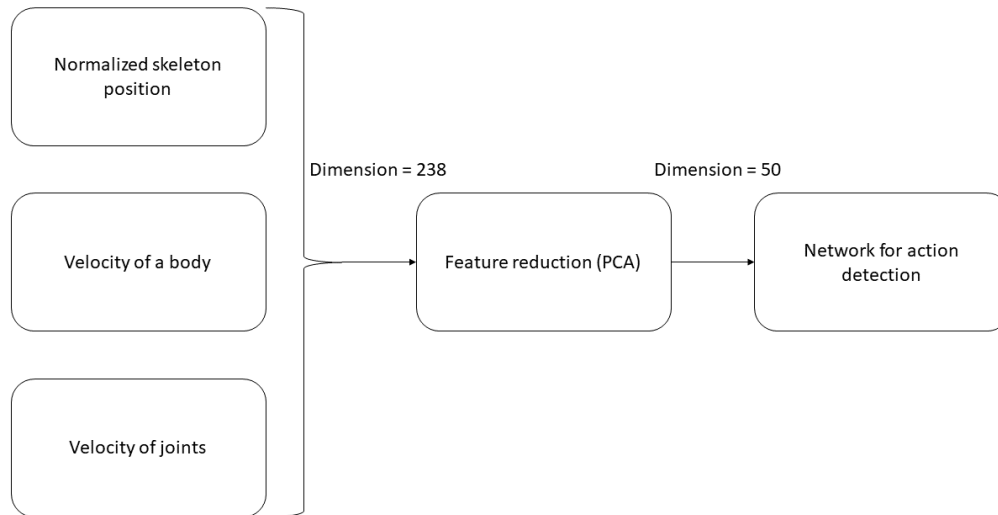


Figure 5.3: Feature extraction steps.

### 5.3.5 Action Classification

Last part of the model is estimating the action of the human. Reduced feature vector derived from three actions is fed into multi layer Perceptron (MLP) classifier. The classifier's parameters are updated iteratively at each time step based on the partial derivatives of the loss function with respect to the model parameters.

Used MLP classifier has three hidden layers. Layers size respectively 20, 30 and 40. For activation function ReLu and for optimisation function Adam is used. Learning rate during classification is constant at 0.001. The output of the classifier is a probability of an action. Special thresh-hold is used as hyper parameter to define weather the action probability is suitable and will be estimated.

# 6 Result and further development

## 6.1 Results

Each model was evaluated based on its accuracy of action prediction with the proposed data set. The accuracy was calculated over the test data set, which was 10 % of all the provided data. The table below shows each models accuracy.

Model	Average accuracy
Multi-Modal approach	70.91 %
Uni-Modal Approach with Skeleton Detection	63.02 %
Uni-Modal Approach	2.4 %

Table 6.1: Presented models average accuracy.

Frames where the skeleton was not predicted and the action was not estimated were counted as false positives. False positives where not taken into account while calculating the accuracy.

The most accurate class was standing, having average accuracy of 81.75 % in the network where multiple neural networks where used. Class run had the lowest accuracy among five classes. The reason behind this is purely the data distribution, as the standing had the most samples among training and run had the least samples (brought out in a section : Proposed data set).

Next section will show the graphs achieved during the learning of each model. First graph describes the training and validation loss of the uni-modal approach. From the graph it can be seen that the validation loss is very high and does not follow the training loss, as it should for the ideal case. After changing the following hyper parameters: hidden layer size, hidden layer dimension, batch and epoch size, the result did not change much. The best achieved graph can be seen from figure 6.1.

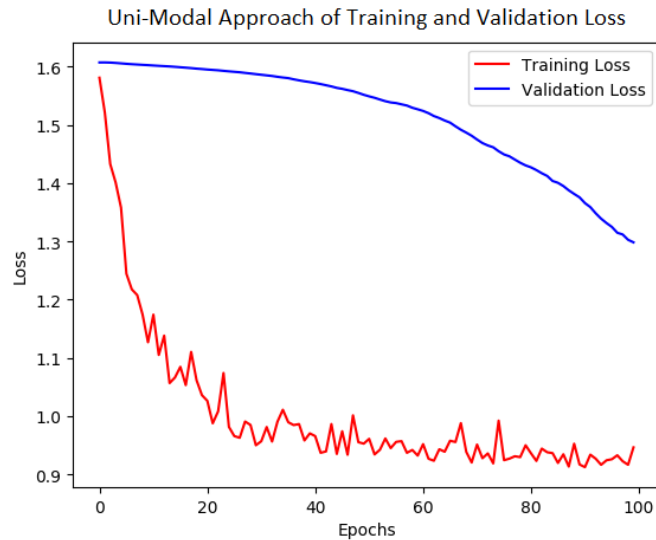


Figure 6.1: Uni-Modal Approach Plot of Training and Validation Loss.

Figure 6.2 describes the training score and validation score of the uni-modal approach with skeleton detection. Cross-validation is used with split parameter of 5. The result is good as the training and validation scores are getting close to each other towards the final epochs. Overall accuracy could be improved with better quality of input data and more data samples. Having more data samples provide better data distribution and the network would become more sensitive. After having better data distribution, network could require new parameter tuning, to improve the accuracy.

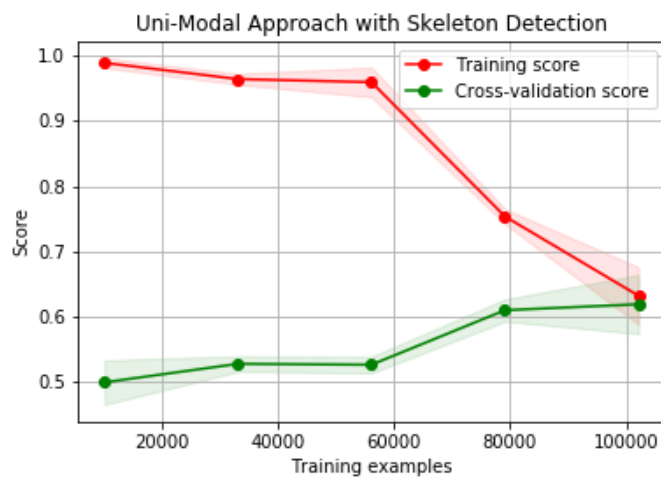


Figure 6.2: Uni-Modal Approach with Skeleton Detection Plot of Training and Validation Loss.

Third method, multi-modal approach, had similar learning curve as uni-modal approach with skeleton detection but with higher over-all accuracy. The final accuracy could be improved by re-training the modal with larger data set and with equal data distribution. From the figure

6.3 it can be seen that the classes that had the most samples(stand) had the best accuracy and the classes that had the least samples(run) had the worst accuracy. It can be observed from the image that the walking and standing classes had the most false-positives. Running was often miss-classified with walking. From a single image it can be hard to classify whether the person is running or walking.

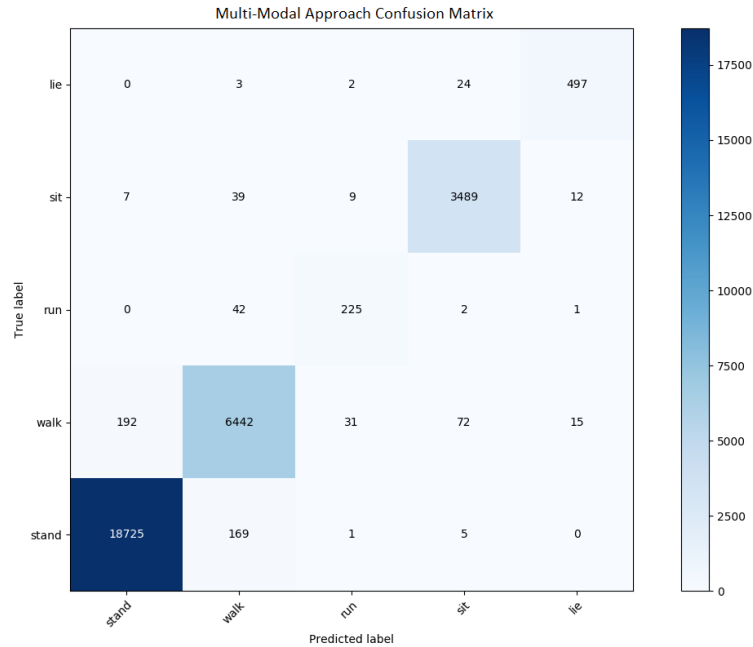


Figure 6.3: Multi-Modal Approach Confusion Matrix.

## 6.2 Further Development

Model using multiple neural neural networks in its pipeline showed the best performance for this particular research and for self driving car concept. The data was most similar to to proposed data set, with similar data distribution. Considering that this section will introduce ways how the model is being improved and the pipeline upgraded.

### 6.2.1 Human Action Direction Estimation

None of analysed models had any information about the movement direction of the human. For autonomous vehicles this is a crucial aspect. Having the information about the humans action without a direction will not help to improve the autonomy. Humans movement prediction can be very useful factor while planning self-driving car path. In order to predict the humans movement direction multiply methods were experimented.

First approach was implementing optical flow on the whole detected human skeleton from OpenPose algorithm. The centre point of the human was tracked with the optical flow. Centre of the human was calculated by minimum and maximum coordinate acquired from the skeleton joints (Equation 6.1).

$$\text{Centre of skeleton } (X, Y) = (X_{min} + X_{max})/2, (Y_{min} + Y_{max})/2 \quad (6.1)$$

Drawback that occurred while validating given method was that human limbs make this approach unstable. While human walks or stands (waving her/his hands for example) the midpoint (red circle shown on the figure 6.4) of the skeleton shifts heavily. Shift is tracked by the optical flow and will cause the system to predict that the person is moving. The output of the method can be seen on the figure 6.4.

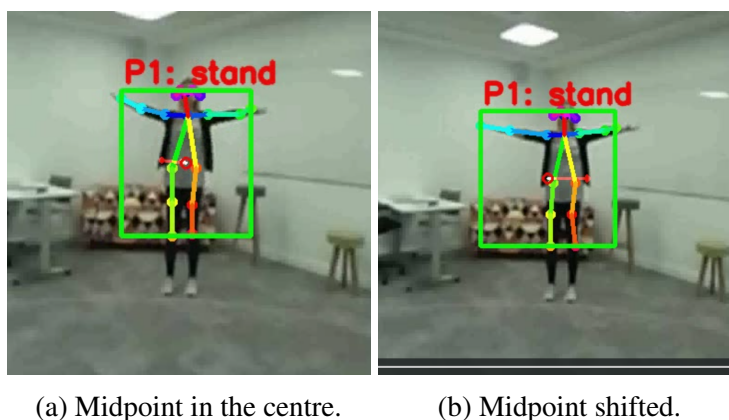


Figure 6.4: First method output.

On the left image (Figure 6.4 - a) it can be seen that the human has moved her hands and the optical flow shows movement of the human to the left (movement shown with arrow). On the right image (Figure 6.4 - b) the coordinate of the minimum and maximum joint values have shifted and so has the midpoint, showing now the movement of the person to the right. On both of the images the person was standing and no movements should have predicted.

To overcome this problem, instead of tracking the whole skeleton, only one certain point is tracked. The main joint of the skeleton for the action detection is neck and if the OpenPose algorithm can not detect the human neck, the whole skeleton will not be predicted on the given frame. That gives us one check for false-negatives, as the movement would not be detected if the skeleton has not been detected. Considering that aspect for the second method human neck is tracked with optical flow. Figure 6.5 shows the result of applying optical flow on neck, red circle is drawn around the neck point.



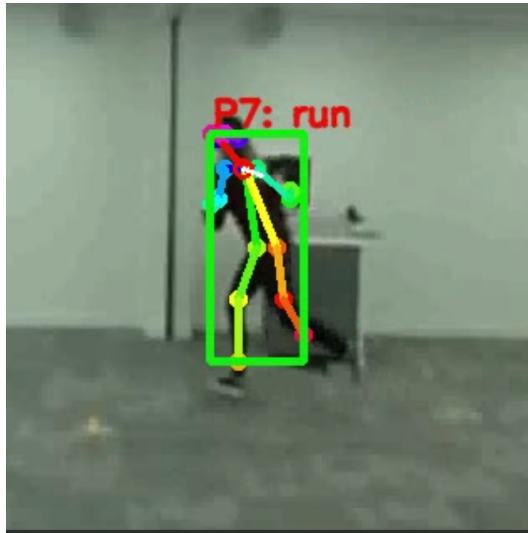


Figure 6.5: Optical flow tracking neck.

The proposed approach worked perfectly on consecutive frames where the pose was estimated. The problems accrued when the pose was not being estimated due to OpenPose algorithm or the modifications that we made in section 6.2.1. Not having the skeleton, means that optical flow does not have the neck coordinates and direction estimation can not be computed.

Next method introduces a new optical flow algorithm that follows YOLOv3 human prediction. A Shi-Tomasi corner detector and good features to track algorithm is used, that detects strong corners from an image. Instead of following one point, the neck, now multiple points are being tracked. The points that are being determined are all in the YOLOv3 human region. Points detected go to optical flow function, using the Lukas-Kanade method. Function tracks where the points have shifted between consecutive frames. Points predicted by the optical flow are iteratively tracked, and the mean shift of all the points are computed. Then the direction can be estimated when the mean shift goes above or below a predefined threshold.

To deal with optical flow detection noise, a back-tracking method is introduced, where the detected points are fed back into the optical flow function to find the original points. If the error between the two predictions are too high (usually due to a level of noise), those predicted points are ignored for the tracks.

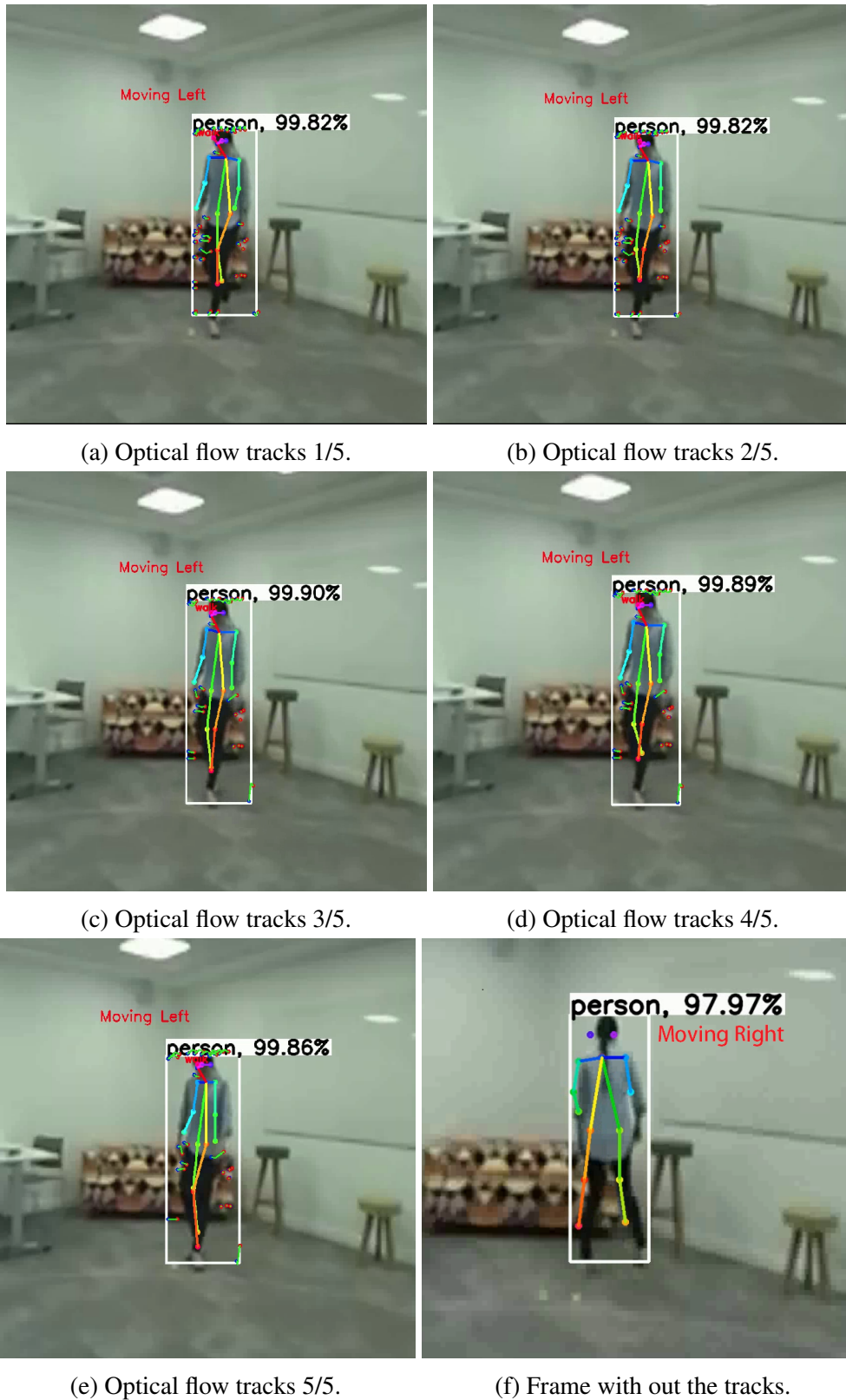


Figure 6.6: Result of mean optical flow.

Figure 6.6 shows the output after applying the new method of optical flow. On the first five images (figure a to e) the points and tracks are shown. Blue points correspond to the oldest tracks that are memorised and the red points correspond to the newest shifted points. Tracks are

drawn in green showing the path how the oldest and newest point has shifted. The movement direction is shown upon the human prediction bounding box. The last image (figure 9.7, image f ) shows the output without the predicted point and tracks.

### 6.2.2 Anomalies

During the validation of the most successful model some problems occurred. Namely OpenPose algorithm was detecting anomalies that falsed the whole direction estimation approach proposed in section 6.2. Detected anomalies can be seen from image 6.7. OpenPose detects human in the region of an image, where there is actually no human. Anomalies keep appearing for different

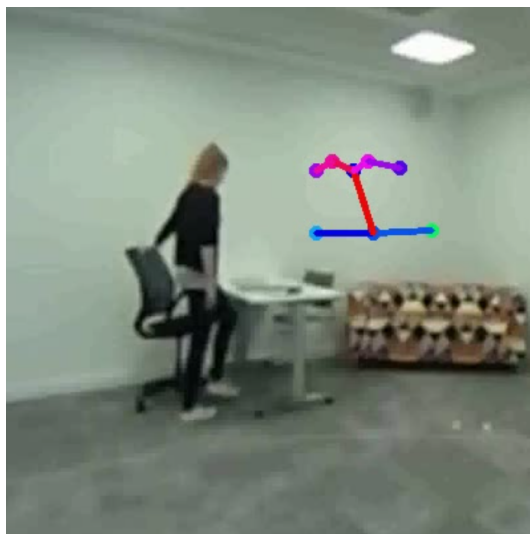


Figure 6.7: Anomalies on video frames.

reasons. After proper investigation two reasons can be brought out: the poor quality of training data and not enough data during training. The poor quality was caused after converting the raw 360 degree footage to 2D.

The fact that there was not enough data is one of the largest problems of neural networks, there is never enough data. To overcome anomaly problem yet another pre-trained network is used. Namely one of the fastest and most accurate object detection methods today, YOLOv3 [14], is being used to detect the humans. The model is trained on COCO data set [54] and has mean average precision ( mAP ) of 57.8 % on 30 FPS on object detection.

First method to overcome anomalies was to preprocess the input data going to the action estimation model. Initial video is passed through the YOLOv3 network, where human is detected. On each frame the human is masked out. Mask is made 5 pixels wider and higher than the original YOLOv3 prediction. Enlarging the YOLOv3 mask will confirm that new mask covers

the whole human on the frame. Mask is finally fed into action recognition network. The result can be seen from the figure 6.8.

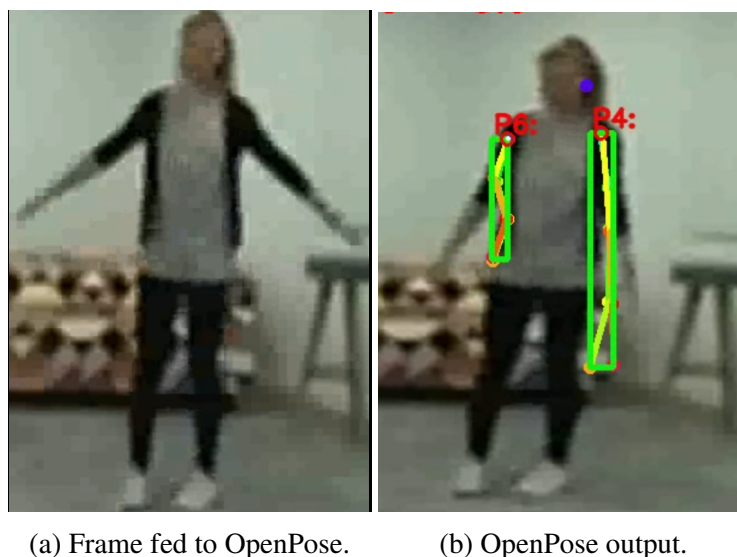


Figure 6.8: Applying OpenPose only on YOLOv3 human bounding area.

The outcome of the experiment was not successful. More anomalies were detected throughout the whole video. After analysing it was found that the mask size of the human differs throughout the frames. It is caused by the human body movement. While the human is facing sideways the mask is smaller and while she/he is facing the camera with hands stretched out wide, mask is bigger. As well as while the human walks or runs her or his hands and legs move, which will affect on the mask size. Neural network algorithm needs a fixed size image as an input, thus there is resizing done. Resizing the image dropped the video resolution. Image resizing to 1080 x 1920 pixels (initial frame size) and 480 x 853 pixels (OpenPose network input size) was tested. Both ended up more or the less with same amount of anomalies.

From the previous experiment one aspect was still achieved. YOLOv3 performed better on human detection than the OpenPose algorithm. Taking that into account new pipeline is presented (pipeline shown of the figure 6.9).

The new proposed pipeline takes in the original 2D video which was transformed from 360 degree footage. The video is feed into the OpenPose network and to the YOLOv3 network. OpenPose predicts the skeleton and YOLOv3 finds a human from the frame. Next step is checking whether the OpenPose predicted skeleton matches the region of YOLOv3 predicted human. If these two regions match, the frame is passed onward to the part of the network where the pose is estimated. No pose is estimated when the regions do not match. Meaning that the current frame is skipped and the next frame is being processed. The outcome of proposed

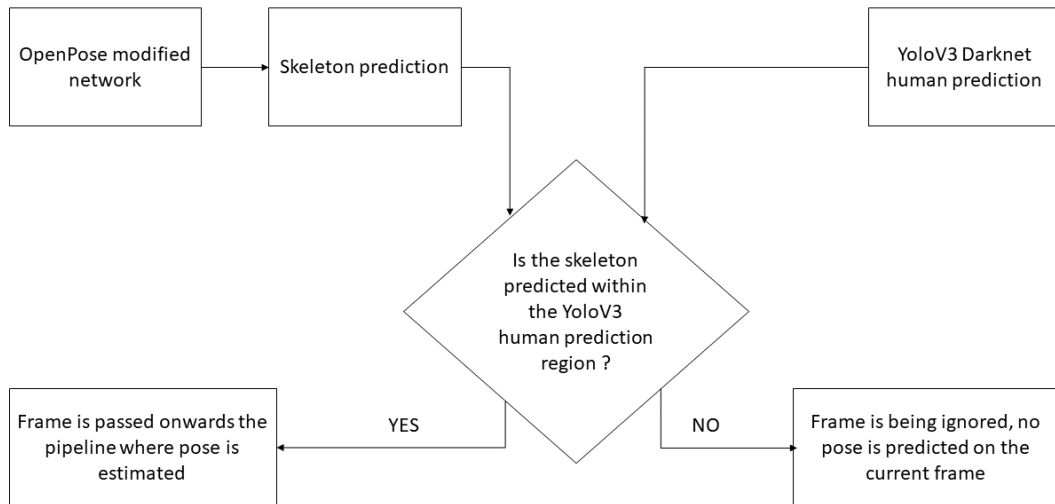
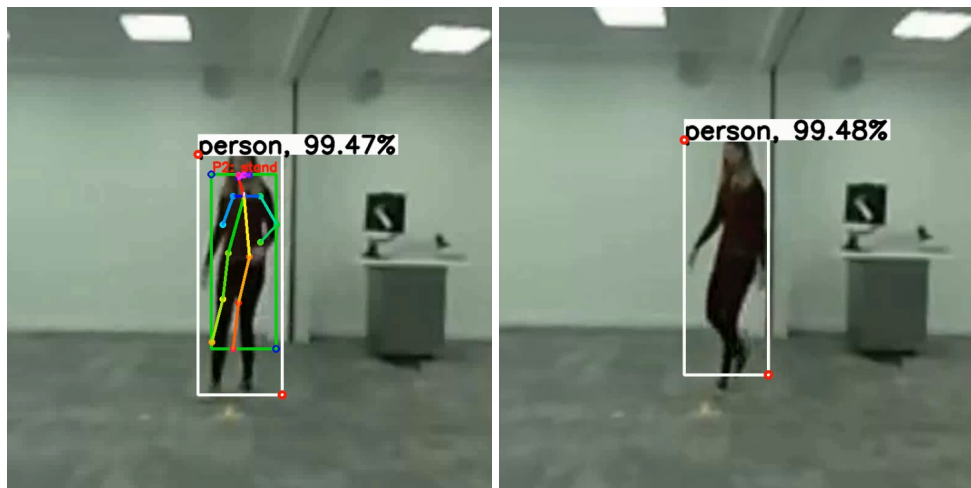


Figure 6.9: New pipeline to detect human and their pose.

pipeline was successful as no more anomalies were detected. The result of this modification can be seen from the figure 6.10. The left image on the figure shows when two predictions match and the pose is being predicted. Right image shows just the YOLOv3 human prediction without pose estimation, meaning the OpenPose skeleton prediction was out of bounds of YOLOv3 prediction area.



(a) Skeleton within boundaries.

(b) Skeleton is neglected.

Figure 6.10: New pipeline outcome.

# 7 Conclusions

## 7.1 Conclusion

The relevant data set is always a decisive factor in any data science field. There is never too much data. More data gives more samples on which the neural network can be trained. Even during this thesis it could be seen that the most accurate class for HAR was standing where it had the most data samples from provided data set. The least accurate was running where as it had the least data samples. Most of the researches so far have focused weather on LiDAR or RBG sensors footage, this thesis focused on applying the HAR on 360 degree camera footage. It compared multiply neural networks models. Each model was retrained with the proposed data set and respective hyper parameters fine tuned, to achieve better performance. The best model, multi-modal approach, was enhanced with more accurate human detection by fusing YoloV3 human prediction with the given model. As the path planning of the human is very crucial aspect on self driving car, the thesis introduced methods to estimate the human movement direction on videos. Best method to track human direction was to implement A Shi-Tomasi corner detector and good features function. Detected points were tracked with optical flow.

## 7.2 Future Work

This thesis considered work only with the footage of the 360 degree camera. As the proposed data set has same information recorded with other sensors, it would be the great interest to see how the proposed method would work with different sensor input data. For example I believe that taking RGB camera footage instead of 360 degree camera, can improve the final accuracy. The reason would be that there would not be data pre-processing, thus the resolution of the data will not drop.

Moreover the sensor data could be fused, to provide a presumably a better data set. The

objective would be to find out which combination of sensors and algorithms provide the best accuracy for the HAR. Fusing data can help with noise problems and filtering out anomalies.

For humans it is hard to classify the action based on one frame. Giving sequence of frames, will make the task much easier. Multi-modal approach showed that recurrent neural networks performed best on human skeleton detection. Therefore it would be really interesting to make a recurrent neural network model for action detection. Instead of classifying an action based on a single frame, the action would be predicted considering multiply iterative frames.

For self driving cars the proposed model needs to be enhanced further in order to detect multiple persons. The OpenPose algorithm is capable of recognising more than one human, but it has not been validated during this thesis. In addition to that it would be interesting to see how the direction of movement would act on the data where multiple persons are present.

While implementing multi-person detection is would be interesting to try dense optical flow instead of used sparse optical flow for action direction estimation. Dense optical flow implementation could possibly help with occlusion, assuming that the input data is with high quality.

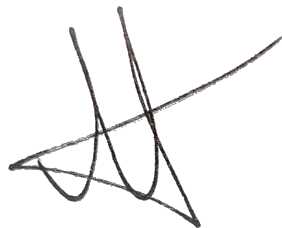
# Acknowledgements

I would like to express my great appreciation to the following:

- My supervisor, Gholamreza Anbarjafari, for guiding me throughout the thesis. I could reach out to him irregardless of the day of the week nor the time.
- My parents who boosted my motivation every time we met.
- My friend, Joan Kangro, and my sister ,Kristin Tammvee, who supported me a lot throughout the process, especially during the writing part.
- My roommate ,Quazi Saimoon Islam, who helped me solving various technical problems I faced and sharing his out of a box ideas for problem solving with me.

Thank You,

Martin Tammvee

A handwritten signature in black ink, appearing to be 'MT' with a large flourish extending to the right.



# Bibliography

- [1] Terrence Fong, Charles Thorpe, and Charles Baur. *Collaborative control: A robot-centric model for vehicle teleoperation*, volume 1. Carnegie Mellon University, The Robotics Institute Pittsburgh, 2001.
- [2] Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.
- [3] Panagiotis G Zavlangas and Spyros G Tzafestas. Industrial robot navigation and obstacle avoidance employing fuzzy logic. *Journal of Intelligent and robotic Systems*, 27(1-2):85–97, 2000.
- [4] Jong Jin Park, Collin Johnson, and Benjamin Kuipers. Robot navigation with model predictive equilibrium point control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4945–4952. IEEE, 2012.
- [5] Clifford A Shaffer and Gregory M Herb. A real-time robot arm collision avoidance system. *IEEE Transactions on Robotics and Automation*, 8(2):149–160, 1992.
- [6] FeedForwardNN. <https://medium.com/@b.terryjack/introduction-to-deep-learning-feed-forward-neural-networks-ffnns-a-k-a-c688d83a309d>. Accessed: 2020-04-05.
- [7] RecurrentNN. <https://medium.com/@george.drakos62/what-is-a-recurrent-nns-and-gated-recurrent-unit-grus-ea71d2a05a69>. Accessed: 2020-04-05.
- [8] LSTM. <https://medium.com/datathings/the-magic-of-lstm-neural-networks-6775e8b540cd>. Accessed: 2020-04-05.
- [9] ResidualNN. <https://medium.com/analytics-vidhya/introduction-to-residual-neural-networks-8af5b7c4afd4>. Accessed: 2020-04-05.
- [10] DNN. <https://medium.com/@RosieCampbell/demystifying-deep-neural-nets-efb726eae941>. Accessed: 2020-04-05.

- [11] CNN. <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>. Accessed: 2020-04-17.
- [12] CNN2. <https://medium.com/@sdoshi579/convolutional-neural-network-learn-and-apply-3dac9acfe2b6>. Accessed: 2020-04-17.
- [13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [14] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [16] Mirco Moencks, Varuna De Silva, Jamie Roche, and Ahmet Konoz. Adaptive feature processing for robust human activity recognition on a novel multi-modal dataset. *arXiv preprint arXiv:1901.02858*, 2019.
- [17] FFmpeg. <https://www.ffmpeg.org/>. Accessed: 2020-03-12.
- [18] Iyiola Olatunji. Human activity recognition for mobile robot. *Journal of Physics: Conference Series*, 1069, 01 2018.
- [19] Dheeru Dua and Casey Graff. UCI machine learning repository. <http://archive.ics.uci.edu/ml>. Accessed: 2020-04-05.
- [20] Tarik Uzunovic, Edin Golubovic, Zlatan Tucaković, Yasin Acikmese, and Asif Sabanovic. Task-based control and human activity recognition for human-robot collaboration. pages 5110–5115, 10 2018.
- [21] Shaufikah Shukri, Latifah Kamarudin, and Mohd Hafiz Fazalul Rahiman. *Device Free Localization for Human and Activity Monitoring*. 09 2018.
- [22] Artur Jordao, Antonio C. Nazare Jr., Jessica Sena, and William Robson Schwartz. Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art, 2018.

- [23] Céline Franco, Jacques Demongeot, Christophe Villemazet, and Nicolas Vuillerme. Behavioral telemonitoring of the elderly at home: Detection of nycthemeral rhythms drifts from location data. pages 759–766, 01 2010.
- [24] Openimage dataset. <https://storage.googleapis.com/openimages/web/index.html>. Accessed: 2020-03-19.
- [25] Kaggle dataset on human. <https://www.kaggle.com/dasmehdixtr/human-action-recognition-dataset>. Accessed: 2020-03-19.
- [26] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.
- [27] Kaiqi Liu, Wenguang Wang, and Jun Wang. Pedestrian detection with lidar point clouds based on single template matching. *Electronics*, 8(7):780, 2019.
- [28] Michalis Vrigkas, Christophoros Nikou, and Ioannis A Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.
- [29] Lawrence Chow and Nicholas Bambos. Real-time physiological stream processing for health monitoring services. In *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, pages 611–616. IEEE, 2013.
- [30] Godwin Ogbuabor and Robert La. Human activity recognition for healthcare using smartphones. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 41–46, 2018.
- [31] Rajesh Kumar Tripathi, Anand Singh Jalal, and Subhash Chand Agrawal. Suspicious human activity recognition: a review. *Artificial Intelligence Review*, 50(2):283–339, 2018.
- [32] Sarita Chaudhary, Mohd Aamir Khan, and Charul Bhatnagar. Multiple anomalous activity detection in videos. *Procedia Computer Science*, 125:336–345, 2018.
- [33] Shyma Zaidi, B Jagadeesh, KV Sudheesh, and Arlene A Audre. Video anomaly detection and classification for human activity recognition. In *2017 International Conference*

on *Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, pages 544–548. IEEE, 2017.

- [34] KinectHAR. <https://medium.com/dsaid-govtech/human-pose-estimation-and-human-action-recognition-experimenting-for-public-good-dabde16521b3>. Accessed: 2020-04-08.
- [35] Abassin Sourou Fangbemi, Bin Liu, Neng Hai Yu, and Yanxiang Zhang. Efficient human action recognition interface for augmented and virtual reality applications based on binary descriptor. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 252–260. Springer, 2018.
- [36] Ana Maqueda, Carlos Del-Blanco, Fernando Jaureguizar, and Narciso Garcia. Human-action recognition module for the new generation of augmented reality applications. pages 1–2, 06 2015.
- [37] J. K. Aggarwal. Human activity recognition - a grand challenge. In *2007 International Symposium on Signals, Circuits and Systems*, volume 2, pages 1–1, 2007.
- [38] J. Graf, S. Puls, and H. Wörn. Incorporating novel path planning method into cognitive vision system for safe human-robot interaction. In *2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, pages 443–447, 2009.
- [39] Christian Braunagel, Enkelejda Kasneci, Wolfgang Stolzmann, and Wolfgang Rosenstiel. Driver-activity recognition in the context of conditionally autonomous driving. 09 2015.
- [40] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F. Wang. Driver activity recognition for intelligent vehicles: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 68(6):5379–5390, 2019.
- [41] Martin Torstensson, Thanh Bui, David Lindström, Cristofer Englund, and Boris Duran. In-vehicle driver and passenger activity recognition. 05 2019.
- [42] Bradley Hayes and Julie Shah. Interpretable models for fast activity recognition and anomaly explanation during collaborative robotics tasks. 05 2017.

- [43] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. A review on human activity recognition using vision-based method. *Journal of Healthcare Engineering*, 2017:1–31, 07 2017.
- [44] Lim J. Noh K.J. Kim G. Jeong H Chung, S. Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning. 2017.
- [45] R. S. Pol and M. Murugan. A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods. In *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pages 1339–1344, 2015.
- [46] David Brogan and Nicholas Johnson. Realistic human walking paths. 04 2003.
- [47] Javier Gómez, Nikolaos Mavridis, and Santiago Garrido. Social path planning: Generic human-robot interaction framework for robotic navigation tasks. 11 2013.
- [48] A. K. Pandey and R. Alami. A framework for adapting social conventions in a mobile robot motion in human-centered environment. In *2009 International Conference on Advanced Robotics*, pages 1–8, 2009.
- [49] Lstm har. <https://github.com/eriklindernoren/Action-Recognition>. Accessed: 2020-01-22.
- [50] Ufc-101 dataset. <https://www.crcv.ucf.edu/data/UCF101.php>. Accessed: 2020-01-22.
- [51] Action recognition based on human skeleton. <https://github.com/smellslikeml/ActionAI>. Accessed: 2020-01-21.
- [52] Action recognition based on human skeleton. [https://github.com/NVIDIA-AI-IOT/trt\\_pose](https://github.com/NVIDIA-AI-IOT/trt_pose). Accessed : 2020 – 01 – 22.
- [53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [54] Coco data set. <http://cocodataset.org/home>. Accessed: 2020-03-7.
- [55] Action recognition based on human skeleton. <https://github.com/felixchenfy/Realtime-Action-Recognition?fbclid=IwAR2SGsNbxZeCvUAxh9qXB1qZ5xIBM0gU9UoaK6Kx4ngOPt6HBV1> 2020 – 01 – 04.

- [56] S. Wei, Y. Sheikh, Z. Cao, T. Simon. Realtime multi-person 2d pose estimation using part affinity fields. 2017.
- [57] Openpose architecture. <https://www.learnopencv.com/multi-person-pose-estimation-in-opencv-using-openpose/>. Accessed: 2020-04-08.
- [58] S. Qiao, Y. Wang, and J. Li. Real-time human gesture grading based on openpose. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–6, 2017.
- [59] Ferhat Attal, Samer Mohammed, Mariam Dedabrishvili, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. Physical human activity recognition using wearable sensors. *Sensors*, 15:31314–31338, 12 2015.
- [60] Małgorzata Słota-Valim. Static and dynamic elastic properties, the cause of the difference and conversion methods – case study. *Nafta-Gaz*, 71:816–826, 11 2015.
- [61] G. Karthikeyan and Prof P Balasubramanie. Ofs-nn: Optimal features-neural network based outlier detection for big data analysis. *Journal of Communications*, 13:396–405, 07 2018.
- [62] Ismael Lemhadri, Feng Ruan, and Robert Tibshirani. A neural network with feature sparsity. 07 2019.
- [63] C. Benedek, B. Gálai, B. Nagy, and Z. Jankó. Lidar-based gait analysis and activity recognition in a 4d surveillance system. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(1):101–113, Jan 2018.
- [64] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.
- [65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

# Appendices

Computer specification to run and re-train each model are brought out below:

- Processor : Intel(R) Core(TM) i5-7500 CPU @ 3.40 GHz.
- Installed memory (RAM) : 8.00 GB.
- Graphics card (GPU) : NVIDIA GeForce GTX 1060 6GB.

Requirements to run Uni-Modal Approach and Uni-Modal Approach with Skeleton Detection are brought out below :

- tensorflow==2.1.0
- scipy
- scikit-learn
- opencv-contrib-python
- pandas
- pillow
- CUDA 10.0

Requirements to run Multi-Modal Approach with YoloV3 are brought out below :

- Pillow
- cython
- matplotlib
- scikit-image

- opencv-python
- h5py
- imgaug
- pandas
- argparse
- IPython[all]
- sklearn
- slidingwindow
- pyyaml
- CMake  $\zeta= 3.12$
- CUDA 10.0
- GPU with CC  $\zeta= 3.0$
- Intel OpenVINO 2019 R1
- OpenCV-dnn
- TVM
- OpenDataCam
- Netron



# Non-exclusive licence to reproduce thesis and make thesis public

I, Martin Tammvee

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

**“Human Activity Recognition Based Path Planning For Autonomous Vehicles”**

supervised by Prof. Gholamreza Anbarjafari

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

*Martin Tammvee*

**19.05.2020**