CHALMERS
UNIVERSITY OF TECHNOLOGY

Pedestrian Intent Prediction Using Deep Machine Learning

Master's thesis in Systems, Control and Mechatronics

Aren Moosakhanian

Sourab Bapu Sridhar

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Pedestrian Intent Prediction Using Deep Machine Learning

Aren Moosakhanian
Sourab Bapu Sridhar

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2021

Pedestrian Intent Prediction Using Deep Machine Learning

Aren Moosakhanian
Sourab Bapu Sridhar

© Aren Moosakhanian
Sourab Bapu Sridhar, 2021.

Examiner: Jiajia Chen, Electrical Engineering
Supervisor: Shen Li, Electrical Engineering

Master's Thesis 2021
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Scene parsing and pedestrian intent prediction on publicly available dataset.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Pedestrian Intent Prediction Using Deep Machine Learning
Aren Moosakhanian
Sourab Bapu Sridhar

Department of Electrical Engineering

Chalmers University of Technology

Abstract

One of the critical requirements for a safe assistive and autonomous driving system is the accurate perception of the ego vehicle's environment. While there have been significant strides in detecting and tracking visible surroundings of the ego vehicle, accurate prediction of vulnerable road users such as pedestrians and cyclists remains a challenge as vulnerable road users can instantly change their direction and speed. Humans make important intuitive decisions based on the interactions in the scene and the sequences of actions to interpret the intent of vulnerable road users. However, the same cannot be assumed for the current assistive and autonomous driving systems, as these intentions are realised through subtle gestures and interactions. Since predicting the future intent of vulnerable road users is essential to warn the driver or automatically perform smoother manoeuvres, our thesis aims to predict pedestrian intent using deep machine learning.

In recent years, the intent prediction problem has been a topic of active research, resulting in several new algorithmic solutions. However, measuring the overall progress towards solving this problem has been difficult. Therefore, this thesis investigates the performance of multiple baseline methods on the joint attention in autonomous driving (JAAD) dataset to tackle this obstacle. Despite achieving state-of-the-art results on curated datasets, most of these methods are developed, disregarding potential deployment in production environments. Our thesis proposes an end-to-end network that attempts to reduce the gap between prototyping and production based on these findings. The proposed end-to-end network predicts the future intent of vulnerable road users up to half a second in the future.

Keywords: Deep Machine Learning, Computer Vision, Intent Prediction, Autonomous Driving, ADAS, AD, JAAD, PIE

Acknowledgements

We would like to thank Semcon for the opportunity to work on this problem. We would also like to thank our supervisors from Semcon, Axel Bender, and Jens Henriksson, for their time and valuable feedback during our thesis. Lastly, we would like to thank our supervisor from Chalmers, Shen Li, and our examiner from Chalmers, Jiajia Chen, for their guidance during the complete process.

Aren Moosakhanian and Sourab Bapu Sridhar, Gothenburg, November 2021

Contents

List of Figures

List of Tables

1
Introduction

This chapter presents an overview of the pedestrian intent prediction problem and introduces the crucial developments in this domain. The organisation of this chapter is as follows: section 1.1 highlights the challenges and importance of pedestrian intent prediction problem, section 1.2 explains the overall objective of the master thesis, section 1.3 explores the current state-of-the-art methods, section 1.4 describes the contribution of the master thesis, section 1.5 outlines the individual contributions of the authors, section 1.6 defines the limitations of the master thesis, and section 1.7 reveals the contents of the following chapters.

1.1 Background

According to the World Health Organisation, nearly half of road traffic fatalities are experienced by vulnerable road users such as pedestrians and cyclists. The reason for this is that they do not have any unique means of protection [1, 2]. As autonomous vehicles become more common on the roads, their progress attracts additional safety concerns for vulnerable road users. Since predicting the intention of vulnerable road users is critical for human driving, the same level of importance must be taken into account by systems providing any driving assistance level, from advanced driver assistant systems to fully autonomous vehicles [3]. With almost half of all fatalities caused by road traffic accidents being pedestrians and cyclists, the ability to successfully predict vulnerable road users' intentions becomes one of the critical requirements to ensure the acceptance of fully autonomous vehicles into our societies.

While there have been significant strides in detecting and tracking visible surroundings to the vehicle, accurate detection and prediction of vulnerable road users' intentions remain challenging. Among these vulnerable road users, accurate detection and prediction of pedestrian intentions become a challenging subproblem as pedestrians do not explicitly indicate their intentions, allowing for higher fatality risk. Due to this reason, the pedestrian intention prediction problem becomes a subset of the much more extensive intent prediction problem where the goal is to automatically estimate a pedestrian's relative position and intention [4]. In other words, pedestrian intent prediction involves predicting pedestrians' positions and intentions to evaluate the ego vehicle's risk. This information is critical for reducing the chance of injuries requiring hospitalisation. Experiments show that initiating an emergency brake with 160 ms of anticipation over a 660 ms time to collision can lessen the

probability of injury requiring hospitalisation from 50% to 35% [5]. Furthermore, information about pedestrian intent is also valuable in lowering collision avoidance alerts and systems' false-positive rates, resulting in a safe and smooth manoeuvre while driving.

One of the most significant reasons for the high complexity of pedestrian intent prediction is that pedestrians are very agile in their movement and can change their direction without a noticeable reduction in speed. We, as humans, make important intuitive decisions based on the sequences of actions and interactions with other people in the scene to achieve safe and smooth navigation. This intuition allows movements that are very dynamic as we can decide what route to take in a very dynamic manner. This simple yet valuable piece of information is crucial for deciding the next step to be taken. On the other hand, machines have a hard time reading human judgments realised by subtle gestures and interactions. This inability to understand humans makes autonomous vehicles very conservative in their driving. This deficiency can cause a lot of starts and stops or jerking movements when driving in city streets. In turn, this can be nauseating for the riders and upsetting for others on the road [6].

However, with the advent of Deep Learning, advanced algorithms that read pedestrian instincts and make judgments are being developed. Various methods ranging from trajectory prediction to behavioural analysis are being explored [1]. At the same time, different input modalities from images to point cloud data are also being examined. In this master thesis, we investigate the idea of using monocular RGB images as core information to recognise the intentions of vulnerable road users.

1.2 Objective

As mentioned in the earlier chapter, with safety being one of the biggest concerns with autonomous vehicles, it is crucial to predict pedestrians' intent accurately. In order to predict pedestrians' intent accurately, precise detection and tracking of ego vehicle's visible surroundings are critical. In this regard, different sensing methods like cameras, radars, and lidars have been used to detect and track visible surroundings of the ego vehicle. Despite the popularity of these methods, none of them is infallible as each method has a specific limitation. Cameras are a widely understood and mature technology. They can readily detect colour information and have an extremely high resolution. However, the accuracy of a camera-based system is highly dependent on the environment and weather conditions. Alternatively, radars are virtually impervious to adverse weather conditions, working reliably in dark, wet, or foggy weather. Radar sensors have a limited resolution, leading to difficulty identifying and reacting to multiple, specific hazards. Lastly, lidar sensors are the only sensors that can provide an incredibly detailed 3D view of the environment around the sensor. The drawback is that it takes enormous processing power to interpret lidar measurements and translate them into actionable data. They are also highly complex and expensive.

In addition to the sensor modality, the availability of quality data and baseline methods is vital to evaluate the performance of the prediction method. In recent years, the pedestrian intent prediction problem has been a topic of active research, resulting in many new algorithmic solutions, which are reviewed in section 1.3. However, a limited number of high-quality datasets can be used to benchmark multiple state-of-the-art methods. Since measuring the overall progress is crucial to solving the intent prediction problem, developing a prediction method based on publicly available high-quality datasets with standard training and evaluation procedures seems reasonable.

Therefore, this thesis aims to design a pedestrian intent prediction method based on 2D images captured from a high-resolution monocular camera by finding methods that use publicly available high-quality datasets, employ standard training and evaluation procedures, and consistently provide accurate predictions. The objective was chosen as all the publicly available high-quality datasets are based on monocular camera data. Additionally, an objective was also set to compare the performance of the novel method against multiple state-of-the-art methods to measure the overall progress in solving the intent prediction problem.

1.3 Related work

Since the pedestrian intent prediction problem has been a topic of active research, various approaches have been taken to tackle this problem. An extensive literature review was carried out, and an overview of the different approaches and the recent work done within these approaches are presented in this section.

1.3.1 Pedestrian detection and tracking

Pedestrian detection and tracking is one of the basic approaches to detect pedestrian intent. This approach aims to predict pedestrian intent based on intrinsic pedestrian features such as future trajectories or poses. Pedestrian detection and tracking based approach usually consist of an object detector followed by an object tracker and a classifier.

When it comes to pedestrian detection methods, a thorough analysis of various pedestrian detection methods based on shallow learning was provided in [7]. However, the accuracy of methods mentioned in [8] drops while detecting pedestrians in a crowd due to occlusion. Recently, various deep learning methods such as [9–12] provide significantly higher accuracy while detecting pedestrians in a crowd [11, 12] or in the presence of occlusion [8]. For pedestrian tracking, multi-person tracking methods to track every person in a crowded scene was employed in [13]. Lately, techniques like people re-identification [14] and pose estimation [15] are being used to solve tracking problems.

Recently, several works have been directed towards designing pedestrian intent prediction with various object detection and tracking algorithms. [16–18] uses different

parts of the body to detect the movement and intent of the pedestrian by zooming into the corresponding body part and using local features to classify whether the pedestrian is crossing or not. At the same time, [19] predicts the intent of the pedestrian by combing CNNs and LSTMs. Lastly, [20] predicts pedestrian intent based on intrinsic pedestrian feature poses by fitting an n-point skeleton to each detected pedestrian and classifying using a support vector machine or a convolutional neural network. Although these features contribute to predicting pedestrian intent, pedestrian detection and tracking based approaches ignore context and interactions with objects in the scene, such as other pedestrians, vehicles, traffic signs, lights, and other environmental factors. Our thesis argues that such relationships can be revealed over time. Therefore, our thesis takes object detection and tracking based approaches for granted and investigates visual reasoning approaches to understand the intent of the pedestrians.

1.3.2 Trajectory prediction

Trajectory Prediction is another closely related approach to detect pedestrian intent. This approach predicts pedestrian intent based on the assumption that accurate prediction of future trajectory indicates the pedestrian's intent. Although the assumption is valid, trajectory prediction is a complex problem as human motion is driven by complex internal and external stimuli. Human motion can be driven by the intent, surrounding objects, social rules, or the environment. Since most factors are not directly observable, future trajectories are hard to predict accurately in real-time and require more annotations and supervision.

Recent works like [21,22] use past trajectories to predict the future trajectories. [23] uses inverse reinforcement learning to predict future trajectories. [24] models social dynamics and crowd interaction to predict future trajectories. Furthermore, some methods utilise human dynamics in different forms to predict trajectories. [25, 26] proposes Gaussian Process Dynamical Models based on the action, speed, location, and heading direction as input to predict future directions and intent. [21,22,27,28] incorporate environmental factors into trajectory prediction. One of the most significant issues with trajectory prediction based approaches is that many methods depend on a top-down view of the scene. In addition, trajectory prediction is not a well-defined problem as future trajectories are often contingent on the initial conditions and cannot be predicted long enough into the future with enough certainty. Therefore, although the methods mentioned above obtain remarkable results, the dependency on the top-down view of the system makes the methods inapplicable to data available for our thesis.

1.3.3 Action prediction

Action prediction can be considered as one of the most suitable approaches for intent prediction. This approach predicts pedestrian intent by modelling the causal relationship between the past, current and potential future information similar to approaches used in activity recognition algorithms. Since the action prediction ap-

proach tries to anticipate the following action by looking at the sequence of previous actions, pedestrian intent prediction can be considered a sub-problem aiming to forecast whether a given pedestrian will cross in the future. Action prediction is an important problem in many domains such as assistive robotics [29–32], surveillance [33–36], sports forecasting [37–39] and autonomous driving systems [40–43]. Action prediction can be either implicit in the form of future trajectories or poses [30, 34, 38, 41, 44] or explicit in terms of predicting future events. [29, 33, 35, 39, 40]

Mainstream strategies for action prediction use sequential temporal tools to model the causal relationship or visual features extracted from recently observed frames to predict the following action. Some commonly employed architectures include recurrent networks [45–47], 3D convolutional networks [48–50], or a combination of both [51]. Among methods based on recurrent networks, high-level semantics are often preprocessed with off-the-shelf algorithms, and data-driven methodologies are employed to learn parameters. Among methods based on convolutional networks, researchers often resorted to deep ConvNet features and learned a classifier from the training data. Most of the methods mentioned above anticipate the next action by looking at the sequence of previous actions [52]. However, other methods build spatiotemporal graphs [6] or use reinforcement learning [53] to predict the next action. One of the most significant issues with the action prediction approach is that many methods depend on the type of data to build a prediction model. Although these methods obtain remarkable results, the thesis aims to build a model that can reason on the scene and estimate the likelihood of crossing or not crossing.

1.3.4 Scene graph parsing and visual reasoning

Scene graph parsing and visual reasoning is relatively a new approach to detect pedestrian intent. A scene graph is a structured representation of a scene that reveals the objects, attributes, and relationships between objects in the scene. As computer vision technology continues to develop, researchers are no longer satisfied with simply detecting and recognising objects in images; instead, researchers look forward to a higher level of understanding and reasoning about visual scenes. Predicting the Spatio-temporal relationship between the various objects in the scene is the principle behind the scene graph parsing and visual reasoning approach for pedestrian intent prediction.

Recently, several works have been directed towards generating scene graphs with global context [54], relationship proposal networks [55], conditional random fields [56], iterative message-passing [57] or recurrent neural networks [58]. Scene graphs built on visual scenes are used for multiple applications such as action recognition [59], image generation [60], trajectory prediction [61], and visual question answering [62].

Since the scene graph parsing and visual reasoning approach considers the pedestrian context and interactions, our thesis proposes an approach for the pedestrian intent prediction problem based on scene graph parsing and visual reasoning. To

achieve this, we extract features from each object in the scene and reason about the relationship between objects through graph convolution techniques. Additionally, our approach creates a scene graph for each time point instead of one single scene graph to model the spatiotemporal relationship between objects. This spatiotemporal modelling captures intrinsic scene dynamics, encoding the sequence of subtle human actions, which are crucial for predicting the intent.

1.4 Thesis contribution

In this master thesis, the authors answer the following research questions:

1. How well can a novel deep machine learning architecture predict pedestrian intent 0.5 seconds, 1 second and 1.5 seconds in the future when applied on a publicly available dataset?

2. What are the key characteristics that explain the differences in the performance of the novel architecture and the existing architecture?

1.5 Individual contribution

Since a considerable effort was spent on formulating the problem, performing the literature survey, designing the problem, implementing the novel architecture and benchmarking the result against the baseline methods, this section outlines the individual contributions of the authors.

Both the authors worked equally on the literature survey (section 1.3, 2.2.3, and 2.3), established the problem formulation (section 1.4), planned the solution approach (section 3.3), and selected the features (section 3.1.2) and the machine learning framework (section 3.2).

The first author predominantly restructured the input dataset into the frame-by-frame structure (section 3.3.3), implemented the graph-frame translation (section 3.3.4), and realised the training algorithm and the initial testing structure (section 2.3). The author also implemented the novel architecture (Section 3.3), trained and tuned model #1, and shared the workload in the training and tuning of model #2.

The second author predominantly examined the various publicly available datasets for the data selection (section 3.1.1), implemented the scene parser algorithm (section 3.3.2), defined the test metrics (section 2.3), and shared the workload in the training and tuning of model #2. The author also organised and trained the baseline methods (section 3.4) and benchmarked all the methods based on accuracy, characteristics and processing times (section 4.2).

1.6 Limitations

The scope of the master thesis is limited because of the following reasons:

1. The master thesis would use the publicly available JAAD dataset [63] for intent prediction. Therefore, the scenarios evaluated in this master's thesis is limited to the scenarios available within the dataset.

2. The data available in the JAAD dataset [63] is limited to images from high-resolution monocular cameras mounted on the ego vehicle. Therefore, the input data is limited by the system configuration. Furthermore, other modalities of input data are not considered.

3. The data available in the JAAD dataset is limited to pedestrians. Hence, cyclists are not considered for evaluating the crossing intent. Henceforth, the phrase *vulnerable road users* only represent the pedestrians in the scope of this master thesis.

1.7 Thesis outline

The thesis has been divided into five chapters: chapter 1 presents an overview of the pedestrian intent prediction problem and provides an introduction to the crucial developments in this domain, chapter 2 introduces various essential terms and theoretical concepts that are relevant to this master thesis, chapter 3 describes the methodology used to predict pedestrian intent, chapter 4 analyses the results obtained from the implementation, discusses the project outcome based on the research questions in section 1.4 and examines the significance of this master thesis, and chapter 5 presents the conclusion for the master thesis and examines the future work.

2
Theoretical Background

This chapter presents a theoretical background behind the machine learning algorithms and neural networks relevant to this master thesis. The organisation of this chapter is as follows: section 2.1 explains the fundamentals of artificial intelligence (AI), machine learning (ML) and deep learning (DL), section 2.2 explains the different types of neural networks used in this master thesis, and section 2.3 defines the evaluation criteria used in this master thesis.

2.1 Artificial Intelligence, Machine Learning and Deep Learning

With lower computational cost and faster communication providing unlimited access to information and a better understanding of the physical world around us, heavily automated decision making such as artificial intelligence is becoming the driving technology of the 21st century. However, artificial intelligence has been the core enabler of many applications such as self-driving cars, digital assistants, and medical imaging, to name a few. Despite this, a correct understanding of this critical technology seems scarce. Additionally, due to the immense hype around this technology, there seem to be many misunderstandings between the terminology. This misinterpretation can be mainly observed when the terms artificial intelligence, machine learning, and deep learning are switched around all the time. Although the terms seem equivalent, the meaning of each term varies, and this section aims to clearly articulate the differences between artificial intelligence, machine learning, and deep learning.

2.1.1 Artificial Intelligence

The term AI was first coined by John McCarthy in 1956 when he held the first academic conference on AI at Dartmouth[64]. According to McCarthy, AI is the science and engineering of making intelligent machines[64]. In other words, AI is a sub-field of computer science, just like quantum physics or organic chemistry, which aims to replicate or simulate human intelligence so that machines can perform tasks like visual perception, speech recognition, and decision-making, which humans typically perform.

The applications of AI can range from Deepmind's Alpha Go [65] to IBM Watson [66]. Therefore, AI is usually classified based on its ability to mimic human

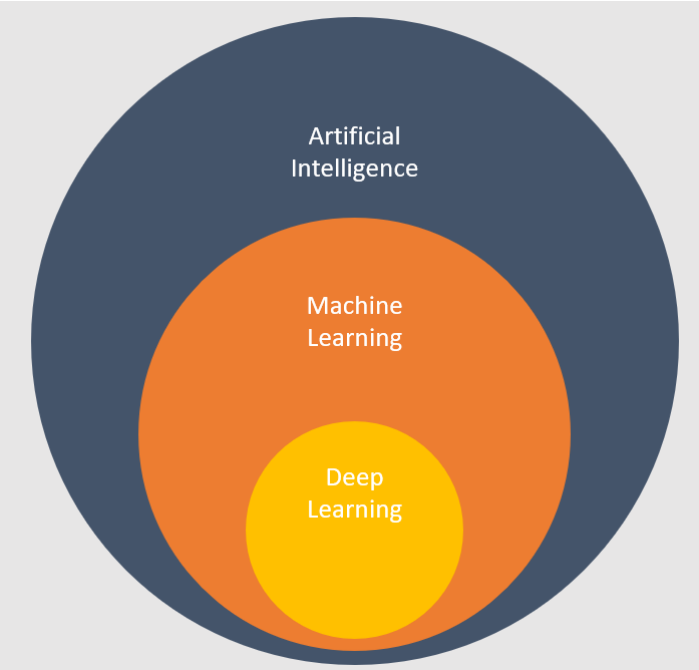

Figure 2.1: Difference between AI, ML and DL

intelligence. Based on the above characteristic, all AI-based systems can be categorised into one of the below three categories:

- Artificial narrow intelligence (ANI)
- Artificial general intelligence (AGI)
- Artificial super intelligence (ASI)

Artificial Narrow Intelligence

ANI, also referred to as weak AI or narrow AI, is the only type of AI humans have successfully realised to date. ANI is goal-oriented and trained to perform a singular task (For example, visual perception or speech recognition) - and outperforms humans at the specific task it is trained to do. Since ANI has a narrow scope of what it can do, even the most intelligent narrow AI in 2021 is nowhere near-human intelligence. Even if some of them can outperform humans in these particular tasks. Some examples of narrow AI are IBM Watson [66], virtual Assistants like Siri by Apple [67], Cortana by Microsoft [68], and others, image recognition software like Google Lens [69], and Tesla's Autopilot system [70].

Artificial General Intelligence

AGI also referred to as strong AI or human-levelled intelligence, is a conceptual AI-based system that mimics human intelligence and can apply its intelligence to solve any problem. AGI can compete with humans on its ability to think, understand, and act indistinguishably from humans in any situation.

AI researchers and scientists are nowhere close to creating AGI. One of the critical challenges to be overcome is intuition. Unlike weak AI-based systems, humans can make intuitive leaps to solve any unknown problem with minimal data. Furthermore, the lack of comprehensive knowledge of the human brain's functionality makes it exponentially difficult for researchers to replicate even primary functions of sight and movement.

Artificial Super Intelligence

ASI is a hypothetical AI-based system that does not just understand human intelligence and behaviour but outperforms even the most intelligent humans in intelligence and ability in all possible fields.

ASI has long been the muse of dystopian science fiction in which robots overthrow and enslave humanity. Since ASI would be better than everything humans do, superintelligent systems' decision-making and problem-solving capabilities would far surpass humans'. Fortunately, AI researchers and scientists do not even dream of creating ASI.

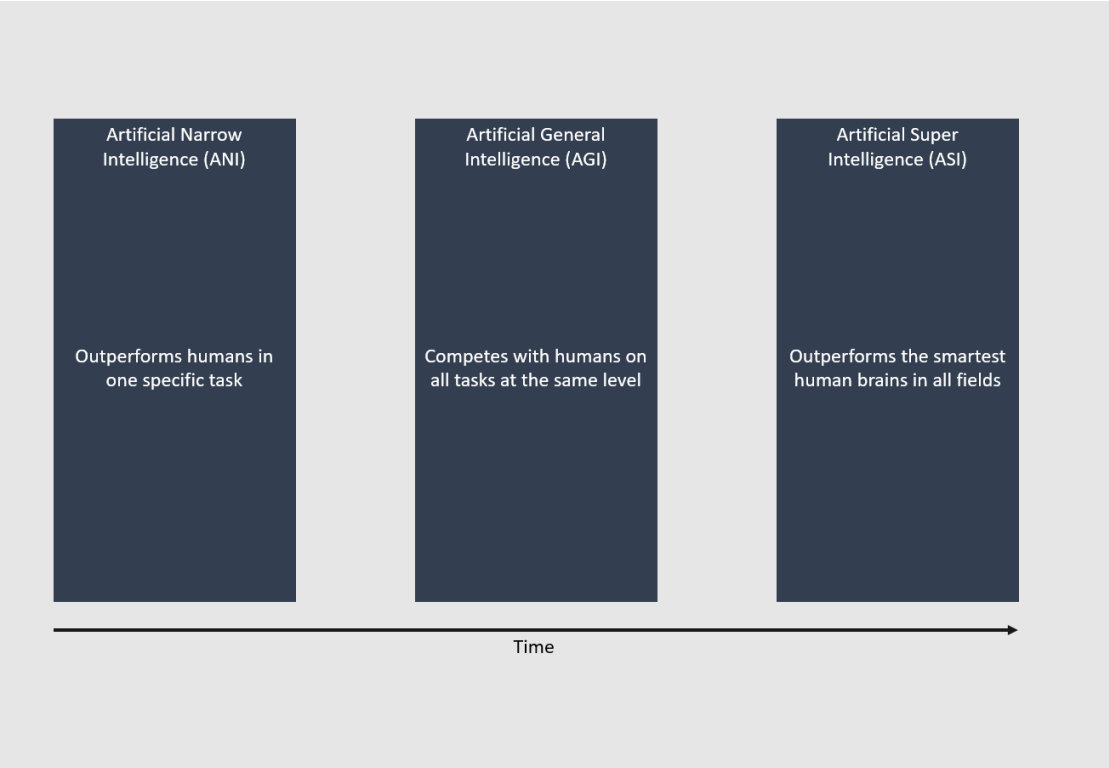

Figure 2.2: Difference between ANI, AGI and ASI

2.1.2 Machine Learning

Arthur Samuel first coined the term ML in 1959. Arthur defined ML as a field of study that gives computers the ability to learn without being explicitly programmed[71]. In other words, the critical idea behind machine learning is to create

algorithms that improve its performance with data. Therefore, the present-day definition of machine learning is as follows: Machine learning is a computer program that learns from experience E for some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [72]. To give an example for the definition mentioned above, to create a machine-learning algorithm that predicts the intent of the pedestrian (T), data with past pedestrian intent patterns (E) must be provided so that the accuracy of the prediction improves over time (P). Other examples of machine learning algorithms used in our day-to-day lives are email spam and malware filters, recommendation engines, and online customer support chatbots.

Machine learning is usually classified into one of the below three categories based on the learning as shown in figure 2.3:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

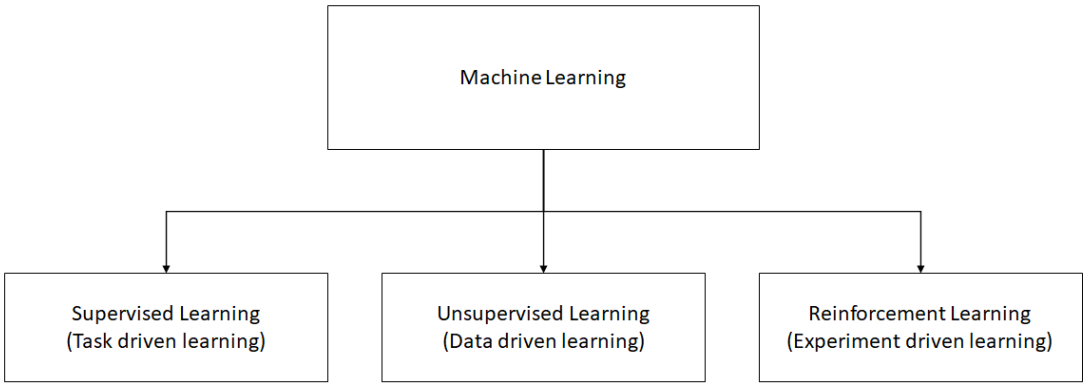

Figure 2.3: Types of ML Algorithms

Supervised Learning

Supervised learning, also called task-driven learning, is a machine learning algorithm that requires labelled datasets to train algorithms that classify data or accurately predict outcomes. In other words, supervised learning is a machine learning algorithm that learns the mapping between input and output variables.

In supervised learning algorithms, training data and the correct outputs are fed to the algorithm, which allows the model to learn the mapping between input and output over time. The trained model is usually run on validation data to evaluate if the model has been trained successfully. The supervised learning algorithm es-

timates its accuracy through a loss function, and the training continues until the model the error has been adequately minimised. Supervised learning is the most commonly used machine learning algorithm, and it is used to solve classification and regression problems.

Some examples of supervised learning algorithms are Naive Bayes algorithm [73], Support Vector Machine [74], Decision Tree [75], and K-Nearest Neighbours [76]. Some problems generally solved by supervised learning algorithms are email spam and malware filters, and cancer detection algorithms.

Unsupervised Learning

Unsupervised learning, also called data-driven learning, is a machine learning algorithm that uses unlabelled datasets to train algorithms. In other words, unsupervised learning is a machine learning algorithm that learns to describe or define the relationship between data.

Unlike supervised learning, unsupervised learning algorithms train only on the input data without output variables. Therefore, unsupervised learning algorithms discover hidden patterns in data without the need for human interference. Unsupervised learning algorithms are instrumental when humans are unaware of the common properties within a dataset. Therefore, unsupervised learning algorithms are used to solve clustering and density estimation problems.

Some examples of unsupervised learning algorithms are K-means clustering algorithm [77], Singular Value Decomposition algorithm [78], and Principal Component Analysis algorithm [79]. Some problems generally solved by supervised learning algorithms are recommender systems and anomaly detection algorithms.

Reinforcement Learning

Reinforcement learning, also called experiment-driven learning, is a machine learning algorithm that uses an environment, which is no fixed training dataset, to achieve a goal or set of goals realised through the agent's actions for which it receives feedback about its performance [80]. In other words, the reinforcement learning algorithm is a machine learning algorithm where an agent operating in an environment must learn to operate using feedback.

Reinforcement learning is similar to supervised learning since the trained agent or model receives feedback signals to optimise its performance, although the feedback may be delayed and statistically noisy, making it challenging for the agent to connect the cause and the effect. Therefore, training reinforcement learning algorithms is comparable to how humans learn: Through trial-and-error. Like humans optimise their behaviour based on the stimuli, agents interacting in an environment use feedback loops to maximise the reward. Therefore, reinforcement learning algorithms are used to solve path planning and motion control problems.

Some examples of reinforcement learning algorithms are Q-Learning algorithms [81], Genetic algorithms [82], DPG algorithm [83], and A3C algorithm [84]. Some common applications for reinforcement learning algorithms are self-driving cars, computer games and resource management problems.

2.1.3 Deep Learning

Igor Aizenberg and his colleagues first coined the term DL in 2000 [85]. Deep learning algorithms are a subset of machine learning algorithms that mimic humans' learning process by teaching the machine to learn by example. Deep learning algorithms use complex multilayer learning structures known as neural networks that learn an implicit representation of the raw data on their own to produce the desired result. In other words, to make traditional machine learning algorithms work, an essential but highly complicated preprocessing step known as feature extraction must be performed manually by domain experts for the algorithms to work. On the other hand, deep learning algorithms learn these extract features automatically as the learning structures within these algorithms optimise to obtain the best possible abstract representation of the input data.

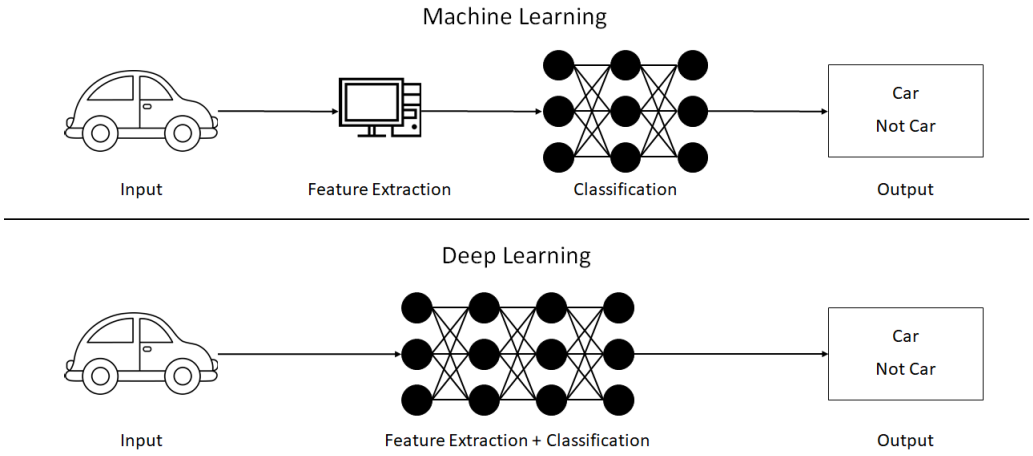

Figure 2.4: Difference between machine learning and deep learning

Due to this, deep learning becomes particularly useful as the majority of the data in the world is unorganised (i.e., it exists in different formats). Another big difference between machine learning and deep learning is that the latter scales better with large amounts of data. In other words, the accuracy of deep learning algorithms tends to increase with an increase in the amount of data, whereas traditional machine learning algorithms stop improving after a saturation point. Due to these reasons, all recent advancements in machine intelligence can be attributed to deep learning algorithms. Deep learning algorithms are the fundamental technology behind voice assistants like Siri [67] and Alexa [86]and are self-driving cars [70].

2.2 Artificial Neural Networks

The simplest definition of an Artificial Neural Network (ANN), according to Dr Robert Hecht-Nielsen, the inventor of the first neurocomputer, is "a computing system made up of several simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs." [87]

Artificial neural networks were initially created as a proof-of-concept attempt to mimic biological neurons in the human brain. Therefore, just as the human brain consists of biological neurons that process the electrical impulses received from adjoining neurons and transmit ahead, artificial neural networks consist of multiple layers of nodes (also known as perceptrons) that process the multiple inputs it receives to produce the output. The output from this single node (or perceptron) can be represented as the equation 2.1.

$$\mathbf{y} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + \mathbf{b} \tag{2.1}$$

As it can be observed from the equation 2.1, the input data to the node is defined as $\mathbf{x}$. The input is received directly from the dataset the neural network is training on or as an output from the previous node represented as $\mathbf{y}_{\mathrm{n}-1}$. The output from the network is represented as $\mathbf{y}$ which is the sum of weighted inputs represented as $\mathbf{w}^{\mathrm{T}}\mathbf{x}$ and the corrective bias $\mathbf{b}$. The parameters weight $\mathbf{w}$ and the corrective bias $\mathbf{b}$ are tunable, and the final values determine the performance of the overall network.

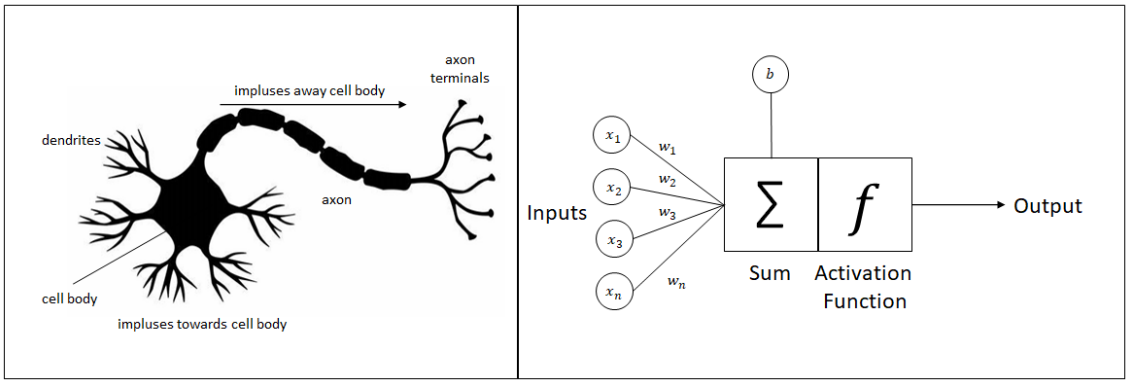

Figure 2.5: Comparison between biological neurons and ANNs

The neural network represented in the figure 2.5 is an example of a feedforward network. Feedforward networks are neural networks in which the connections between the nodes do not form a loop. Feedforward networks are considered one of the simplest types of neural network architectures that are considered the quintessential deep learning models [88]. As an example, convolutional neural networks (further explained in the section 2.2.1) are a specific variety of feedforward neural networks that are used for image data. Since there are no feedback connections in a feedforward network, feedforward networks are considered a stepping stone to recurrent neural networks (further explained in section 2.2.2) that are used for time-series data.

Minsky and Papert show that a single layer neural network cannot solve problems in which the data is not linearly separable, such as the XOR problem [89]. Since most of the data available today are highly unorganised and non-linearly separable, adding one or more layers to the neural network would enable it to solve problems in which data is non-linearly separable. Another reason for adding additional layers is [90], according to which training a single layer neural network that represents any function is highly difficult if not impossible. Hence, it is a common practice to add additional layers to a neural network. The total number of layers in a neural network defines the "depth" of the neural network.

2.2.1 Convolutional neural networks

Convolutional Neural Networks (CNN) are a class of neural networks mainly used for processing visual data. Convolutional neural networks are frequently used for image processing, object classification, and automatically processing and correlating data with a known, grid-like topology [91]. Convolutional neural networks achieve this by using a specialised linear mathematical operation called convolution, which is represented as the equation 2.2.

$$\mathbf{y} = \mathbf{x} * \mathbf{w} \tag{2.2}$$

Similarly to the equation 2.1, the data to the node is defined as $\mathbf{x}$ and output from the network is represented as $\mathbf{y}$. The biggest difference is how the weight $\mathbf{w}$, also known as the kernel, is applied to the input $\mathbf{x}$. Instead of applying the weights through a general multiplication operation, the weights are applied through a convolution operation that is defined as "computing the weighted average of a point of data by its adjacent points of data" [88]. An example of a CNN applied on visual data can be observed in the figure 2.6.

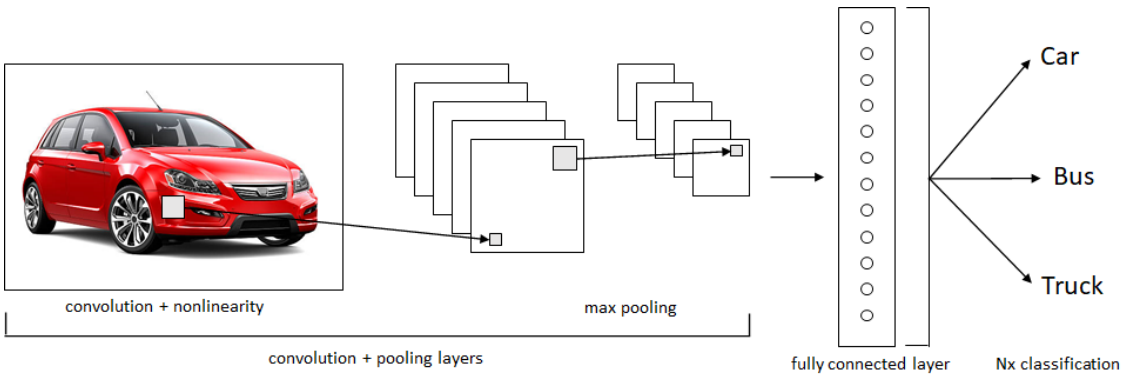

Figure 2.6: An example of a CNN structure applied on visual data. The model analyzes the image in patches through the layers and reduces those values down the line to a classification layer that makes an estimation or a prediction.

2.2.2 Recurrent neural networks

Recurrent Neural Networks (RNN) are another class of neural networks that allow previous outputs to be used as inputs while having hidden states. This property of

recurrent neural networks allows it to process sequential or time-series data that is typically represented as x^1, x^2, x^3, ... x^τ. Unlike feedforward neural networks such as convolutional neural networks, where the inputs and outputs are independent, recurrent neural networks are characterised by their memory parameter, which allows the recurrent neural network to acquire knowledge from prior inputs to modify the current input and output. In other words, the output of the recurrent neural network depends on the historical information in the sequence. Although recurrent neural networks can process historical information, the network has difficulties accessing information from long ago. Additionally, recurrent neural networks are computationally very slow. Due to these reasons, additional specialisations are needed in recurrent neural networks to process long sequences. An example of a recurrent neural network is represented in the figure 2.7.

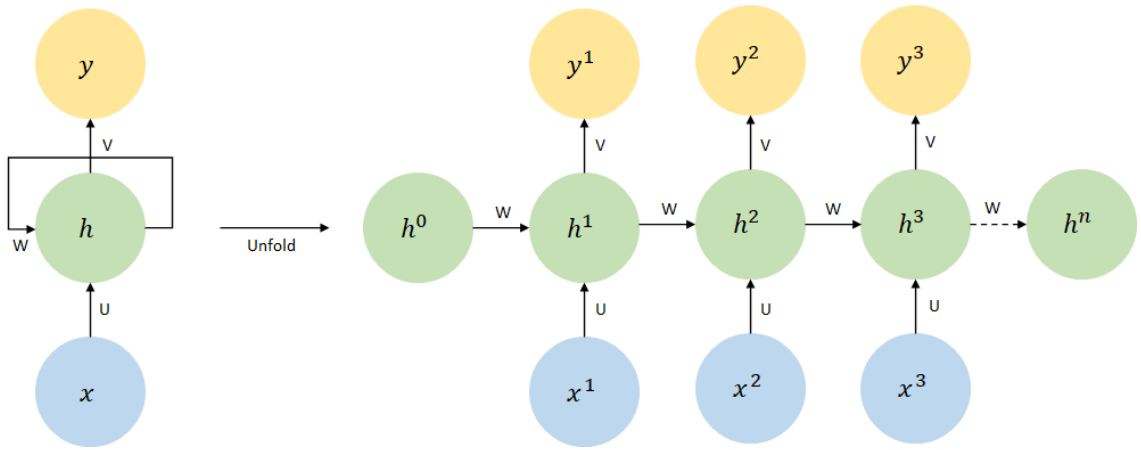

Figure 2.7: An example of how a simple RNN can be structured. The estimation of the previous input is included in the estimation of the current input.

2.2.3 Graph neural networks

Graph Neural Networks (GNN) are a relatively newer class of neural networks that leverage the structure and properties of graphs. Graphs are considered a specific kind of data structure representing the relations (also known as edges) between a collection of entities (also known as nodes). Deep learning models like convolutional neural networks typically take rectangular or grid-like arrays as input. Therefore, graphs are not straightforward to represent in a format that is compatible with deep learning. One of the biggest challenges with graph data structures is the representation of the connectivity between nodes. Once all the properties of graphs are represented in a format compatible with deep learning models, graph neural networks are used to perform an optimisable transformation of graph attributes that preserve the graph symmetries. In other words, graph neural networks are a class of neural networks that accept a graph as an input, with information loaded into its nodes, edges and global context that progressively transform these embeddings without changing the connectivity of the input graph. [92] In recent years, researchers across various disciplines have significantly increased research on graph neural networks. An important reason for this is that graphs can be used to denote

a large number of systems across various areas, including social science, natural science, knowledge graphs and many other research areas [93]. As a unique non-euclidean data structure for machine learning, graph neural networks focus on node classification, link prediction, and clustering tasks.

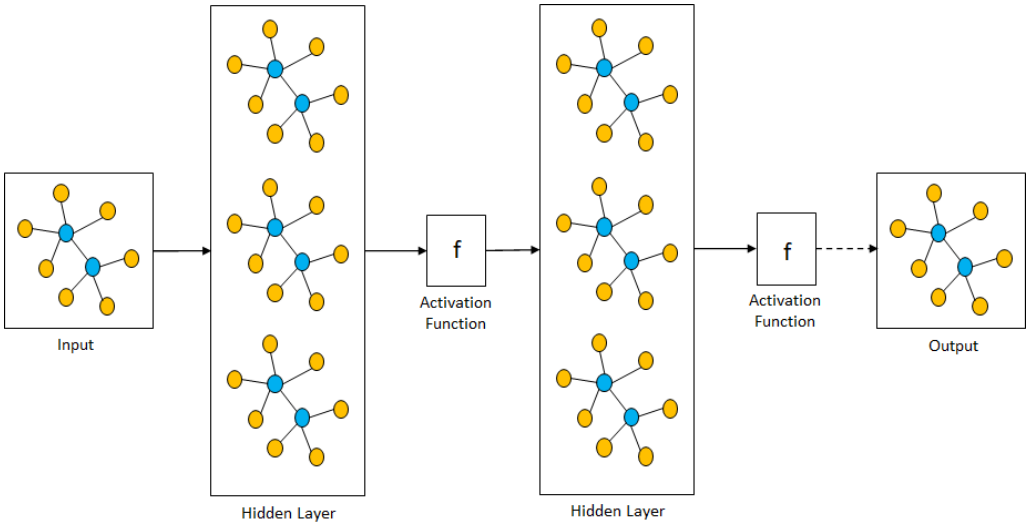

Figure 2.8: An example of a simple GNN

2.2.4 Activation function

In artificial neural networks, the activation function of a node represents the activation pattern of the node. In other words, the activation function determines whether a node would be active or inactive based on the sum of the weighted inputs and the corrective bias to the node. A simple real-world example of an activation function could be an electrical circuit that turns its output ON or OFF based on the combination of inputs. The purpose of activation functions in a neuron is to add non-linearity to the output of neurons. Since neural networks are essentially a combination of nodes where the output of one node is the input to the other, the activation functions get concatenated over time, eventually leading to a highly nonlinear function, which, along with the neural network parameters, represent the training data.

2.2.5 Forward propagation

In artificial neural networks, forward propagation refers to the operations that compute the output of a neural network from the input data. In other words, forward propagation processes the input data through each hidden layer to compute the final output. Forward propagation is necessary for calculating the final value of network parameters that minimise the loss function. This subsection provides a detailed explanation of the forward propagation with a simple example.

Let us consider a simple example of a simple neural network with one input layer,

one hidden layer and an output layer as represented in the figure 2.9. For simplicity, let us also assume that the neural network does not include the corrective bias term.

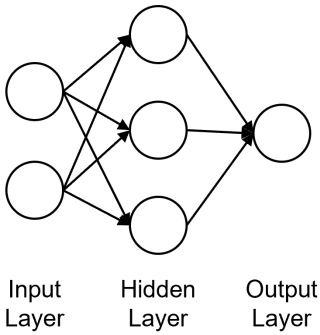

Figure 2.9: A simple neural network to explain forward propagation

When the input data is propagated through the first hidden layer, the intermediate output is represented as the equation 2.3.

$$\mathbf{z}_2 = \mathbf{w}_1^{\mathrm{T}}\mathbf{x} \tag{2.3}$$

Based on the equation 2.3, it can be understood that the input data is represented as $\mathbf{x}$, the weights of the hidden layer are represented as $\mathbf{w}_1$ and the intermediate output of the hidden layer is represented as $\mathbf{z}_2$. On applying the activation function $\mathbf{f}_1$ on the intermediate output of the hidden layer, the activation output from the hidden layer is represented as 2.4.

$$\mathbf{a}_2 = \mathbf{f}_1(\mathbf{z}_2) \tag{2.4}$$

Once the intermediate activation output from the hidden layer is propagated to the output layer, the overall output of the neural network is represented as the equation 2.5.

$$\begin{aligned} \mathbf{z}_3 &= \mathbf{w}_2^{\mathrm{T}}\mathbf{a}_2 \\ \mathbf{y} &= \mathbf{f}_2(\mathbf{z}_3) \end{aligned} \tag{2.5}$$

Where the weights of the output layer are represented as $\mathbf{w}_2$, intermediate output of the output layer is represented as $\mathbf{z}_3$, activation function of the output layer is represented as $\mathbf{f}_2$, and the output of the neural network is represented as $\mathbf{y}$. This process of calculating the neural network output from the input data is called forward propagation. In addition to the overall output, forward propagation calculates the overall loss of the neural network $\mathbf{L}$, which is crucial to tune the overall parameters of the network.

2.2.6 Backward propagation

In artificial neural networks, backward propagation refers to tuning the neural network parameters based on the loss gradient calculated with respect to the network parameters. In other words, backward propagation traverses the network in reverse order, i.e., from the output layer to the input layer, to calculate the loss gradient

and fine-tune the network parameters in a way that reduces the overall loss. In the previous example of figure 2.9, we know that the output $\mathbf{y}$ is a function of the intermediate output $\mathbf{z}_3$, which is then an intermediate output of the weights $\mathbf{w}_2$. In this case, the gradient of output $\mathbf{y}$ with respect to the weights $\mathbf{w}_2$ can be calculated using the chain rule, which is represented in the equation 2.6.

$$\frac{\partial \mathrm{Y}}{\partial \mathrm{W}_2} = \mathbf{prod}\left(\frac{\partial \mathrm{Y}}{\partial \mathrm{Z}_3}, \frac{\partial \mathrm{Z}_3}{\partial \mathrm{W}_2}\right) \tag{2.6}$$

Where the **prod** operator is used for multiplying the arguments after necessary operations such as transposition and swapping. For vectors, the **prod** operator is simply matrix-matrix multiplication. The **prod** operator is used here to hide all the notation overhead. Similar to the equation 2.6, the gradient of output $\mathbf{y}$ with respect to the weights $\mathbf{w}_1$ can be calculated using the chain rule, which is represented in the equation 2.7.

$$\frac{\partial \mathrm{Y}}{\partial \mathrm{W}_1} = \mathbf{prod}\left(\frac{\partial \mathrm{Y}}{\partial \mathrm{Z}_3}, \frac{\partial \mathrm{Z}_3}{\partial \mathrm{A}_2}, \frac{\partial \mathrm{A}_2}{\partial \mathrm{Z}_2}, \frac{\partial \mathrm{Z}_2}{\partial \mathrm{W}_1}\right) \tag{2.7}$$

This process of calculating loss gradients with respect to overall network parameters is known as backward propagation. Based on the values of these loss gradients calculated in the backward propagation, the overall network parameters are updated during training to reduce the overall loss.

2.2.7 Regularization

Regularisation refers to the various techniques that discourage learning complex neural network models, thus reducing the risk of overfitting the training data. This subsection describes the most prevalent and efficient regularisation techniques: weight regularisation, batch normalisation, and dropout regularisation.

Weight regularization

Weight regularisation, also known as weight decay or ride regression, is one of the most common regularisation techniques where the essential idea is to encourage simpler models by penalising larger weights. During the weight regularisation, the loss function of the neural network is extended by a so-called regularisation term. The regularisation term can either be dependent on the sum of absolute values of the weights, also known as L1 regularisation, or the sum of squared values of the weights, also known as L2 regularisation. Since larger weights result in a more significant penalty, the optimisation algorithm pushes the model to have smaller weights, thus simplifying the model. The regularisation term is also scaled by the hyperparameter α, determining how much regularisation is required for the model. The updated loss functions based on L1 and L2 regularisation are represented as the equations 2.8 and 2.9 respectively.

Loss function with L1 regularisation:

$$\widehat{\mathrm{L}}(\mathrm{w}) = \mathrm{L}(\mathrm{w}) + \alpha \left\|\mathrm{w}\right\|_1 = \mathrm{L}(\mathrm{w}) + \alpha \sum_{\mathrm{i}} \left|w_i\right| \tag{2.8}$$

Loss function with L2 regularisation:

$$\widehat{\mathrm{L}}(\mathrm{w}) = \mathrm{L}(\mathrm{w}) + \frac{\alpha}{2} \left\|\mathrm{w}\right\|_2^2 = \mathrm{L}(\mathrm{w}) + \frac{\alpha}{2} \sum_{\mathrm{i}} \mathrm{w}_{\mathrm{i}}^2 \tag{2.9}$$

Batch normalization

Batch normalisation is another commonly used regularisation technique where the neural networks are made faster and more stable by adding additional layers to the neural network that performs the normalisation operation by recentering and rescaling the inputs of a layer. Batch normalisation techniques are primarily used to mitigate the problem of internal covariate shift, where the distribution of network activations change due to change in network parameters during training [94]. In other words, batch normalisation limits the distribution of inputs to a layer when the parameters of the preceding layers change. These covariate shifts are highly problematic with an increasing number of layers, resulting in a reduction in model accuracies. Batch normalisation normalises a layer input by subtracting the batch mean and dividing it by standard deviation. The normalisation ensures that the layer inputs have a mean and standard deviation of 0 and 1, respectively, thereby fixing the problem of internal covariate shift. However, the normalisation of layer inputs also compromises the nonlinear relationship the neural network learns during the training process. This reduction in non-linearity is mitigated by adding additional trainable parameters that scale and shift the normalised values to accommodate the distribution of the input dataset.

Dropout regularisation

Dropout regularisation is another famous and powerful regularisation technique where the key idea is to randomly drop nodes along with their connections from the neural network during training [95]. During the training of neural networks with dropout regularisation, multiple "thinned" networks are created with a unique combination of nodes dropped randomly in the hidden layers. In other words, multiple "thinned" neural networks are created based on probability hyperparameter P at each update of the gradient. During testing, the approximate effect of averaging the predictions from all the "thinned" neural networks is replicated by using one neural network with smaller weights. Therefore, dropout regularisation prevents nodes from co-adapting too much to the dataset, thus, significantly reducing overfitting.

2.3 Evaluation of model

Assessing a machine learning model is as important as building it. Since machine learning models are developed to perform on unseen data, a meticulous evaluation is required to create a robust model. This section provides a brief description of various evaluation metrics used to determine model performance.

One of the most straightforward evaluation metrics is accuracy which determines how many predictions are correct. The formula for accuracy is represented as the equation (2.10).

$$\text{accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{2.10}$$

Although accuracy is a straightforward metric, it can provide an unreliable assessment with unbalanced datasets. For example, if the dataset containing two classes comprising 95% samples from class A and 5% samples from class B, and the model only correctly predict samples from class A, the model's accuracy would be 95%, which can be a highly misleading conclusion if samples from class B are of interest.

In the example mentioned above, additional insights like classwise accuracy could have helped identify the inherent problems in the model. A confusion matrix is a table that precisely does that. A confusion matrix does not evaluate the model's performance but provides further insights by displaying the number of correct and incorrect predictions for each class. Therefore, for the example mentioned above, the confusion matrix is a $\mathbb{R}^{2\times2}$ matrix represented as the figure (2.10).

Based on the number of True Positives (TP), False Negatives (FN), False Positives (FP) and True Negatives (TN) estimated from the confusion matrix, additional performance metrics like Precision and Recall can be calculated. Precision measures the performance of the model when the prediction is positive. In other words, Precision determines how many positive predictions are true [96]. Therefore, the formula for Precision is represented as the equation (2.11).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.11}$$

Another important metric that determines the performance of the model is Recall. Recall, also known as sensitivity, measures the performance of the model while predicting positives. In other words, Recall signifies how many actual positives are identified correctly [96]. Therefore, the formula for Recall is represented as the equation (2.12).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.12}$$

Both Precision and Recall must be calculated to determine the performance of the model. However, it is essential to note that both Precision and Recall cannot be maximised simultaneously as both metrics operate in a zero-sum game framework. In other words, increasing Precision reduces Recall and vice-versa. Therefore, either Precision or Recall can be maximised based on the task.

Another metric that can determine the model's performance based on both Precision and Recall is the F1 score. F1 score provides a weighted average of Precision and Recall. The formula for the F1 score is represented as the equation (2.13).

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{2.13}$$

As it can be observed from the equation (2.13), the F1 score is the harmonic mean of Precision and Recall. Since the F1 score combines both Precision and Recall, the F1 score is an essential metric for imbalanced datasets as it requires both Precision and Recall to have a reasonable value. Therefore, the highest possible value of an F1-score is 1, indicating perfect Precision and Recall, whereas a 0 indicates the value of either Precision or Recall to be zero [97].

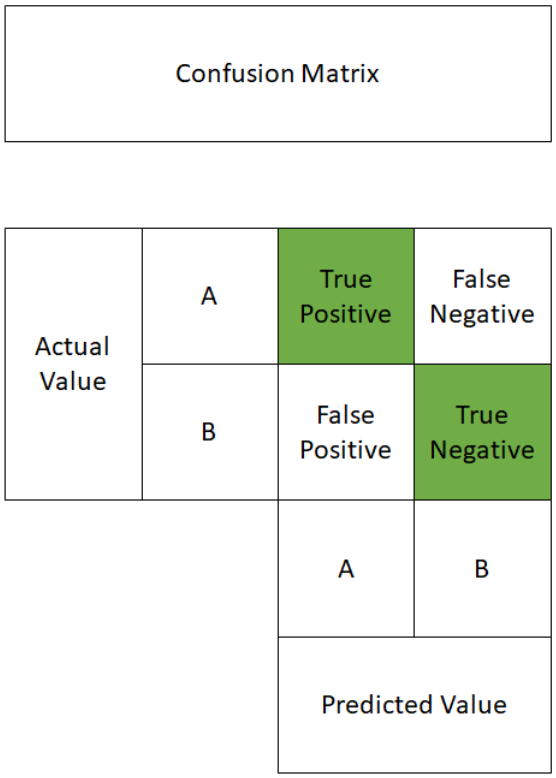

Figure 2.10: Confusion matrix

3
Methods

This chapter presents the methodology used to predict pedestrian intent. The organisation of this chapter is as follows: section 3.1 presents the data extraction method utilized for the available datasets, section 3.2 introduces the machine learning framework used, 3.3 analyzes the novel architecture employed to predict pedestrian intent, and section 3.4 illustrates the experimentation and benchmarking methodology.

3.1 Data extraction and pre-processing

The data extraction and pre-processing involve selecting an appropriate dataset and extraction and pre-processing of required features.

3.1.1 Dataset selection

Since the pedestrian intent prediction problem has been a topic of active research, various datasets are being continually developed to evaluate the performance of the state of the art methods. Therefore, to measure the overall progress in solving the problem, choosing a suitable dataset is critical. Presently, two high-quality datasets are publicly available: JAAD and PIE. Both datasets were created to study pedestrian intent but have a slightly different area of focus. This section provides a brief introduction to both datasets and explains the reasoning behind the final selection.

Joint attention in autonomous driving dataset

JAAD is a dataset for studying joint attention in the context of autonomous driving. The focus is on pedestrian and driver behaviours at the crossing point and the factors that influence them. To this end, the JAAD dataset provides a richly annotated collection of 346 short video clips (5-10 sec long) extracted from over 240 hours of driving footage. These videos filmed in North America and Eastern Europe represent scenes typical for everyday urban driving in various weather conditions. Bounding boxes with occlusion tags are provided for all pedestrians making this dataset suitable for pedestrian detection. Behaviour annotations specify behaviours for pedestrians that interact with or require the attention of the driver. Several tags (e.g., weather and locations) and timestamped behaviour labels from a fixed list (e.g., stopped, walking, and looking) are available for each video. Also, a list of demographic attributes is provided for each pedestrian (e.g., age, gender, and the

direction of motion), and a list of visible traffic scene elements (e.g., stop sign and traffic signal) is provided for each frame [63].

Pedestrian intent estimation dataset

PIE is a relatively newer dataset for studying pedestrian behaviour in traffic. The focus of the dataset is on intent estimation and trajectory prediction. To this end, PIE contains over 6 hours of footage recorded in typical traffic scenes with an on-board camera. The dataset also provides accurate vehicle information from the OBD sensor (vehicle speed, heading direction, and GPS coordinates) synchronised with video footage. Rich spatial and behavioural annotations are available for pedestrians and vehicles that potentially interact with the ego-vehicle and the relevant infrastructure elements (traffic lights, signs, and zebra crossings). The dataset contains over 300K labelled video frames with 1842 pedestrian samples making it one of the most extensive publicly available datasets for studying pedestrian traffic. The PIE dataset also contains 898 examples of people who intend to but do not cross, 512 pedestrians to cross who eventually cross in front of the vehicle, and 430 pedestrians with no crossing intention [41].

Selection of Dataset

In order to select an appropriate dataset for the thesis, the decision-matrix method was used to compare the datasets. The decision-matrix method or the Pugh analysis is a technique that helps identify the most probable solution among all alternatives. [98] The method refines a list of alternatives using a matrix-based process to weigh and compare the approaches. Since comparing and evaluating alternatives can be tedious, using a systematic approach like the Pugh analysis helps reduce bias from the decision-making process and provide a consistent approach for selecting among several concepts [99].

In order to create the decision-matrix, the following steps were followed [98]:

1. **Define evaluation criteria:** Evaluation criteria are the parameters on which the alternatives are compared.
2. **Determine the significance of each criterion:** Add weights to evaluation criteria to signify the relative importance of each criterion.
3. **Specify different approaches:** Define different approaches to be compared.
4. **Rate different approaches:** For each criterion, rate each alternative +1 for better, 0 for same, and -1 for worse.
5. **Aggregate the scores:** Calculate the weighted sum for each alternative.
6. **Select the best alternative:** Select the approach with the highest score.

Based on the steps mentioned above, the JAAD and the PIE datasets were compared based on the ease of implementation, availability of resources, the number of pedestrians, types of features and variance of the dataset. The overall table is presented below:

Decision-Matrix				
#	Criterion	Weights	JAAD	PIE
1	Ease of implementation	2	+1	-1
2	Availability of resources	15	+1	-1
3	Number of pedestrians	5	-1	+1
4	Types of features	3	-1	+1
5	Variance of the dataset	5	-1	+1
Overall Score			+4	-4

Table 3.1: Decision-matrix to select the appropriate dataset

Based on the overall score from the 3.1, the JAAD dataset was selected to predict pedestrian intent.

3.1.2 Feature selection

Once the appropriate dataset is selected to predict pedestrian intent, it is crucial to select the features used to predict pedestrian intent. The JAAD dataset provides a large number of features for pedestrians that they divide into three categories as represented in the table 3.2.

Categorisation of features available for pedestrians		
#	Category	Features
1	Pedestrian Behaviour	Bounding Box, Occlusion, Action, Reaction, Nod, Hand Gesture, Cross, Look
2	Pedestrian Appearance	Pose, Clothes, Backpack, Bag, Object
3	Pedestrian Attribute	Age, Crossing, Crossing Point, Decision Point, Group Size, Motion Direction

Table 3.2: Categorisation of features available for pedestrians in the JAAD dataset

As an initial approach, all the features available from the JAAD dataset for each pedestrian was used to train the intent prediction model.

3.1.3 Cleaning the dataset

Since the JAAD dataset contains behaviour information only for a small subset of all the annotated pedestrians available in the dataset, the dataset had to be filtered to only consider the pedestrian with behavioural information. Therefore, all pedestrians without behavioural information were removed from the original dataset to train the intent prediction model. Additionally, videos that did not have any pedestrians were also removed to clean the dataset further.

3.2 Machine learning framework

Once an appropriate dataset is selected and processed, selecting a suitable machine learning framework is crucial to implement the solution. Presently, the three most

popular open-source machine learning frameworks are Tensorflow, Keras and Pytorch. This section provides a brief introduction to the different machine learning frameworks and explains the reasoning behind the final selection.

Tensorflow is an end-to-end open-source framework for machine learning developed by Google. Tensorflow has a broad ecosystem of tools, libraries, and community resources that allows researchers and developers to develop state-of-the-art machine learning applications [100]. Tensorflow is primarily based on dataflow and differentiable programming and provides both high and low-level interfaces to develop various machine learning applications. Although TensorFlow was initially developed to conduct machine learning research, the system is generic enough to apply in a wide variety of other domains. Tensorflow provides stable interfaces in both Python and C++ [101].

Keras is another open-source neural network framework developed by Google. Keras provide a high-level interface written in Python that can run on top of Tensorflow, CNTK, and Theano. Keras contains various implementations of commonly used neural network building blocks such as neural network layers, optimisers, activation functions, and a host of tools to make development with various data formats more convenient to simplify the development of deep neural networks [102]. Keras has gained favour for its ease of use and syntactic simplicity, facilitating fast development. Keras also provides consistent interfaces, clear error messages, and thorough documentation and developer guides [101].

Pytorch is another popular open-source machine learning framework that the Facebook AI Research lab primarily develops. Pytorch accelerates the path from research prototyping to production deployment as Pytorch enables fast, flexible experimentation and efficient production through a user-friendly front-end, distributed training, and ecosystem of tools and libraries [103]. Although Pytorch was released much later than Tensorflow, researchers are increasingly adopting PyTorch as their preferred framework because of its Pythonic interface and the ease of building highly complex neural networks [101].

Since neither of the three frameworks is objectively better than the other, the decision-matrix method was used to select the suitable framework based on ease of usage, library support, training duration, flexibility and debugging capabilities. The overall table is presented below:

Decision-Matrix					
#	Criterion	Weights	Keras	Tensorflow	Pytorch
1	Ease of usage	5	+1	0	0
2	Library support	10	0	0	+1
3	Training Duration	3	-1	0	+1
4	Flexibility	5	-1	-1	+1
5	Debugging Capabilities	10	-1	-1	+1
Overall Score			-13	-15	+28

Table 3.3: Decision-Matrix to select the appropriate machine learning framework

Based on the overall score from the 3.3, Pytorch was selected as the suitable machine learning framework.

3.3 Novel architecture

In this section, we will define and motivate the general structure of our novel architecture. We present an overview of the structure in section 3.3.1 while the subsequent sections will break down each part in more detail.

3.3.1 Architecture structure

As stated in the introduction, the goal of the thesis is to create an architecture that would be as close as possible to real-world implementation, even if testing this is not part of the scope of this thesis. Therefore, the thesis attempts to create an architecture that reduces the gap between prototyping and production by proposing an end-to-end network that adopts the real-world data pipeline.

Based on the literature survey for this master thesis, the scene graph parsing and visual reasoning approach was considered the most appropriate approach for this master thesis. Since we argue that pedestrian intent is primarily dependent on context and interactions in the scene rather than implicit features, creating a spatiotemporal model that captures intrinsic scene dynamics by encoding the sequence of subtle human actions is crucial to this master thesis. One such model is the [6], where pedestrian intent is predicted by parsing the scene graph and modelling the spatiotemporal relationship between various objects in the scene. Although the pedestrian intent prediction model developed by [6] is closest to our master thesis, one of the most significant drawbacks with [6] is that the method is not developed considering real-time processing of inputs. Since the goal of the thesis is to create a model that would be as close to real-time processing as possible, the novel architecture developed in this master thesis is an improvement upon the model developed in [6] that adopts an end-to-end structure to process input videos in real-time. The structure of the novel architecture is represented as the figure 3.1.

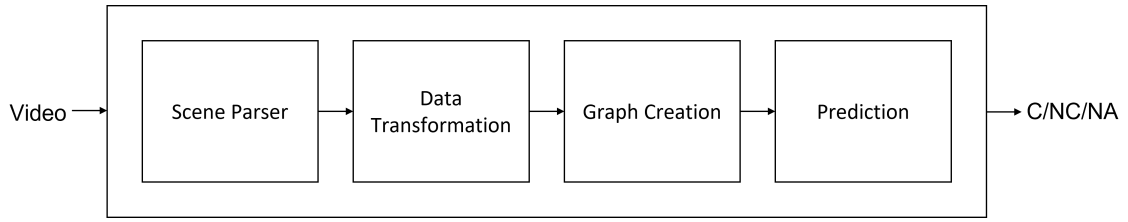

Figure 3.1: Novel architecture proposed in this master thesis

3.3.2 Scene Parser

As it can be observed from the figure 3.1, a scene parser algorithm is used to parse the input video frame to identify spatial information about various objects present in the frame. The table 3.4 lists all the objects of interest that are considered by the scene parser in our thesis.

Objects of interest for scene parser		
#	Category	Features
1	Vehicle	Car, Bus, Bike, Motorbike, Cycle, and others
2	Road users	Pedestrians

Table 3.4: Objects of interest for scene parser

Based on the table 3.4, an "off-the-shelf" YOLOv4-Deepsort algorithm was used as the scene parser algorithm. A high confidence threshold of 0.8 and intersection-over-union (IOU) threshold of 0.5 was used to reduce false positives. With the spatial features for the objects collected from the scene parser, the spatial features are combined with the behavioural, appearance and attribute features collected from the JAAD dataset [63] to create the overall feature set. This comprehensive feature set is then restructured to a frame-by-frame structure to make the feature set compatible with the graph creation algorithm. Based on this restructured feature set, the overall features are transformed into graphs by the graph creation algorithm and fed to the novel prediction model before predicting pedestrian intent. The prediction model is designed to process each graph frame sequentially to train the weights and predict pedestrian intent. The model predicts crossing 15 frames, 30 frames and 45 frames in advance for each pedestrian, corresponding to 0.5 seconds, 1 second and 1.5 seconds in advance. Sequentially connecting these algorithms emulates an end-to-end architecture that attempts to reduce the gap between prototyping and production.

3.3.3 Data transformation

Since the prediction model predicts pedestrian intent based on intrinsic scene dynamics, encoding the spatiotemporal relationship between various objects in the scene is crucial to this master thesis. Since the output from the scene parser algorithm is incompatible with the graph creation algorithm, multiple data transformations are necessary to model the spatiotemporal relationship between various objects in the

scene and make the modelled scene dynamics compatible with the prediction model. This section describes the various data transformations performed to make the output of the scene parser algorithm compatible with the graph creation algorithm.

As mentioned in section 3.1.1, the JAAD dataset [63] provides a large number of pedestrian attributes. However, the biggest drawback of the dataset is the structure of the annotations. Instead of structuring the dataset as a frame-by-frame structure, the annotations are structured as a pedestrian-focused structure. In other words, the dataset follows the complete sequence of one pedestrian at a time, regardless of the total number of pedestrians in the input video or frames at that time. Since this thesis aims to create an end-to-end intent prediction model, restructuring the dataset from a pedestrian-focused structure to a frame-by-frame structure becomes an absolute necessity. A frame-by-frame structure allows the model to abstract any implicit social relations or interactions between pedestrians to improve the prediction. No pre-existing sorting algorithm that accomplishes the goal mentioned above was found; a new algorithm was developed from scratch to achieve this goal. The input to the algorithm is the object detections from the scene parser algorithm along with the pedestrian features from the JAAD dataset [63]. The overall set of features are then restructured to a frame-by-frame structure so that each frame of each video would directly contain all the information available in sequence to allow for real-time processing of information. A visual representation of the overall data transformation process is represented as the figure 3.2.

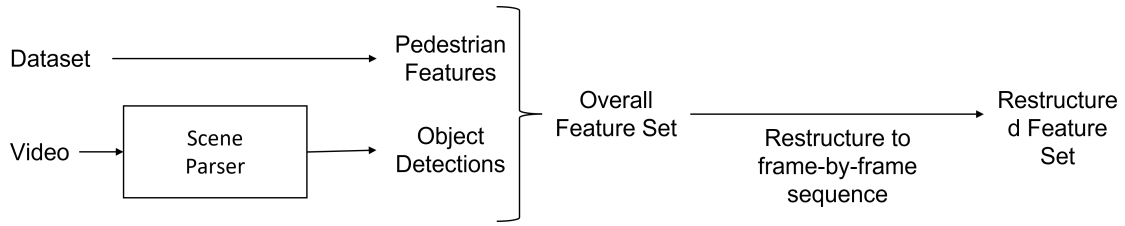

Figure 3.2: Visual representation of data transformation process

3.3.4 Graph frame creation

As mentioned in the previous sections, the prediction model is incompatible with the restructured feature set generated by the data transformation algorithm. Therefore, to model the spatiotemporal relationship between various objects in the scene, the restructured "frame-by-frame" feature set is further translated to a sequence of graph frames to be compatible with the prediction model. As the name suggests, a graph frame models the spatial relationship between all objects and the pedestrians present in the frame graphically. Therefore, a graph frame is essentially a graph structure corresponding to a specific video frame. A visual representation of the overall graph creation process is represented as the figure 3.3.

As it can be observed from the figure 3.3, each pedestrian or object parsed by the scene parser is modelled as a graph node, whereas the graph edges are used to reflect the spatial relationship between various objects in the graph frame. Furthermore,

each graph frame is modelled using two types of node: Pedestrian node and object node. Each pedestrian node is defined by its implicit features such as behaviour, appearance and attribute, whereas an objects attribute defines the object nodes. The vector representation of the pedestrian node and object node is represented as the equation 3.1.

$$\begin{aligned} \text{ped}_\text{i} &= \begin{bmatrix} x_i^{max} & y_i^{max} & x_i^{min} & y_i^{min} & occ_i & beh_i & app_i & att_i \end{bmatrix} \\ \text{obj}_\text{i} &= \begin{bmatrix} x_i^{max} & y_i^{max} & x_i^{min} & y_i^{min} \end{bmatrix} \end{aligned} \tag{3.1}$$

Since the graph frame encodes every information from the visual frame, the shape of the graph frame may vary due to the varying number of objects in the scene. A simple workaround was achieved by considering a static graph with a maximum of fifty objects per frame to model the pedestrian intent.

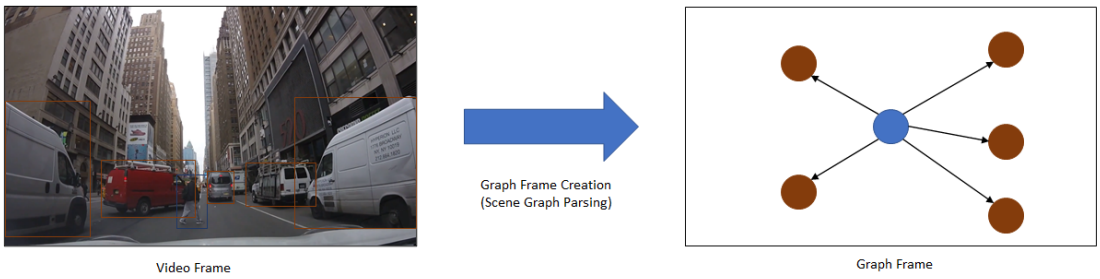

Figure 3.3: Visual representation of graph frame creation process: The graph nodes represented in the graph frame represents a pedestrian or an object from the visual frame whereas the edges represent the spatial relationship between the nodes

3.3.5 Prediction architecture

Once each graph frame corresponding to a particular video frame is constructed to model the spatial relationship between various objects in the scene, the temporal relationship between different frames is modelled using Graph Convolutional Long Short-Term Memory (i.e., GCLSTMs). However, these temporal connections among various objects across frames are not drawn directly from the graph frames. Instead, abstracted information from Graph Convolution layers (i.e., GCONV) is first generated to model the temporal connections. To achieve this spatiotemporal modelling, the prediction model performs graph convolution twice on each observed frame, where the features for the nodes are used for modelling the temporal connections. Lastly, the temporal connections are connected to classifier layers to classify the pedestrian as crossing, not crossing, or not applicable. Interestingly, this classifier layer was achieved with two variants: the first variant achieved the classifier layer without the softmax function, and the second variant achieved the classifier layer with the softmax layer. Different training methodologies were used to train these variants to compare the performance of the prediction architectures. The overall structure of the prediction model is represented as the figure 3.4.

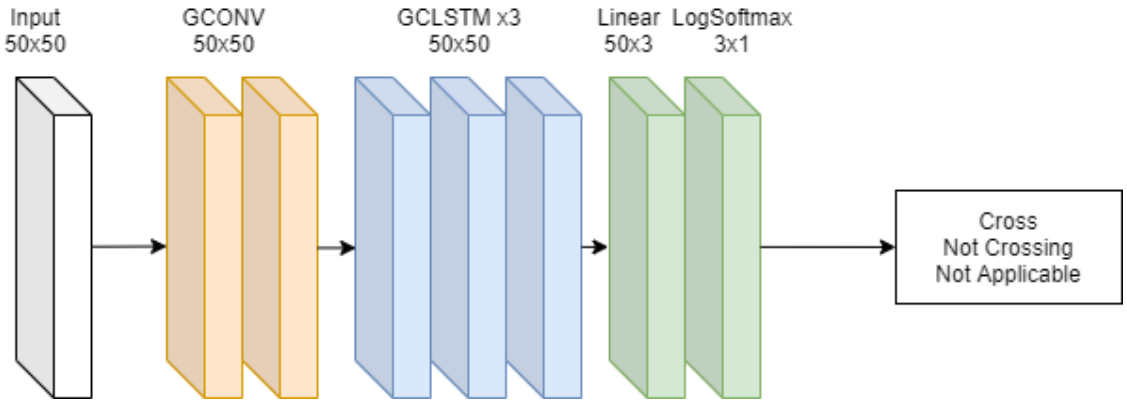

Figure 3.4: Overall prediction architecture

3.4 Experimentation and benchmarking

As the proposed architecture resembles an end-to-end structure, the first 264 videos are used to train the model, whereas videos from 265 to 346 are used to test the model. In addition to training and testing the novel architecture, this master thesis aims to measure the overall progress towards solving the pedestrian prediction problem. The master thesis achieves this by investigating the performance of multiple baseline methods on the JAAD dataset. Since standard training and testing procedures are crucial to evaluate the performance of baseline methods, a common data split was utilised for training and testing other baseline models. In this master thesis, the performance of the novel architecture was benchmarked against five different models: the Fusion of Spatio-temporal Skeletons for Intention Prediction model (FUSSI) [4], the Static model (Static) [104], the Stacked with multilevel Fusion RNN model (SFRNN) [104], the Convolutional 3D model (C3D) [104], and the Pedestrian Crossing Prediction with Attention model (PCPA) [104]. Out of the five above mentioned models, only the FUSSI model [4] is an end-to-end model. Other non-end-to-end models are trained and tested using a common standard but different from the one used by the end-to-end models. The training parameters for all the models are represented in the table 3.5.

Training Parameters							
#	Models	Dataset	Batch	Loss	LR	Optimiser	Epochs
1	Static [104]	JAAD	32	BCE	1e-6	ADAM	5
2	SFRNN [104]	JAAD	32	BCE	1e-7	ADAM	40
3	C3D [104]	JAAD	16	BCE	5e-6	ADAM	40
4	PCPA [104]	JAAD	8	BCE	5e-5	ADAM	20
5	Novel (#1)	JAAD	1	MSE	1e-3	ADAM	5
6	Novel (#2)	JAAD	1	NLL	1e-3	ADAM	5

Table 3.5: Experimental setup from training models

4

Results and Discussion

This chapter presents the results from the novel intent prediction model developed in this master thesis. The organisation of this chapter is as follows: section 4.1 presents the results from the novel intent prediction model, section 4.2 compares the results from the novel intent prediction model with the baseline models, and section 4.3 discusses the project outcome based on the research questions in section 1.4 and examines the significance of this master thesis.

4.1 Novel intent prediction model

The goal of this section is to analyse the results obtained from the novel intent prediction model developed in this master thesis and discuss the project outcome based on the research questions in section 1.4. The results are primarily analysed on the effect of time-to-event on the accuracy of the model.

Figure 4.1: Examples of intent prediction for crossing and not-crossing pedestrians

Effect of time-to-event

Since the JAAD dataset has quite some variations in the time-to-event, figure 4.2 documents the variation of accuracy with changes in the time-to-event. Time-to-event or the prediction length indicates how early the intent is predicted. For this master thesis, the model's accuracy is calculated at three different time-to-event or prediction lengths: 0.5 seconds, 1 second and 1.5 seconds.The results are presented in the figure 4.2.

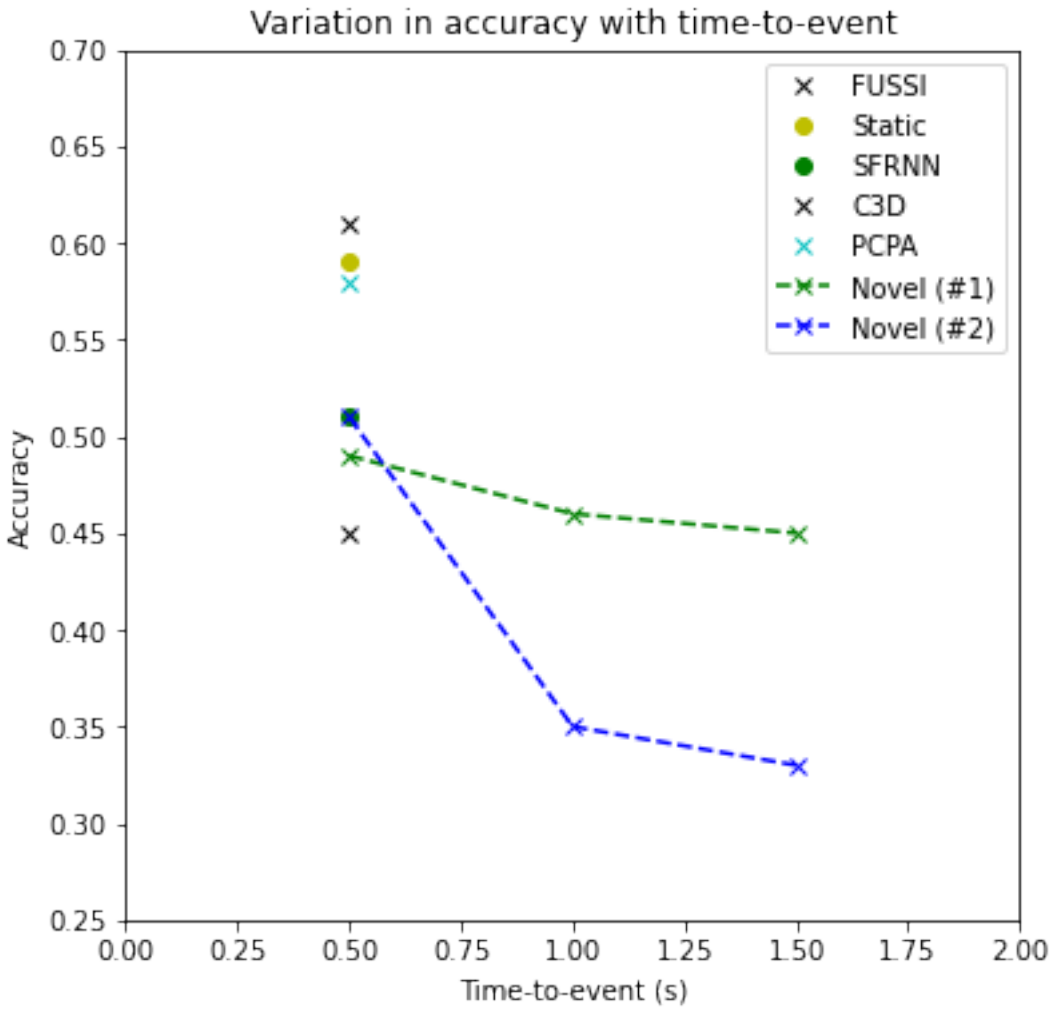

Figure 4.2: Effect of time to event on accuracy of the novel intent prediction model

As it can be observed in the figure 4.2, the first variant of the proposed intent prediction model achieves an accuracy of 49% while predicting pedestrian intent 0.5 seconds in advance, 46% while predicting pedestrian intent 1 second in advance, and 45% while predicting pedestrian intent 1.5 seconds in advance. Similarly, the second variant of the proposed intent prediction model achieves an accuracy of 51% while predicting pedestrian intent 0.5 seconds in advance, 35% while predicting pedestrian intent 1 second in advance, and 33% while predicting pedestrian intent 1.5 seconds in advance. At the same time, the accuracies of various baseline models

can also be observed in the figure. Since most of the baseline methods were trained to predict only 0.5 seconds in advance, it can be observed that there are no accuracy metrics for baseline methods at 1 second and 1.5 seconds. Another interesting trend observed in the figure is the drastic reduction in accuracy when the time-to-event increases from 0.5 seconds to 1 second. On further analysis, this severe reduction in accuracy metrics can be attributed to a lack of scene information. However, it can also be observed that the accuracy remains relatively the same when the time-to-event increases from 1 second to 1.5 seconds. On analysing this result further, it was observed that most pedestrian crossing events are between 0.5 seconds to 1 second, which means that the crossing event is complete by 1.5 seconds, which the model seems to be predicted well.

4.2 Comparison with baseline methods

This section aims to analyse the results obtained from benchmarking the novel intent prediction model developed in this master thesis against the baseline models and investigate the key characteristics that explain the differences in the performance between the novel and baseline models.

Table 4.1 presents the results obtained from benchmarking the novel intent prediction model developed in this master thesis against the baseline models. It can be observed from the table 4.1 that the C3D model [104] has the best accuracy and recall scores, whereas the FUSSI model has the best AUC, precision and F1 score of all the methods. A high percentage of precision indicates that a high number of positive predictions are true, whereas a high percentage of recall signifies how many actual positives are identified correctly. In the case of the novel model, one of the reasons for the lower precision and recall scores could be the simultaneous prediction for multiple pedestrians. Unlike datasets based on complete crossing sequences, a frame-by-frame dataset provides all the information in the frame, which means that predictions for multiple pedestrian crossing simultaneously are expected.

Benchmarking of models						
#	Model	Accuracy	AUC	F1	Precision	Recall
1	FUSSI [4]	0.45	**0.89**	**0.83**	**0.79**	0.89
2	Static [104]	0.59	0.52	0.71	0.63	0.82
3	SFRNN [104]	0.51	0.45	0.63	0.61	0.64
4	C3D [104]	**0.61**	0.51	0.75	0.63	**0.91**
5	PCPA [104]	0.58	0.5	0.71	0.67	0.75
6	Novel Architecture (#2)	0.51	-	0.39	0.33	0.49

Table 4.1: Benchmarking results of the novel intent prediction model developed in this master thesis against the baseline models

Additionally, since the information is provided frame-by-frame, the model only has access to partial information at any given time. Another reason for the lower preci-

sion and recall score could be the design of the proposed model. Since the complete end-to-end architecture is used to test the videos, the performance of the scene parser algorithm also influences the precision and recall values of the overall network. Lastly, the lower precision and recall values could be caused due to the method by which the novel prediction model predicts pedestrian intent. Since the novel architecture is designed end-to-end, the model starts predicting pedestrian intent even without processing the minimum number of graph frames required to predict pedestrian intent accurately, resulting in a higher number of incorrect results at the beginning, which could also decrease overall precision and recall scores.

Similar to the results obtained in the table 4.1, a comparison between the novel intent prediction model developed in this master thesis and the baseline methods were made to understand the key characteristics that explain the differences in performances on the different methods. Table 4.1 presents the key difference between the various methods.

Comparision between models						
#	Model	Category	Observation Endpoint	Observation Length (s)	Prediction Length (s)	Simultaneous Prediction
1	FUSSI [4]	Pedestrian Detection and Tracking	before event	0.5	0.5	individual
2	Static [104]	Action Prediction	complete sequence	0.5	next frame	individual
3	SFRNN [104]	Action Prediction	before event	0.5	2	individual
4	C3D [104]	Action Prediction	complete sequence	0.5	next frame	individual
5	PCPA [104]	Action Prediction	complete sequence	0.5	next frame	individual
6	Novel Architecture	Scene Graph Parsing and Visual Reasoning	before event	0.5	0.5/1/1.5	multiple

Table 4.2: Comparison of characteristics between models

As it can be observed from the table 4.1, one of the critical differences between the baseline methods and the novel method is the observation endpoint. Unlike most other baseline models, which samples the complete crossing sequence data to make a prediction, the novel architecture only considers the frames before the event to make a prediction. Furthermore, it can also be observed that none of the baseline methods is equipped to handle multiple pedestrians simultaneously, which the novel architecture can handle. In addition, it can also be observed that most baseline models are formulated as action prediction problems. At the same

time, another striking difference can be observed between the input data. Most of the models sample crossing sequence data, whereas the novel architecture and the FUSSI [4] are the only two models that sample frame-by-frame data. This finding aligns with the decisive characteristic considered while designing the novel model: real-time processing of input data. Since the model does not need to wait for a certain number of samples to predict pedestrian intent, the novel method would be much faster than the baseline methods. However, since the novel method is based on frame-to-frame inputs, it is more prone to failures. These trade-offs are common while implementing a real-life application and must be resolved before implementing the solution.

4.3 Discussions

Since the previous sections analyse the novel intent prediction model's performance compared to the baseline models, this section aims to reflect on the proposed model, the ethical aspects associated with this problem, and the contribution of this master thesis.

4.3.1 Model performance

The first variant of the proposed intent prediction model achieves an accuracy of 49% while predicting pedestrian intent 0.5 seconds in advance, 46% while predicting pedestrian intent 1 second in advance, and 45% while predicting pedestrian intent 1.5 seconds in advance. Similarly, the second variant of the proposed intent prediction model achieves an accuracy of 51% while predicting pedestrian intent 0.5 seconds in advance, 35% while predicting pedestrian intent 1 second in advance, and 33% while predicting pedestrian intent 1.5 seconds in advance. One of the significant reasons for the lower accuracy scores is caused due to real-time processing of data. The difference in processing speeds can be observed explicitly in table 4.3 where it can be observed that the novel architecture is much faster than the FUSSI network [4].

Model performance					
#	Model	Accuracy	AUC	F1	Processing times
1	FUSSI [4]	0.45	**0.89**	**0.83**	1 fps
2	Static [104]	0.46	0.45	0.54	31 ped/sec
3	SFRNN [104]	0.51	0.5	0.74	39 ped/sec
4	C3D [104]	0.61	0.51	0.75	6.7 ped/sec
5	PCPA [104]	**0.58**	0.5	0.71	6.4 ped/sec
6	Novel Architecture	0.51	-	0.39	6 fps

Table 4.3: Comparison between model performances at 15 frames/0.5 seconds

There can be several reasons for this difference in processing speeds: The FUSSI network [4] processes image data over the complete prediction pipeline, whereas

the novel architecture converts the image data into graphs before predicting the pedestrian intent. Furthermore, the FUSSI network [4] requires pose estimation for at least 15 frames (corresponding to 0.5 seconds) before predicting pedestrian intent. On the other hand, the novel architecture uses the scene dynamics coded as graphs to predict pedestrian intent. Additionally, it can also be observed that the other models predict pedestrian intent based on sequence data rather than frame-by-frame data. Therefore, the processing speeds are measured in pedestrians per second rather than frames per second.

4.3.2 Difficulties

One of the most significant limiting factors of this thesis work was the input dataset. Adopting the JAAD dataset [63] for this master thesis was motivated more by the hardware resources available to process the dataset rather than the dataset's contents. This choice proved to be severely restricting while training the model. The JAAD dataset did have a high number of attributes. However, the number of pedestrians containing all the attributes was few, and the dataset was severely biased towards crossing pedestrians. On analysing the results further, it was found that the model was biased despite reweighing the sample classes, causing the model accuracy to be moderate.

Another challenge that forced us to change our approach was the lack of ground truth data for pose estimation. The pedestrian intent prediction problem has been a topic of active research, and most approaches have utilised a combination of intrinsic pedestrian features such as pose and another feature choice to tackle this problem. To achieve this, most of the methods have hand-annotated the ground truth for pose estimation, but none of these ground truths was publically available. Therefore, not applying the pose estimation data also restrained the accuracy of the model.

Lastly, the problem that complicated the thesis was making the network end-to-end with total online processing of inputs. A genuinely end-to-end network that processes the inputs online would process it according to time. Therefore, we were surprised that most available models do not predict intent based on incomplete inputs as they appear in real-time but predict using complete crossing sequence after saving complete sequences. Due to this, every publically available dataset was designed on complete sequences rather than frame-by-frame values. Since the motivation of our model was to bridge the gap to deployment, the proposed neural network processes the inputs frame-by-frame, as they would appear in real-time. The thesis had to redesign the input dataset as none of the publicly available models provided frame-by-frame information. Since this redesigned dataset was adopted for training the network, the redesigned dataset could have contributed to the lower accuracy scores. One of the most important reasons for this could be the simultaneous prediction for multiple pedestrians. Unlike datasets based on complete crossing sequences, a frame-by-frame dataset provides all the information in the frame, which means that predictions for multiple pedestrian crossing simultaneously are

expected. Ideally, models handing frame-by-frame information must be designed to handle dynamic inputs. However, this thesis works around that problem by setting a maximum number of pedestrians that the model can predict at any time instance. Another reason for the lower accuracy scores could be the design of the proposed model. Since the complete end-to-end architecture is used to test the videos, the scene parser algorithm's performance could also negatively influence the accuracy scores of the overall network. Another reason for the lower accuracy scores could be the prediction method itself. Since the novel architecture is designed end-to-end, the model starts predicting pedestrian intent even without processing the minimum number of graph frames required to predict pedestrian intent accurately, resulting in a higher number of incorrect results at the beginning, which could also decrease the overall accuracy scores. Having a complicated design proved highly challenging for this thesis.

4.3.3 Ethical Aspects

One of the potential ethical issues that could arise from implementing the pedestrian intent prediction model on real-world data could be a breach of privacy for individuals. Since people are unaware of the video data being collected while crossing the road, a potential breach of privacy for individuals who do not want their information revealed could be an issue. Since deep learning models that could potentially save lives, like the pedestrian intent prediction model, would be required to be constantly improved even after deployment to cover all types of edge cases, accumulating real-world data to improve the model, a common practice within the deep learning community, could further exacerbate the issue.

Since deep learning models like pedestrian intent prediction models contributes to the UN Sustainability Goal number eleven, which ensures that cities and human settlements are inclusive, safe, resilient, and sustainable, issues like privacy possess a much bigger question regarding the approach towards machine intelligence. Should humanity opt for a more precautionary approach where no new technology should be adopted until proved safe, or should humanity opt for a more utilitarian approach, where one can argue that the safety of a larger group of people outweighs the privacy concerns for a smaller group of people. Ethical questions like the one mentioned above would be needed to be answered first before deploying any deep learning models like the pedestrian intent prediction model in the real world.

In this master thesis, since the pedestrian intent prediction models are based on the publically available dataset, such ethical issues do not exist as the data is collected considering these ethical issues.

4.3.4 Contribution

Although pedestrian intent prediction has been a topic of active results, resulting in many new algorithmic solutions and benchmarking datasets, real-time processing of pedestrian intent prediction based on scene graph parsing and visual reasoning

does not appear to be done. Most of the existing methods seem to process the data offline, which results in better outcomes, but cannot be for practical use cases. The most famous pedestrian intent prediction approach comprises a detector-tracker model, where an object detector and tracking algorithm followed by a classifier is used to detect pedestrian intent. Other approaches include trajectory prediction and action prediction, where the former is typically suited for a top-down static view of the scene, whereas the latter is generic to handle all use cases. However, newer approaches like action prediction and visual reasoning algorithms have yielded better results; the contribution of a thesis like ours advances the understanding of modern deep learning algorithms.

5

Conclusion and Future Work

This chapter presents the conclusion for the master thesis and examines the future work. The organisation of this chapter is as follows: section 5.1 presents the conclusion for the master thesis, and section 5.2 examines the challenges faced during the master thesis and the corresponding future work.

5.1 Conclusion

This thesis explores the various factors that can help predict pedestrian intent. A pedestrian intent prediction model is developed using video data with pedestrian behaviour, appearance, and attribute as inputs by modelling pedestrian intent on context, interactions and scene dynamics rather than implicit pedestrian features. The model is achieved through an object detector-tracker algorithm like YOLOv4-Deepsort, graph convolutional networks and graph convolutional recurrent networks. The first variant of the proposed intent prediction model achieves an accuracy of 49% while predicting pedestrian intent 0.5 seconds in advance, 46% while predicting pedestrian intent 1 second in advance, and 45% while predicting pedestrian intent 1.5 seconds in advance. Similarly, the second variant of the proposed intent prediction model achieves an accuracy of 51% while predicting pedestrian intent 0.5 seconds in advance, 35% while predicting pedestrian intent 1 second in advance, and 33% while predicting pedestrian intent 1.5 seconds in advance.

The lower accuracy is caused due to a combination of multiple reasons such as real-time processing of data, simultaneous predictions for multiple pedestrians, the performance of the scene parser algorithm, and premature prediction of pedestrian intent due to the inherent design of the network. Since pedestrian intent prediction has been a topic of active research, this master thesis aims to advance the overall progress in solving the intent prediction problem by developing an algorithm focusing on real-time data processing. The spatial relationship between various objects in the scene carries a solid connection to pedestrian intent, indicating that environment and intrinsic scene dynamics significantly affect pedestrian intent. Alternatively, it could be due to the dataset being biased. Nevertheless, further algorithmic solutions and more complicated benchmarking datasets must be developed to simulate real-life use cases and solve the intent prediction problem.

5.2 Future Work

Since the JAAD dataset had a limited number of pedestrians containing all the features, the model proved biased towards samples. Therefore, to improve the model's generality, the model can be trained on more complex publically available datasets like PIE [41]. In addition to the dataset, intrinsic features such as pose can be included using OpenPose. Although using an off-the-shelf algorithm to calculate pose would not be as accurate as having pose annotations, it could be attempted in the future to improve intent prediction.

Additionally, this thesis attempts to implement an end-to-end network that reduces the gap between prototyping and production. Although the thesis could not implement a perfect end-to-end network with total online processing of inputs, the same could be attempted in the future to bridge the gap to deployment.

Another approach that could be attempted in the future is the development of a generic algorithm to restructure existing datasets to provide frame-by-frame information. Since the algorithm developed to restructure the JAAD dataset was tailor-made for our use case, a generic algorithm can be developed to redesign all future datasets. Lastly, since a frame-by-frame dataset provides all the information in the frame, handling of dynamic inputs to simultaneously predict multiple pedestrians in a frame can be attempted in the future.

In conclusion, although this thesis attempts to bring real-time intent prediction algorithms closer to deployment, much work needs to be done to solve this problem in real-world environments.

Bibliography

[1] S. Ahmed, M. N. Huda, S. Rajbhandari, C. Saha, M. Elshaw, and S. Kanarachos, "Pedestrian and cyclist detection and intent estimation for autonomous vehicles: A survey," *Applied Sciences*, vol. 9, 06 2019.

[2] J. Breene, M. Khayesi, R. McInerney, A. Sukhai, T. Toroyan, D. Ward, and K. Iaych, "Global status report on road safety 2018," *Geneva: World Health Organization*, Jun 2018.

[3] Z. Fang and A. M. López, "Intention recognition of pedestrians and cyclists by 2d pose estimation," *CoRR*, vol. abs/1910.03858, 2019.

[4] F. Piccoli, R. Balakrishnan, M. J. Perez, M. Sachdeo, C. Nuñez, M. Tang, K. Andreasson, K. Bjurek, R. D. Raj, E. Davidsson, C. Eriksson, V. Hagman, J. Sjöberg, Y. Li, L. S. Muppirisetty, and S. Roychowdhury, "Fussi-net: Fusion of spatio-temporal skeletons for intention prediction network," *CoRR*, vol. abs/2005.07796, 2020.

[5] Z. Fang, D. Vázquez, and A. López, "On-board detection of pedestrian intentions," *Sensors*, vol. 17, p. 2193, Sep 2017.

[6] B. Liu, E. Adeli, Z. Cao, K. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *CoRR*, vol. abs/2002.08945, 2020.

[7] D. Geronimo and A. M. Lopez, *Vision-Based Pedestrian Protection Systems for Intelligent Vehicles*. Springer Publishing Company, Incorporated, 2013.

[8] C. Zhou and J. Yuan, "Bi-box regression for pedestrian detection and occlusion estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[9] G. Li, J. Li, S. Zhang, and J. Yang, "Learning hierarchical graph for occluded pedestrian detection," in *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, (New York, NY, USA), p. 1597–1605, Association for Computing Machinery, 2020.

[10] W. Liu, S. Liao, W. Hu, X. Liang, and X. Chen, "Learning efficient single-stage pedestrian detectors by asymptotic localization fitting," in *ECCV*, 2018.

[11] H. Xie, W. Zheng, and H. Shin, "Occluded pedestrian detection techniques by deformable attention-guided network (dagn)," *Applied Sciences*, vol. 11, no. 13, 2021.

[12] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," *CoRR*, vol. abs/1711.07752, 2017.

[13] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," *CoRR*, vol. abs/1608.05404, 2016.

[14] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," *CoRR*, vol. abs/1803.10859, 2018.

[15] U. Iqbal, A. Milan, and J. Gall, "Pose-track: Joint multi-person pose estimation and tracking," *CoRR*, vol. abs/1611.07727, 2016.

[16] D. Varytimidis, F. Alonso-Fernandez, B. Durán, and C. Englund, "Action and intention recognition of pedestrians in urban traffic," *CoRR*, vol. abs/1810.09805, 2018.

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

[18] S. Koehler, M. Goldhammer, S. Bauer, S. Zecha, K. Doll, U. Brunsmann, and K. Dietmayer, "Stationary detection of the pedestrians intention at intersections," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 4, pp. 87–99, 2013.

[19] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian prediction by planning using deep neural networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5903–5908, 2018.

[20] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila, "A new benchmark for vision-based cyclist detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1028–1033, 2016.

[21] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 618–633, Springer International Publishing, 2014.

[22] G. Habibi, N. Jaipuria, and J. P. How, "Context-aware pedestrian motion prediction in urban intersections," 2018.

[23] K. Saleh, M. Hossny, and S. Nahavandi, "Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2018.

[24] Y. Xu, Z. Piao, and S. Gao, "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] R. Quintero, I. Parra, D. F. Llorca, and M. Sotelo, "Pedestrian path prediction based on body language and action classification," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 679–684, 2014.

[26] Y. Hashimoto, G. Yanlei, L.-T. Hsu, and K. Shunsuke, "A probabilistic model for the estimation of pedestrian crossing behavior at signalized intersections," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 1520–1526, 2015.

[27] N. Jaipuria, G. Habibi, and J. P. How, "A transferable pedestrian motion prediction model for intersections with different geometries," *CoRR*, vol. abs/1806.09444, 2018.

[28] Y. F. Chen, M. Liu, and J. P. How, "Augmented dictionary learning for motion prediction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2527–2534, 2016.

[29] C. Park, J. Ondřej, M. Gilbert, K. Freeman, and C. O'Sullivan, "Hi robot: Human intention-aware robot planning for safe and efficient navigation in crowds," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3320–3326, 2016.

[30] J. Bütepage, H. Kjellström, and D. Kragic, "Anticipating many futures: Online human motion prediction and synthesis for human-robot collaboration," *CoRR*, vol. abs/1702.08212, 2017.

[31] R. Luo and L. Mai, "Human intention inference and on-line human hand motion prediction for human-robot collaboration," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5958–5964, 2019.

[32] S. Zhou, M. J. Phielipp, J. A. Sefair, S. I. Walker, and H. B. Amor, "Clone swarms: Learning to predict and control multi-robot systems by imitation," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[33] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," *CoRR*, vol. abs/1902.03748, 2019.

[34] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. D. Reid, S. H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *CoRR*, vol. abs/1907.03395, 2019.

[35] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Predicting the future: A jointly learnt model for action anticipation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[36] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 759–776, Springer International Publishing, 2020.

[37] C. Lu, M. Hirsch, and B. Scholkopf, "Flexible spatio-temporal networks for video prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[38] M. Qi, J. Qin, Y. Wu, and Y. Yang, "Imitative non-autoregressive modeling for trajectory forecasting and imputation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[39] J. Chen, W. Bao, and Y. Kong, "Group activity prediction with sequential relational anticipation model," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 581–597, Springer International Publishing, 2020.

[40] A. Rasouli, "Deep learning for vision-based prediction: A survey," *CoRR*, vol. abs/2007.00095, 2020.

[41] A. Rasouli, I. Kotseruba, T. Kunic, and J. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6261–6270, 2019.

[42] S. Malla, B. Dariush, and C. Choi, "TITAN: future forecast using action priors," *CoRR*, vol. abs/2003.13886, 2020.

[43] Z. Zhang, J. Gao, J. Mao, Y. Liu, D. Anguelov, and C. Li, "Stinet: Spatio-temporal-interactive network for pedestrian detection and trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[44] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[45] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident db," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[46] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident db," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[47] H. Zhao and R. P. Wildes, "On diverse asynchronous activity anticipation," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX* (A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, eds.), vol. 12374 of *Lecture Notes in Computer Science*, pp. 781–799, Springer, 2020.

[48] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Proceedings of The 2nd Conference on Robot Learning* (A. Billard, A. Dragan, J. Peters, and J. Morimoto, eds.), vol. 87 of *Proceedings of Machine Learning Research*, pp. 947–956, PMLR, 29–31 Oct 2018.

[49] P. Gujjar and R. Vaughan, "Classifying pedestrian actions in advance using predicted video of urban driving scenes," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2097–2103, 2019.

[50] Q. Ke, M. Fritz, and B. Schiele, "Time-conditioned action anticipation in one shot," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[51] E. Alati, L. Mauro, V. Ntouskos, and F. Pirri, "Help by predicting what to do," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 1930–1934, 2019.

[52] Y. A. Farha, A. Richard, and J. Gall, "When will you do what? - anticipating temporal occurrences of activities," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5343–5352, 2018.

[53] L. Chen, J. Lu, Z. Song, and J. Zhou, "Part-activated deep reinforcement learning for action prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[54] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[55] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[56] W. Cong, W. Y. Wang, and W.-C. Lee, "Scene graph generation via conditional random fields," *ArXiv*, vol. abs/1811.08075, 2018.

[57] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[58] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[59] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 413–431, Springer International Publishing, 2018.

[60] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1219–1228, 2018.

[61] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 2688–2697, PMLR, 10–15 Jul 2018.

[62] D. Teney, L. Liu, and A. van den Hengel, "Graph-structured representations for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[63] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 206–213, 2017.

[64] J. Veisdal, "The birthplace of ai," *Medium*, Jun 2021. [Online; accessed 30-October-2021].

[65] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 01 2016.

[66] C. Doyle, "Ibm watson assistant - virtual agent," *IBM Watson Assistant Blog*, 2021. [Online; accessed 30-October-2021].

[67] E. Sadun and S. Sande, *Talking to Siri: Mastering the Language of Apple's Intelligent Assistant*. Que Publishing Company, 3rd ed., 2014.

[68] Z. Paul, "Cortana-intelligent personal digital assistant: A review," *International Journal of Advanced Research in Computer Science*, vol. 8, pp. 55–57, 08 2017.

[69] R. Patel, "Google lens: Real-time answers to questions about the world around you," *Google*, May 2018.

[70] M. Dikmen and C. Burns, "Trust in autonomous vehicles: The case of tesla autopilot and summon," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1093–1098, 2017.

[71] J. McCarthy and E. A. Feigenbaum, "In memoriam: Arthur samuel: Pioneer in machine learning," *AI Magazine*, vol. 11, p. 10, Sep. 1990.

[72] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[73] I. Rish, "An empirical study of the naïve bayes classifier," *IJCAI 2001 Work Empir Methods Artif Intell*, vol. 3, 01 2001.

[74] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the fifth annual workshop on Computational learning theory - COLT 92*, 1992.

[75] P. Yadav, "Decision tree in machine learning," Sep 2019. [Online; accessed 30-October-2021].

[76] B. W. Silverman and M. C. Jones, "E. fix and j.l. hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951)," *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, p. 233, 1989.

[77] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (L. M. L. Cam and J. Neyman, eds.), vol. 1, pp. 281–297, University of California Press, 1967.

[78] P. Luboobi, "Foundations of machine learning : Singular value decomposition (svd)," Feb 2018.

[79] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.

[80] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. The MIT Press, 2018.

[81] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, 2013. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.

[82] A. Sehgal, H. M. La, S. Louis, and H. Nguyen, "Deep reinforcement learning using genetic algorithm for parameter optimization," *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pp. 596–601, 2019.

[83] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, p. I–387–I–395, JMLR.org, 2014.

[84] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1928–1937, PMLR, 20–22 Jun 2016.

[85] I. N. Aizenberg, N. N. Aizenberg, and J. P. Vandewalle, *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. USA: Kluwer Academic Publishers, 2000.

[86] A. Acharya, S. Adhikari, S. Agarwal, V. Auvray, N. Belgamwar, A. Biswas, S. Chandra, T. Chung, M. Fazel-Zarandi, R. Gabriel, S. Gao, R. Goel, D. Hakkani-Tür, J. Jezabek, A. Jha, J. Kao, P. Krishnan, P. Ku, A. Goyal, C. Lin, Q. Liu, A. Mandal, A. Metallinou, V. I. Naik, Y. Pan, S. Paul, V. Perera, A. Sethi, M. Shen, N. Strom, and E. Wang, "Alexa conversations: An extensible data-driven approach for building task-oriented dialogue systems," *CoRR*, vol. abs/2104.09088, 2021.

[87] K. Gurney, *An Introduction to Neural Networks*. USA: Taylor & Francis, Inc., 1997.

[88] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[89] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969.

[90] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2, pp. 303–314, dec 1989.

[91] C. Thomas, "An introduction to convolutional neural networks," *Medium*, May 2019.

[92] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A gentle introduction to graph neural networks," *Distill*, 2021. https://distill.pub/2021/gnn-intro.

[93] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *CoRR*, vol. abs/1812.08434, 2018.

[94] J. Huber, "Batch normalization in 3 levels of understanding," Nov 2020. [Online; accessed 30-October-2021].

[95] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[96] D. Powers and Ailab, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation," *J. Mach. Learn. Technol*, vol. 2, pp. 2229–3981, 01 2011.

[97] O. Barak, "Is f1 the appropriate criterion to use?," *Medium*, May 2021. [Online; accessed 30-October-2021].

[98] I. contributors, "Pugh matrix – isixsigma," 2017. [Online; accessed 30-October-2021].

[99] R. Baxter, "Concept selection with a pugh matrix," May 2015. [Online; accessed 30-October-2021].

[100] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasude-

van, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[101] A. Systems, "Tensorflow vs keras vs pytorch: Which framework is the best?," Jun 2020. [Online; accessed 30-October-2021].

[102] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[103] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[104] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Benchmark for Evaluating Pedestrian Action Prediction," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1258–1268, 2021.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY