

ABSTRACT

The critical conditions for a secure assistive and independent driving system is the accurate perception of the pride vehicle's pedestrian. While there have been significant strides in detecting and tracking visible enclosing of the pride vehicle and accurate prediction of vulnerable road druggies similar as climbers and cyclists remains a challenge as vulnerable road druggies can incontinent change their direction and speed. people make important intuitive opinions grounded on the relations in the scene and the sequences of conduct to interpret the intent of vulnerable road druggies. However, the same can't be assumed for the current assistive and independent driving systems, as these intentions are realized through subtle gestures and relations. Since prognosticating the unborn intent of vulnerable road druggies is essential to advise the motorist or automatically perform smoother manoeuvre, our paper aims to prognosticate rambler intent using deep machine learning. Many times, the intent prediction problem has been a content of active exploration, performing in several new algorithmic results. Overall, measuring the overall progress towards working this problem has been delicate. Thus, this paper investigates the performance of multiple baseline styles on the common attention in independent driving(JAAD) datasets to attack this handicap. Despite achieving state of the art results on curated datasets, almost of these styles are developed, disregarding Implicit deployment in product surroundings. Our paper proposes an end to end network that attempts to reduce the gap between prototyping and product grounded on this findings. The proposed end to end network predicts the unborn intent of vulnerable road druggies up to half a alternate in future.

KEY WORDS: Deep Machine Learning, Computer Vision, Prediction, Autonomous Driving, ADAS, AD, JAAD, PIE

TABLE OF CONTENTS

Chapter 1 Introduction.....	1
1.1 Background.....	1
1.2 objective.....	3
1.3 Related work.....	5
1.4 Thesis Contribution.....	9
1.5 Individual contribution.....	9
1.6 Limitation.....	11
1.7 Thesis outline.....	11
Chapter 2 Theoretical Background.....	13
2.1 Artificial Intelligence,Machine learning,and Deep Learning	13
2.2 Artificial Neural Networks.....	21
2.3 Evaluation of model.....	33
Chapter 3 Methods.....	35
3.1 Data Extraction and proceesing.....	35
3.2 Machine learning framework.....	39
3.3 Novel architecture	41
3.4 Experimentation and benchmarking.....	45
Chapter 4 Result and Discussion	47
4.1 Novel intent prediction model.....	47
4.2 Compare with baseline methods	51
4.3 Discussions.....	53
Chapter 5 Conclusion and Future Work	57
5.1 Conclusion.....	57
5.2 Future work.....	58
Acknowledgements	59
References	60

TABLE OF FIGURES

2.1	Difference among AI, ML and DL.....	14
2.2	Difference among ANI, AGI and ASI.....	17
2.3	Types of ML Algorithms.....	19
2.4	Difference among machine learning and deep learning.....	21
2.5	Comparison among biological neurons and ANNs.....	23
2.6	Example of a CNN structure applied on visual data. The model analyzes.....the layers and reduces those values down the line to a classification layer that make sanestimation.....	25
2.7	An example of how a simple RNN can be structured. The estimation of the previous input is included in estimation of current inputmodel..	27
2.8	An example of a simple GNN.....	29
2.9	A simple neural network to explain forward propagation.....	31
2.10	Confusion matrix.....	34
3.1	Novel architecture proposed in this papers.....	42
3.2	Visual representation of data transformation process.....	43
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 among the nodes.....	44
3.4	Overall prediction architecture.....	45
4.1	Examples of intent prediction for crossing and not crossing pedestrians.....	47
4.1.1	Object motion tracking.....	49
4.1.2	Object tracking & trajectory.....	49
4.2	Effect of time to event on accuracy of the novel intent predictionmodel.....	50

List of Tables

3.1 Decision matrix to select the appropriate dataset.....	38
3.2 Categorisation of features available for pedestrians in the JAAD dataset.....	38
3.3 Decision Matrix to select the appropriate machine learning framework.....	41
3.4 Objects of interest for scene parser.....	42
3.5 Experimental setup from training models.....	45
4.1 Benchmarking results of the novel intent prediction model developed.....	51
4.2 Comparison of characteristics among models.....	52
4.3 Comparison among model performances at 15 frames/0.5 seconds.....	53

Chapter 1

Introduction

1.1 Background

According to the WHO, 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. 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. 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.

There have been significant strides in detecting and tracking visible surroundings to the vehicle, accurate discovery and prediction of vulnerable road users' intentions remain challenging. Among these vulnerable road users, accurate discovery and prediction of pedestrian intentions become a challenging sub problem as pedestrians do not explicitly indicate their intentions, allowing for higher fatality danger. The pedestrian intention prediction problem becomes a subset of the much more extensive intent prediction problem where the target is to automatically estimate a pedestrian's relative position & intention. This information is critical for reducing the chance of injuries requiring hospitalization. Experiments show that initiating an emergency brake with 160 ms of anticipation over a 660 ms time to collision can less than

presumably of accident taking hospitalization from 50 to 35(percentage) . Thus, information about pedestrian intent is also valuable in lowering collision avoidance cautions & systems false positive rates, working in a safe and smooth manoeuvre while driving.

We as humans make important intuitive opinions grounded on the sequences of conduct and relations 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 realized 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.

With Deep Learning, advanced algorithms that read pedestrian instincts and make good sense are being raised. Various methods ranging from trajectory prediction to behavioral analysis are being explored. At a same time, different input modalities from images to point cloud data are also being examined. In paper, we proved the knowledge of using monocular RGB images as core information to fete the intentions of vulnerable road druggies.

1.2 Objective

In the earlier chapter, with safety being one of the biggest concerns with autonomous vehicles, it is crucial to predict pedestrians intent accurately. Methodically to predict pedestrians intent accurately, precise detection and tracking of pride vehicle's visible are unfavorable. this regard, different sensing styles like cameras, radars, and lizards have been used to detect and track visible surroundings of the pride 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. 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, lizard 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 lizard 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. Previous years, the pedestrian intent prediction problem has been a content of active research, solving in many new algorithmic solutions, which are showing in part 1.3, However, few number of high quality datasets can be used to benchmark multiple state-of-the-art methods. Overall progress is crucial to solving the intent prediction problem, developing a prediction method grounded on publicly accessible high quality datasets with standard training and evaluation procedures seems reasonable.

Thus, this paper aims to style a pedestrian intent prediction method grounded on 2D images captured from a high resolution monocular camera by finding methods that use publicly accessible best quality datasets, employ standard training and evaluation procedures, and consistently provide smooth 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 content of good research, various approaches have been taken to attack 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 paper.

1.3.1 Pedestrian detection & tracking

Pedestrian detection and tracking is 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 grounded on shallow learning was provided in . However, the accuracy of methods mentioned in drops while detecting pedestrians in a crowd due to occlusion. Recently, various deep learning methods such as provide significantly higher delicacy while detecting pedestrians in a crowd. For pedestrian tracking, multi person tracking methods to track every person in a crowded scene was employed. Lately, techniques like people re-identification and pose estimation are being used to solve tracking problems.

In several works have been directed towards designing pedestrian intent prediction with many object detection and tracking algorithms. 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, predicts the intent of the pedestrian by combining CNN's and Holst's. Lastly, predicts pedestrian intent based on intrinsic pedestrian feature poses by fitting point skeleton to each detected pedestrian and classifying using a support vector machine or a convolution neural network. Although these features impart to predicting pedestrian intent, pedestrian detection and tracking grounded approaches ignore context and interactions with objects in the scenes, such as others pedestrians, vehicles, traffic signs, lights, and other environmental factors. Our paper argues that such relationships can be revealed over time. Thus, our paper 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 related approach to detect pedestrian intent. This approach predicts pedestrian intent grounded on the assumption that accurate prediction of future trajectory indicates the pedestrian 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 use past trajectories to predict the future trajectories. uses inverse reinforcement learning to predict future trajectories. models social dynamics and crowd interaction to predict future trajectories. Furthermore, some methodicalness human dynamics in different forms to predict trajectories. proposes Gaussian Process Dynamical Models based on the action, speed, location, and heading direction as input to predict future directions and intent. 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 situations and can't be predicted long enough into the future with enough certainty. Thus, 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 this paper.

1.3.3 Action prediction

Action prediction can be considered one of the most best approaches for intent prediction. This predict pedestrian intent by modelling the causal connection between the previous current & potential future information same to approaches used in design granting algorithms. Hereafter the action prediction

approach tries to anticipate 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 future. The prediction is an important problem in many domains such as assistive robotics, surveillance sports forecasting and autonomous driving systems. Action prediction can be either indirect in the form of hereafter trajectories or indirect in terms of predicting future events.

Prevailing strategies for action prediction use sequential temporal tools to model the causal relationship. Some commonly employed architectures include recurrent networks 3D convolution networks, or a combination of both. Among methods based on recurrent networks, high level semantics are often processed with off the shelf algorithms, and data driven methodologies are employed to learn parameters. Among methods based on convolution networks, researchers often resorted to deep Convnet features and learned a classifier from the training data. Most methods mentioned above anticipate the next action by looking at the sequence of previous actions. However, other methods build extemporization graphs or use reinforcement learning 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 this methods obtain remarkable results, paper 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 & visual reasoning

Scene graph parsing & visual reasoning is relatively newly approach to detect pedestrian intent. A scene graph is a framework characterization of a scene that reveals the objects, assigns, and connections between objects in the scene. As computer vision technology continues to develop, researchers are no longer satisfy with simply detecting and recognizing objects in images, instead & researcher look forward to a higher level of understanding and reasoning about visual scenes. Predicting the Extemporization relationship between the various objects in the scene is the principle behind the scene graph parsing & visual reasoning approach for pedestrian intent prediction.

Recently, several works have been directed towards generating scene graphs with global context with relationship proposal networks conditional random fields iterative message-passing or recurrent neural networks. Scene graphs built on visual scenes are used for multiple applications.

the scene graph parsing & visual reasoning approach consider the pedestrian factor and interactions, thesis proposes an approach for the pedestrian and prediction problem grounded on scene graph parsing & visual reasoning.

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

1.4 Thesis contribution

This paper, the author answer the following research questions:

#How can be novel deep machine learning architecture predict pedestrian intent 0.5 seconds, 1 second and 1.5 seconds in the future times when applies on publicly accessible datasets?

#What are the key fact that explain the among 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 bench marking the result against the baseline methods, this section outlines the individual contributions by author. 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 author predominantly restructured the input datasets into the frame-by-frame structure section 3.3.3, implemented the graph-frame translation section 3.3.4, and realized the training algorithm and the initial testing structure section 2.3. also implemented the novel architecture Section 3.3, trained and tuned and shared the workload in the training and tuning of model. since the various publicly accessible 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 result in the training and tuning of model. also organized and trained the base line methods section 3.4 & bench marked all the methods based on delicacy characteristics and processing times section 4.2.

1.6 Limitations

The scope of the paper is limited because of the following reasons:

This paper would use the publicly accessible JAAD datasets for intent prediction. Thus, the scenarios evaluated in paper is shorted to the scenarios available within the datasets.

data accessible in the JAAD datasets is limited to images from high resolution monocular cameras mounted on the pride vehicle. Thus, the input data is limited by the system configuration. Further, other modalities of input data are not considered.

data accessible in the JAAD datasets is short to pedestrians. Here, cyclists aren't considered for evaluating the crossing intent. Hence, the phrase vulnerable road druggies only represent the pedestrians in the scope this paper.

1.7 Thesis outline

This paper has been divided into 5 chapters: chapter 1, presents an overview of the pedestrian intent prediction difficulties and describes an introduction to the crucial developments in this domain. chapter 2, introduces various essential terms and theoretical concepts that are applicable to this paper, chapter 3, describes the methodology used to predict pedestrian intent, chapter 4, An describes the results obtained from the implementation, discussion the project outcome based on the research questions & examines the significance of this paper, and chapter 5, presents the conclusion for the paper and examines the future work.

Chapter 2

Theoretical Background

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 AI is becoming the driving technology of the 21st century. However, artificial intelligence has been the core enable 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. According to McCarthy, AI is the science and engineering of making intelligent machines. 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.

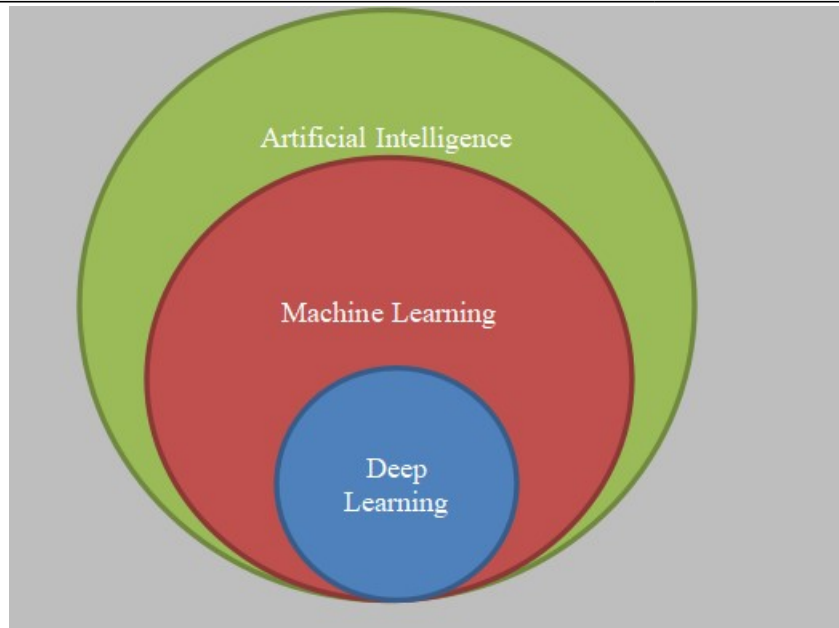


Figure 2.1: Difference among AI, ML and DL

intelligence. Based on the above characteristic, all AI-based systems can be categorized 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 understood to date. ANI is goal-oriented and trained to perform a singular task for e.x, 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. If any of them can outperform humans in these particular tasks. Some examples of narrow AI are IBM Watson, virtual Assistants like Siri by Apple, Cortana by Microsoft, and others, image recognition software like Google Lens, and Tesla's Autopilot system.

***Artificial General Intelligence**

Artificial General Intelligence (AGI) refers to highly autonomous systems that possess the cognitive capabilities to understand, learn, and perform any intellectual task that a human being can do. AGI represents a level of artificial intelligence that surpasses narrow AI systems, which are designed to perform specific tasks.

As of my knowledge cutoff in September 2021, true AGI does not yet exist. While there have been significant advancements in AI research and technology, achieving AGI remains an ongoing challenge. AGI would require a system capable of generalizing knowledge across domains, adapting to new situations, exhibiting common sense reasoning, and possessing a high level of self-awareness. Researchers and organizations are actively working towards the development of AGI, but its creation remains a complex and uncertain endeavor. It is a topic of great interest and speculation in the field of artificial intelligence.

***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 robots overthrow and enslave humanity. Since ASI would be better than everything humans do, super intelligent systems decision-making and problem-solving capabilities would far surpass humans'. Fortunately, AI researchers and scientists do not even dream of creating ASI.

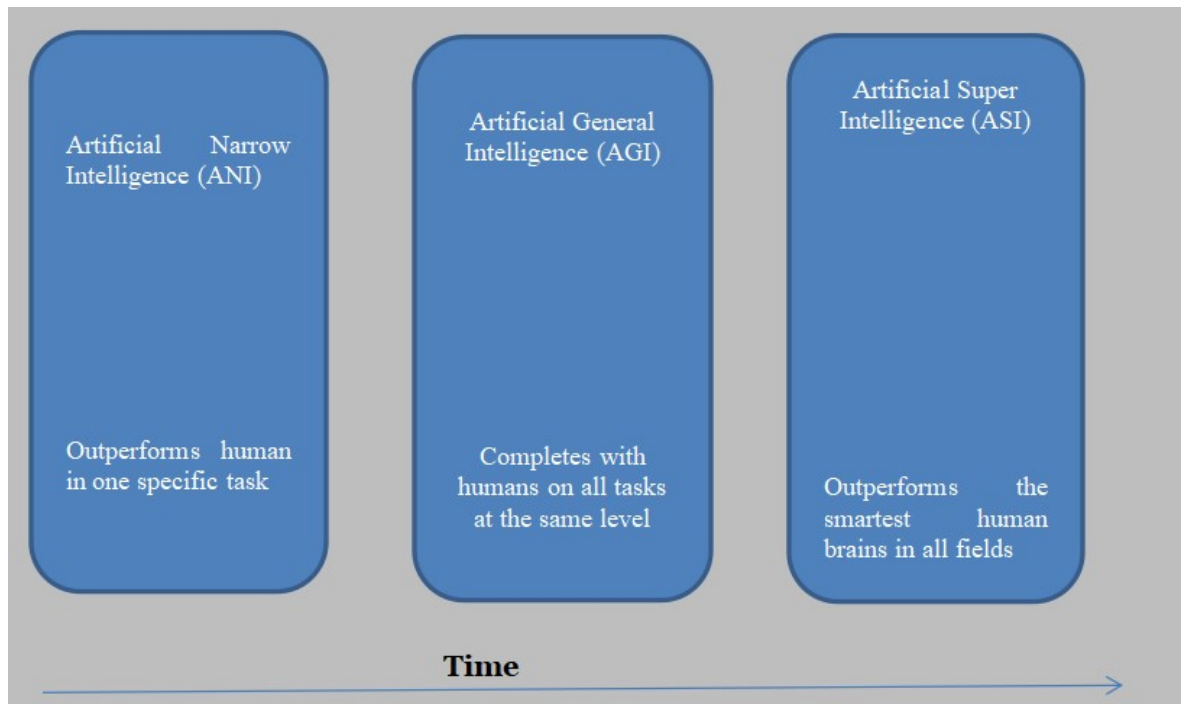


Figure 2.2: Difference among ANI, AGI and ASI

2.1.2 Machine Learning

Arthur Samuel first make up term ML in 1959. Arthur defined ML as space of study that gives computers able to learn without being explicitly programmed.

Thus, the present day definition of machine learning is as follows: Machine learning is a computer program that learns from experience E for few class of tasks T and performance measure P , if it's performance at tasks in T , as measured by P , prosper with experience. An example for the definition mentioned above, to make a machine learning algorithm that predicts the intent of the pedestrian, data with past pedestrian intent patterns must be provided so that the delicacy of the prediction improves over time. Another examples of machine learning algorithms used in our day to day lives are email spam and malware filters, recommendation engines, and online customer support chat-bots.

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

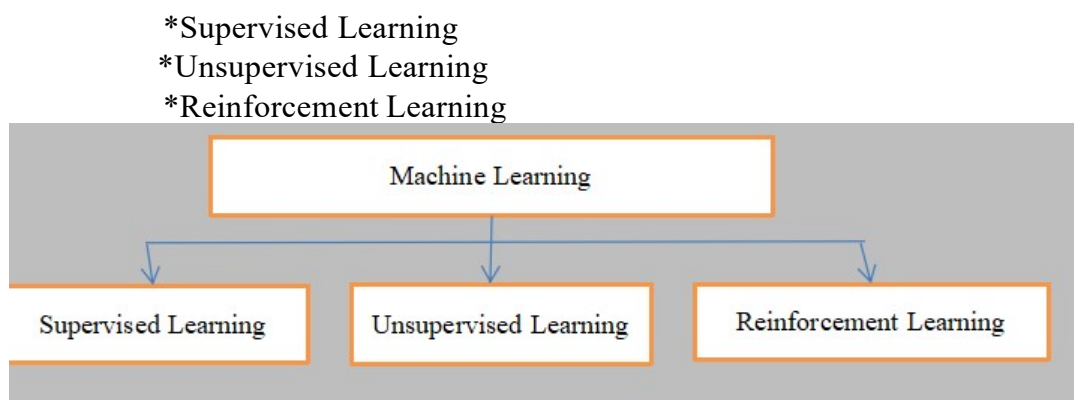


Figure 2.3: Types of ML Algorithms

***Supervised Learning**

Supervised learning, also called task driven learning, is a machine learning algorithm that requires labeled 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 among input and output variables.

In supervised learning algorithms, training data and the correct outputs are 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

estimates its accuracy through a loss function, and the training continues until the model the error has been adequately minimized. 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, Support Vector Machine ,Decision Tree and K-Nearest Neighbour. 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 unlabeled 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. Thus, unsupervised learning algorithms detect hidden patterns in data without need for human interference. Unsupervised learning algorithms are instrumental when humans are unaware of the common properties within a datasets. Thus, unsupervised learning algorithms are used to solve clustering and density estimation problems.

Some examples of unsupervised learning algorithms are K-means clustering algorithm , Singular Value Decomposition algorithm, and Principal Component Analysis algorithm. Some problems generally solved by supervised learning algorithms are recommended 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 datasets, to achieve a goal or set of goals realized through the agent's actions for which it receives feedback about its performance.

Reinforcement learning is similar to supervised learning since the trained agent or model receives feedback signals to optimize 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 optimize their behaviour based on the stimuli, agents interacting in an

environment use feedback loops to maximize the reward. Thus, reinforcement learning algorithms are used to solve path planning and motion control problems. Some examples of reinforcement learning algorithms are Q Learning algorithms, Genetic algorithms, DPG algorithm and A3C algorithm. Some common applications for reinforcement learning algorithms are self driving cars, computer games and resource management problems.

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 neural computer, is “a computing system made up of highly inter connected processing elements, which process information by their energetic state reaction to external inputs.”

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 perceptron) that process the multiple inputs it receives to produce the output. The output from this single node can be represented as the equation 2.1.

$$\mathbf{y} = \mathbf{w}^T \mathbf{x} + \mathbf{b} \quad (2.1)$$

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 datasets the neural network is training on or as an output from the previous node represented as \mathbf{y}_{n-1} . The output from the network is represented as \mathbf{y} which is the sum of weighted inputs represented as $\mathbf{w}^T \mathbf{x}$ and the corrective bias \mathbf{b} . The parameters weight \mathbf{w} and the corrective bias \mathbf{b} are trainable, and the final values determine the performance of the overall network.

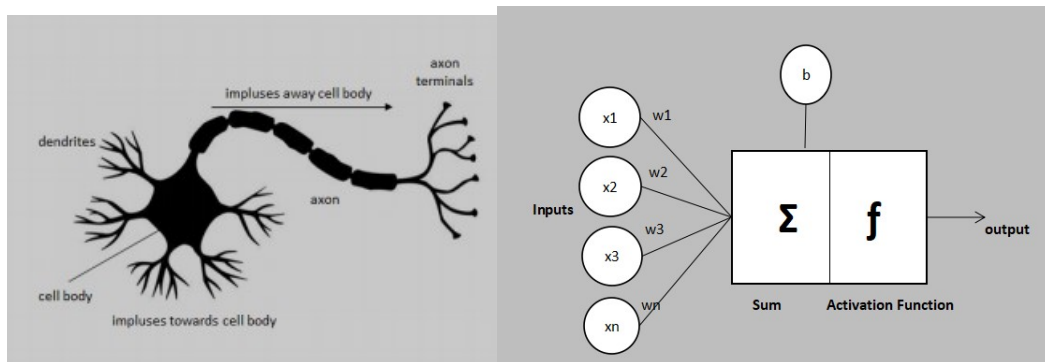


Figure 2.5: Comparison among biological neurons and ANN's

The neural network represented in the figure 2.5 is an example of a feed forward network. Feed forward networks are neural networks in which the connections among the nodes do not form a loop. Feed forward networks are considered one of the simplest types of neural network architectures that are

considered the quintessential deep learning models. As an example, convolution neural networks (further explained in the section 2.2.1) are a specific variety of feed forward neural networks that are used for image data. Since there are no feedback connections in a feed forward network, feed forward 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. Since most of the data available today are highly unorganized 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 , 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.

2.1.3 Convolution neural networks

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

$$\mathbf{y} = \mathbf{x} * \mathbf{w} \quad (2.2)$$

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”. An example of a CNN applied on visual data can be observed in the figure 2.6.

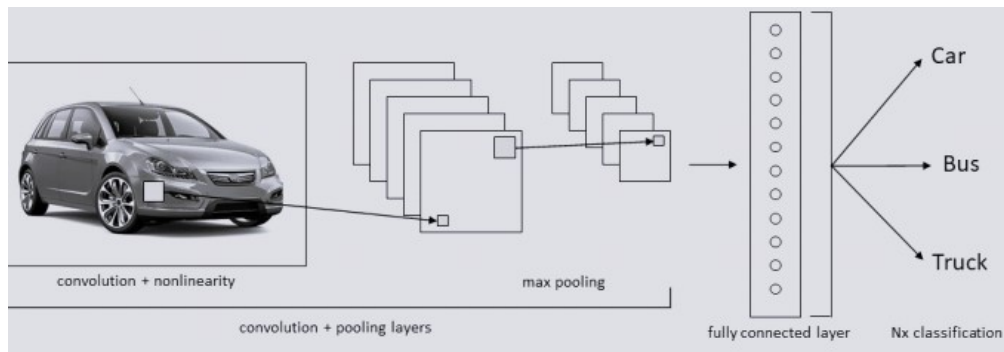


Figure 2.6: An example of CNN structure applied visual data. The model analyzes the image in patches through the layers and reduces those values down the line to a classification layer that makes an estimation.

2.1.4 Recurrent neural networks

Recurrent Neural Networks (RNN) are another class of neural networks that

allow previous output to be used as input while having hidden states. This property of recurrent neural networks allows it to process sequential or time-series data that is typically represented as $x^1, x^2, x^3, \dots, x^T$. Unlike feed forward neural networks such as convolution neural networks, where the inputs and outputs are independent, recurrent neural networks are characterized 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 specializations are needed in recurrent neural networks to process long sequences. An example of a recurrent neural network is represented in the figure 2.7.

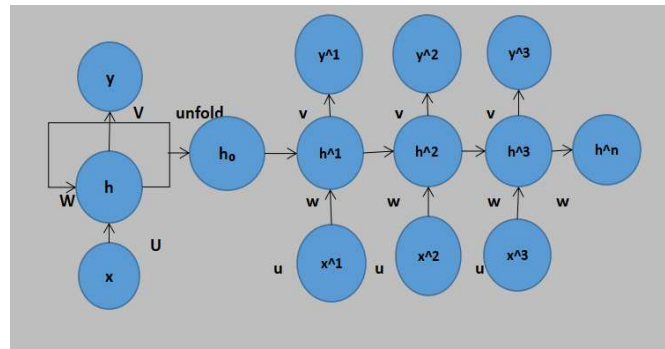


Figure 2.7: An example of how a simple RNN can be structured. The estimation of the previous input is included in the estimation of the current input.

2.1.5 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 among collection of entities. Deep learning models like convolution neural networks typically take rectangular or grid like arrays as input. Thus, graphs are not straightforward to represent in a format that is compatible with deep learning. One of the most 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 optimistically transformation of graph attributes that preserve the graph symmetries. In other words, graph neural networks are 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 embedding without changing the connectivity of

the input graph. In recent years, researchers across various disciplines have significantly increased research on graphneural networks. An important reason for this is that graphs can be used to denotes large number of systems across various areas, including social science, natural science, knowledge graphs and many other research areas. As a unique non- euclidean data structure for machine learning, graph neural networks focus on node classification, link prediction, and clustering tasks.

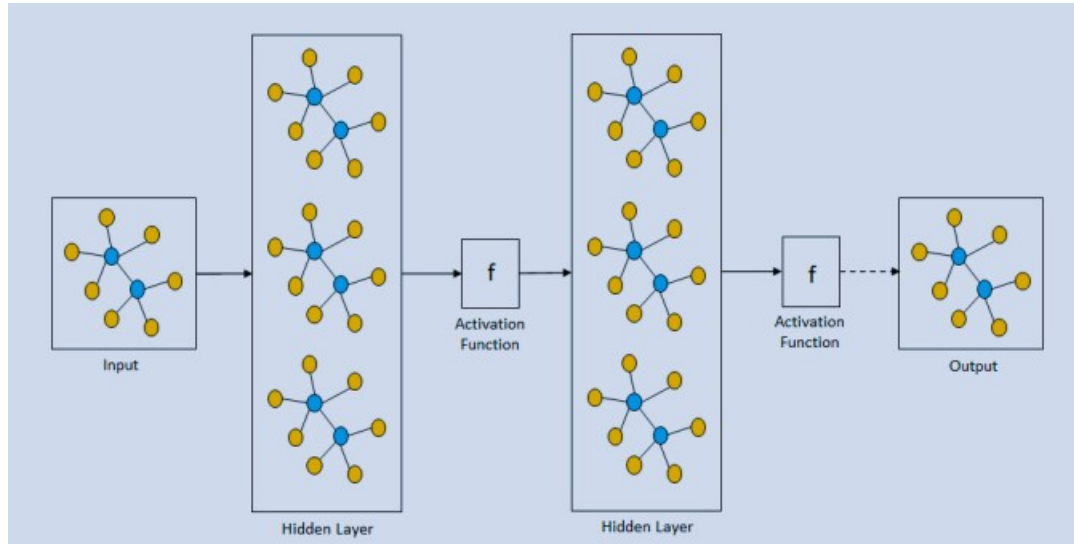


Figure 2.8: An example of simple GNN

2.1.6 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.

2.1.7 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 minimize 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.

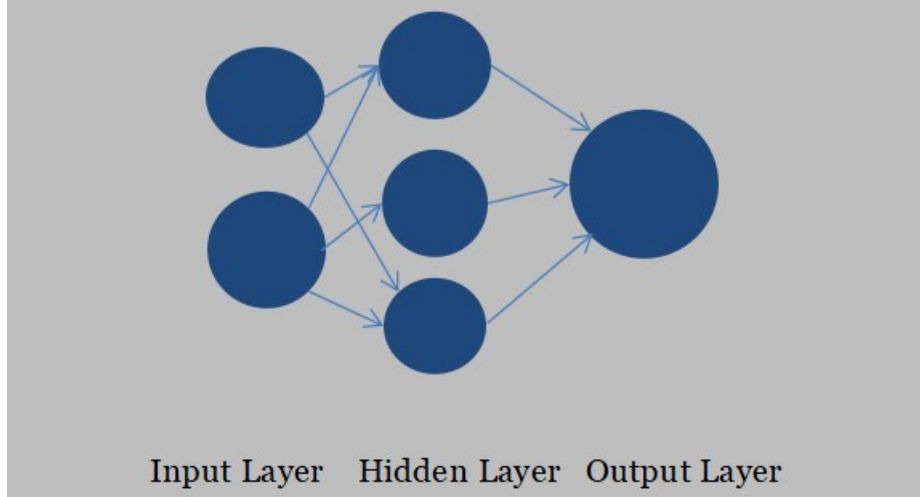


Figure 2.9: Simple neural network to processing forward propagation

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

$$\mathbf{Z}^2 = \mathbf{w}^T \mathbf{x} \quad (2.3)$$

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.

$$\mathbf{A}^2 = \mathbf{f}^1(\mathbf{z}^2) \quad (2.4)$$

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.

$$\begin{aligned} \mathbf{Z}^3 &= \mathbf{w}^T \mathbf{a}^2 \\ \mathbf{y} &= \mathbf{f}^2(\mathbf{z}^3) \end{aligned} \quad (2.5)$$

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

Artificial neural backward propagation refer 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. 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.

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

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.

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

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 calculation in the backward propagation, the overall network parameters are updated during training to reduce the overall loss.

2.1.8 Regularization

Regularization refers to the various techniques that discourage learning complex Neural network models, thus reducing the risk of over fitting the training data. This subsection describes the most prevalent and efficient regularization techniques: weight regularization, batch normalization, and dropout regularization.

Weight regularization

Weight regularization, also known as weight decay or ridge regression, is one of the most common regularization techniques where the essential idea is to encourage simpler models by penalizing larger weights. During the weight regularization, the loss function of the neural network is extended by so called regularization term. The regularization term can either be dependent on the sum of absolute values of

the weights, also known as L1 regularization, or the sum of squared values of the weights, also known as L2 regularization. Since larger weights result in a more significant penalty, the optimization algorithm pushes the model to have smaller weights, thus simplifying the model. The regularization term is also scaled by the hyper parameter α , determining how much regularization is required for the model. The updated loss functions based on L1 and L2 regularization are represented

as the equations 2.8 and 2.9 respectively.

Loss function with L1 regularization:

$$\hat{L}(w) = L(w) + \alpha \|w\|_1 = L(w) + \alpha \sum_i w_i \quad (2.8)$$

Loss function with L2 regularization:

$$\hat{L}(w) = L(w) + \frac{\alpha}{2} \|w\|^2 = L(w) + \frac{\alpha}{2} \sum w^2 \quad (2.9)$$

2.1.9 Deep Learning

Igor Aizenberg and his colleagues first make up the term DL in 2000. 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 multi layer learning structures known as neural networks that learn an implicit representation of the raw data on their own to produce the desired result.

Essential but highly complicated processing step known as feature extraction must be performed manually by domain experts for the algorithms to work. Others, deep learning algorithms learn these extract features automatically as the learning structures within this algorithms optimize to obtain the best possible abstract representation of the input data.

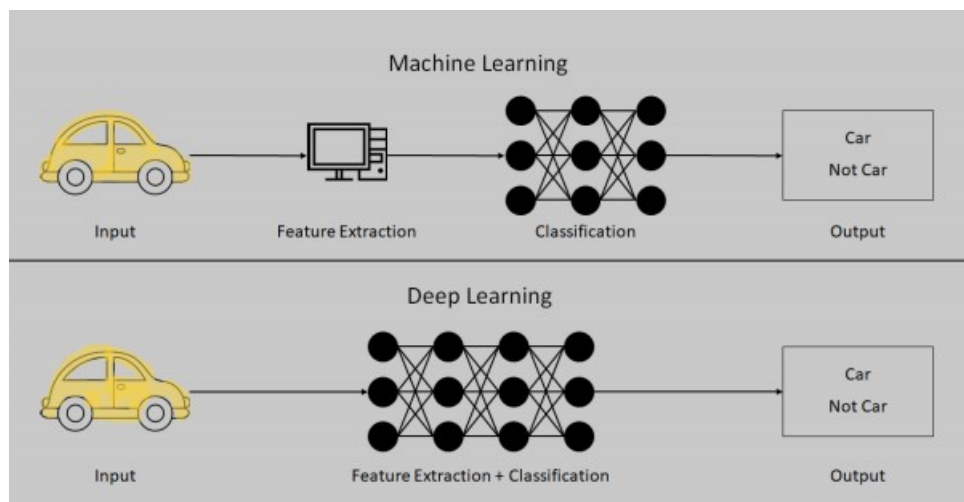


Figure 2.4: Difference among machine learning and deep learning

Due to this, deep learning becomes particularly useful as the majority of the data in the world is unorganized. 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 and Alexa and are self-driving cars.

Batch normalization

Batch normalization is another commonly used regularization technique where the neural networks are made faster and more stable by adding additional layers to the neural network that performs the normalization operation by reentering and resealing the inputs of a layer. Batch normalization techniques are primarily used to mitigate the problem of internal co-variate shift, where the distribution of network activation changes due to changes in network parameters during training. In other words, batch normalization limits the distribution of inputs to a layer when the parameters of the preceding layers change. These co-variate shifts are highly problematic with an increasing number of layers, resulting in a reduction in model accuracies. Batch normalization normalizes a layer input by subtracting the batch mean and dividing it by standard deviation. The normalization ensures that the layer inputs have a mean and standard deviation of 0 and 1, respectively, thereby fixing the problem of internal co-variate shift. However, the normalization 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 normalized values to accommodate the distribution of the input datasets.

***Dropout regularization**

Dropout regularization is another famous and powerful regularization technique where the key idea is to randomly drop nodes along with their connections from the neural network during training. During the training of neural networks with dropout regularization, 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. Thus, dropout regularization prevents nodes from adapting too much to the datasets, thus, significantly reducing overfitting.

2.3 Evaluation of model

Since machine learning models are developed to perform on hidden data, a meticulous evaluation is required to create a robust model. This section provides a brief description of many evaluation metrics used to determine model performance.

$$\text{Accuracy} = \frac{\text{data of correct predictions}}{\text{total number of predictions}}$$

2.9

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).

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 predicts 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 class-wise accuracy could have helped identify the inherent problems in the model. A confusion matrix doesn't evaluate model performance but provides further insights by displaying the number of correct and incorrect predictions for each class. Thus, for the example mentioned above, the confusion matrix is a $R^{2 \times 2}$ 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. Therefore, the formula for Precision is represented as the equation (2.11).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.11)$$

Also known as, measures the performance of the model while predicting positives. In other words, Recall signifies how many actual positives are identified correctly. Thus, the procedure for Recall is represented as the equation (2.12).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.12)$$

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 maximized simultaneously as both metrics operate in a zero-sum game framework. Other's increasing Precision minimize Recall and vice versa. Thus, either Precision or Recall can be maximized based on the task.

Another metric that can determine the model's performance grounded on both Precision and Recall is the F1 score. F1 score provides a weighted standard of Precision and Recall. The procedure for the F1 score is showing as the equation (2.13).

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

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. Thus, the maximum possible value of F1 score is 1 indicating perfect Precision and Recall, whereas a 0 indicates the value of either Precision or Recall to be null.

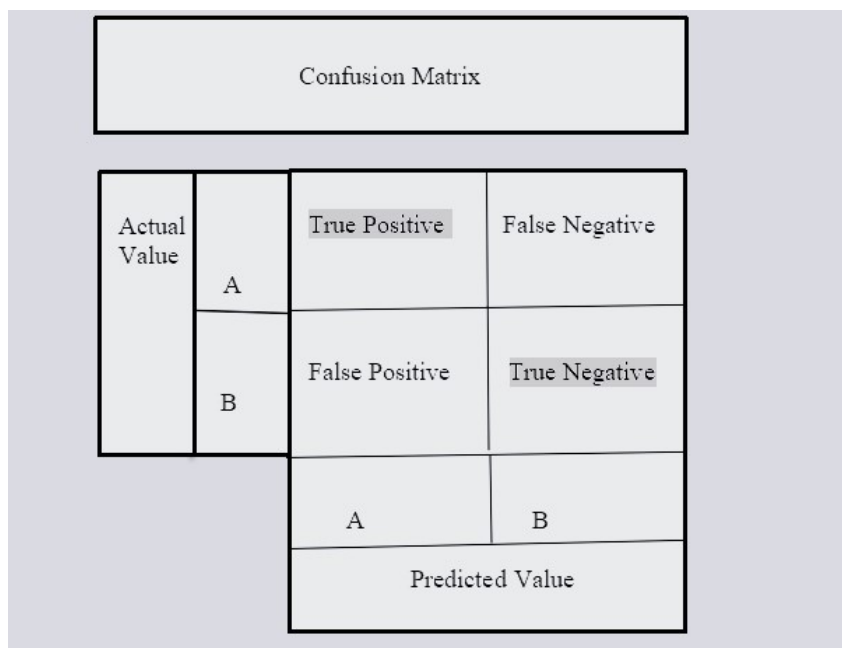


Figure 2.10: Confusion matrix

Chapter 3

Methods

3.1 Data extraction and processing

The data extraction and processing involve selecting an appropriate datasets and extraction and processing of required features.

3.1.1 Datasets 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. Thus, to measure the overall progress in solving the problem, choosing a suitable datasets 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 datasets

JAAD is a datasets for joint attention in the context of autonomous driving. The pedestrian and driver behaviour at the crossing point and the factors that influence them. the JAAD datasets provides a richly annotated taken of 346 short video clips (5 to 10 sec long) extracted from over 240 hours of driving footage. These videos filmed in North America and Eastern Europe represent scenes regularly for everyday urban driving in various weather conditions. Bounding boxes with occlusion target are provided for all pedestrians making this datasets satisfactory for pedestrian detection. Behaviour annotations specify attention for pedestrians that interact with the attention of the driver. Several tag and timestamped behaviour labels from a fixed list (e.x. stopped, walking, and looking) are available for each video. Also, a list of demo attributes is provided for each pedestrian (e.x. age,

direction of motion), and a list of visible traffic scene elements is provided for each frame.

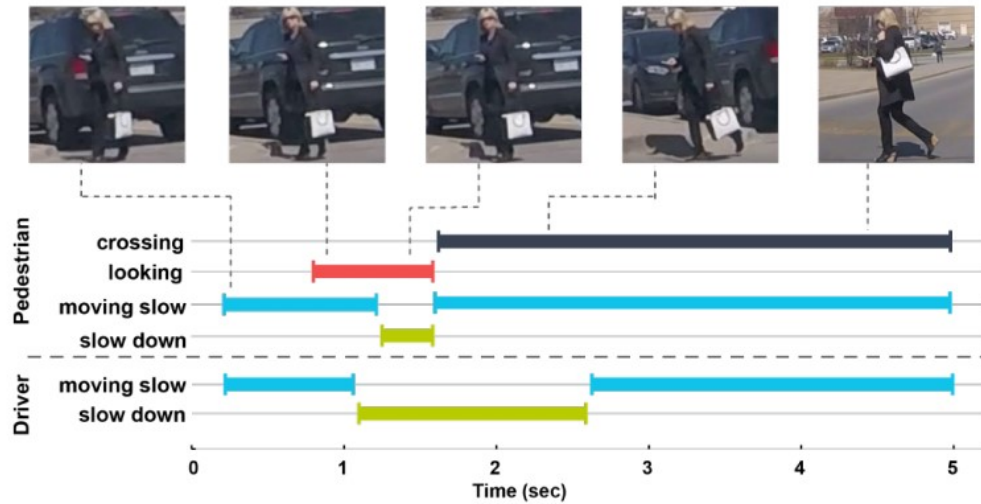


Figure 3. Datasets JAAD

The data used for this project is the JAAD datasets, especially of the data is annotation of pedestrian things, some of which are: Crossing, Looking, Direction, and Pedestrian Id.

In addition this paper mentioned upper, which is XML file. all data processing algorithms are available through Python scripts ready in the GitHub repository. The first step is to convert the videos to images and finally, the data of the images must match the image labels. With these data, we can implement more models to predict pedestrian behaviors.

Pedestrian intent estimation datasets

PIE is a relatively newer datasets for studying pedestrian behaviour in traffic. The focus of the datasets is on intent estimation and trajectory prediction. PIE contains over 6 hours of video recorded in typical traffic scenes with an on board camera. The datasets also supply accurate vehicle data from the OBD sensor (vehicle speed, heading direction, and GPS coordinates) synchronized with video frame. Behavioral annotations are accessible for pedestrians and vehicles that potentially interact with the pride-vehicle and the apposite infrastructure elements (traffic lights, signs, and zebra crossings). The datasets contains over 300K labeled video frames with 1841 pedestrian samples making it one of the most extensive publicly accessible datasets for studying pedestrian traffic. The PIE datasets also contains 896 examples of people who intend to but don't cross, 511 pedestrians to cross who eventually cross in front of the vehicle, and 429 pedestrians with no crossing intention.

Selection of Datasets

In order to select an appropriate datasets for this paper, 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.

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.

The decision matrix, the following steps were followed:

1. Evaluation criteria are the parameters on which the alternative are compared.
2. Add weights to evaluation criteria to signify the relative importance of each criterion.
3. Define different advanced toward to be compared.
4. For each criterion, rate each alternative +1 for better, 0 for same, and - 1 for worse.
5. Calculation the weighted sum for each alternative.
6. Select the advanced toward with the highest score.

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

Decision Matrix				
	Criterion	Weight	JAAD	PIE
1	Ease of implementation	2	+1	-1
2	Availability of resources	14	+1	-1
3	Number of pedestrians	5	-1	+1
4	Types of features	2	-1	+1
5	Variance of the datasets	4	-1	+1
Overall Score			+4	-4

Table 3.1: Decision matrix to select the appropriate datasets

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

3.1.2 Feature selection

The JAAD datasets provides maximum number of features for pedestrians that they divide into three categories:

Categorization of features available for pedestrians		
	Category	Features
	Pedestrian Behaviour	Bounding Box, Action, Reaction, Cross, Look
	Pedestrian Appearance	Pose, Object
	Pedestrian Attribute	Crossing, Crossing Point, Decision point Motion Direction

Table 3.2: Categorization of features available for pedestrians in the JAAD datasets

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

3.1.3 Cleaning the datasets

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

3.2 Machine learning framework

Once an appropriate datasets 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, Ke-ras 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. TensorFlow is primarily based on data flow 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++.

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 unfriendly front end, distributed training, and ecosystem of tools and libraries. Although PyTorch was released much later than TensorFlow, researchers are increasingly adopting PyTorch as their preferred framework because of its Python interface and the ease of building highly complex neural networks.

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	Ke-ras	TensorFlow	PyTorch
	Ease of usage	5	+1	0	0
	Library support	10	0	0	+1
	Training Duration	3	-1	0	+1
	Flexibility	4	-1	-1	+1
	Debugging Capabilities	10	-1	-1	+1
Overall Score			-13	-14	+27

Table 3.3: Decision Matrix to appropriate machine learning framework

3.2 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 both part in more details.

3.1.4 Architecture structure

As stated in the introduction, the goal of this paper is to create an architecture that would be as close as possible to real-world implementation, If testing this is not part of the scope of this paper. Thus, the paper 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 paper, the scene graph parsing and visual reasoning approach was considered the most appropriate approach for this paper. Since we noticed that pedestrian intent is basically dependent on context and interactions in scene rather than implicit features, creating a spatial model that captures intrinsic scene dynamics by encoding the sequence of subtle human actions is crucial to this paper. One such model is the, where pedestrian intent is predicted by parsing the scene graph and modelling the spatial relationship between various objects in the scene. Although the pedestrian intent prediction model developed by is closest to our thesis, one of the most significant drawbacks with 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 paper is an improvement upon the model developed in 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

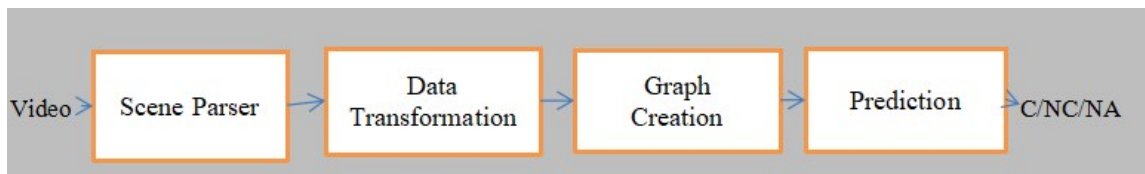


Figure 3.1: Novel architecture proposed

3.1.5 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 paper.

Objects of interest for scene parser		
	Category	Features
	Vehicle	Motorbike, Cycle, and others
	Road users	Pedestrians

Table 3.4: Object of interest for scene parser

In table 3.4, An off-the-shelf, YOLO v4, deep sort algorithm used as the scene parser algorithm. At a confidence of 0.7 and intersection over union (IOU) of 0.4 was used to reduce false positives. This features for the objects collect from the scene parser, this special features are combined with the behaviour, appearance and assign features collect from the JAAD datasets to make the overall feature set. This feature set is then restructured to a frame by frame structure to make the feature set suitable with the graph creation algorithm. grounded on this restructured quality set, the all features are changed into graphs by the graph creation algorithm and to the novel prediction model previous predicting pedestrian intent. The prediction model is designed to process both graph frame in order to trained the weights and predict pedestrian intent. At a moment this model crossing 14 frames, 31 frames and 44 frames in process for both pedestrian, corresponding to 0.5 seconds, 1 second and 1.5 seconds in process. In order connecting these algorithms emulates an end to end architecture that attempts to reduce the gap between prototyping and production.

3.1.6 Data transformation

This prediction model predicts pedestrian intent grounded on intrinsic scene dynamic encoding the spatial relationship among different objects in the

scene is difficult to this paper. Since the output from the scene parser algorithm is incompatible with the graph make algorithm, multiple data transformations are essential to model the spatial relationship between many objects in the scene and make the modeled scene dynamics compatible with the prediction model. This section describes the many data transformations performed to make the output of the scene parser algorithm essential with the graph creation algorithm.

Section 3.1.1, the JAAD datasets provides a large number of pedestrian attributes. However, the biggest drawback of the datasets is the structure of the annotations. This paper aims to create an end to end intent prediction model, restructuring the datasets 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 between pedestrians to improve the prediction. No preexisting sorting algorithm that accomplishes the goal mentioned above was found, a new algorithm was developed from scratch to achieve this goal. The input algorithm is the object detect from the scene parser algorithm through with the pedestrian part from JAAD datasets. 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.

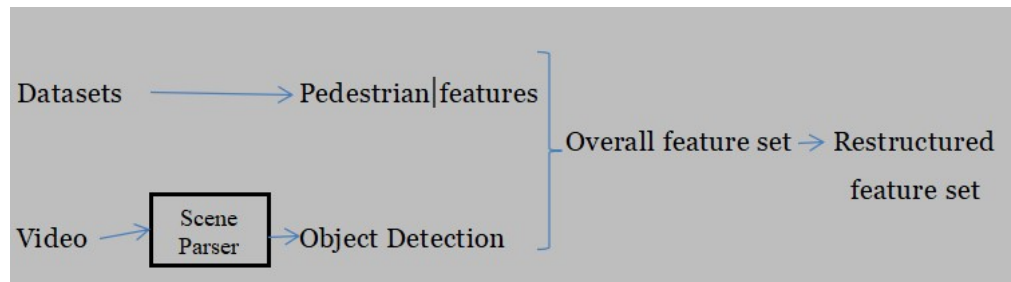


Figure 3.2: Visual representation of data transformation process

3.1.7 Graph frame creation

The prediction model is incompatible with the restructured feature set generated by the data transformation algorithm. Thus, the extemporization relationship between many 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. Thus, 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 figure 3.3.

In figure 3.3, both pedestrian by the scene parser is model as a graph node, where as the graph edges are used to reflect the spatial relationship among many objects in the graph frame. Further, both graph frame is modeled using two types of node: Pedestrian node and object node. both pedestrian node is marked by consistent marking such as behaviour, appearance and attribute, where as an objects attribute identify the object nodes. The vector delegation of the pedestrian node and object node is Showed as the equation 3.1.

$$\text{Pedi} = \hat{h}_i^{\max} \chi_i^{\min} \gamma_i^{\min} \text{occ}_i \text{beh}_i \text{app}_i \text{atti}_i \quad (3.1)$$

$$\text{Obj}_i = h_i^{\max} \chi_i^{\max} \gamma_i^{\min} \chi_i^{\min}$$

Since the graph frame encodes each data from the visual frame, the shape of the graph frame may vary due to the varying number of objects in the scene. A normal solved was solved by considering a fixed graph with a highest of fifty objects per frame to model pedestrian intent.

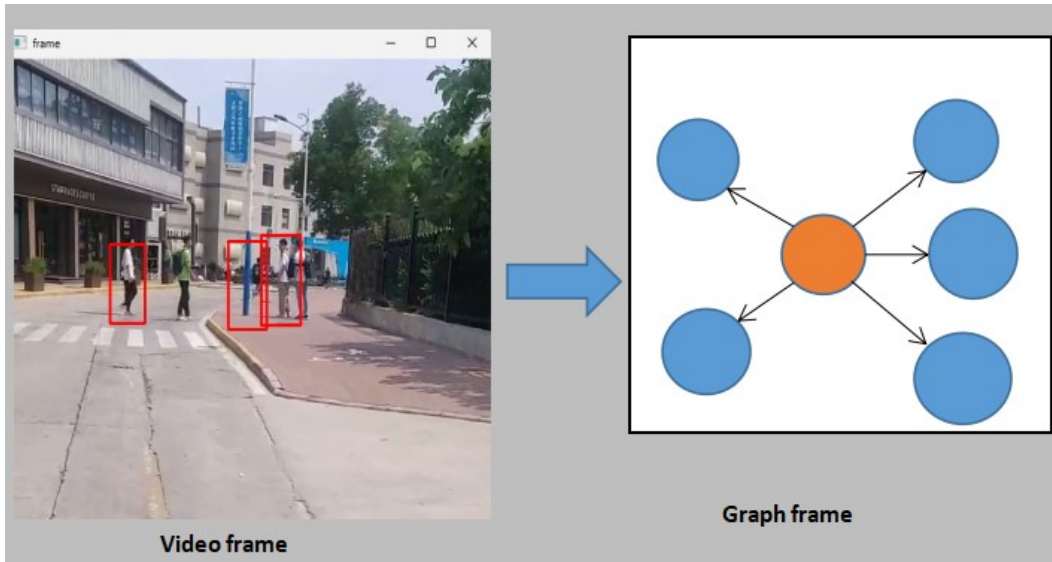


Figure 3.3: Visual representation of graph frame creation process: The graph nodes represented in the graph frame represents a pedestrian or an object from the visual frame whereas the edges represent the spatial relationship among the nodes.

3.1.8 Prediction architecture

Once each graph frame corresponding to a particular video frame is constructed to model the spatial relationship among various objects in the scene, the temporal relationship among different frames is model using Graph Convolution LST Memory. 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 spatial 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. The temporal connections are connected to classifier layers to align the pedestrian as crossing, not crossing, or not applicable. This classifier layer was solved with two variants, The first variant achieved the classifier layer without the soft max function, and the second variant achieved the classifier layer with the soft max 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.

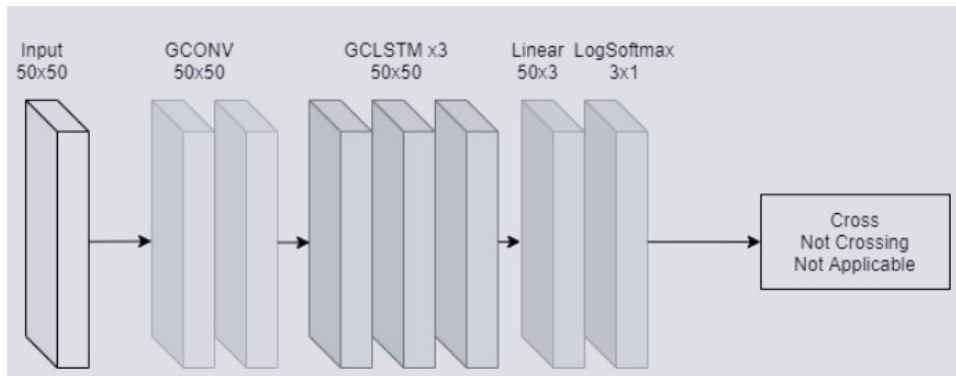


Figure 3.4: Overall prediction architecture

3.3 Experimentation and bench marking

The proposed architecture resembles an end to end structure, the first 266 videos are used to train the model, whereas videos from 266 to 347 are used to test the model. In addition to training and testing the novel architecture, this paper aims to measure the overall progress towards solving the pedestrian prediction problem. The paper achieves this by investigating the performance of multiple baseline methods on the JAAD datasets. Since standard training and testing procedures are crucial to evaluate the performance of baseline methods, a common data split was utilized for training and testing other baseline models. In paper, the performance of the novel architecture was bench marked against five different models: the Fusion of Spatial Skeletons for Intention Prediction model (FUSSI), the Static model (Static), the Stacked with multilevel Fusion RNN model (SFRNN), the Convolution 3D model (C3D) and the Pedestrian Crossing Prediction with Attention model (PCPA). Out of the five above mentioned models, only the FUSSI model 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 below 3.5:

Training Parameters							
	Models	Datasets	Batch	Loss	LR	Optimizer	Epochs
	Static	JAAD	32	BCE	1e-6	ADAM	10
	SFRNN	JAAD	32	BCE	1e-7	ADAM	40
	C3D	JAAD	16	BCE	5e-6	ADAM	40
	PCPA	JAAD	8	BCE	5e-5	ADAM	20
	Novel 1	JAAD	2	MSE	1e-3	ADAM	5
	Novel 2	JAAD	1	NLL	1e-3	ADAM	5

Table 3.5: Experimental setup from training models

Chapter 4

Results and Discussion

4.1 Novel intent prediction model & Result

The goal of this section is to analyse the results obtained from the novel intent prediction model developed in this paper and discussion the project outcome based on the research questions in section 1.4. The results are primarily analyzed on the effect of time to event on the delicacy of the model.





	Not-Crossing	Crossing	
CORRECT			CORRECT
INCORRECT			INCORRECT

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

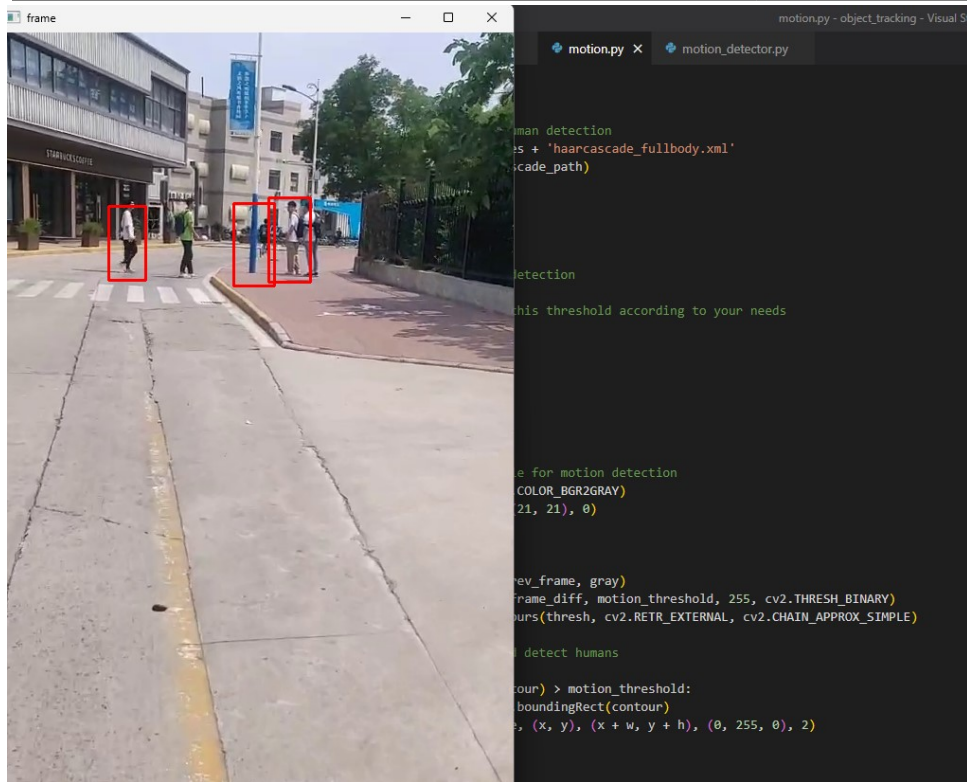


Figure 4.1.1: Object motion tracking

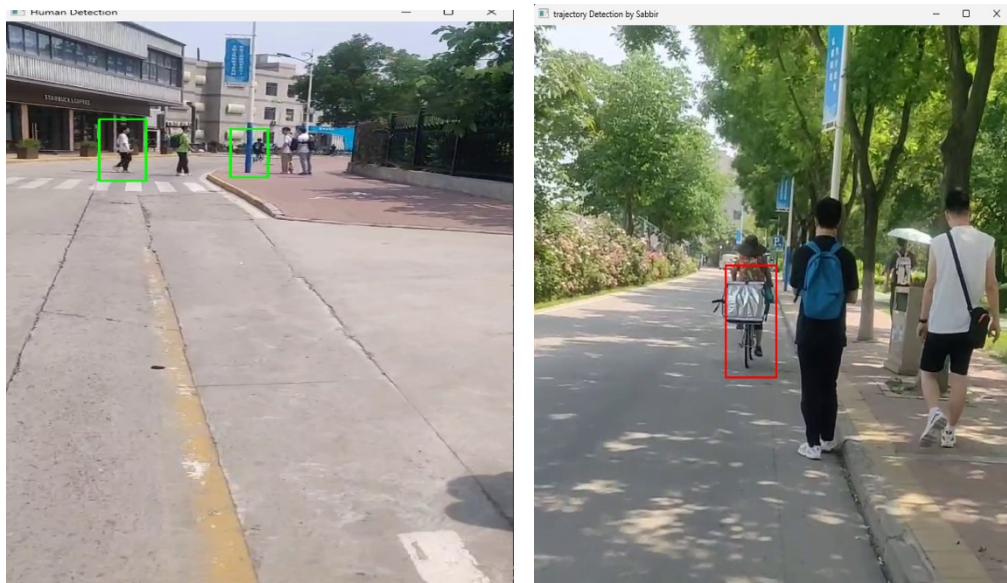


Figure 4.1.2: Object tracking & trajectory

Effect of time-to-event

Since the JAAD datasets 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 paper, the model delicacy 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.

Variation in delicacy with time to time

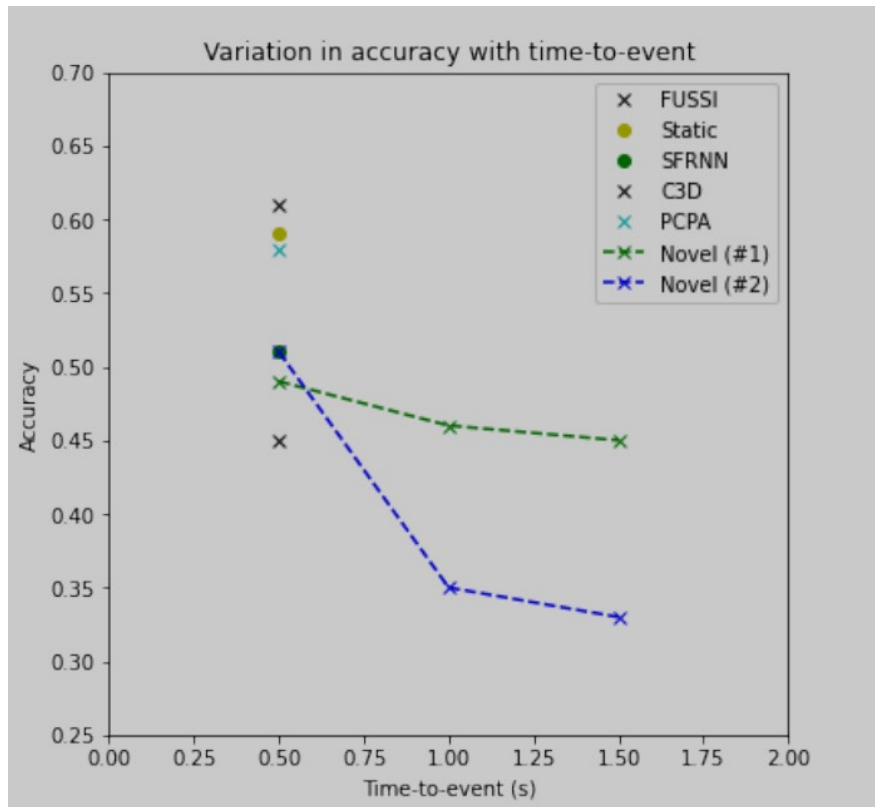


Figure 4.2: Effect of time to event on delicacy of the novel intent prediction model

As it can be observed in the figure 4.2, the first variant of the proposed intent prediction model achieves an delicacy of 49% while predicting pedestrian intent 0.5 seconds in approach, 46% while predicting pedestrian intent 1 second in approach, and 45% while predicting pedestrian intent 1.5 seconds in approach. Since, the second variant of the proposed intent prediction model achieves an delicacy of 51% while predicting pedestrian intent 0.5 seconds in approach, 35% while predicting pedestrian intent 1 second in approach, and 33% while predicting pedestrian intent

1.5 seconds in advance. At the same time, the delicacy 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 delicacy 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 be observed that the delicacy remains relatively the same when the time to event increases from 1 second to 1.5 seconds. On analyzing 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 paper aims to analyse the results obtained from bench marking the novel intent prediction model developed in this paper 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 bench marking the novel intent prediction model developed in paper against the baseline models. It can be observed from the table 4.1 that the C 3D model has the best delicacy and recall scores, where the FUSSI model has the best AUC, precision & 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, most reasons for the lower precision and recall scores could be the contemporaneous prediction for multiple pedestrians. Unlike datasets based on complete crossing sequences, a frame by frame datasets 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
	FUSSI	0.45	0.51	0.83	0.79	0.82
	Static	0.58	0.52	0.71	0.65	0.84
	SFRNN	0.52	0.47	0.64	0.63	0.67
	C3D	0.64	0.52	0.74	0.63	0.92
	PCPA	0.59	0.54	0.72	0.65	0.77
	Novel Architecture	0.53	-	0.38	0.34	0.48

Table 4.1: Benchmarking results of the novel intent prediction model developed against the baseline model

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 precision 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 lower precision & recall values could be because 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 recycling the minimum number of graph frames required to predict pedestrian intent accurately, resulting in a higher number of incorrect result the beginning, which could also decrease overall perfection and recall scores.

Similar to the results obtained in the table 4.1, a comparison between the novel intent prediction model developed in this paper 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 among the various methods.

Comparison among models						
	Model	Category	Observation Endpoint	Observation Length (s)	Prediction Length (s)	Simultaneous Prediction
	FUSSI	Pedestrian Detection and Tracking	before event	0.5	0.5	individual
	Static	Action Prediction	complete sequence	0.5	next frame	individual
	SFRNN	Action Prediction	before event	0.5	2	individual
	C3D	Action Prediction	complete sequence	0.5	next frame	individual
	PCPA	Action Prediction	complete sequence	0.5	next frame	individual
	Novel Architecture	Scene Graph Parsing and Visual Reasoning	before event	0.5	0.5/1/1.5	multiple

Table 4.2: Comparison of characteristics among 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. Further, it can be observed that none of the baseline methods is equipped to handle multiple pedestrians simultaneously, which is 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 models sample crossing cycling data, where as the novel architecture & the FUSSI are 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

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

4.2.1 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. Since, the second variant of the proposed intent prediction model achieves an delicacy of 51% while predicting pedestrian intent 0.5 seconds in approach, 35% while predicting pedestrian intent 1 second in approach, and 33% while predicting pedestrian intent 1.5 seconds in approach. One of the significant reasons for the lower delicacy 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.

Model performance					
	Model	Accuracy	AUC	F1	Processing times
	FUSSI	0.44	0.87	0.83	1 fps
	Static	0.46	0.45	0.54	31 ped/sec
	SFRNN	0.51	0.51	0.73	39 ped/sec
	C3D	0.60	0.52	0.75	6.7 ped/sec
	PCPA	0.58	0.52	0.71	6.4 ped/sec
	Novel Architecture	0.51	-	0.39	6 fps

Table 4.3: Compare among model performances at 15 frames/0.5 seconds

There can be numerous reasons for this difference in processing pets. The FUSSI network processes image data over the complete prediction channel, whereas the new armature converts the image data into graphs before prognosticating the prediction intent. The FUSSI network requires a disguise estimation for at least 15 frames corresponding to 0.5 second before prognosticating pedestrian intent. Other hand, the new armature uses the scene energetic enciphered as graphs to prognosticate rambler intent. Since, it can be observed that the other models prognosticate prediction intent predicated on cycling data rather than frame by frame data. thus, the processing pets are measured in pedestrian per second rather than frames per second.

4.2.2 Difficulties

One of the most significant limiting factors of paper work was the input datasets. The JAAD datasets for this paper was motivated by the hardware resources accessible to reuse the datasets rather than the datasets contents. This select proved to be severely restricting while training the model. The JAAD datasets did have a high number of attributes. However, the number of pedestrians containing all the attributes was few, and the datasets was severely biased towards crossing pedestrians. On analyzing 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 groundtruth data for pose estimation. The pedestrian intent prediction problem has been a topic of active research, and most approaches have utilized a combination of 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 publicly available. Thus, not applying the pose estimation data also restrained the accuracy of the model.

Lastly, the problem that complicated the paper 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. Thus, we were surprised that most accessible models do not predict intent based on incomplete inputs as they appear in real-time but predict using complete crossing sequence after resulting complete sequences. Due to this, every publicly available datasets 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. This paper had to redesign the input datasets as none of the publicly accessible models provided frame-by-frame information. Since this redesigned datasets was adopted for training the network, the redesigned datasets 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 grounded on complete crossing sequences, a frame by frame datasets provides all the data in the frame. which is the predictions for multiple pedestrian crossing expected. Normally, models handling frame by frame information must be designed to handle dynamic inputs. However, this paper works around that problem by setting a maximum number of pedestrians that model can predictably any time instance. Another reason for the lower delicacy 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 delicacy scores could be the prediction method. The novel architecture is designed end to end, the model raised predicting pedestrian intent indeed without cycling the minimum number of graph frames required to predict pedestrian intent delicacy, solved maximize number of incorrect results at the beginning, which could also decrease the overall delicacy scores. Having a complicated design proved highly challenging for paper.

4.2.3 Ethical Aspects

The potentially 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 needed to be constantly improved even after deployment to cover all types of edge cases, accumulating real-world data to ameliorate the model, a common practice within the deep learning community could further complicate the issue.

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 over would be demanded to be answered first before planting any deep learning models like the pedestrian intent prediction model in the real world.

In paper, since the pedestrian intent prediction models are grounded on the intimately publicly accessible datasets, such ethical issues don't exist as the data is collected considering these ethical issues.

4.2.4 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 generally suited for a top down fixed view of the scene, whereas the latter is generic to handle all use cases. However, new approaches like action prediction and visual logic algorithms have yielded better results; the contribution of paper like our approach the understanding of modern deep learning algorithms.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This paper explores the colorful 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, Deep sort, graph complication networks and graph complication recurrent networks. The first variant of the proposed intent prediction model achieves an delicacy 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. Since, the second variant of the proposed intent prediction model achieves an delicacy 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 delicacy is caused due to a combination of multiple reasons such as real time processing of data, simultaneous predictions for multiple pedestrians, the result of the scene parser algorithm, and premature prediction of pedestrian intent due to the essential design of the network. Since pedestrian intent prediction has been a topic of active research, this paper aims to advance the overall progress in working the intent prediction problem by developing an algorithm focusing on real time data processing. The spatial relationship between colorful objects in the scene carries a solid connection to pedestrian intent, indicating that environment and intrinsic scene dynamics significantly affect pedestrian intent. Alternately, it could be due to the datasets being biased. further algorithmic results and more complicated bench marking datasets must be developed to simulate real life use cases and break the intent prediction probable.

5.2 Future Work

Since the JAAD datasets had a limited number of climbers containing all the features, the model proved prejudiced towards samples. thus, to ameliorate the model generality, the model can be trained on more complex intimately available datasets like PIE. In addition to the datasets, natural features similar as disguise can be included using Open Pose. Although using an out the shelf algorithm to calculate disguise would not be as accurate as having disguise reflections, it could be tried in the future to ameliorate intent prediction. also, this paper attempts to apply an end- to- end network that reduces the gap between prototyping and product. Although the paper couldn't apply a perfect end- to- end network with total online processing of inputs, the same could be tried in the future to ground the gap to deployment. Another approach that could be tried in the future is the development of a general algorithm to restructure being datasets to give frame by frame in- conformation. Since the algorithm developed to restructure the JAAD datasets was knitter- made for our use case, a general algorithm can be developed to redesign all unborn datasets. Incipiently, since a frame by frame datasets provides all the information in the frame, running of dynamic inputs to contemporaneously prognosticate multiple climbers in a frame can be tried in the future. In conclusion, paper attempts to bring real time intent prediction algorithms closer to deployment, important work needs to be achieved to break this problem in real world surroundings .

Acknowledgments

Acknowledgments should be words of appreciation to supervisors and other organizations or individuals who guided or assisted in the completion of the graduation project.

References

- [1] S. Ahmed, M. N. Huda, S. Rajbhandari, C. Saha, M. Elshaw, and S. Kanarachos, ‘Pedestrian and cyclist discovery and intent estimation for independent vehicles: A survey’ *Applied Sciences*, vol. 9, 06 2019.
- [2] J. Breene, M. Khayesi, R. McInerney, A. Sukhai, T. Toroyan, 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 climbers and cyclists by 2d pose estimation’ *CoRR*, vol. abs/1910.03858, 2019.
- [4] F. Piccoli, R. Balakrishnan, M. J. Perez, M. Sachdeo, C. Nuñez, K. Andreasson, K. Bjurek, R. D. Raj, E. Davidsson, C. Eriksson, V. Hagman, J. Sjöberg, Y. Li, L. S. Muppirisetty, and S. Roy chowdhury, ‘Fusion of spatiotemporal skeletons for intention prediction network’ *CoRR*, vol. abs/2005.07796, 2020.
- [5] Z. Fang, D. Vázquez, and A. López, ‘On board detect of pedestrian intentions’ *Sensor*, vol. 17, p. 2193, Sep 2017.
- [6] B. Liu, E. Adeli, Z. Cao, K. Lee, A. Sheno, 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 System with Intelligent Vehicles*. Springer Publishing Company, Incorporated, 2013.
- [8] C. Zhou and J. Yuan, ‘Bounding-box regression for pedestrian detection and occlusion estimation’ in *Proceedings of the European Conference Computer Vision*, 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 (New York, NY, USA)*, p. 1597–1605, Association

- [10] W. Liu, S. Liao, W. Hu, X. Liang, and X. Chen, ‘Learning efficient single stage pedestrian sensors by asymptotic localization fitting’ in ECCV, 2018.
- [11] E. Ristani and C. Tomasi, ‘Features for multi target multi-camera tracking and re-identification’ vol. abs/1803.10859, 2018.
- [12] U. Iqbal, A. Milan, and J. Gall, ‘Joint multi-person pose estimation and tracking’ vol. abs/1611.07727, 2016.
- [13] D. Varytimidis, F. Alonso-Fernandez, B. Durán, and C. Englund, ‘Action and intention recognition of pedestrians in urban traffic’ vol. abs/1810.09805, 2018.
- [14] N. Dalal and B. Triggs, ‘Histograms of oriented gradients for human detect’ Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886–893, 2005.
- [15] 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.
- [16] 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, pp. 5903–5908, 2018.
- [17] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrilu, ‘A new benchmark for vision-based cyclist detection’ in 2016 IEEE Intelligent Vehicles Symposium, pp. 1028–1033, 2016.
- [18] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrilu, ‘Context-based pedestrian path predictio’ in Computer Vision – ECCV 2014 T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), pp. 618–633, Springer International Publishing, 2014.
- [19] G. Habibi, N. Jaipuria, and J. P. How, ‘Context-aware pedestrian motion prediction in urban intersections’ 2018.
- [20] 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, pp. 1–8, 2018.
- [21] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, ‘Repulsion loss: Detecting pedestrians in a crowd’ vol. abs/1711.07752, 2017.
- [22] S. Tang, B. Andres, M. Andriluka, and B. Schiele, ‘Multi-person tracking by multi cut and deep matching’ vol. abs/1608.05404, 2016.
- [23] 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, pp. 679–684, 2014.
- [24] 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 ITS, pp. 1520–1526, 2015.

-
- [25] N. Jaipuria, G. Habibi, and J. P. How, ‘A transferable pedestrian motion prediction model for intersections with different geometries’ vol. abs/1806.09444, 2018.
- [26] Y. F. Chen, M. Liu, and J. P. How, ‘Augmented dictionary learning for motion prediction,’ in 2016 IEEE International Conference on Robotics and Automation , pp. 2527–2534, 2016.
- C. Park, J. Ondřej, M. Gilbert, K. Freeman, and C. O’Sullivan, ‘Human intention-aware robot planning for safe and efficient navigation in crowds’ in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems , pp. 3320–3326, 2016.
- [27] J. Bütetage, H. Kjellström, and D. Kragic, ‘Anticipating many futures: Online human motion prediction and synthesis for human-robot collaboration’ vol. abs/1702.08212, 2017.
- [28] 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, pp. 5958–5964, 2019.
- [29] 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 , Nov 2019.
- [30] J. Bütetage, H. Kjellström, and D. Kragic, ‘Anticipating many futures: Online human motion prediction and synthesis for human-robot collaboration’ vol. abs/1702.08212, 2017.
- [31] R. Luo and L. Mai, ‘Human intention inference and online human hand motion prediction for human-robot collaboration’ in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems , pp. 5958–5964, 2019.
- [32] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, ‘Predicting the future: A jointly learn model for action anticipation’ in Proceedings of the IEEE/CVF, (ICCV), October 2019.
- [33] 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
- [34] 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, pp. 5958–5964, 2019.
- [35] 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’ vol. abs/1902.03748, 2019.
- [36] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. D. Reid, S. H. Rezatofighi, and S. Savarese, ‘Multi model trajectory forecasting using bicycle and graph attention networks’ vol. abs/1907.03395, 2019.

