A Comparative Performance Analysis of Accident Anticipation with Deep Learning Extractors

by

Alfi Mashab Mostak
22341078
Nayna Jahan Neha
19101223
Azwaad Labiba Mohiuddin
19101032
Adiba Tabassum
19101211

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Alfi

Alfi Mashab Mostak
22341078

Nayna Jahan

Nayna Jahan Neha
19101223

Azwaad Labiba.

Azwaad Labiba Mohiuddin
19101032

Adiba Tabassum

Adiba Tabassum
19101211

Approval

The thesis titled "A Comparative Performance Analysis of Accident Anticipation with Deep Learning Extractors" submitted by

1. Alfi Mashab Mostak (22341078)
2. Nayna Jahan Neha (19101223)
3. Azwaad Labiba Mohiuddin (19101032)
4. Adiba Tabassum (19101211)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 22, 2022.

Examining Committee:

Supervisor:
(Member)

Iqbal

Muhammad Iqbal Hossain, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Mohammed Abid Abrar
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

____

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Accident anticipation has become a major focus to avert accidents or to minimise their impacts. Over the years, several network systems are being developed and applied in self-driving technology. Despite the fact that advancement in the autonomous industry is fast-growing, major efficiency is required in the network systems that are gradually emerging. Recent research has proposed a novel end-to-end dynamic spatial-temporal attention network (DSTA) by combining a Gated Recurrent Unit (GRU) with spatial-temporal attention learning network, to identify an accident video in 4.87 seconds before the occurrence of the accident with 99.6% accuracy when tested on the Car Crash Dataset (CCD). However, DSTA has not been able to provide efficient results on the Dashcam Accident Dataset (DAD) dataset. Moreover, the GRU model integrated in the DSTA network has a weak information processing capability and low update efficiency amid several hidden layers. The decision-making process of the accident anticipation network may be understood using the high quality saliency maps produced by the Grad-CAM and XGradCAM approaches. In this paper, we evaluate that using Wide ResNet network enhances the performance mechanism of feature extraction to increase accident anticipation precision. This change improves the capacity to process information and the learning efficacy. In addition, we suggest employing a Gated Recurrent Unit (GRU) network which will serve as a prominent feature to train the model to recognize data's sequential properties and apply patterns to forecast the following likely event. Hence, we plan to incorporate Wide ResNet50, a system for extracting features which will identify the vehicles at risk by using wider residual blocks. These neural networks generate labels for identifying hazardous conditions in driving environments in order to anticipate accidents.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Muhammad Iqbal Hossain sir for his consistent support and valuable advice in our work. He helped us whenever we needed help.
Thirdly, we would like to express our sincere gratitude to our co supervisor Mohammed Abid Abrar sir for his kind guidance and technical support.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration **i**

Approval **ii**

Abstract **iv**

Acknowledgment **v**

Table of Contents **vi**

List of Figures **viii**

List of Tables **x**

Nomenclature **x**

1 Introduction **1**

1.1 Research Problem 1

1.2 Research Objectives 3

1.3 Thesis Structure 3

2 Literature Review **5**

2.1 Related Works 5

2.2 Algorithms and Models 11

2.2.1 Recurrent Neural Network (RNN) 11

2.2.2 CNN Architectures for Feature Extraction 11

3 Methodology **12**

3.1 Dataset 12

3.2 Feature extraction in image processing 13

3.2.1 Feature Extraction with VGG16 13

3.2.2 Feature Extraction with ResNet50 14

3.2.3 Feature Extraction with Wide_ResNet50_2 15

3.2.4 Selection of Feature Extractor 17

3.3 Accident Prediction 18

3.3.1 Convolution Neural Network 18

3.3.2 Long Short-Term Memory (LSTM) 18

3.3.3 Gated Recurrent Unit (GRU) 18

4 Implementation 20
4.1 Dataset Preparation 20
4.1.1 Frame extraction 20
4.1.2 Data Preprocessing and Classification 20
4.1.3 Dataset Splitting 20
4.2 Network description 21
4.2.1 Wide Residual Networks for feature extraction 21
4.2.2 Utilising GRU for Spatio-Temporal Relational Learning 22

5 Result Analysis and Experimental Evaluation 24
5.1 Evaluation Metrics 24
5.1.1 Average Precision 24
5.1.2 Time-to-Accident 24
5.2 Evaluation of Test Results 25
5.2.1 Performance of feature extractor 25
5.2.2 Comparative Analysis considering the State-of-the art approaches 27
5.2.3 Evaluating generated saliency maps using GRAD-CAM 27

6 Conclusion 29

Bibliography 29

List of Figures

2.1 *Diagram depicting the suggested AdaLEA, LEA (Loss for Early Anticipation) and EL (conventional work). Here, the ATTC is 3.65 seconds, compared to 2.99 seconds for a conventional EL.* 5

2.2 *The model proposed by Bao et al. Graph representations G(X_t, A_t), time step t, the latent relational representations Z_t , accident score at* a_t 7

2.3 *Rundown of the DSTA network,* $O_t \longrightarrow$ *Object Features* $F_t \longrightarrow$*Frame features with reduced dimensions* o_t *passes through a dynamic spatial attention module to gain weighted* o'$_t$ *object level features.* $X_t = o'_t + f_t$ *goes into the GRU.* h'$_{t-1}$ $\longrightarrow$ *hidden representation with a focus on the hidden states' temporal evolution. Inputs to the DSTA network are :* $_t + h'_{t-1}$ *, and returns* $a_t \longrightarrow$ *probability of frame having a crash.* 8

2.4 *Multi-Net framework with two parallel Multi-label architecture* 9

2.5 *Overview of the proposed Grad-CAM method* 9

2.6 *Illustration of the proposed method for generating spatio-temporal attention guidance.* 10

2.7 *DRIVE Model* 11

3.1 *Overview of our CNN-RNN network* 12

3.2 *The summary of the VGG-16 model* 14

3.3 *Architecture of ResNet50 on ImageNet features* 15

3.4 *Architecture of ResNet50 on ImageNet features* 15

3.5 *Structure of wide residual networks. For clearance, the final classification layer is skipped.* 16

3.6 *Different Residual Blocks* 16

3.7 *Extracted features of an input frame with accident using Wide_ResNet50_2* 17

4.1 *Feature Extraction from specified layer* 22

5.1 *Comparison of AP between model trained using ResNet50 (blue line) and VGG16 (orange line)* 25

5.2 *Comparison of mTTA between model trained using ResNet50 (blue line) and VGG16 (orange line)* 25

5.3 *Comparison of AP between model trained using ResNet50 (orange line) and Wide_ResNet50 (blue line)* 26

5.4 *Comparison of mTTA between model trained using ResNet50 (blue line) and Wide_ResNet50 (orange line)* 26

5.5 *Human attention maps* 28

5.6 *Grad-CAM method saliency maps* 28

List of Tables

5.1 TABLE 1 26
5.2 TABLE 2 27
5.3 TABLE 3 27

Chapter 1

Introduction

Human beings are susceptible to gradual changes and the latest change that we have been trying to adapt to is the use of autonomous driving vehicles. With the passage of time, as autonomous driving vehicles are now more popular than ever and are frequently being tested to ensure their capability of being on the road, it is very necessary to anticipate accidents. The ability of the human brain to select relevant sensory information for preferential processing, which improves performance in visual and cognitive activities, is referred to as visual attention. Driver attention prediction in critical situations is a particularly complicated computer vision issue, yet it is necessary for autonomous driving. While driving, important visual cues, such as a pedestrian, a bike, traffic light changes, or unusual activity of different other vehicles, influence human visual attention. As a result, the gaze behaviour of drivers can be utilised as a proxy for their attention.

Predicting the possibilities of an accident will help us to prevent accidents to a great extent or at least minimise their impacts on the people involved. Action anticipation is done by the means of analysing the driving behaviours and responses from the various video based datasets. These datasets in combination to different neural models have given notable results to determine the possibilities of accidents. However, it still requires more accuracy and a bigger time margin to make sure that there is enough time to react before an accident takes place.

Our proposed strategy includes a system for extracting features that will identify the vehicles at risk by using wider residual blocks, and employs a Gated Recurrent Unit network that will serve as a crucial feature to train the model to recognize data's sequential properties to forecast the upcoming event. In order to foresee accidents as early as possible, these neural networks produce labels for identifying hazardous circumstances in driving contexts.

1.1 Research Problem

Over the years, improving road safety has been a primary goal. Approximately 1.3 million people fall into arms of death each year as a consequence of traffic accidents, stated in the global status report on 2018 by the World Health Organisation on road safety. Not only the related people, and their families, but also the nation as a whole suffers huge economic and infrastructural losses due to traffic accidents.

Recently, Advanced Driver Assistance Systems (ADAS) equipped autonomous cars along with other autonomous vehicles have gained popularity. Furthermore, these cars completely share the road with ordinary vehicles driven by humans. As a result, autonomous cars and driver assistance systems must predict traffic incidents from natural driving scenarios in order to assure guaranteed safety for passengers, other vehicles, and pedestrians on the road. Furthermore, dashcam video footage contains visual cues for predicting a future accident, complicating the dynamic spatial-temporal interaction between traffic agents.

When autonomous driving makes life easier for consumers and meets critical industrial demands, it also raises worries about traffic accidents. There were about twenty-nine incidents where human drivers had to take control of the vehicle to avoid a possible accident [28].

Accident prediction is a difficult subject, and the computer vision group has been studying it for a few years. Traditionally, forecasting accidents using computer vision required evaluating live dashcam video data, which typically contains complicated spatial-temporal interactions between traffic participants and a dynamic background. With crowded traffic scenes and few visual indications, predicting how long an accident will last from early observed frames is extremely difficult. For improved prediction, most present techniques are designed to learn properties of accident-relevant agents while disregarding aspects of depth maps and human visual attention.

The quality of the scene analysis and how clearly the scenes are annotated determines the accuracy of accident anticipation. Due to the lack of comprehensively annotated road scene cues, development of the network systems for predicting crashes earlier becomes complicated. Although, localization and mapping [9], motion planning [6], are the parts of autonomous driving technology that have been considerably studied, the other important segments of this technology for instance the behavioural decision [5], and scene understanding of the roads [11], have not been broadly researched yet. Understandably, for scenes involving multiple tasks, scene categorization, object tracking and semantic segmentation are the important sub tasks of scene analysis; for which it is complex and crucial to combine these aspects together to train a network model. In spite of the fact that numerous computer vision tasks can be accomplished successfully by the utilisation of Convolutional Neural Networks (CNNs), complex scene analysis is still in need to produce multiple labels efficiently for the classification of complex driving scenes. To effectively extract the required cues, we propose to use the wide residual network features for extracting important cues from the driving scenes [25].

The prime motivation of any prediction model is to anticipate actions as early as possible. On the other hand, low prediction performance is one of the most critical challenges faced by the Recurrent Neural Network (RNN) models. Hence, Gated recurrent unit (GRU) [4] can be introduced as an optimal prediction model with a view to inaugurate optimization space and intensify the weights of important cues for strengthening the accuracy and time.

1.2 Research Objectives

To gain more precision in accident anticipation, we propose to use the GRU Model along with Wide ResNet50, a system for extracting features that will identify the vehicles at risk by using wider residual blocks, in the technology of autonomous cars to aid the drivers, passengers and pedestrians with safety to a great extent.

- We want to integrate wide residual networks as a feature extraction process to extract significant features as a notable way to train the model.
- Refine the information processing ability and improve the efficiency of learning by integrating it with the GRU which optimises the learning mechanism.
- Comprehend the key regions of interest important for prediction in a frame using GRAD-CAM generated saliency maps.
- A detailed comparison of viability and constraints of different combinations of feature extractors integrated with CNN and RNN models to predict accidents.

Thus, we can assess how quickly our suggested GRU integrated with the wide residual networks can anticipate automobile crashes in comparison to earlier state-of-the-art models.

1.3 Thesis Structure

This study has been divided into six sections to walk through an attempt to improve accident anticipation in order to reduce accidents or minimise their impacts.
Chapter 01 introduces the research problem that is going to be solved in the course of this study along with the objectives that the study plans to gradually accomplish. Our focus is to draw a comparison on different feature extractor combinations and integrate GRU into the network to improve its learning mechanism to anticipate accidents at the better rates and reduce the losses caused by them.
Chapter 02 discusses the works relevant to this particular study and the neural network models and algorithms essential for the research. The state of the art in this case is the novel end-to-end dynamic spatial-temporal attention network (DSTA) which manages to predict accidents 4.87 seconds earlier with an accuracy of 99.6% on the Car Crash Dataset (CCD), however, is unable to provide efficient results (3.66 seconds) on the Dashcam Accident Dataset (DAD). The network employed for this particular work can be called a CNN-RNN hybrid architecture combining the lightweight and faster RNN GRU model with various CNN based extractors to train the data.
Chapter 03 is the research methodology which emphasises on the gradual procedure employed to conduct the research. The dataset used in this regard is the Dashcam Accident Detection (DAD) dataset also known as SA dataset. The feature extraction for image processing involves extractors, namely, VGG16, ResNet50 and Wide_ResNet50_2. This chapter further explores the accident prediction part of the network for which we use the Gated Recurrent Unit (GRU) as it is a better, lightweight version of LSTM which strengthens the networks by long and short term memory using its three-gate architecture.

Chapter 04 provides a description of the training and testing process. Features extracted using the extractors from each frame flows into the GRU and the model estimates the probability using the hidden representation of whether there is an accident in that frame. For videos without accidents, cross entropy is the only loss function there. Loss of each frame is totaled, averaged, and then back propagated for the entire video clip.

Chapter 05 emphasises on the experimental evaluation and the data analysis. The foundation of our evaluation consists of two primary metrics, namely, Average Precision (AP) and Time To Accident (TTA). The model with the highest scores on the validation dataset is the only one kept. Our analysis shows that the Wide Residual Network, rather than both VGG and ResNet, is a favoured option as it anticipates accidents 4.11 seconds earlier on average.

Chapter 06 draws a conclusion to the study by providing an overview of the entire research procedure.

Chapter 2

Literature Review

2.1 Related Works

In this segment, we intend to discuss the works that have been previously done in respect to anticipating accidents using various techniques and their rate of success. Furthermore, we will review the challenges faced by each of the networks and how these network systems tried to overcome them.
In their paper [17] T Suzuki et al. used Adaptive Loss for Early Anticipation (AdaLEA) in an unique approach for traffic accident prediction with self-annotated Near-miss Incident DataBase (NIDB).

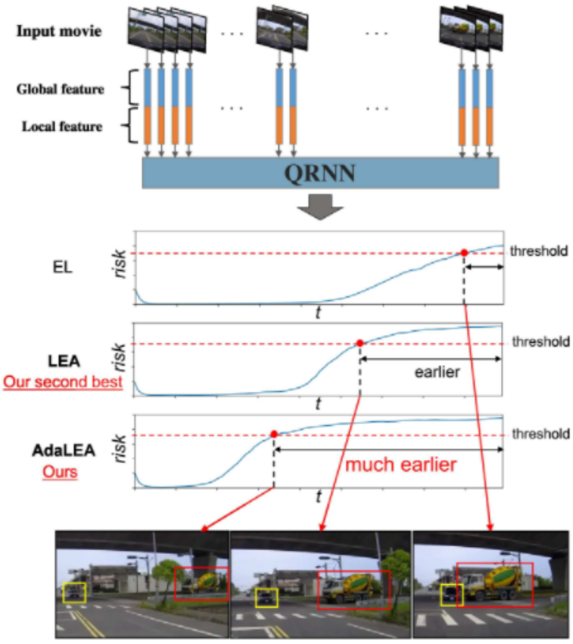


Figure 2.1: *Diagram depicting the suggested AdaLEA, LEA (Loss for Early Anticipation) and EL (conventional work). Here, the ATTC is 3.65 seconds, compared to 2.99 seconds for a conventional EL.*

During this phase of the training process, the model picks up new information gradually. The model's capacity to predict an accident and how early it can do so throughout each epoch are used by the loss adaptivity to establish penalty weights.

This approach draws its inspiration from curriculum, which employ planned and sequential data for training. Depending on how far ahead of time the model can anticipate a traffic collision, they update the weight value adaptively at each learning epoch. The Curriculum Learning[1] enhances generalisation of a model by ranging from easy to tough data samples in training time.

From each frame, the method retrieves global and local data. Then performs temporal analysis on the frames and calculates risk rate which predicts the possibility of a future tragedy. For temporal analysis in anticipation related tasks they used QRNN [8] instead of LSTM as it is faster and with temporal convolution on successive features, it is able to create stable anticipation. The model(AdaLEA) is trained using the revised loss function. The Exponential Loss(EL) used with the training method modifies the penalty weight in line with the difficulty at each frame to stabilise anticipatory learning. But EL does not encourage early expectation because it consistently assigns higher weights near the accident that is why an early anticipation method was proposed. Positive (a clip featuring a road accident) and negative (no accident, a routine driving scene) samples are used to calculate the proposed losses. While the loss in the negative sample is conventional cross-entropy, the weighting value in the positive sample slowly grows as a video frame approaches an accidental frame. Furthermore, they include a range of simple to challenging samples in training period which also boosts model adaptability. Furthermore, By referencing to ATTC, AdaLEA delivers adaptive penalty values based on the anticipation time. ATTC reflects how far ahead of time a model predicts on average. The Near-miss Incident Database(NIDB) will assess traffic risk forecasting and risk factor prediction.

Chen et al. [18] investigated an efficient way to recognize road scenes by developing a multi-label neural network that is trained in an extensive dataset introduced in this research containing driving scenes of currently occurring situations, road structures and weather condition with regard to impactful categorised samples of captured conditions with its scene dynamics and image resolutions. The system includes single and multi-class classification labels, in which the multi-category prediction is learnt from the multiple labelled objects. On the other hand, supervised-learning of samples that need to be taken care of during the time of training, are learnt using the single labels. In particular, the imbalanced categories are sampled by the boosting function of the proposed deep data integration method. Over and above, this research plays a significant role to the self-driving car technology by classifying multi-class labels effectively as this system can extract features from various input images using resolution adaptive mechanisms maintaining the most image information.

Fatima et al. [24] proposed a new Feature Aggregation block that reinforces every query object attribute by adding to the query object a weighted total of all object features in a particular video frame. The attention weights are established by appearance relations between distinct items in a particular frame, whereas the weighted sum represents global context unique to the query object. It is critical for the network to grasp the global context around an item in a particular frame in order to identify accidents. For action anticipation they used sequence modelling power of an RNN architecture, LSTM network, which gives an anticipation probability value. The dataset used here is the street accident (SA) dataset which includes videos captured with a frame rate of 20 frames per second from six cities in Taiwan.

In implementation of their model, they used Faster-RCNN to extract the objects. They applied LSTM with a dropout of 0.5 and a hidden state size of 512. Demonstrating that combining an FA block with an LSTM can offer us with additional information in both the spatial and temporal domains of a video stream. Bao et al. [22] proposes a traffic accident anticipation model which considers features specific to the agents and their spatio-temporal relations by using Graph Convolutional Networks (GCN) and Recurrent Neural Network (RNN) to exhibit the results at each time step. Dashcam videos are given as input in the framework from which a graph is constructed in combination of the identified objects and associated features at each time span. At each time step, in a cyclic process, the latent relational data are merged with the associated object features as input to an RNN cell to upgrade the hidden state from both spatial and temporal viewpoints. The predictive uncertainties are naturally formed because of the usage of the Bayesian neural network (BNN) to predict the scores of accidents. This paper shows the collection of the new Car Crash Dataset (CCD), which is formulated by labelling accident videos which are collected from Youtube with time-based annotations, distinctive environmental attributes, involvement of ego-vehicles and other participants in the accidents, as well as the reasons of the accidents which was more informative than the existing DAD and A3D datasets.

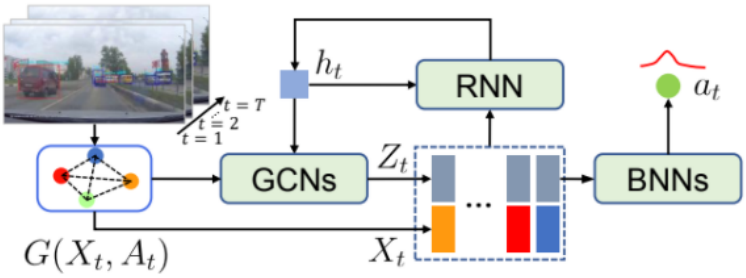


Figure 2.2: *The model proposed by Bao et al. Graph representations $G(X_t, A_t)$, time step t, the latent relational representations Z_t , accident score at a_t*

In a paper by [30] Karim et al., the Dynamic Spatial-temporal Attention (DSTA) network was proposed to analyse streaming dashcam video information that consist complex spatial-temporal connections of traffic agents in a dynamic backdrops in order to select different temporal snippets of a video sequence using the Dynamic Temporal Attention (DTA) network and to concentrate on informative spatial regions of frames with the use of the Dynamic Spatial Learning (DSA) network. To learn spatial temporal relational features along with scene appearance features, the Gated Recurrent Unit network has been implemented in place of the ordinary RNN. The DSTA framework reads the indices of the video frames. The DSTA then receives the temporal and spatial associations of the significant items in the dynamic scene as inputs, from which the network repeatedly learns and outputs the likelihood that subsequent frames will contain crashes. A 16-layered deep CNN network called VGG-16 feature extractor is used by the object detector to extract the features of the objects from each frame and detect a certain number of objects from each video frame with the highest detection scores. Consequently, the trained network system can classify one thousand objects from the photos. Using a fully connected embedding layer, dimensions of object and frame features can also be captured. After that, these features flow into the GRU, which is integrated with spatial and temporal attention modules. To say nothing of, GRU is a certain kind of RNN (Recurrent Neural Network) that masters the spatial-temporal relations among aspirant objects as the spatial-temporal relations of important objects and context information provides necessary information for accident anticipation. Thus, the dimension-reduced object and frame features from the DAD and CCD datasets with various environmental attributes containing the temporal and spatial relations are the inputs to the DSTA network. As a result, this model updates the likelihood of a future frame having a crash. During the training the network attempts to maximise the mean Time to accident by using backpropagation of the loss function and it can anticipate an accident as early as 3.66 seconds. Furthermore, modifying the DSTA network for different regions and different countries is an achievable way to advance the ability to predict accidents early with higher accuracy.

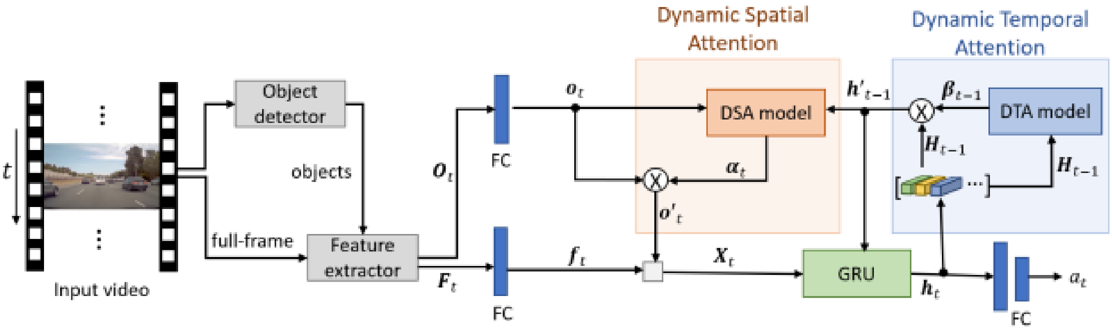


Figure 2.3: *Rundown of the DSTA network, $O_t \longrightarrow$ Object Features F_t $\longrightarrow$Frame features with reduced dimensions o_t passes through a dynamic spatial attention module to gain weighted o'_t object level features. $X_t = o'_t + f_t$ goes into the GRU. h'_{t-1} $\longrightarrow$ hidden representation with a focus on the hidden states' temporal evolution. Inputs to the DSTA network are : $_t + h'_{t-1}$, and returns a_t $\longrightarrow$ probability of frame having a crash.*

Karim et al. in their paper [25], emphasised on how comprehensive scene analysis is an integral part of effective anticipation of crashes and developed a system called MultiNet having two multi-task neural networks that looks for potential conditions for crashes from its environment and gives environmental alerts to autonomous cars or human drivers. Additionally, the Multi-Net system generates different labels in order to classify the driving views, in which real time object detector Yolo V3 [16] is used to identify specific objects in videos or images and DeepLab V3 is integrated as instance segmentation tool[14] for labelling each pixel in the image in order to categorise objects that are likely for occurring any sort of crash. In order to mitigate the scarcity of datasets, two entirely new datasets have been formed containing different roadway structures and accident scenes that were later transformed into image frames. The Multi-Net system incorporates two aligned multi-tasking network systems that performs image classification by splitting into two branches to provide labels for four variables namely crash likelihood, road functions, time and weather of the day.

Dash cameras of the cars that capture RGB images are the inputs of the first network that are processed to identify the frames that are critical to scene analysis and those

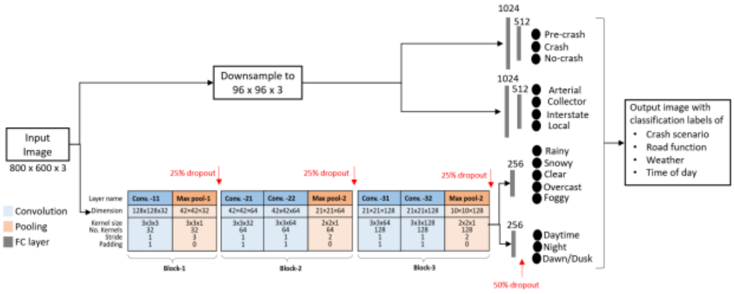


Figure 2.4: *Multi-Net framework with two parallel Multi-label architecture*

are fed to the two branches of Multi-Net after downsampling in order to labelize the likelihood of crash and function of the road. However, the second network is CNN for categorising both the time and the weather of the day. 90% of the times the classifier could successfully predict the crashes. Even though the research introduces the system as a vision sensor based network system especially for complex driving scene analysis, this classifier struggles to distinguish between the time of day due to having similarity in the features. Apart from that, if the speed of the inferencing function of the segmentation task is improved, the real time scene analysis can be achieved more efficiently.

Karim et al. [29] train deep neural networks (GRU) to learn the spatio-temporal relationships from traffic accident videos which can anticipate accidents as early as 4.57s before occurrence with an accuracy of 94.02%. A sequence of frames from the collected videos are fed into a feature extractor which maps it into a feature map and ultimately generates a feature vector which is sent into the GRU. The GRU learns the hidden representations of the features and predicts the accident probability scores which are further calculated into gradient maps with respect to the previously obtained feature maps. Finally, the importance weights are calculated and fed into the Grad-CAM to determine the final saliency maps. The results demonstrate that their Grad-CAM method is particularly effective in producing an increased visual explanation for the network's predictions.

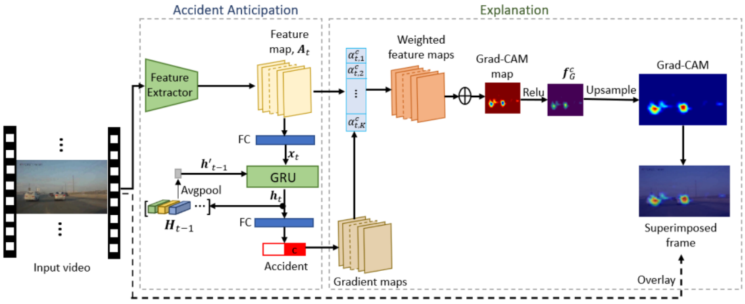


Figure 2.5: *Overview of the proposed Grad-CAM method*

In this paper [26] by Y Li et al. Instead of classification research, the research paper investigates the characteristics and the distribution of variables of accident statistics so that guidance can be generated depending on the possibility of a crash. An approach for differentiating things recognized from footage of dashcams based on their potential of being involved in fatal crashes is scenario-wise, spatio-temporal attention guidance.

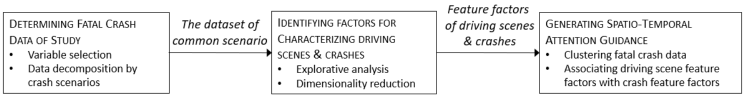

Determining Fatal Crash Data of Study
• Variable selection
• Data decomposition by crash scenarios
The dataset of common scenario
Identifying factors for Characterizing driving scenes & crashes
• Explorative analysis
• Dimensionality reduction
Feature factors of driving scenes & crashes
Generating Spatio-Temporal Attention Guidance
• Clustering fatal crash data
• Associating driving scene feature factors with crash feature factors

Figure 2.6: *Illustration of the proposed method for generating spatio-temporal attention guidance.*

This paper used the fatal crash data of the U.S. retrieved from FARS (2013-2017). 162,104 fatal crashes were recorded during the period, having more than 40 variables. Mainly, four categories (Time-related, location-related, environment-related, special crash types) were used to choose thirteen variables to characterise driving scenes that should be recognised by the vehicle-mounted systems or Computer Vision. Five of those thirteen variables used to evaluate driving scenarios are applicable to a tiny percentage of fatal collisions. The variables are coded in binary where zero carries most value. According to one claim made in this paper, certain scenarios have specific patterns of fatal crashes that are more distinct than the general trend. By classifying fatal crashes by their road type and road configuration, the dataset was significantly lowered from 128,149 cases to 184 spatially-specified groups. This notion is supported by the exploratory analysis and this work devised a strategy for merging such groupings into bigger groups. Each of the larger groupings should have its own attention guidance if they are sufficiently distinct. In a two-step process, the184 particular spatial groups were merged collectively, the 68 groups with a size more than 100 were clustered using a down-to-top hierarchical clustering algorithm in the first stage. The second round of clustering analysis centred on the remaining 22,582 fatal crashes in the 158 single-group groups, as well as the 116 groups of no more than 100 and the 42 groups remaining from the first round. Association rule mining was used to acquire the clusters' spatio-temporal attention guidance. The guidelines specify exactly what sort of tragedy is most likely to happen, as well as how probable it is that a deadly crash will take place at a given location and time. If the results of the assessment of driving scenes are combined with attention guidance, they have the ability to improve drivers' awareness of items that require extra attention for safety reasons.

In this paper [23] , they present a way for simulating both bottom-up and top-down visual attention mechanisms in a dashcam observation environment, allowing the suggested stochastic multi-task agent's decision to be visually explained by attentive regions. Using a deep reinforcement learning (DRL) method, this model concurrently learns accident anticipation and fixation prediction rules. The agent performs actions at each time step to anticipate the likelihood of a future collision, as well as the fixation point. DADA-2000 and DAD, two traffic accident datasets, are used to test this technique.

Taking video as an input from the dashcam, the stochastic multi-tasking agent outputs the accident score and the next fixation at each time step. These outputs are created based on the environment's observation status. They suggest a dense anticipation reward and a sparse fixation reward as scalar rewards from a driving environment to steer the agent's learning.

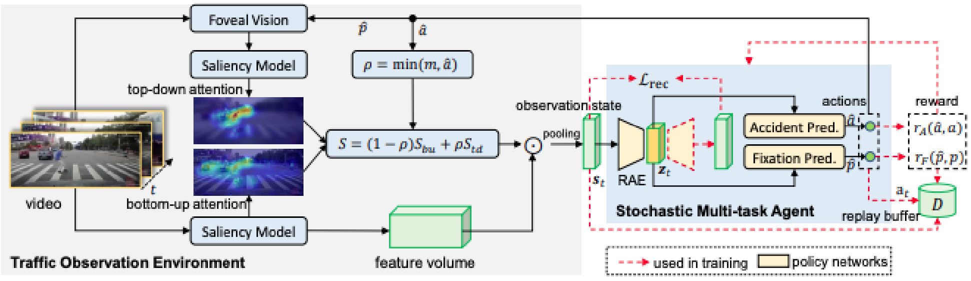


Figure 2.7: *DRIVE Model*

2.2 Algorithms and Models

2.2.1 Recurrent Neural Network (RNN)

An effective method for learning sequentially through space and time is the recurrent neural network (RNN). The inputs in RNNs are recurrently given from the outputs of the previous steps. A sequence is developed based on the connections between the nodes which basically shows a temporal dynamic behaviour. Therefore, capability to retain memory and learn data sequences can be achievable by RNNs [8]. RNN does not provide efficient performance as the gap length rises due to having issues with vanishing gradients [3], making it challenging to train an RNN to recognize long-term relationships.

2.2.2 CNN Architectures for Feature Extraction

A neural network called Convolutional Neural Network retrieves image features from input. A CNN is made up of a classifier network and a feature extraction network. During training, the weights of both networks are established. Instead of doing it manually, CNN utilises a feature extractor in the training phase. Then the neural network uses the extracted feature signals for classification. The feature extractor used by CNN is made up of unique neural network types, the weights of which are determined during training. The output of the neural network classification is then created based on the image features. Convolution layer piles and sets of pooling layers are included in the neural network for feature extraction. A few examples of convolutional neural networks are the well-known AlexNet, GoogLeNet, ResNet, and VGG models.

Chapter 3

Methodology

Accident anticipation is a subfield of action anticipation in deep learning. The purpose of the proposed model for predicting car accidents in the field of deep neural networks is to give a probability of the occurrence of crash as early as possible. In order to do so, data regarding various road scenes with and without accidents are needed to be extracted and preprocessed. From the extracted frames, multiple objects are detected at frame-level and object-level through a feature extractor to reduce the amount of the redundancy in the dataset. The input to our optimised neural network model will be the concatenation of the weighted aggregation of object-level and frame-level features. To train our model, we use the existing datasets introduced in our mentioned research works.

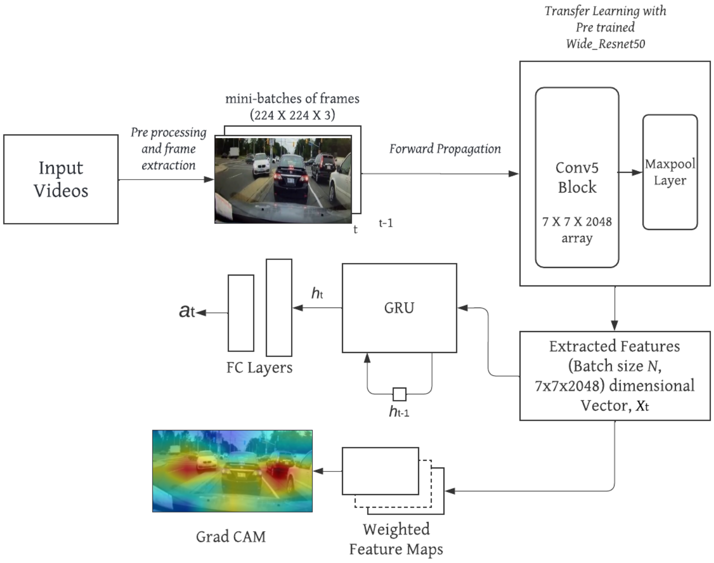


Figure 3.1: *Overview of our CNN-RNN network*

3.1 Dataset

For our model we will be using the Dashcam Accident Dataset (DAD) dataset which is a collection of 20fps videos shots. This dataset has 1,284 training videos (455 accident positives and 829 accident negatives) and 466 testing videos (165 accident positives and 301 accident negatives). Each video in this collection is five seconds long, allowing for about one-hundred frames each video. The primary reason to go for this particular dataset is the inability of other models to produce a favourable and efficient result on this dataset. In comparison to the CCD dataset, the DAD dataset has a much lower rate in action anticipation in the previous state-of-the-art models such as DSTA [30]. In the accident positive videos of DAD, the accidents generally occur in the last 0.5 seconds, the network trains on a few indicators. The DSTA obtains a low Average Precision (nearly 36%) score if it is trained and tested alone on CCD or DAD dataset. Hence, early accident prediction becomes challenging.

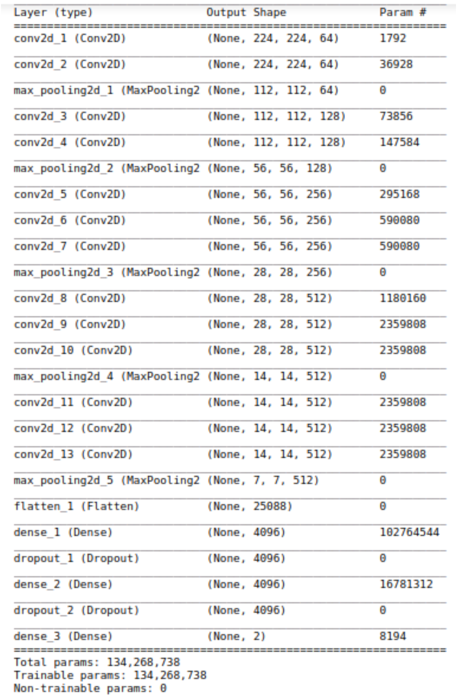
3.2 Feature extraction in image processing

Clearly discernible cues of moving things that might be implicated in an accident can be seen in observations of spatially specified objects. Intuitively, appearance and motion cues are crucial for anticipating accidents. Our approach starts by initially identifying objects in individual video frames using various convolution neural net (CNN) architectures to evaluate the performance of various architectures and choose the best one.

3.2.1 Feature Extraction with VGG16

When categorising 1000 photos into 1000 different categories, the object detection and classification technique based on CNN architecture VGG16 has an accuracy rate of 92.7% [20]. With transfer learning, it is a popular technique for identifying photos and is straightforward to use. For each frame, we extract a fixed 4096-dimension feature using a pre-trained VGG16 [27] network at 20 frames per second in order to extract single-frame-based cues. After detecting the spatially distributed objects of each clip of video frames, the ones with the highest detection score are passed to the VGG-16 convolution neural net for feature extraction and aggregation. The network's input is an image of dimensions (224, 224, 3). The VGG-16 features 4096 dimensions. The dimension was reduced to 512 by passing these features through fully connected embedding layers.
The features we extracted using the pre-trained ImageNet VGG-16 model are:

- det: Shape of bounding boxes detected (50, 19, 6), last dimension denotes (x1, y1, x2, y2, prob, cls)
- labels : labels to determine whether or not the video involves an accident. If accident exists (positive), it shows [0, 1] and if accident does not exist (negative), it shows [1, 0]
- ID: video name
- data: shape of 4096-dimensional extracted features is (50, 20, 4096). It contains 19 box-level feature with shape (50, 19, 4096) and frame-level features with shape (50, 1, 4096)



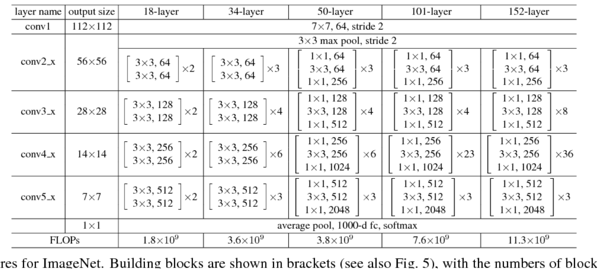
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 224, 224, 64)      1792
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 224, 224, 64)      36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 112, 112, 64)      0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 112, 112, 128)     73856
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 112, 112, 128)     147584
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 56, 56, 128)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 56, 56, 256)       295168
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 56, 56, 256)       590080
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 56, 56, 256)       590080
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 28, 28, 256)       0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 28, 28, 512)       1180160
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 28, 28, 512)       2359808
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 512)       0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 7, 7, 512)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              102764544
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_2 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dropout_2 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 8194
=================================================================
Total params: 134,268,738
Trainable params: 134,268,738
Non-trainable params: 0
```

Figure 3.2: *The summary of the VGG-16 model*

The dimension of both object features and frame-level features are reduced using fully connected layers yielding lower-dimensional object features. The VGG-16 neural network model will be used to create various labels for categorising images of driving scenes and to develop a scene analysis system for the driving scenarios using vision sensors. Basically it is used as a CNN feature extractor, passed through fully connected layers.
The frames with the highest detection score are fed to the VGG16 for feature extraction which gives us features of both frame-level and object level.

3.2.2 Feature Extraction with ResNet50

The ResNet50 network, which was launched using parameters pre-trained on ImageNet [12], is another base feature extractor utilised to compare in this study. Eight times deeper than VGG nets but yet less complicated, residual nets are evaluated on the ImageNet dataset with a maximum depth level of 152 layers. An aggregation of these residual nets results in an error of 3.57% on the ImageNet test set. The model is improved using the DAD dataset after it has been initialised by the ImageNet classification models. ResNet50 or a framework for residual learning makes it simpler to train networks that are much deeper than those previously employed. A feature map,$A \in R^{K \times U \times V}$, was extracted using the applied ResNet50. In other words, the feature map includes K channels, each of which has a height U, and width V. This feature map becomes a D-dimensional feature vector, $x_t \in R^D$, when it is flattened by a dense layer. The bottleneck blocks in this model necessitate downsampling. The pre-trained ResNet50 model anticipates small batches of normalised input images having 3-channel RGB of dimension (3 x H x W), where 224 is the value of H and W. After being loaded in the range between [0, 1] the images need to be normalised of mean values of [0.485, 0.456, 0.406] and standard values of [0.229, 0.224, 0.225]. The weights are set up initially as in [7] and training of models with a learning rate starts at 0.1 which is divided by 10 once the error has plateaued. ResNet features have no hidden fc layers, in contrast to the VGG-16 employed in [4].

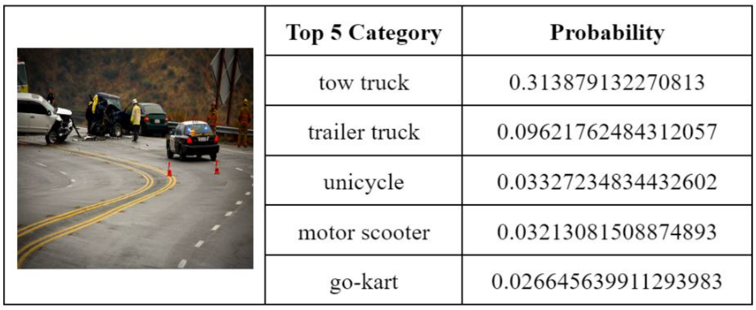


layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

:ures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of block

Figure 3.3: *Architecture of ResNet50 on ImageNet features*

On different recognition challenges, ResNet50 performs well in terms of generalisation. To extract ResNet-50 features, a pre-trained model first detects potential items. Here, the advantages of replacing VGG-16 are what we're interested in. Since all models' detection implementations are identical, the improvements can only be attributed to improved networks. The features below are identified for a single frame of an accident scene.



Top 5 Category	Probability
tow truck	0.313879132270813
trailer truck	0.09621762484312057
unicycle	0.03327234834432602
motor scooter	0.03213081508874893
go-kart	0.026645639911293983

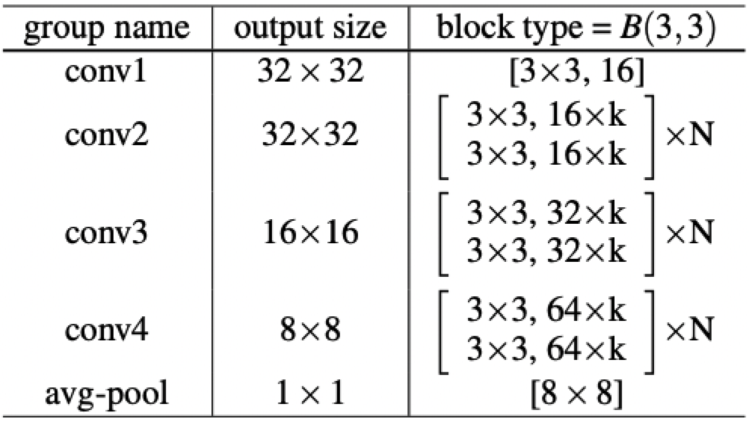
Figure 3.4: *Architecture of ResNet50 on ImageNet features*

The observation above is from a frame of an accident scene that shows the likelihood of each aspect.

3.2.3 Feature Extraction with Wide_ResNet50_2

Apart from the bottleneck number of channels, which is doubled for each block, the wide residual architecture is identical to the usual deep residual network as in ResNet. In outer 1x1 convolutions, the number of channels is equal, but the final block of deep residual network or the ResNet-50 contains 2048-512-2048 channels, on the other hand Wide ResNet-50-2 has 2048-1024-2048. ResNets are modified to create Wide Residual Networks, which widen the residual networks while reducing their depth. Dropout is applied between convolutional layers to regularise training and prevent overfitting because widening residual blocks leads to an increase in the number of parameters. Wide residual network experiments reveal that approach produces constant gains and even cutting-edge new findings.

Factor k [13] (corresponding to k = 1) determines network breadth. The first layers of the convolution groups conv3 and conv4 conduct downsampling, and the groups'



group name	output size	block type = $B(3,3)$
conv1	32 × 32	[3×3, 16]
conv2	32×32	$\begin{bmatrix}3\times3, 16\times k\\3\times3, 16\times k\end{bmatrix}\times N$
conv3	16×16	$\begin{bmatrix}3\times3, 32\times k\\3\times3, 32\times k\end{bmatrix}\times N$
conv4	8×8	$\begin{bmatrix}3\times3, 64\times k\\3\times3, 64\times k\end{bmatrix}\times N$
avg-pool	1 × 1	[8 × 8]

Figure 3.5: *Structure of wide residual networks. For clearance, the final classification layer is skipped.*

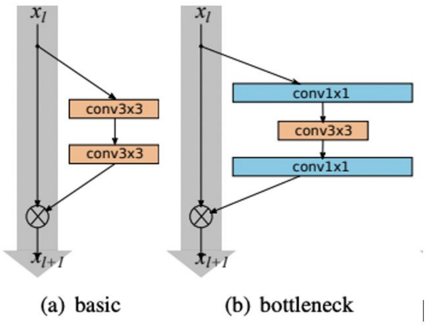
convolutions are presented in parentheses with the number N indicating how many blocks exist in each block.

We have implemented pretrained Wide_ResNet50_2 based on Torch. Identity mappings in residual blocks that train very deep networks were presented in recent follow-up research that looked at the order of activations in [13] residual networks. The following equation can be used to represent the identity map of a block of residual:

$$\mathrm{r}_{l+1} = r_l + G(r_l, W_l)$$

In this, W_l are the block's parameters, G is a function of residual, and r_{l+1} is the input and r_l is the output in the network's l-th unit. The wider residual network is made up of more filters in each residual block that are piled in order. Residual networks in [13] included two types of blocks:

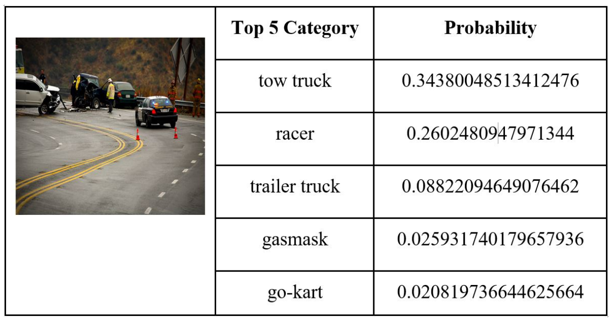
- basic: utilising two subsequent 3×3 convolutions, batch normalisation, and ReLU before the convolution: conv3×3-conv3×3.
- bottleneck: having a single 3×3 convolution surrounded by 1×1 convolution layers that increase and decrease dimensionality: conv1 × 1-conv3 × 3-conv1 × 1.



(a) basic (b) bottleneck

Figure 3.6: *Different Residual Blocks*

Here we implemented the pre-trained Wide_ResNet50_2 using the weights trained on imagenet for the same frame of accident scene that was tested for ResNet50. Here, the prediction was more accurate and included more scene-related information.



Top 5 Category	Probability
tow truck	0.34380048513412476
racer	0.2602480947971344
trailer truck	0.08822094649076462
gasmask	0.025931740179657936
go-kart	0.020819736644625664

Figure 3.7: *Extracted features of an input frame with accident using Wide_ResNet50_2*

3.2.4 Selection of Feature Extractor

For our study, we utilised Wide_ResNet50_2 as our feature extractor. Due to the gradient signal either inflating (being very huge) or vanishing (becoming very low) as it gets back-propagated through numerous layers, training extremely deep neural networks is challenging. We combine the wide residual network with the Gated Recurrent Unit method to achieve the optimal outcome.

- It is not worthwhile to use VGG16. Due to the approximately 138 million trainable parameters, it is computationally expensive and takes a long time to train. The wide ResNet model, on the other hand, has almost the same accuracy and just 68 million trainable parameters.

- Deep residual networks have proven to be scalable to thousands of layers and can still obtain better results. Regrettably, because feature reuse is diminishing, exceedingly deep residual networks are challenging to train because each fraction increase in accuracy necessitates about doubling the number of layers. With a view to resolving these challenges, wide residual network architecture is used, where the depth of the residual networks is reduced but the width is expanded. Compared to their more often utilised thin and very deep equivalents, these networks are far better. For example, all previous deep residual networks, up to and including 1,000-layer deep networks, are outperformed even by a basic 16-layer-deep wide residual network in both precision and performance, attaining cutting-edge performance on our SA (Street Accident) or Dashcam Accident dataset (DAD).

- The number of layers makes up a neural network's "depth," but its "width" is often the number of neurons per layer or, in the case of convolutional layers, the number of feature maps per layer. Wide resnets are just resnets with additional feature mappings in their convolutional layers. More distinct traits can be learned from a wider layer. However, a wider layer will have more parameters that need to be tuned and will be more prone to overfitting, which is why dropout is applied between convolutional layers to regularise training and prevent from overfitting and diminishing feature reuse.

Since the GPU is far more effective at doing parallel computations on huge tensors, widening the layers is more computationally efficient than having thousands of little kernels. Thus, transfer learning can further effectively make use of the features extracted from Wide_ResNet50_2 network architecture by having an optimal number to widening factor ratio of ResNet blocks.

3.3 Accident Prediction

A deep neural network must be used to learn the spatio-temporal correlations between the visual components of accidents in the video series in order to properly predict a traffic accident before it happens. Patterns associated with diverse situations can be recognized by creating reliable prediction models capable of automatically separating distinct accidental instances [2].

3.3.1 Convolution Neural Network

A multi-layer artificial neural network architecture, convolutional neural network (CNN) is specifically made to process pixel input and is used in pattern recognition systems. The essential modules for a convolution neural network's feature extraction function are the convolution layer in which the DAD dataset on street accidents is provided having numerous records of scenes as frames and the pool sampling layer. To remove redundant fields that are not important in the frames, the convolution layer adjusts the input and implements filtration. In the pooling layer, after eliminating any irrelevant information, the input data set was passed to the new layer by determining the convolutional layer's local sensitivity and secondary feature extraction. The weights for the extracted features are allotted randomly and then the fully connected layer moves these extracted features. The back propagation system of the network effectively reaches the given threshold value based on the error computation. By training iteratively for an optimised number of epochs and suppressing the loss function to update the network's weight parameters, the network model can improve the performance of the network. When CNN is chosen, GRU will not be used after the feature extraction.

3.3.2 Long Short-Term Memory (LSTM)

The LSTM is a recurrent neural network model that can learn order dependence for problems with predictions in sequence. It recognized the issue of RNN long-term dependence, that occurs when the RNN fails to forecast the recorded data in long-term memory yet makes reasonable predictions based on the most recent data. It creates a dedicated memory storage system that uses a back propagation based on time mechanism to train the data. By default, LSTM keeps information for a lengthy period of time and can be used for time-series data processing, prediction, and classification.

3.3.3 Gated Recurrent Unit (GRU)

In the disciplines of deep learning and time series prediction, RNN has made significant progress. Additional variations have been created in response to RNN challenges such gradient disappearance and gradient explosion, including the Gated Recurrent Unit (GRU), which decreases the number of gating units in the LSTM model while strengthening network structure. We can create better outcomes thanks to the deep GRU neural network model's efficient learning and precise prediction capabilities. Gated recurrent unit (GRU) networks outperform typical recurrent neural networks (RNNs) by overcoming the difficulties of disappearing and explosion of gradients. The memorising process is handled by a gating technique in this algorithm.

Our proposed GRU concentrates on spatially specific observations related to the presence of vehicles, pedestrians, or other items extracted from Wide Residual CNN based network at each frame in the scene in order to anticipate accidents.

Algorithm 1

```
Notation:
V: total number of videos, indexed by $v$
T: total number of frames in a video v, indexed by $t$
X: total features of a frame
$R_t$ : reset gate
$Z_t$ : update gate
$H_{t-1}$ : hidden state (number of hidden units: $h$ )
$H'_t$ : candidate hidden state at time step $t$
$H_t$ : new hidden state
$W_{xh}, W_{hh}, W_{hr}, W_{hz}, W_{xr}, W_{xz}$: are weight parameters
$b_h, b_r, b_z$: are bias parameters
$\sigma$: sigmoid activation function
$\emptyset$: FC layer
$a_v$ : accident prediction video v

Input: dashcam video, $V$
for $v$ in $V$ do
    for $t_v$ in $T_v$ do
        $X_t \longleftarrow Resnet50$
        GRU:
        $R_t = \sigma(X_tW_{xr} + H_{t-1}W_{hr} + b_r)$
        $Z_t = \sigma(X_tW_{xz} + H_{t-1}W_{hz} + b_z)$
        $H'_t = tanh(X_tW_{xh} + (R_t \odot H_{t-1})W_{hh} + b_h)$
        $H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot H'_t$
        if $Z_t \approx 1$
            $H_{t-1} \longleftarrow output$
        if $Z_t \approx 0$
            $H_t \longleftarrow output$
        $a_v \longleftarrow \emptyset(h_t)$
```

Chapter 4

Implementation

The model was trained and tested by adjusting the Gated Recurrent Neural Network in a virtual environment using PyTorch [26]. Using a memory of 32GB along with Nvidia RTX 3070 ti GPU, training and testing were carried out. The implementation of the model consists of input data preprocessing, extracting features with wide residual convolutional neural network and aggregation of the extracted features to the model training on optimised GRU and testing to predict the occurrence of accident. Pytorch framework is used to access the torch library in the TorchEnv environment. This section also provides the results of the implementation of the feature extraction with wide residual network and accident prediction for the proposed model for accident anticipation in deep learning. Visual Studio is used to apply the TorchEnv environment and train and run the test using part of the proposed input data and to obtain the results.

4.1 Dataset Preparation

4.1.1 Frame extraction

A video is made up of a continuous stream of sequential images. We must first extract the frames from a video before we can work with it. To process a video, we must conduct operations on each frame individually. OpenCV is an open source library that was used to handle video data that generated 20 images per second, and our dataset contains 5 second long clips. As a result, each movie yields approximately 100 frames. We import the video from the destination, execute a loop, and report a success value to ensure the function is reading the frames correctly. It runs until the value is true, producing separate .jpg images that can be used as frames.

4.1.2 Data Preprocessing and Classification

To preprocess the DAD dataset, we resize the frames into 224 x 224 x 3 dimensions. The data extracted is then annotated and classified into Accident and Not Accident classes.

4.1.3 Dataset Splitting

The dataset is split between training and testing halves in a 75:25 ratio.

4.2 Network description

We proceeded to determine the mean and standard deviation (std) of our dataset to normalise it after obtaining the train/test splits. In our study, we used the data's intended mean and standard deviation values, mean: [0.5,0.5,0.5] and std: [0.5,0.5,0.5]. The train data is first loaded from the train folder. Only the training data must be used to compute the mean and standard deviation. By default, this will load images, therefore we pass the ToTensor transform to scale all of the frames from 0-255 to 0-1. To load our data we need to make sure that our images are the same size and normalisation as those used to train the model because we will be utilising the models from torchvision. The network's input is an image of dimensions (224, 224, 3). We create the validation split after loading our data and applying our transforms. Then, replace the validation transforms while performing a recursive copy to prevent this from also affecting the training data transforms.

4.2.1 Wide Residual Networks for feature extraction

We initiate our procedure by first loading wide residual network pre-trained on IMAGENET for each frame that is extracted at 20 frames per second in order to obtain single-frame-based appearance cues provided by our Street Accident dataset with a required input of dimensions (224, 224, 3). We use this pre-trained network as our feature extractor for extracting features, allowing the input frames to propagate through all layers before halting before the fully connected layer and extracting the outputs from the pool layer as our features. A frame can contain N number of objects. All of the popular ResNet variations have pre-trained models available from Torchvision. With its PyTorch models and weights, such as wide_resnet50_2, Wide_ResNet50_2_Weight, trained with a 32 GB RAM Nvidia RTX 3070 ti graphics card, the Wide ResNet is implemented using transfer learning. Before being fed into the RNN network, each mini-batch of 10 training samples goes through the wide residual network. The Wide ResNets are initialised with settings such as a kernel of 3x3, stride of 1, and dilation of 1 padding when the IMAGENET1K V1 weight is specified. This initiation technique makes it possible for models to converge to a global minimum more quickly and effectively. The weights are tuned through mini-batch gradient descent. The video from the dashcam enters the wide residual network backbone as a flow of frames denoted by t. A feature map A_t is extracted from frame t via the feature extractor. The feature map transforms into a feature vector, X_t, of object dictionary as a list of JSON and it is passed to the network of GRU to learn the hidden representation, h_t, of frame t. The feature vector and the labels are then appended to the data list. Features were retrieved from the wide resnet's global average pooling layer, which has a 2048 pixels in size. These features' dimension was reduced using a linear embedding before being sent as an input to the GRU network. In this study, Wide ResNets establish a more direct channel for information to flow throughout the GRU network for feature extraction and localisation in the frames, which ultimately results in higher performance.

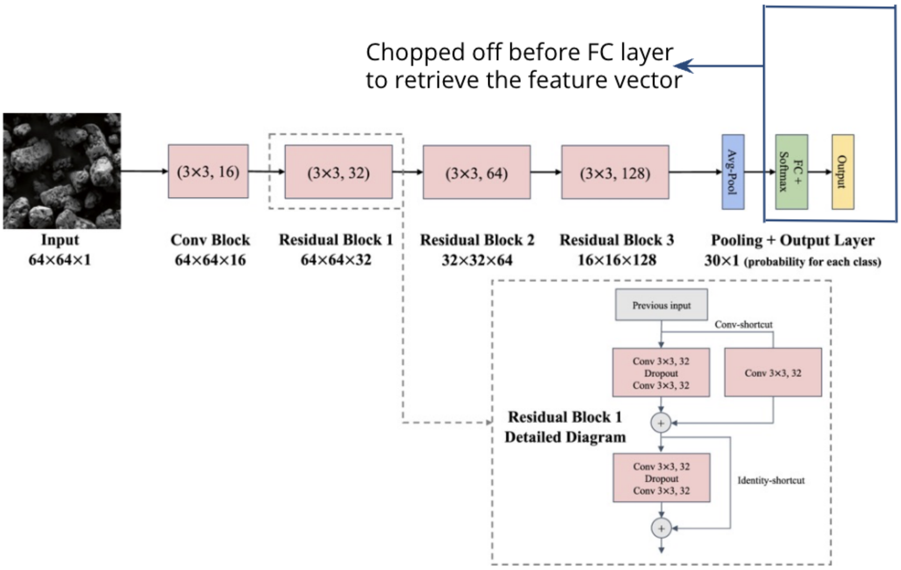


Figure 4.1: *Feature Extraction from specified layer*

4.2.2 Utilising GRU for Spatio-Temporal Relational Learning

The recurrent neural network (RNN) is a critical weapon for sequential training in spatio-temporal dimensions. Important insights in order to prevent collisions can be found in the spatio-temporal interactions between image features.

This work updates every frame's hidden representation d_t, using GRU, a specific form of RNN, to learn spatio-temporal correlations among the features. A reset gate ($f(reset)t$) and an update gate ($f(update)t$) are two gates in GRU that keep the most important information from the video sequence while discarding the rest. Equations which are used to represent the data going through the GRU in mathematics are as follows (1-4):

$$f_t^{(reset)} = \sigma(W_f^{(reset)} x_t + B_f^{(reset)} d'_{t-1}), \tag{4.1}$$

$$reset_t = tanh(W_{reset} x_t + B_{reset}(f_t^{(reset)} \circ d'_{t-1})), \tag{4.2}$$

$$f_t^{(update)} = \sigma(W_f^{(update)} x_t + B_f^{(update)} d'_{t-1}), \tag{4.3}$$

$$d_t = (1 - f_t^{(update)}) \circ reset_t + f_t^{(update)} \circ d'_{t-1'} \tag{4.4}$$

In the above equations, σ signifies the sigmoid activation and $\circ$ represents element-wise product operators. The median pooled hidden representation of the previous N frames is d'_{t-1} :

$$d'_{t-1} = avgpool([d_{t-1'}, d_{t-2'}......., d_{t-n'}]) \tag{4.5}$$

During training, a one-hot encoded vector known as a video-level label y is assigned to each frame. To anticipate the accident probability of each video frame, the spatio-temporal relations of the retrieved features from wide_resnet50 are learned using the GRU. Each frame's feature, X_t, flows into the GRU and the model estimates the probability using the hidden representation, a_t, of whether there is an accident in the video of each frame.
For accident recordings, the exponential loss function is utilised, and it gives frames near to the accident greater weight, resulting in increased anticipation odds ratios for those frames. For the whole video clip, the loss for each frame is summed, averaged, and then back propagated.Through backpropagation of the loss function, this network optimised the parameters. We used GRU with 256 hidden states and a 0.5 dropout. The model was trained with the Adam optimizer using 30 training epochs on the DAD dataset with a learning rate of 0.0001. The batch size for the training was set to 10.

Chapter 5

Result Analysis and Experimental Evaluation

5.1 Evaluation Metrics

5.1.1 Average Precision

Average Precision mainly determines the accuracy of identifying whether an accident has occurred or not from a subject video. The videos would be labelled in binary as positive (1) or negative (0) for accident instances. The goal of this metric is to evaluate how correctly an accident has been identified from a video. At any instantaneous time step t, if the average precision is higher than a determined threshold, then the frame is supposed to be an accident positive frame, else negative. This not only allows us to understand the Average Precision (AP) but also get the precision (positive predictive value) and recall (sensitivity).

5.1.2 Time-to-Accident

Time to accident (TTA) is the most important metric in our case since we want to anticipate accidents earlier in order to be able to avoid it entirely or at least minimise the damage caused by it. It relies on the accident positive predictions to determine the time after which an accident is supposed to happen. For a range of threshold values, multiple TTA results can be obtained along with the corresponding recall rates. To evaluate the accident occurrence earlier than it actually happens, mTTA and TTA@0.8 is used. Here by mTTA we mean the mean Time to accident or the average of the TTA values and the TTA at 80% recall rate is denoted by TTA@0.8. It is to be mentioned that a greater number of false positives predictions can produce considerably greater TTA results while producing a lower AP. It refers to the fact that the model may produce positive predictions for random inputs because of being overfitting on the accident videos. Therefore, to evaluate a proper prediction the TTA with greater AP is to be considered, because it is pointless to obtain high TTA without achieving a high AP.

5.2 Evaluation of Test Results

To assess the model's generalisation potential and make required corrections, it is re-evaluated after each epoch on a new validation set that it has never observed before. The only model kept and used for further analysis is the one with the highest scores on the validation dataset. In an effort to increase both the AP and mTTA during training, the loss function will be back propagated and the parameters will be optimised by the network. We aim for larger AP and longer mTTA spanning several epochs but having high TTA is useless if high AP cannot be ensured, which results in a tradeoff between the accuracy and earliness of prediction. Particularly when it comes at the expense of extremely low precision, a very high mTTA may be impractical.

5.2.1 Performance of feature extractor

The study evaluates the effectiveness of resnet50 and VGG16 as a feature extractor for the proposed model on DAD dataset. As seen from the figure 5.1, using resent50 as an extractor yields better results by obtaining higher AP (68.78%) with mTTA 4.40, than VGG16 (56.98%) with mTTA 4.42.

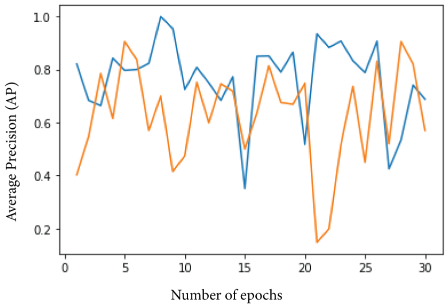


Figure 5.1: *Comparison of AP between model trained using ResNet50 (blue line) and VGG16 (orange line)*

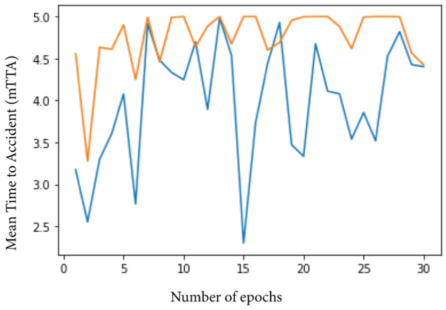


Figure 5.2: *Comparison of mTTA between model trained using ResNet50 (blue line) and VGG16 (orange line)*

However, the model trained on wide_resnet50 is anticipating on average 4.11 seconds earlier before an accident occurs, side by side keeping the competitive AP performance at 78.54% in comparison to models trained using resnet50 and VGG16. Since it is useless to acquire high TTA if high AP cannot be ensured, we primarily report TTA metrics when the highest AP is attained in table 5.1 .

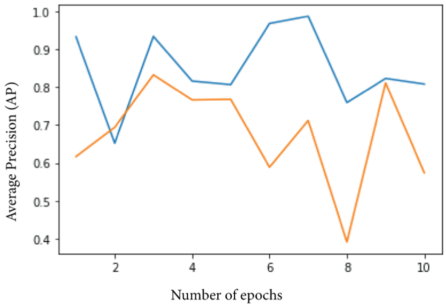


Figure 5.3: *Comparison of AP between model trained using ResNet50 (orange line) and Wide_ResNet50 (blue line)*

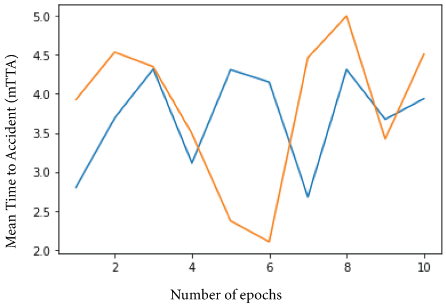


Figure 5.4: *Comparison of mTTA between model trained using ResNet50 (blue line) and Wide_ResNet50 (orange line)*

Table 5.1: TABLE 1

FEATURE EXTRACTOR COMPARISON ON DAD DATASET

Feature Extractor	mTTA(s)	AP(%)
VGG16	4.42	56.98
Resnet50	4.40	68.78
Wide_Resnet50	4.11	78.54

This study demonstrates that, wide residual network, rather than both VGG and resnet, is a preferred option given all these feature extraction models based on convolutional neural networks because overfitting of the model on the accident frames could result in good predictions for any input. Thus, aside from fair comparison with existing approaches, we primarily report TTA metrics when the highest AP is attained because it is pointless to achieve high TTA if high AP cannot be ensured.

Table 5.2: TABLE 2

TTA COMPARISON WITH HIGHEST AP FOR GRU AND CNN

Feature Extractor	**TTA (GRU)**		**TTA (CNN)**	
	TTA	**AP (%)**	**TTA**	**AP (%)**
VGG16	4.653	56.98	4.759	75.20
ResNet50	4.509	68.78	3.931	70.89
Wide_ResNet50	4.315	78.54	3.954	79.40

5.2.2 Comparative Analysis considering the State-of-the art approaches

Approaches as state-of-the-art [[30]]-[[22]], [[17]] that aim for a longer mTTA are compared to our model. The comparison study's findings are summarised in TABLE 5.1. DSA released in 2016 [8]obtained mTTA of 1.34 seconds and AP of 48.1% on the DAD dataset. Other researchers have gradually raised the value to 53.7% for AP and lengthened 3.66 seconds for mTTA during the last four years [21], [19], and [22]. By leveraging the AP by an additional 22.4% and the mTTA by 0.65 seconds, our suggested network achieves the AP at 78.54% and for mTTA, it obtained 4.11 seconds. Thus, it illustrates that even with a challenging dataset, our network can enhance AP as well as mTTA. On the DAD dataset, the suggested network has surpassed the state-of-the-art efficiency.

Table 5.3: TABLE 3

MODEL COMPARISON ON DAD

Datasets	**Year**	**Methods**	**mTTA(s)**	**AP(%)**
DAD	2016	DSA[10]	1.34	48.1
	2017	L-RAI[15]	3.01	51.4
	2018	adaLEA[17]	3.43	52.3
	2020	GCRNN[22]	3.53	53.7
	2021	DSTA[30]	3.66	56.1
	2022	(ours)	4.11	78.54

5.2.3 Evaluating generated saliency maps using GRAD-CAM

In order to clearly display the accident anticipation possibility the recommended accident anticipation network produced, the saliency maps are additionally overlayed with their corresponding input images. Humans pay attention to cars or pedestrians who could be engaged in or affected by a traffic collision, as shown by the human attention maps figure 5.5. The hottest regions in the Grad-CAM method saliency maps figure 5.6 closely resemble the areas where people focus. This shows that the suggested accident anticipation network successfully generates the forecast by envisioning highly salient values on the traffic agents involving collision or influenced by it.

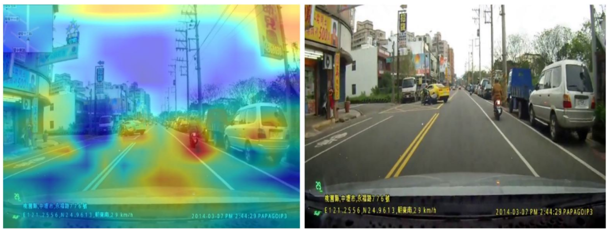


Figure 5.5: *Human attention maps*

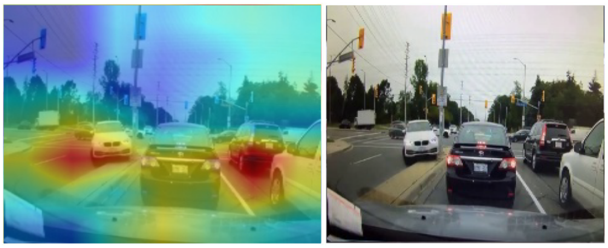


Figure 5.6: *Grad-CAM method saliency maps*

According to the comparison analysis based on figure 5.5 and 5.6 , by emphasising on the areas that are most salient, just like a human would, the accident anticipation network developed in this study accurately predicts a future accident. High quality saliency maps generated by the Grad-CAM and XGradCAM algorithms can be used to describe the decision-making process of the proposed accident anticipation network.

Chapter 6

Conclusion

While working on this subject for a year, we faced different challenges over the time. The first and foremost problem we faced was finding the correct dataset and manipulating it according to our criterias. This included manual labelling of over 1,284 videos of DAD dataset. The second challenge that we faced was running our accident prediction code and training our model with limited computing power initially. This also had an effect on the overall time period of our thesis. Finally, we encountered difficulties including the feature extractors in our code and getting those settled with our model.

To solve the challenge of accident detection and anticipation, researchers have applied computer vision and deep learning techniques and videos from dashboard-mounted cameras fixed in vehicles are included in the dataset. There are two distinct groups for the tasks of detection and anticipation. Accident detection is similar to accident recognition where the network has access to the entire temporal context at test time. Because all practical systems are causal, forecasting an accident requires using just a limited amount of temporal knowledge. This makes accident anticipation a difficult endeavour. The anticipation problem involves how early the network foresees the accident. Therefore, preventing accidents should be approached differently than solving detection issues.

To conclude, demand for guaranteed safety in the self-driving system is progressively increasing. Most importantly, when lives are at stake, potential for death and serious injuries are always a prominent risk. Hence, predicting accidents as early as possible resolves the potential risks of traffic crashes and ensures the safety of the passengers of the vehicles as well as for the pedestrians. Consequently, the necessity arises to emerge efficient systems that can predict traffic accidents with most accuracy at the earliest time. Therefore, in our proposed model we opt to use GRU with Wide Residual Networks by extracting features to classify between crash labels in order to anticipate accidents as early as possible. Efficient systems for accident anticipation can not only reduce injury, death or loss of property, but also improve the road network safety management.

Bibliography

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[2] S. Shanthi and R. G. Ramani, "Gender specific classification of road accident patterns through data mining techniques," in *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012)*, IEEE, 2012, pp. 359–365.

[3] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, PMLR, 2013, pp. 1310–1318.

[4] K. Cho, "Van merriã «nboer b, gulcehre c, bahdanau d, bougares f, schwenk h, bengio y," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[5] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, IEEE, 2014, pp. 458–464.

[6] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[8] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," *arXiv preprint arXiv:1611.01576*, 2016.

[9] C. Cadena, L. Carlone, H. Carrillo, *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[10] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Asian Conference on Computer Vision*, Springer, 2016, pp. 136–153.

[11] M. Cordts, M. Omran, S. Ramos, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, Springer, 2016, pp. 630–645.

[14] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[15] K.-H. Zeng, S.-H. Chou, F.-H. Chan, J. Carlos Niebles, and M. Sun, "Agent-centric risk assessment: Accident anticipation and risky region localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2222–2230.

[16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[17] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident db," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3521–3529.

[18] L. Chen, W. Zhan, W. Tian, Y. He, and Q. Zou, "Deep integration: A multi-label architecture for road scene recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4883–4898, 2019.

[19] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-level semantic feature detection: A new perspective for pedestrian detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5187–5196.

[20] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.

[21] J. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, "Proceedings of the ieee/cvf international conference on computer vision," 2019.

[22] W. Bao, Q. Yu, and Y. Kong, "Uncertainty-based traffic accident anticipation with spatio-temporal relational learning," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2682–2690.

[23] W. Bao, Q. Yu, and Y. Kong, "Drive: Deep reinforced accident anticipation with visual explanation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7619–7628.

[24] M. Fatima, M. U. K. Khan, and C.-M. Kyung, "Global feature aggregation for accident anticipation," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 2809–2816.

[25] M. M. Karim, Y. Li, R. Qin, and Z. Yin, "A system of vision sensor based deep neural networks for complex driving scene analysis in support of crash risk assessment and prevention," *arXiv preprint arXiv:2106.10319*, 2021.

[26] Y. Li, M. M. Karim, R. Qin, Z. Sun, Z. Wang, and Z. Yin, "Crash report data analysis for creating scenario-wise, spatio-temporal attention guidance to support computer vision-based perception of fatal crash risks," *Accident Analysis & Prevention*, vol. 151, p. 105 962, 2021.

[27] P. Naveen and B. Diwan, “Pre-trained vgg-16 with cnn architecture to classify x-rays images into normal or pneumonia,” in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, IEEE, 2021, pp. 102–105.

[28] K. Wiggers, *Waymo’s driverless cars were involved in 18 accidents over 20 months, 2020*, 2021.

[29] M. M. Karim, Y. Li, and R. Qin, “Toward explainable artificial intelligence for early anticipation of traffic accidents,” *Transportation research record*, vol. 2676, no. 6, pp. 743–755, 2022.

[30] M. M. Karim, Y. Li, R. Qin, and Z. Yin, “A dynamic spatial-temporal attention network for early anticipation of traffic accidents,” *IEEE Transactions on Intelligent Transportation Systems*, 2022.