REAL-TIME ACCIDENT DETECTION USING VIT-B/32

A PROJECT REPORT

Submitted by

SAKTHI PRIYA M (TKM20MCA-2033)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report REAL-TIME ACCIDENT DETECTION USING VIT-B/32, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof.NATHEERA BEEVI M . This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kollam

Date: 22-07-22

SAKTHI PRIYA M

DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

CERTIFICATE

This is to certify that, the project report entitled **"REAL-TIME ACCIDENT DETECTION USING VIT-B/32"** is submitted by **SAKTHI PRIYA M (TKM20MCA-2033)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications,is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor   Head of the Department   External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of our project.

I am extremely grateful to **Dr.Fousia M Shamsudeen**, Head of the Department, for providing us with best facilities.

I would like to thank our coordinator, **Prof. Vaheetha Salam** and project guide **Prof. Natheera Beevi M**, Department of Computer Applications, who motivated me throughout the project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout our course of study.

I owe my thanks to our friends and all others who have directly or indirectly helped us in the successful completion of this project.

SAKTHI PRIYA M

ABSTRACT

One of the most unfortunate risks in today's busy society is traffic accidents. Each year, traffic accidents cause a large number of casualties, illnesses, and deaths in addition to suffering huge financial losses. Given the quick growth of embedded surveillance video systems for tracking traffic accidents, it is necessary to distribute systems with high detection accuracy and speed. Recent advancements in vision-based accident detection methods have been extremely successful thanks to deep convolutional neural networks' potent capabilities. The preferred architecture for computer vision tasks has long been CNNs. However, current CNN-based approaches ignore any information and treat accidental classification of all image pixels as equal. As a result, this may result in a low accuracy rate and detection delays.

This study uses a Vision Transformer-based accident detection method in place of CNN to improve detection speed and achieve high accuracy. Transformers deal with images as a series of patches as opposed to convolutional networks, selectively focusing on various visual components according to context. Additionally, the transformer's attention mechanism addresses the issue with low probability, enabling early accident identification. In this project, traffic accidents were found utilizing video footage and the Vision Transformer (VIT-B/32) transformer. For the accident root analysis, additional roadside actions are also categorized. On the publicly accessible dataset, Vision Transformer achieves a classification accuracy of about 92%. The model is a video-based accident detection coupled with sms service to deliver notifications to the appropriate authorities.

Contents

1 INTRODUCTION **1**

1.1 Problem Definition 3

1.2 Objective 3

2 LITERATURE SURVEY **4**

2.1 Purpose of the Literature Review 4

2.2 Related Works 5

3 METHODOLOGY **8**

3.1 Proposed system 9

3.2 System Architecture 10

3.2.1 Dataset 11

3.2.2 Video to frame conversion 11

3.2.3 Converting image to patch segments 12

3.2.4 Training Accident Detection Model Using VIT-B/32 12

3.2.5 Testing 12

3.2.6 Deploying the model in Flask API 12

3.2.7 Save the VIT-B/32 model 12

3.3 VIT B/32 TRANSFORMER 12

3.3.1 Training model using the VIT-B/32 14

3.4 CLIP 16

3.5 Flask API 17

3.6 Software Requirement and Specification 18

3.6.1 Python 19
3.6.2 Spyder 21
3.6.3 HTML 22
3.6.4 CSS 22
3.6.5 JavaScript 23
3.6.6 PyTorch 23
3.6.7 OpenCV 24

4 RESULTS AND DISCUSSIONS 25

4.1 Model Evaluation 25
4.1.1 Training Result 25
4.1.2 Accuracy 26
4.1.3 Error rate 26
4.2 User Interface 27
4.2.1 Graphical User Interface 27

5 CONCLUSION 30

5.1 Advantages 30
5.2 Future Enhancement 31

REFERENCES 32

APPENDIX 34

A Screenshots 34

List of Figures

3.1 Block diagram of the proposed system 9

3.2 Block diagram of web portal 10

3.3 Dataset 11

3.4 Vision Transformer Architecture 14

3.5 CLIP Image Processing 16

4.1 Training the model 26

4.2 User Interface 27

4.3 Cars Detection 28

4.4 Accident Detection 28

4.5 Sms Alert 29

A.1 Video Upload 34

A.2 Detection Result 35

A.3 Detection 35

Chapter 1

INTRODUCTION

Numerous automobile accidents around the world resulted in more than a million fatalities and numerous minor injuries. Even though these nations only account for half of the world's automobiles, numerous studies have indicated that many developing and impoverished countries have the highest incidence of fatal road accidents. According to data available, there are 140,000 deaths annually in India, or an average of 13 each hour. The key objective is to make it possible for the system to identify accidents based on the video sequence that the camera transmits. a device that, by spotting the accident early and alerting the authorities afterward, assists accident victims who need it. There are several factors that might lead to road accidents. The geometry of the road, the local environment, drunk driving, and speeding are some of the most frequent elements that raise the likelihood of them happening. Even though the majority of them only result in material damage, each of these events has an impact on people's quality of life by reducing both personal safety and traffic mobility.

Video cameras are now a tool for managing and regulating traffic in urban areas, thanks to technological advancements. They enable analysis and observation of the city's traffic patterns. The number of cameras required to carry out these jobs has, however, grown dramatically over time, making control challenging if automation techniques are not applied because more specialists are required to adhere to all the points. The control and follow-up process can be automated using a number of different methods. A system using traffic-based video camera surveillance is an illustration of this. These can be utilised to estimate the speeds and trajectories of items of interest in order to predict and prevent traffic accidents in the vicinity.

Different methods to identify traffic accidents have been given by the scientific community. These consist of machine learning, deep learning, sensor data, social network data analysis, and statistics-based techniques. These modern methods have led to advancements in a number of scientific domains, including video-based problem resolution (video processing). In order to find a method for the identification and categorization of traffic accidents using video, it is crucial to explore these strategies.

Since the development of convolutional layers in the field of neural networks, the resolution of problems involving digital image processing has improved. Deep learning algorithms, particularly for image understanding and analysis, have demonstrated outstanding performance in a wide range of applications. These layers take advantage of the spatial relationship that the input data have, which dense neural networks cannot do because of the amount of the data. However, the success of CNNs is at the expense of limiting the computation to leveraging geographically constrained data. The performance of CNNs has gradually reached saturation, and the computer vision research community has begun looking towards alternative designs.

While using fewer computer resources for pre-training, Vision Transformer (ViT) outperforms convolutional neural networks (CNN) in terms of performance. When training on fewer datasets, Vision Transformer (ViT) shows a generally lesser inductive bias compared to convolutional neural networks (CNN), which increases reliance on model regularisation or data augmentation (AugReg). Similar to the sequence of word embeddings employed by transformers when converting text to text, the ViT model depicts an input image as a collection of image patches and predicts the image's class labels. When sufficiently trained, ViT outperforms a similar state-of-the-art CNN despite requiring four times fewer computer resources. In contrast to ViT, CNN divides the photos into visual tokens. By dividing a picture into fixed-size patches, accurately embedding each one, and including positional embedding as an input to the transformer encoder, the visual transformer can change an image. Additionally, ViT models surpass CNNs in terms of accuracy and computing efficiency by roughly four times.

1.1 Problem Definition

One of the greatest problems with traffic management is the number of vehicle accidents on major roads and highways. Accidents must be reported as soon as they happen so that appropriate action can be taken quickly. Automatic accident detection assists in restoring regular traffic, and if additional medical aid is required, it can be quickly requested. The proposed methodology for real-time single-vehicle accident identification examines each vehicle's motion and using heuristics to determine whether the movement pattern is consistent with single-vehicle accidents. This model uses visual transformer in order to detect accident in real time with high accuracy and efficiency.

1.2 Objective

The project's major purpose is to:

- Use the video data to identify the traffic accident
- Report an accident to the appropriate authority so that it can be addressed
- Deploy a model with higher accuracy and lower error rate
- Reduce human interventions
- Identify the social activities happened in the accident spot

Chapter 2

LITERATURE SURVEY

A literature review is a synopsis of previous written works on a particular topic. This phrase may be used to describe a full academic document or a particular section of an academic work, such as a book or an essay. It provides a thorough summary of prior research on a subject. The literature review researches scholarly works, journals, and other pertinent sources in a particular field of study. This past study should be mentioned, described, summed up, critically evaluated, and clarified in the review. By recognising the contributions of previous researchers, the literature review reassures the reader that your study was designed with care. When a prior research is referenced, it is presumed that the author has read, evaluated, and integrated it into the present work. By offering a ”landscape,” a literature review provides the reader with a thorough understanding of the subject’s evolution. From this view, the reader can infer that the author has truly included all (or the vast majority) of earlier, important works on the subject into his or her own work.

2.1 Purpose of the Literature Review

1. By choosing high quality research papers or studies that are pertinent, significant, important, and valid and compiling them into a single comprehensive report, it makes it simple for readers to get information on a certain issue.

2. By requiring them to describe, assess, and compare original research in that particular field, it gives researchers starting out in a new field a great place to start.

3. It makes sure that previous work is not repeated by researchers.

4. It can suggest topics to focus on or give hints about the direction that future study should take.

5. It highlights the key findings.

6. It points up gaps, discrepancies, and inconsistencies in the literature.

7. It offers a helpful evaluation of the methods and strategies used by other researchers.

2.2 Related Works

Mubariz et al. [1] used a decision-level combination of machine and deep learning models to assess the seriousness of the incidents. The key elements affecting unintentional severity are distance, temperature, wind chill, humidity, visibility, and wind direction. This study suggests an ensemble of deep learning and machine learning models called RFCNN, which combines Random Forest and Convolutional Neural Network, to predict the severity of traffic accidents. The performance of the model-based approach is compared using a number of basic learner classifiers.

A method for object detection was created by Josh Beal et al. [2] using the Toward Transformer. ViT-FRCNN was modified by the author for object detection. It is a competitive object identification solution that uses a transformer backbone, indicating that there are enough novel architectures that are realistic alternatives to the well-researched CNN backbone for advancement on challenging vision tasks. They also look into enhancements over a conventional detection backbone, such as better performance on photos that are outside of their respective domains, improved performance on large objects, and a reduced reliance on non-maximum suppression. This study demonstrates that neural network-based object detections are facing stiff competition from transformer-based detections.

To pinpoint the primary causes of a road and traffic accident, Sachin et al. [3] carried out an examination of accident data. As a preliminary assignment, the author segmented 11,574 traffic accidents that occurred between 2009 and 2014 on the road network of Dehradun, India, using the K-modes clustering technique.The multiple elements that are connected to the occurrence of an accident are identified using both the whole data set (EDS) and the clusters discovered using the K-modes clustering approach. The results of both the cluster-based analysis and the analysis of the complete data set are then compared. The findings show that k mode clustering and association rule mining work together to provide information that would otherwise be hidden if segmentation hadn't been done before association rules were developed.

In order to assess the seriousness of accidents, Shakil et al. [4] created a machine learning model that accounts for often ignored human factors including drunkenness, drug use, age, and gender. In this research, single-mode and ensemble-mode machine learning (ML) approaches were compared in terms of their predicted accuracy, precision, recall, F1 score, and area under the receiver operator characteristic (AUROC) curve. Road accident severity may be determined in one of two methods in this study's analysis of the issue: (i) binary classification (such as grievous and non-grievous), or (ii) multiclass classification (fatal, serious, minor, and non-injury).

To create a target group for creating the accident countermeasures, Max Cameron et al.[5] studied the accident data. The accident countermeasures are developed using the high risk group, high accident severity groups, and accident involved clusters. Each category, including the vehicle, driver, passenger, environmental elements, and the primary accident factors among them, was recognised by the author. A countermeasure is developed for each category by grouping together all the high risk elements.

To find the high risk factors that cause serious accidents, Mohamed et al. [6] did data analytics on accident injury data. The author used cutting-edge data analytics techniques to forecast injury severity levels and assess their effectiveness. The findings of this research showed that strategies based on trees, like XGBoost, perform better than those based on regression, like ANN.

Mehdizadeh et al. [7] presented a thorough review of the use of data analysis in road safety. The two types of analytics models are (a) explanatory or predictive models, which aim to understand and quantify crash risk, and (b) optimization techniques, which aim to reduce crash risk via route/path selection and rest-break planning. According to their study, utilising open data sources and descriptive analytical techniques may make routing safer (such data summaries, visualisation, and dimension reduction). They also provided code that experts and academics could use to collect and analyse data.

Sharma et al. [8] used support vector machines with different Gaussian kernel functions for crash to figure out what factors were most important in causing accidents. In the paper, neural network and support vector machines were compared to each other. The paper said that SVMs are better at getting things right. But the SVMs method has the same flaws as ANN when it comes to predicting how bad a traffic accident will be.

J.Ma et al. [9] developed the XGBoost-based technique and investigated the association between collision, time, environmental and geographical characteristics, and fatality rate. Compared to existing machine learning approaches, the proposed method has the best modelling performance, as shown by the results. The research identified eight factors that influence road fatalities.

Chapter 3

METHODOLOGY

Road accidents are among the most unfortunate dangers in today's frantic world. Each year, road accidents cause numerous casualties, injuries, and fatalities, as well as substantial economic damages. Given the rapid growth of embedded surveillance video systems for monitoring road accidents, it is necessary to distribute systems with high precision and detection speed. Current CNN-based accident classification systems treat all image pixels as equivalent, disregarding any information. Thus, this can result in a low rate of accuracy and a delay in detection. Transformers, unlike convolutional networks, operate on images as a succession of patches, selectively focusing on various image regions based on context. In addition, the transformer's attention mechanism solves the problem with a low probability, allowing for early accident identification. In this research, the Vision Transformer (VIT-B/32) is used to detect traffic accidents from video footage.

Convolutional neural networks (CNN) are outperformed by Vision Transformer (ViT), while using less computing power during pre-training. The inductive bias of the Vision Transformer (ViT) is often lower than that of convolutional neural networks (CNN), necessitating more model regularisation or data augmentation (AugReg) while training on less samples.

The construction of a text-based task-specific transformer serves as the basis for the ViT visual model. The ViT model encodes an input image as a series of image patches and predicts image class labels directly, similar to the order of word embeddings used when applying transformers to text. When trained on sufficient data, ViT demonstrates outstanding performance, surpassing a similar state-of-the-art CNN with four times fewer processing resources. When paired with NLP models, these transformers have a high success rate for photo recognition applications. While

ViT divides images into visual tokens, CNN employs pixel arrays.The visual transformer divides a picture into patches of a specified size, embeds each patch correctly, and augments the encoder's input with positional embedding. Additionally, ViT models beat CNNs by nearly a factor of four in terms of computation accuracy and efficiency.

The proposed system sends text messages to the authority using the Twilio sms API service. The Twilio APIs power its communications platform (Application Programming Interfaces). The software layer that links and optimises global communications networks is hidden behind these APIs, enabling your users to call and contact anybody in the globe. The SMS API provided by Twilio is a versatile building element that may take you from sending your first text message to sending and receiving millions.

3.1 Proposed system

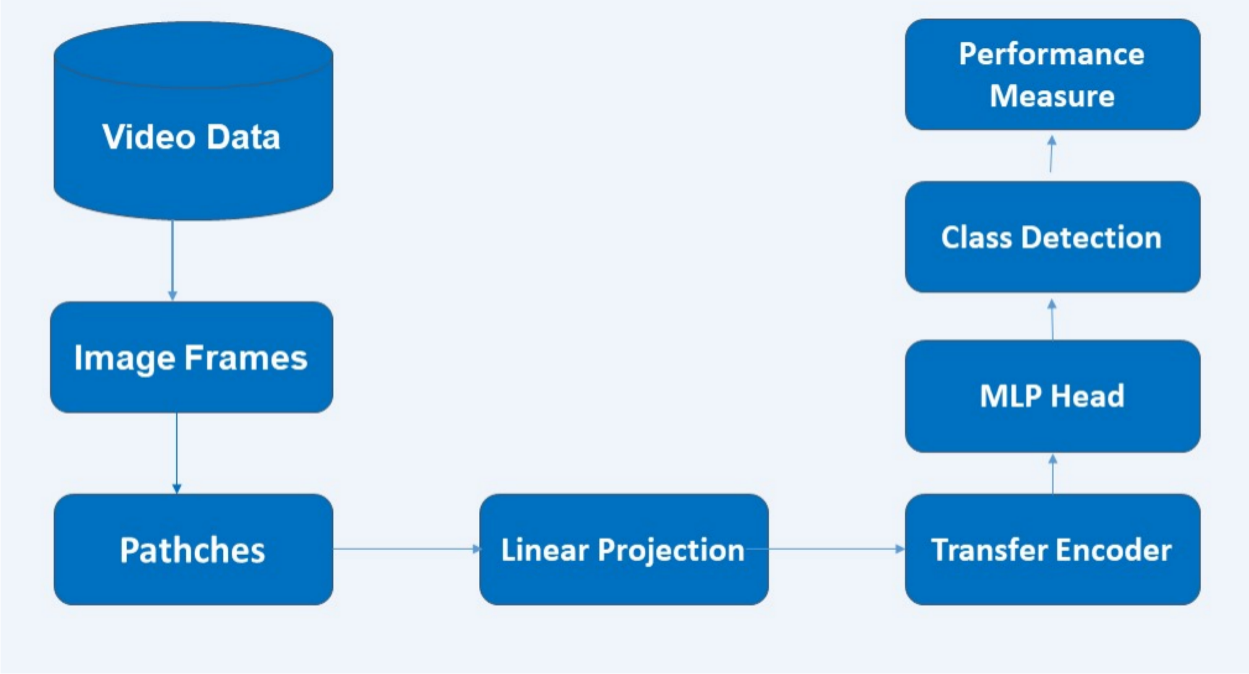


Figure 3.1: Block diagram of the proposed system

3.2 System Architecture

This project identifies the road accident and the surrounding activities. The flow of the project is shown in figure 3.1. The proposed system consist of five major phases:

1. Converting the video into frames
2. Dividing the image into the patch segments
3. Input the patches into the VIT-B/32 Transformer
4. Model creation and evaluate the model
5. Developing intelligent web portal for accident detection

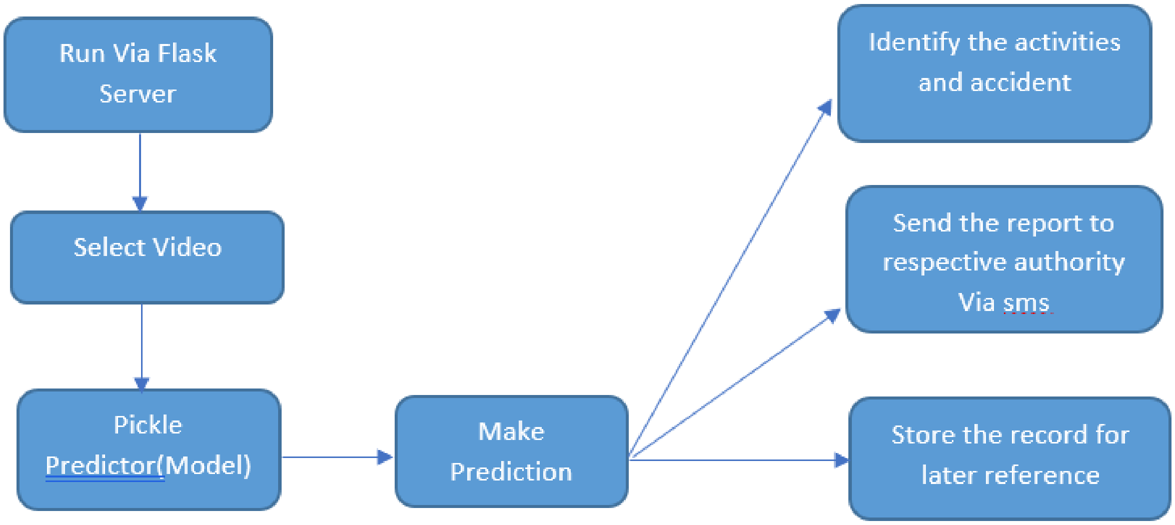

Run Via Flask Server
Select Video
Pickle Predictor(Model)
Make Prediction
Identify the activities and accident
Send the report to respective authority Via sms
Store the record for later reference

Figure 3.2: Block diagram of web portal

The steps involved in the intelligence web portal is described in the figure 3.2.

3.2.1 Dataset

This project implemented using the Accident images dataset from the github [13]. This dataset consists of 10480 images organised into three folders titled Accident –Detection, Vehicles-in-Accidents, Accident-Severity, and Social activities that often occur on the side of the road. The dataset generally contains accident detection, severity and vehicles classes. Some other labels also derived from the images to encounter the social surrounding activities that happens. For example, fight on a street, fire on a street, street violence, person walking cross to the vehicles.. etc.

The figure 3.3 describes about the predictive attributes that are available in the accident data and its count and description.

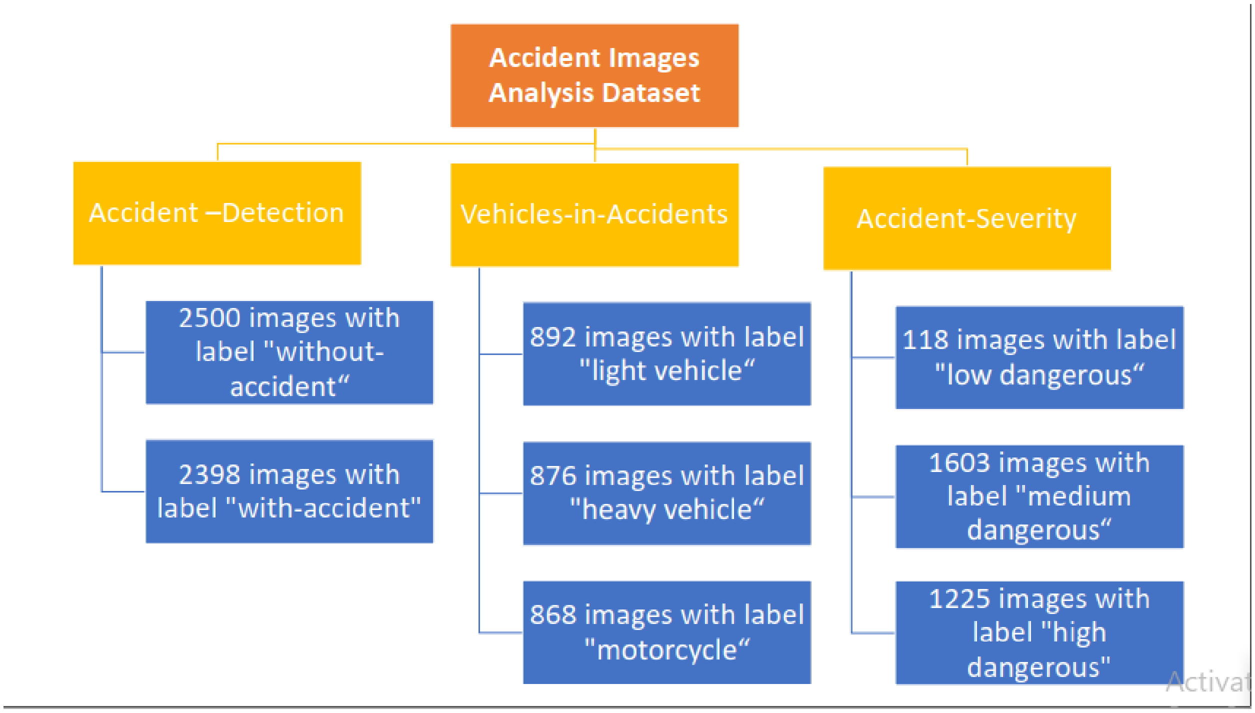

Accident Images Analysis Dataset
Accident –Detection
Vehicles-in-Accidents
Accident-Severity
2500 images with label "without-accident“
2398 images with label "with-accident"
892 images with label "light vehicle“
876 images with label "heavy vehicle“
868 images with label "motorcycle“
118 images with label "low dangerous“
1603 images with label "medium dangerous“
1225 images with label "high dangerous"
Activat

Figure 3.3: Dataset

3.2.2 Video to frame conversion

The input video are divided into the set of frames by using the opencv library. It reads all the video and convert each segments into the frames to detect the scenario that is happening in each second.

3.2.3 Converting image to patch segments

A patch is a small picture with specific characteristics. Finding the patch/template in a picture is the aim of template matching.

3.2.4 Training Accident Detection Model Using VIT-B/32

One of the most used transformers in the deep learning space is the vision transformer. Convolutional neural networks had to be used in computer vision before the invention of the vision transformers for challenging tasks.

3.2.5 Testing

The preprocessed data is splitted into two set. 70% of the data considered as the training data and the rest 30% used to evaluate the model. In the testing phase the model evaluated with the testing data as well as with the unknown data.

3.2.6 Deploying the model in Flask API

Here, we developed a basic API using Flask where user can upload video footages which is to be detected and the system road accident and surrounding activities.

3.2.7 Save the VIT-B/32 model

To implement the model in an website it is need to be stored. Hence, the model extracted and stored in the disk for future use.

3.3 VIT B/32 TRANSFORMER

The basis of the vision transformer (ViT), a transformer used in the area of computer vision, is laid out by the operating principles of the transformers used in the field of natural language processing. By examining the links between input token pairs, the transformer internally learns new information. In computer vision, we may use the patches of pictures as the token.

One of the most used transformers in the deep learning space is the vision transformer. Convolutional neural networks had to be used in computer vision before the invention of the vision transformers for challenging tasks. As with BERT and GPT for complex NLP tasks, vision transformers give us one more powerful model for computer vision challenges.

The basis of the vision transformer (ViT), a transformer used in the area of computer vision, is laid out by the operating principles of the transformers used in the field of natural language processing. By examining the connections between input token pairs, the transformer internally learns new information. In computer vision, we may use the patches of pictures as the token. This connection may be found by highlighting the network. This may be achieved by either replacing some of the convolutional network's components or doing so in combination with one. The classification of images may be done using these network architectures.

By dividing a picture into fixed-size patches, accurately embedding each one, and including positional embedding as an input to the transformer encoder, the visual transformer can change an image. Additionally, ViT models outperform CNNs by roughly four times in terms of accuracy and computing efficiency. The architecture of the vision transformer is depicted in figure 4.1.

The overall architecture of the vision transformer model is given as follows in a step-by-step manner:

1. Divide a picture into patches (fixed sizes)
2. Flatten the patched-up images.
3. From these flattened picture patches, produce lower-dimensional linear embeddings.
4. Add positional embeddings.
5. Input the sequence into a cutting-edge transformer encoder.
6. Use picture labels to pre-train the ViT model, which is subsequently fully supervised on a large dataset.
7. Improve the image classification dataset's downstream dataset

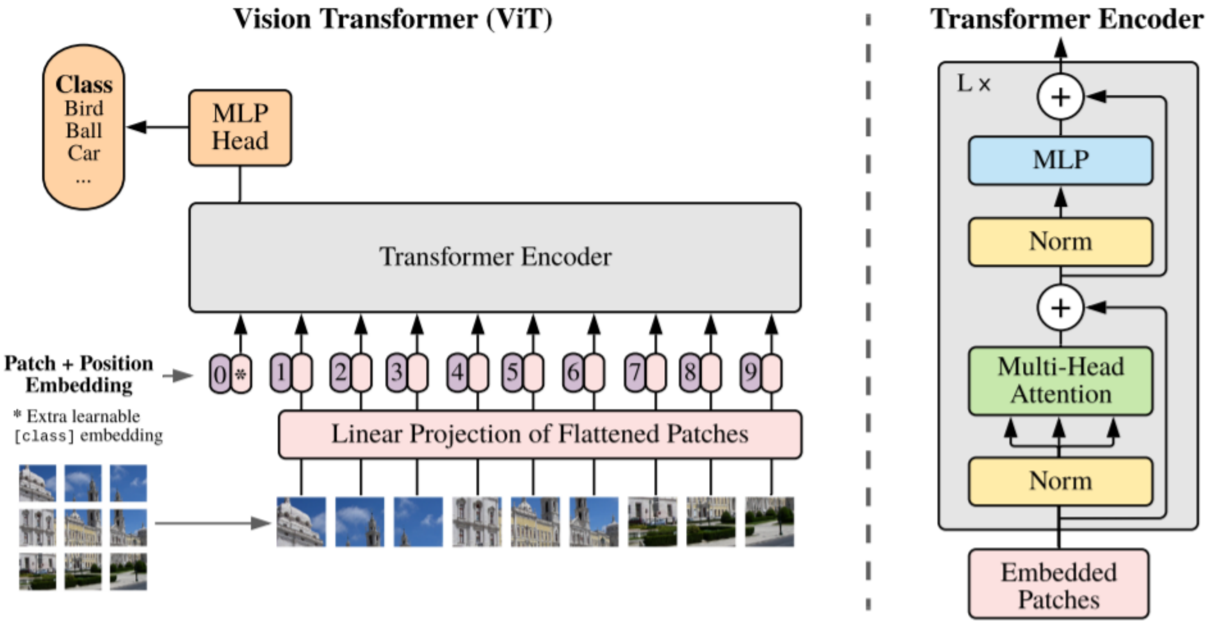


Figure 3.4: Vision Transformer Architecture

3.3.1 Training model using the VIT-B/32

Import Data

The model data set for training consists of 10480 photos, which are divided into classes such accidents, non-accidents, and surrounding activities.

Configure Hyper Parameters

learning_rate = 0.001

weight_decay = 0.0001

batch_size = 256

num_epochs = 100

image_size = 72

patch_size = 6

*num_patches =(image_size // patch_size) ** 2* *projection_dim = 64*

num_heads = 4

*transformer_units = [projection_dim * 2,projection_dim,]*

transformer_layers = 8

mlp_head_units = [2048, 1024]

Taking into account the abovementioned parameters, we can state that the process will involve 100 training epochs, resizing the image, and patching it. The learning rate controls the step size at each iteration of an optimization algorithm as it advances toward a minimum of a loss function. A regularisation method used on a neural network's weights is called weight decay or regularisation.

Data Augmentation

In the augmentation, the photos are first resized and normalised before being randomly flipped. This procedure will be completed in sequential methods and using the Keras provided layers.

Building Network

A multilayer perceptron (MLP) is a type of artificial neural network called a feedforward network. It takes a set of inputs and turns them into a set of outputs. An MLP is made up of many layers of input nodes that are linked together as a directed graph between the input and output layers. Backpropogation is how MLP trains the network.

The patch maker converts the images into patch segments. The patch encoder will linearly transform the image patches and add to the projected vector a learnable position embedding.

Building vision transformer

Building blocks for the vision transformer are created in this step. We'll use the enhanced data that passes via the patch maker block and then the patch encoder block, as it was implemented above. We'll apply a self-attention layer to the patch sequences in the transformer block. A classification head will process the output from the transformer block, aiding in the creation of the final outputs.

After that, the classifier employing the vision transformer in which we have offered the techniques for data augmentation, patch creation, and patch encoding Our final contribution to the transformer as an image representation will be encoded patches. We can alter the output's shape by flattening the layer.

Compile and train the model

The image encoder of the model is based on the ViT-B/32 Transformer architecture. Each batch of photos must first be passed into the ViT feature extractor in order to get embeddings before being fed into the model. To achieve this, we must first apply transformations to the batch to make sure it fits with the necessary feature extractor requirements. A list of images is necessary for the feature extractor.

3.4 CLIP

CLIP (Contrastive Language-Image Pre-Training) is a neural network trained with numerous (image, text) pairings. Similar to the zero-shot capabilities of GPT-2 and GPT-3, it can be instructed in natural language to anticipate the best appropriate text snippet given an image without directly optimising for the job.

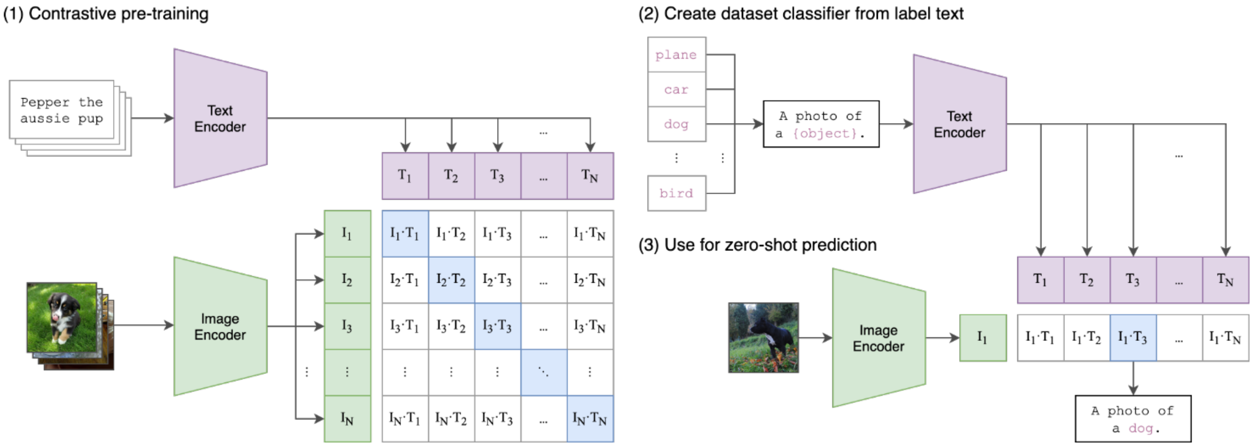


Figure 3.5: CLIP Image Processing

The only thing an ImageNet model can do ”out of the box” is forecast the 1000 ImageNet categories. An ML practitioner must create a new dataset, include an output head, and fine-tune the model if they need to conduct any additional activity. Contrarily, CLIP does not require extra training samples to handle a wide range of visual classification tasks. We just need to ”teach” CLIP’s text-encoder the names of the task’s visual ideas in order to apply CLIP to a new task; after that, CLIP will produce a linear classifier of its visual representations. This classifier’s accuracy frequently matches that of fully supervised models.

3.5 Flask API

A Python-based web application framework is called Flask. It was made by Armin Ronacher, who also founded the Pocco global community of Python aficionados. Flask is supported by the Jinja2 template engine and the Werkzeug WSGI framework. They’re both Pocco projects. Installation of Flask frequently calls for Python 2.6 or later. Although Python 3 (versions 3.3 and later) works well with Flask and its dependencies, many Flask extensions do not properly support it. Flask should be installed on Python 2.7 as a result. The Python API Flask makes it possible to create web apps. The system was created by Armin Ronacher. Since less fundamental code is needed to build a straightforward web application with Flask than it is with Django, it is more explicit and simpler to understand. The WSGI (Web Server Gateway Interface) toolkit and the Jinja2 template engine are both used by Flask.

Flask was designed to be user-friendly and expandable. The goal of Flask is to offer a reliable framework for web applications of various levels of complexity. You are then free to add any more extensions you think are required. You are also permitted to design your own modules. Flask is helpful in many different situations. For prototyping, it is quite helpful. The Werkzeug WSGI toolkit and the Jinja2 template engine are two external libraries that Flask needs to run.

- It is simple to transform Flask into the necessary web framework by adding a few lightweight extensions because of its lightweight and modular structure.
- The Flask documentation is comprehensive, full of examples, and organised effectively. To get a sense for Flask, use sample apps as well.
- It is quite easy to deploy Flask in a production environment.
- Functionality for managing HTTP requests

- Strong Flexibility
- The setup is much more adaptable than Django's, providing you with a variety of options to meet every production requirement.

HTTP Methods

GET - This is used to deliver data to the server in plain text without encryption.
POST - Sends the server the form's data. The server does not cache data obtained using the POST method.

Routing

Nowadays, routing techniques are offered by web frameworks so that users may remember their URLs. Directly to the web page rather than going through the Home page is helpful. The following route() decorator is used to link the URL to a function.

Handling Static files

A static file, such a javascript or CSS file, is often needed by a web application to show the presentation of a web page in the browser. However, during development, these files are made available as static directories either within the package or next to the module. Usually, the web server configures them.

3.6 Software Requirement and Specification

The tools used for the project are :

- Python
- Spyder
- HTML
- CSS
- Javascript
- Pytorch

- OpenCV

3.6.1 Python

Guido Rossumin designed Python, an object-oriented programming language, in 1989. It is great for rapidly developing prototypes of complex applications. It provides interfaces to many OS system APIs and libraries and may be extended to C or C++. Numerous well-known businesses, such as NASA, Google, YouTube, and BitTorrent, use Python as a programming language. Artificial intelligence, natural language processing, neural networks, and other cutting-edge computer science topics make extensive use of Python programming. Guido van Rossum created Python, a high-level, open-source programming language, in the late 1980s. The Python Software Foundation currently manages Python. Early in his career, he co-created the ABC language, marking the beginning of everything. Python is a robust programming language that can be used to create games, online applications, and graphical user interfaces. It is a sophisticated language. Reading and writing statements in Python differ significantly from standard English. Python is not written in a machine-readable language, thus it must first be processed by computers. Python is a commonly used programming language. Before a programme is executed, its interpreter analyses the source code and turns it into byte code, which a computer can understand.

Python is an object-oriented programming language that allows users to design and execute programmes by managing objects or data structures. The quality of Python is consistent throughout. Python treats every class, data type, function, and method similarly. Programming languages are established to offer users and programmers with a helpful tool for constructing programmes that change people's lives, ways of life, economies, and cultures. They improve the quality of life by boosting power, communication, and productivity. When a language fails to meet expectations, it dies and is replaced by a superior language, hence becoming obsolete. Python is a computer language that has stood the test of time and remained popular among programmers, company owners, and average consumers. It is a dynamic, active, and incredibly useful language that is highly recommended as a primary programming language for anybody desiring to learn programming and begin coding.

• Python is Interpreted: At runtime, Python is processed by the interpreter. You don't need to build your software before launching it. This is similar to PERL and PHP.

• Python is Interactive:You can really sit at a Python prompt and communicate to the interpreter right away while building programmes in the language.

• Python is Object-Oriented :Python is compatible with object-oriented programming, which encapsulates code in objects.

• Python is a Beginner's Language: Python is a great programming language for beginners since it makes it easy to create a variety of applications, from basic text processing to game development to web browsers.

Pandas

Pandas is a set of tools for data analysis and manipulation designed specifically for Python. It provides particular methods and data structures for dealing with mathematical tables and time series. It is free software provided in accordance with the three clauses of the BSD licence.

Scikit learn

Scikit-learn is a free machine learning package for Python. It supports a number of techniques, including support vector machines, random forests, and k-neighbors, as well as Python's NumPy and SciPy libraries.

Collections

In Python, collections are containers for holding data collections, such as list, dict, set, tuple, etc. They are constructed collections. List, tuple, and dict are examples of built-in container data types that are replaced by the collections module. A number of modules have been created that offer other data structures for storing data sets.

MatplotLib

It is a Python visualisation package for arrays in 2D charts. Built on NumPy arrays and intended to interact with the whole SciPy stack, it is a multiplatform data visualisation framework. One of visualization's main benefits is that it allows us visual access to enormous amounts of data in easily understandable ways. In Matplotlib, there are several plot types, such as line, bar, scatter, and histogram.

NumPy

NumPy is a general-purpose toolkit for working with arrays. It provides a very quick multidimensional array object along with the ability to interact with these arrays. The foundational Python module for scientific computing, to put it simply.

3.6.2 Spyder

Computer programmers are made more efficient by integrated development environments (IDEs), which combine basic tools like code editors, compilers, and debuggers into a single software package. The environment is already provided by an IDE, thus users do not need to install the language's compiler or interpreter on their computers.

Spyder is a specific Python IDE. It is a popular IDE because of its many useful features. By and for scientists, engineers, and data analysts, Spyder is a free and open source scientific environment created in Python. The extensive editing, analysis, debugging, and profiling capabilities of an all-inclusive programming tool are combined with the data exploration, interactive execution, deep inspection, and outstanding visualisation characteristics of a scientific package to provide a novel solution.

A straightforward, lightweight, and effective interactive development environment for scientific Python programming is called Spyder (Scientific PYthon Development EnviRonment; formerly known as Pydee). This programme is cross-platform and open source. On Windows, it may be installed using Python(x,y), WinPython, or Anaconda; on Mac OS, it can be installed using Anaconda or MacPorts. Additionally, Spyder may be included into popular Linux versions (Ubuntu, Debian, Fedora, OpenSuse, Gentoo, ArchLinux).

Some of Spyder's more notable characteristics include the ones listed below:

- It examines the code to offer go-to definitions, automated code completion, and horizontal and vertical splitting.

- It works nicely with data science libraries like NumPy since it was created primarily for data scientists.
- You may use it to launch the IPython terminal.
- It comes with a powerful debugger.
- It has a built-in documentation browser.

3.6.3 HTML

HTML is the most often used markup language for web-based content. Cascading style sheets (CSS) and JavaScript are two programming languages that may benefit. Web browsers transform HTML files into multimedia web pages after receiving them from a web server or local storage. HTML originally defined the semantic structure of a web page as well as recommendations for how the information should be displayed.

The basic units of an HTML page are called HTML elements. HTML structures may be used to include images and other objects, such as interactive forms, into the finished page. By giving text components with structural semantics, such as headers, paragraphs, lists, links, and other elements, HTML makes it possible to create organised texts.

3.6.4 CSS

A style sheet language called CSS is used to regulate how a page written in a markup language like HTML is presented. CSS is a crucial part of the World Wide Web, much like HTML and JavaScript. The goal of CSS is to keep presentational elements like layout, colour, and font apart from text. With this division, content accessibility may be increased, presentation attributes can be specified with greater freedom and control, different web pages can share formatting, and the.css file can be cached to speed up page loads for sites that use it. This separation eliminates complexity and redundancy in the structure content by supplying the necessary CSS in a separate .css file.

3.6.5 JavaScript

Along with HTML and CSS, the computer language JavaScript, sometimes known as JS, is one of the core components of the World Wide Web. Over 97 percent of websites employ JavaScript for client-side web page behaviour, and third-party libraries are often used. Every major web browser has a specialised JavaScript engine to run code on consumer electronics. JavaScript is an ECMAScript compliant high-level, usually just-in-time, compiled programming language. It has prototype-based object orientation, first-class functions, and dynamic typing. Programming paradigms such as imperative, functional, and event-driven are all supported. The Document Object Model, regular expressions, dates, text, and regular expressions are all supported through APIs (DOM). JavaScript engines, which were once primarily utilised in web browsers, are now a standard component of many servers and applications.

3.6.6 PyTorch

PyTorch is an open source machine learning framework that is built on the Torch library and used for applications like computer vision and natural language processing. PyTorch was largely developed by Meta AI. It is open-source software distributed under the Modified BSD licence and is free to use.

PyTorch is an open source machine learning (ML) framework built on Python and the Torch library. This is one of the most widely used platforms for deep learning research. The framework has been created to speed up the process of moving from a research prototype to actual implementation.

Better visualisation provided by TensorFlow enables developers to more effectively debug and monitor the training process. But PyTorch only offers a little amount of visualisation. Because of the TensorFlow Serving framework, TensorFlow outperforms PyTorch when it comes to delivering learned models into the real world.

The PyTorch framework supports more than 200 different mathematical operations.PyTorch's popularity is growing as more people use it to construct artificial neural network (ANN) models. The primary uses of PyTorch are in research, data science, and artificial intelligence (AI).

3.6.7 OpenCV

OpenCV is a free to use software library for computer vision and machine learning. To hasten the inclusion of artificial intelligence into products, OpenCV was used to provide a common infrastructure for computer vision applications.

The OpenCV-Python library of Python bindings was developed to handle computer vision difficulties. The popularity of Guido van Rossum's general-purpose programming language Python has grown rapidly, mostly as a result of its usability and understandable code.

Images are loaded into CV2 in the specified format, including with the alpha channel. The value of the alpha channel, which holds transparency information, determines how opaque a pixel is.

Chapter 4

RESULTS AND DISCUSSIONS

Taking into account the quick development of embedded surveillance video systems for tracking traffic accidents This study uses a Vision Transformer-based accident detection method in place of CNN to improve detection speed and achieve high accuracy. Transformers work with pictures as a series of patches, selectively focusing on certain visual components depending on the circumstances.

Every frame of the video the system is taking from the camera is taken into account once it has started running, and it is then run through the suggested model.The VIT-B/32 Transformer will then receive the patches as input. The model is then developed and assessed. It can send a alert message to the authority/nearby ambulance service when it detects an accident.

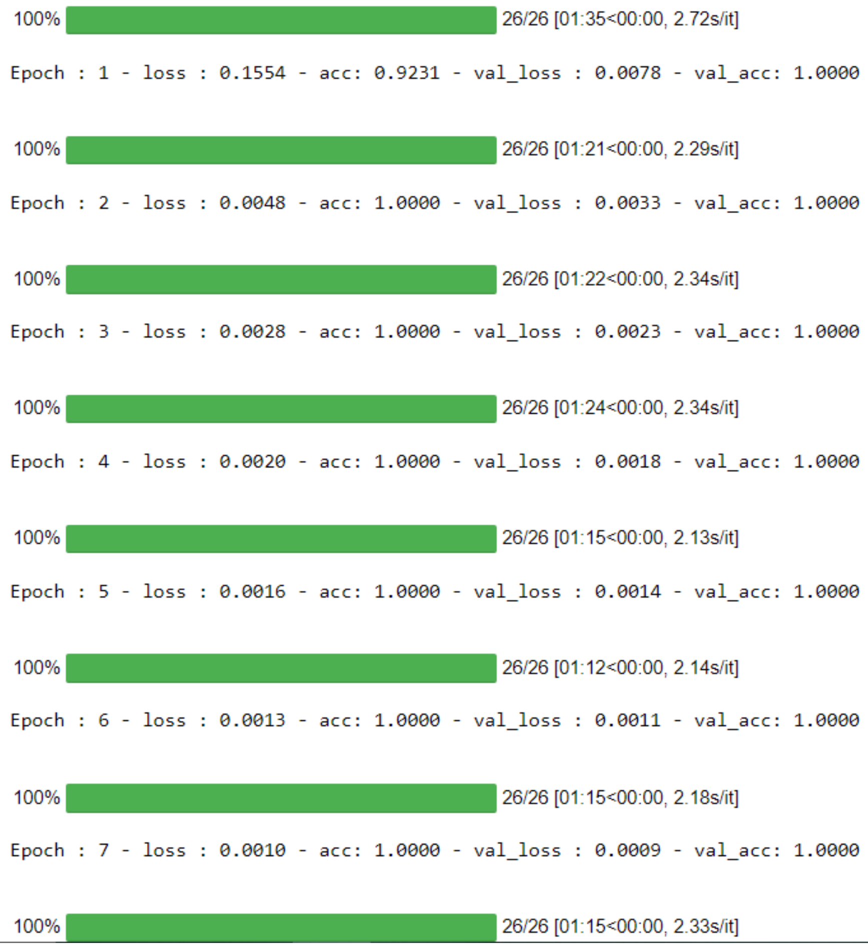
4.1 Model Evaluation

4.1.1 Training Result

The model is trained for 15 epochs, with a batch size of 64 for the training phase which amounts to 1 step per epoch for training. Thus, training for 15 such epochs yielded optimized results, with high accuracy of 92% for VIT-B/32 and corresponding loss of 8%. The figure 4.1 shows the models training .

4.1.2 Accuracy

The ratio of accurately predicted instances to all examples is how accuracy is measured. While running all the epoches, the VIT-B/32 accident detector has a classification accuracy of up to 92



```
100% 26/26 [01:35<00:00, 2.72s/it]

Epoch : 1 - loss : 0.1554 - acc: 0.9231 - val_loss : 0.0078 - val_acc: 1.0000

100% 26/26 [01:21<00:00, 2.29s/it]

Epoch : 2 - loss : 0.0048 - acc: 1.0000 - val_loss : 0.0033 - val_acc: 1.0000

100% 26/26 [01:22<00:00, 2.34s/it]

Epoch : 3 - loss : 0.0028 - acc: 1.0000 - val_loss : 0.0023 - val_acc: 1.0000

100% 26/26 [01:24<00:00, 2.34s/it]

Epoch : 4 - loss : 0.0020 - acc: 1.0000 - val_loss : 0.0018 - val_acc: 1.0000

100% 26/26 [01:15<00:00, 2.13s/it]

Epoch : 5 - loss : 0.0016 - acc: 1.0000 - val_loss : 0.0014 - val_acc: 1.0000

100% 26/26 [01:12<00:00, 2.14s/it]

Epoch : 6 - loss : 0.0013 - acc: 1.0000 - val_loss : 0.0011 - val_acc: 1.0000

100% 26/26 [01:15<00:00, 2.18s/it]

Epoch : 7 - loss : 0.0010 - acc: 1.0000 - val_loss : 0.0009 - val_acc: 1.0000

100% 26/26 [01:15<00:00, 2.33s/it]
```

Figure 4.1: Training the model

4.1.3 Error rate

The ratio of erroneously predicted instances to all examples is known as the error rate. While executing all the epoches, the VIT-B/32 accident detector experienced a 8% misclassification error rate.

4.2 User Interface

4.2.1 Graphical User Interface

The primary goal of user interface development is to implement the extracted VIT-B/32 learning model in a website. Any end user may use the model to forecast the severity of an accident with this implementation, even if they have no technical background in machine learning.

The application consist of two phases: • Front End

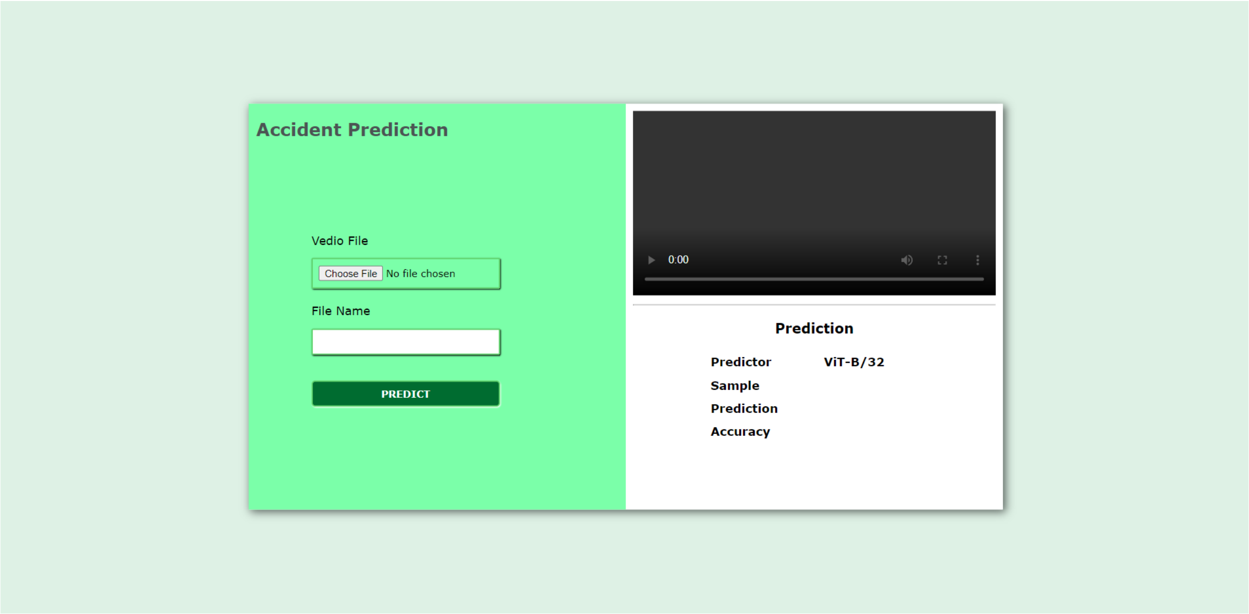
• Programming Interface

Front End

The front end of this project is developed using the HTML, CSS and the JavaScript.

Programming Interface

The backend is created using Flask. It serves as the web browser's and the VIT-B/32 model's interface.



Predictor ViT-B/32
Sample
Prediction
Accuracy

Figure 4.2: User Interface

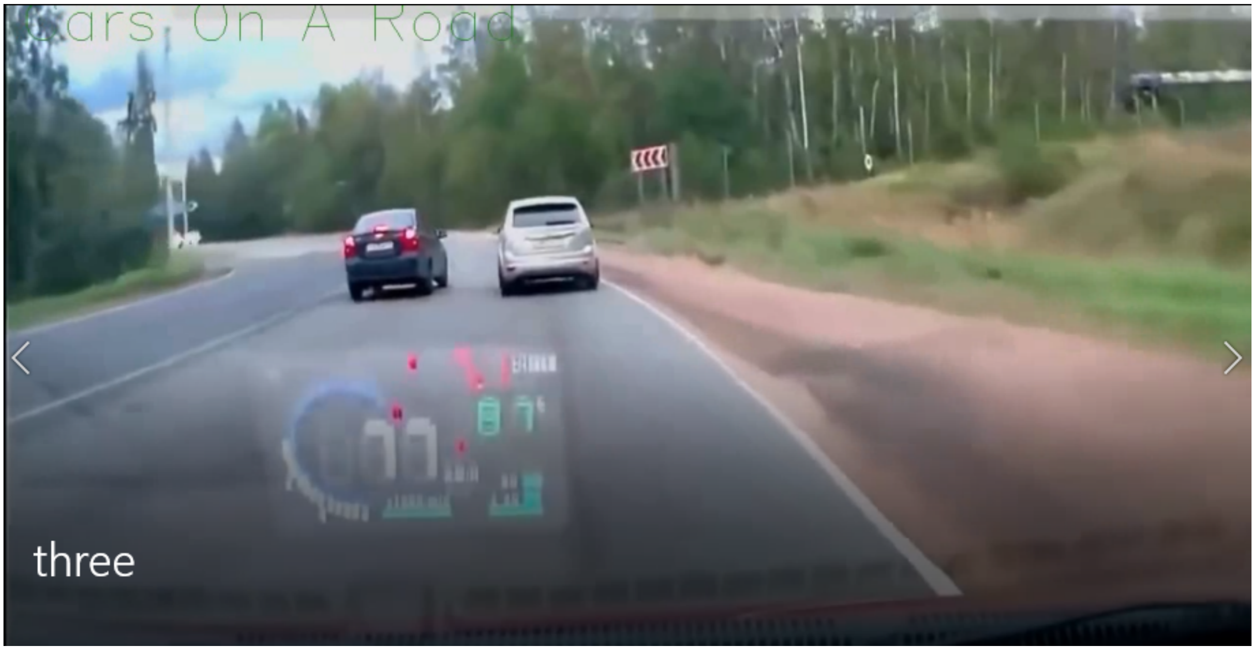


Figure 4.3: Cars Detection

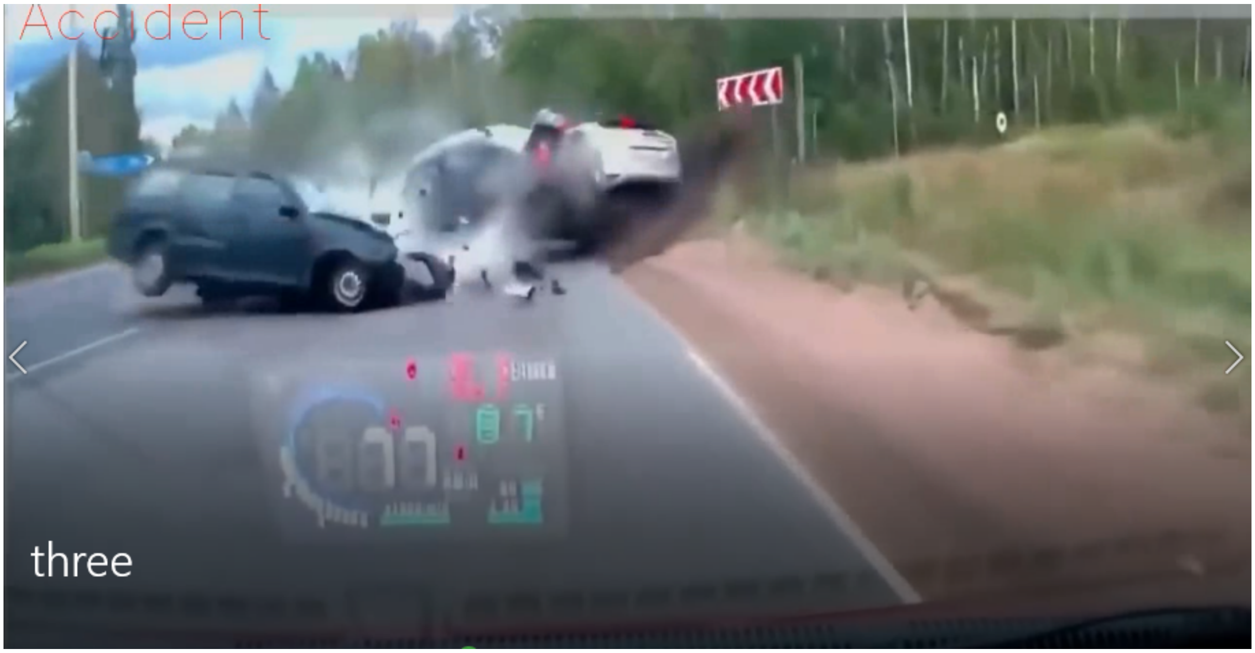
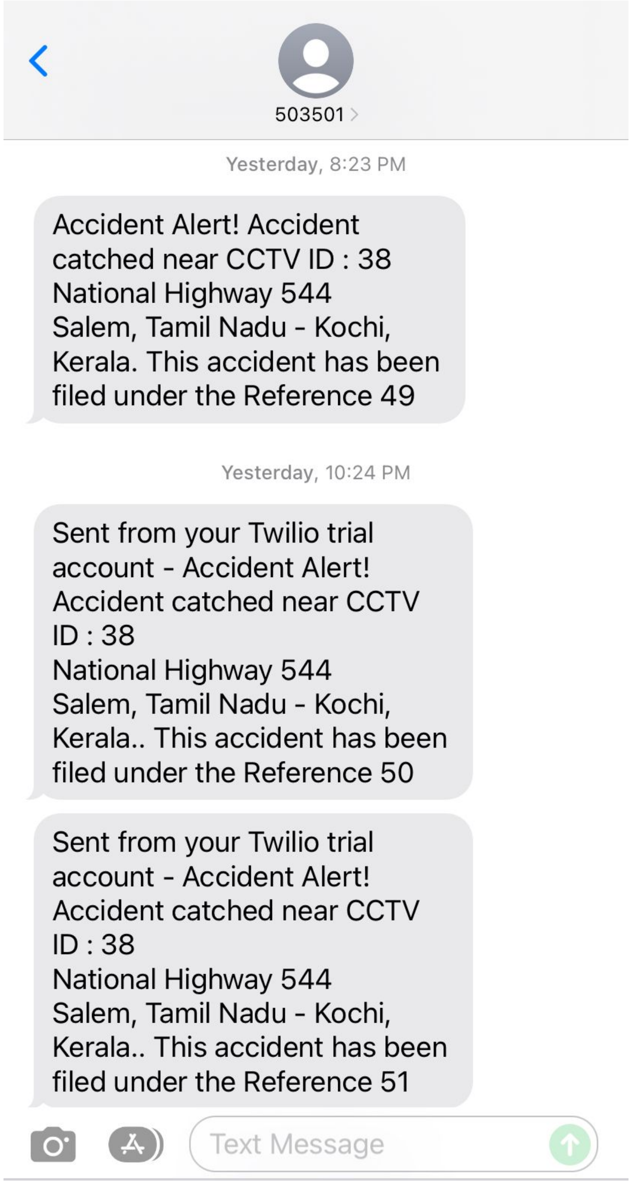


Figure 4.4: Accident Detection



Yesterday, 8:23 PM

Accident Alert! Accident catched near CCTV ID : 38
National Highway 544
Salem, Tamil Nadu - Kochi, Kerala. This accident has been filed under the Reference 49

Yesterday, 10:24 PM

Sent from your Twilio trial account - Accident Alert! Accident catched near CCTV ID : 38
National Highway 544
Salem, Tamil Nadu - Kochi, Kerala.. This accident has been filed under the Reference 50

Sent from your Twilio trial account - Accident Alert! Accident catched near CCTV ID : 38
National Highway 544
Salem, Tamil Nadu - Kochi, Kerala.. This accident has been filed under the Reference 51

Figure 4.5: Sms Alert

Chapter 5

CONCLUSION

Accidents, which cause the loss of both life and property, are one of the most prevalent problems that mankind faces on a daily basis. The suggested strategy provides a very useful and effective solution to this problem. The suggested automobile accident detection system can identify an accident as it occurs and sends a sms notification about the event, together with details like location and CCTV ID , to the nearby hospitals and police stations. The suggested system is far more cost-effective, failsafe, and accurate than its competitors thanks in large part to a model-based approach, unlike existing systems in use that comprise pricy sensors and unnecessary hardware. Images have been used in experimentation, testing, and validation, and the findings demonstrate that this approach does indeed produce better sensitivity and accuracy; as a consequence, it is a realistic choice for deploying this system on the majority of the state and national highways in the nation. Thus, the project contributes to a social cause and aids in the development of a system that ensures no person is abandoned or rendered helpless in the case of an accident, thereby ensuring and upholding the highest standards of quality of life.

5.1 Advantages

The main merits of proposed model are:

- No specialised expertise necessary.
- Can access the model anytime and anywhere

- Can analyse the underlying causes and external factors
- Can send an warning sms to appropriate authority.

5.2 Future Enhancement

The methodology can eventually combine supervised and unsupervised techniques to enhance the system. To extract the accidents from the frames that unsupervised methods label as unusual, we can employ supervised learning models. This model may also be used to determine risk factors and prevention measures. The system of upcoming technologies can also be improved by using additional photos and including hybridization in the vision transformers.There is a time lag in detetcion as the model needs high system configuration.This can be eliminated in future. The system may also use automated messaging to send text messages or audio messages to several phones simultaneously.

References

[1] Mubariz mansoor, Muhammad umar , Saima sadiq , Abid isaq, Saleem ullah, Hamza, and Carmen, ”RFCNN: Traffic Accident Severity Prediction Based on Decision Level Fusion of Machine and Deep Learning Model”, IEEE Access, September 23, 2021.

[2] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, Dmitry Kislyuk,” Toward Transformer-Based Object Detection”,doi.org/10.48550/arXiv.2012.09958, Computer Vision and Pattern Recognition, 2020

[3] Sachin Kumar and Durga Toshniwal, ”A data mining framework to analyze road accident data”, DOI 10.1186/s40537-015-0035-y,20th International Conference on Ubiquitous Computing and Communications,2021

[4] Shakil Ahmed, Md Akbar Hossain, Md Mafijul Islam Bhuiyan, Sayan Kumar Ray, ”A Comparative Study of Machine Learning Algorithms to Predict Road Accident Severity”,IEEE DOI 10.1109/IUCC-CIT- DSCI-SmartCNS55181.2021.00069, 2021

[5] Mohamed K Nour,Atif Naseer, ”Accident Data Analysis to Develop Target Groups For Countermeasures”, Max Cameron,International Journal of Advanced Computer Science and Applications, Vol. 11, No. 12, 2020

[6] Mohamed K Nour, Atif Naseer, Basem Alkazemi, Muhammad, ”Road Traffic Accidents Injury Data Analytics”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 12, 2020

[7] A. Mehdizadeh, M. Cai, Q. Hu, M. A. A. Yazdi, N. Mohabbati-Kalejahi, A. Vinel, S. E. Rigdon, K. C. Davis, and F. M. Megahed, “A review of data analytic based applications in road traffic safety. Part 1: Descriptive and predictive modeling,” Sensors (Switzerland), vol. 20, no. 4, pp. 1–24, 2020.

[8] Q. Hu, M. Cai, N. Mohabbati-Kalejahi, A. Mehdizadeh, M. A. A. Yazdi, A. Vinel, S. E. Rigdon, K. C. Davis, and F. M. Megahed, “A review of data analytic applications in road traffic safety. Part 2: Prescriptive modeling,” Sensors (Switzerland), vol. 20, no. 4, pp. 1–19, 2020.

[9] J. Ma, Y. Ding, J. C. Cheng, Y. Tan, V. J. Gan, and J. Zhang, “Analyzing the Leading Causes of Traffic Fatalities Using XGBoost and Grid-Based Analysis: A City Management Perspective,” IEEE Access, vol. 7, pp. 148 059–148 072, 2019.

[10] N. Zagorodnikh, A. Novikov, and A. Yastrebkov, “Algorithm and software for identifying accident-prone road sections,” Transp. Res. Procedia, vol. 36, pp. 817–825, 2018. https://doi.org/10.1016/j.trpro.2018.12.074

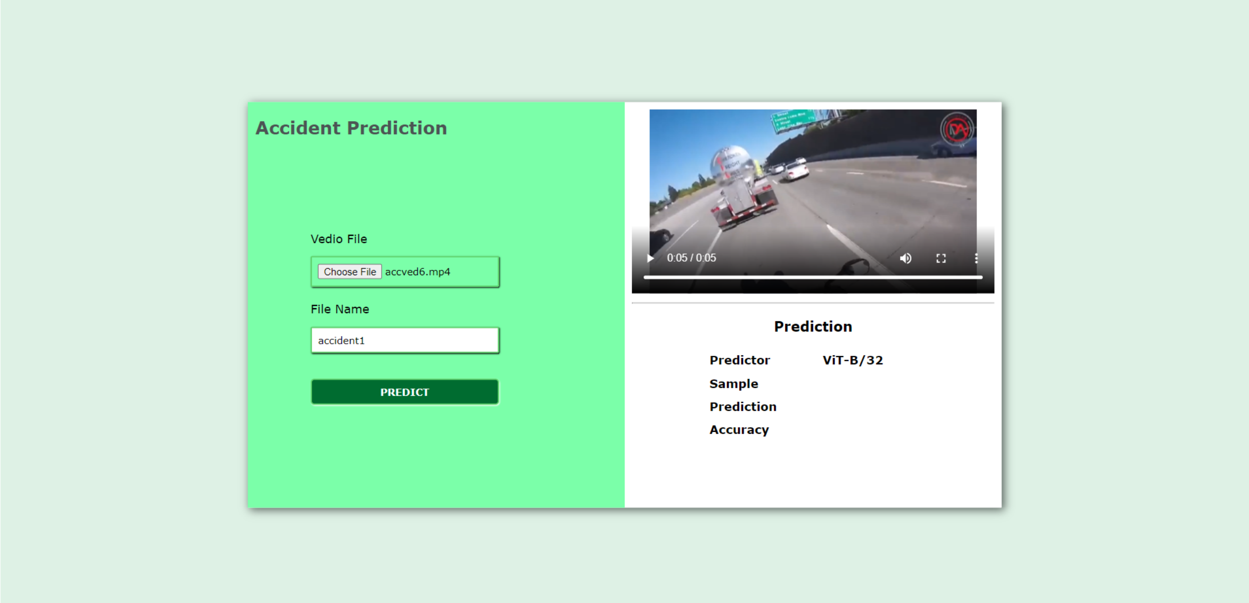
[11] L. G. Cuenca, E. Puertas, N. Aliane, and J. F. Andres, “Traffic Accidents Classification and Injury Severity Prediction,” in 2018 3rd IEEE Int. Conf. Intell. Transp. Eng. ICITE 2018, 2018, pp. 52–57

[12] Karan V Deve, Object Detection Using Vision Transformers – Airoplane Detect https://keras.io/examples/vision/object_detection_using_vision_transformer/

[13] Dataset : https://github.com/mghatee/Accident-Images-Analysis-Dataset

APPENDIX

A Screenshots



Accident Prediction

Vedio File

Choose File accved6.mp4

File Name

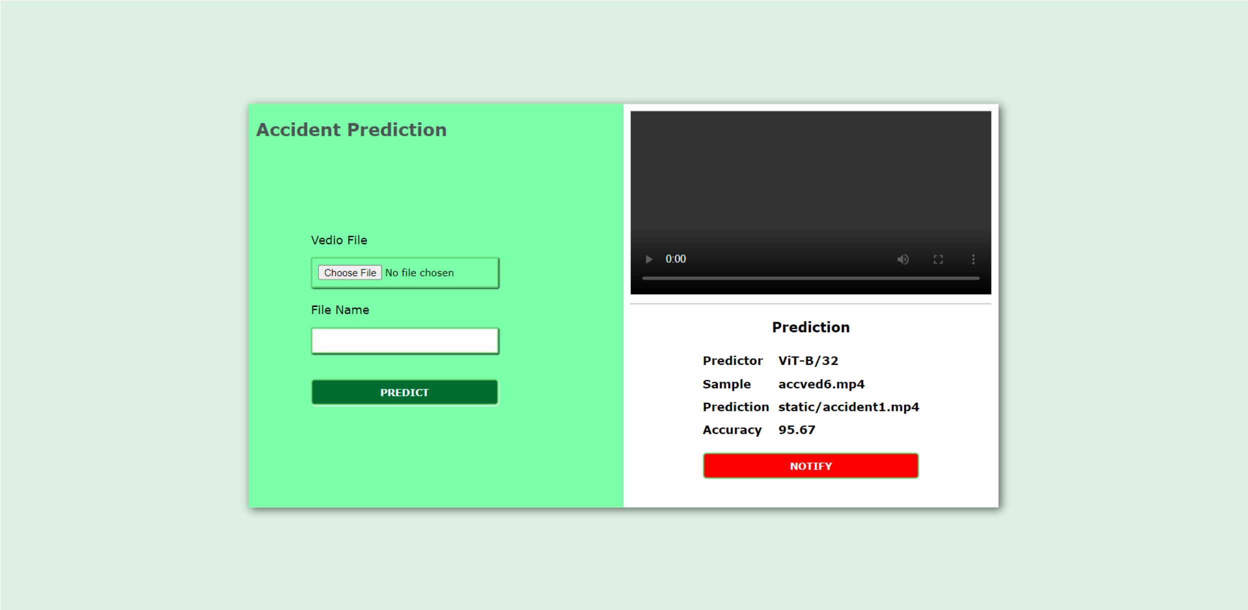
accident1

PREDICT

Prediction

Predictor	ViT-B/32
Sample	
Prediction	
Accuracy	

Figure A.1 : Video Upload



Accident Prediction

Vedio File

Choose File No file chosen

File Name

Prediction

Predictor	ViT-B/32
Sample	accved6.mp4
Prediction	static/accident1.mp4
Accuracy	95.67

Figure A.2 : Detection Result

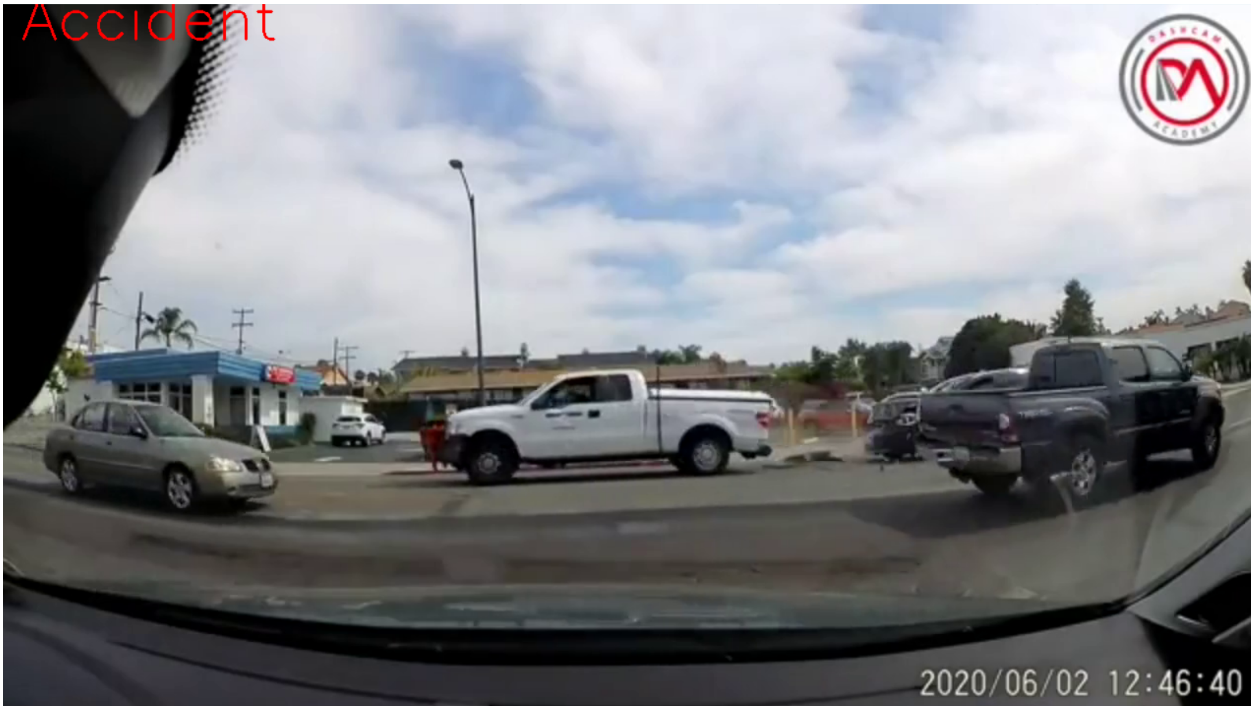


Figure A.3 : Detection