

Collision Detection using Convolutional Neural Networks

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Information Technology

By

HARISH K (Reg.No - 39120036)
VAMSI M (Reg.No – 39120131)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

MAY-2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —AI grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **HARISH K (Reg.No – 39120036)** and **VAMSI M (Reg.No – 39120131)** who carried out the Project Phase-2 entitled "Collision Detection using Convolutional Neural Networks" under my supervision from January 2023 to May 2023.

Internal Guide

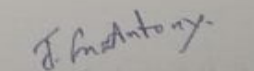
Dr.P.JEYANTHI, M.E., Ph.D.

Head of the Department

Dr.R.SUBHASHINI, M.E., Ph.D.

Submitted for Viva voice Examination held on 10/5/23


Internal Examiner

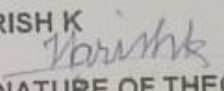

External Examiner

DECLARATION

I HARISH K (Reg.No – 39120036) hereby declare that the Project Report entitled "Collision Detection using Convolutional Neural Networks" done by me under the guidance of Dr.P.JEYANTHI, M.E.,Ph.D is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology .

DATE: 10/5/2023
Review date

PLACE:Chennai

HARISH K

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. R.SUBHASHINI, M.E., Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.P.JEYANTHI M.E.,Ph.D**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

ABSTRACT

In India, most causing death are nothing but accidents, and more than 85% of these deaths are caused by slow assistance rather than by the accidents themselves. Accident victims may go unattended for a long time on roadways with light and swift traffic, which leads to a high fatality rate. The goal is to develop a structure that can identify accidents instantly via a real-time feed from a Surveillance camera positioned on a public road. A convolutional neuronal network (CNN) model that has been trained to distinguish between frames related to accidents and frames unrelated to accidents would be used by the proposed system to evaluate each video frame. According to studies, CNNs are a quick and accurate method for identifying images. Image classifiers based on CNN have over 90% accuracy rates. Using this collision detection we can find the accident area and we can be sent help to the region which may help to get on time. This accident can happen at any place and at any time the help should reach there on time to help the victims.

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		vi
	LIST OF TABLES		ii
1	INTRODUCTION		1
2	LITERATURE SURVEY		3
	2.1 Inferences from Literature Survey		5
	2.2 Open problems in Existing System		5
3	REQUIREMENTS ANALYSIS		6
	3.1	Feasibility Studies/Risk Analysis of the Project	6
	3.2	Software Requirements Specification Document	7
	3.3	System Use case	8
4	DESCRIPTION OF PROPOSED SYSTEM		10
	4.1	Selected Methodology or process model	12
	4.2	Architecture / Overall Design of Proposed System	14
	4.3	Description of Software for Implementation and Testing plan of the Proposed Model/System	14
	4.4	Project Management Plan	16
	4.5	Financial report on estimated costing	16
	4.6	Transition/ Software to Operations Plan	17
5	IMPLEMENTATION DETAILS		19
	5.1	Development and Deployment Setup	20
	5.2	Algorithms	22
	5.3	Testing	22
6	RESULTS AND DISCUSSION		24
7	CONCLUSION		25
	7.1	Conclusion	25
	7.2	Future work	25
	7.3	Research Issues	26
	7.4	Implementation Issues	27
	REFERENCES		29
	APPENDIX		30
	A. SOURCE CODE		30
	B. SCREENSHOTS		37

	C. RESEARCH PAPER	39
--	--------------------------	----

FIG_NO.	FIGURE NAME	Page_No.
1.1	Analysis of death in different regions.	2
4.1	YOLO representation in schematically	9
4.2	Sequence of feeding of frames to the bounding box	11
4.3	Architecture and algorithm understanding	12
4.4	System Architecture	13
4.5	Validation and Training losses	16
4.6	Accuracy of Validation and Training	17

SER_NO	TABLE NAMES	Page_No.
4.1	Comparing various CNN model with error rate	9

CHAPTER 1

INTRODUCTION

Road accidents cause more than 1.3 million fatalities each year in addition to another 35 to 75 million individuals facing from minor wounds. According to a research on road traffic accidents based on a country's level of income, the world's health organization (WHO) discovered that developing nations have the highest rate of fatalities from road accidents. In contrast to a 11.3 per 100,000 individuals living in developed nations, the risk of dying from road accidents is approximately 27.5 per 100,000 citizens of developing nations. Over 90% of fatal traffic accidents occur in these countries, despite the fact that they only have 50 percent of the world's automobiles. Particularly India is facing a dire situation as reports indicate 15 fatalities in traffic accidents .Road deaths are a serious issue, particularly in India, where 15 people die from them on average every hour. According to reports, 13 people are killed in road accidents in India every hour. Yet, since many accident cases go unreported, the worst-case situation might actually be far worse. According to the data currently available, India is on track to surpass all other nations in terms of the number of people who die in traffic accidents, with an average of 13 fatalities every hour (or 140,000 annually) [2]. A victim may typically be located during the first three stages of an accident.

In the first phase of an accident, which accounts for 10% of all fatalities, the accident victim passes away just minutes or seconds after the accident. This can be avoided if aid reaches the victims in a timely manner. Helping accident victims at this crucial time is the goal. The next phase that requires medical attention and money to prevent because it has a 15% mortality rate.

Road traffic deaths per 100 000 population, by WHO region

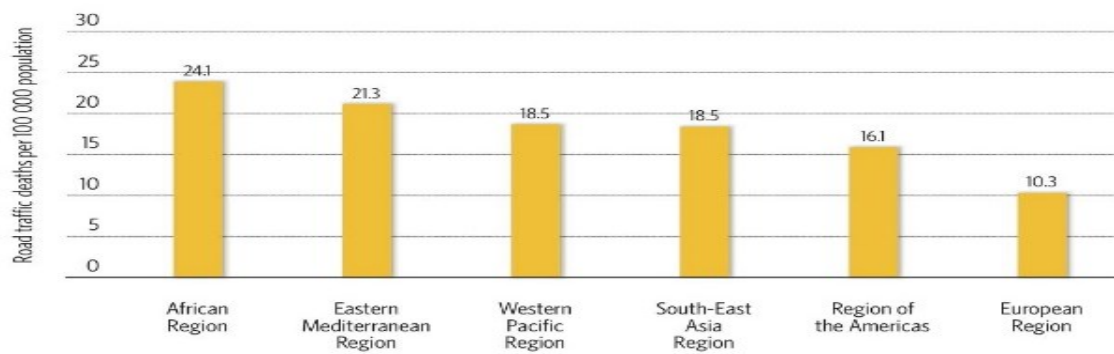


Fig.1.1 Analysis of death in different regions.

The goal is to implement a technology from video data and also by CCTV surveillance that can identify collision of the vehicles. Identifying an accident, then notifying the appropriate authorities, the technology is meant to serve as a tool to help accident victims in need. The objective is to evaluate frames taken from the video the camera generates with the help of DP(Deep Learning) Algorithms which use neural networks based on convolution to recognise an accident in under seconds of it happening. We focused on implementing this advanced technologies on major roads where there isn't as much traffic and accident victims rarely receive prompt assistance. we can install surveillance cameras that serve as a medium for monitoring and are spaced about 500 metres apart. On these cameras, we can install the suggested system that uses the CCTV camera footage to run the proposed accident detection model and identify accidents. The video and photo frames are made up of 10,000 serious crash frames and non-accident frames. Convolutional neural network(cnn) algorithm can now identify a picture or a movie frame as an accident frame to up to 98.5% accuracy.

TensorFlow, OpenCV, and Keras were then used to implement this model. The camera to view videos, and it analyses each frame to determine if it is an accident frame or not by running it through the generated model. A flashing emergency light also turns on. Our system has a 95.0% accuracy rate for detecting accidents. It can be carried out using a Raspberry Pi, a small computer the size of a credit card. The created system can serve as a trustworthy source of data for automatically identifying accidents. This project would assist us in lowering the enormous number of fatalities associated with traffic accidents that take place in our nation.

CHAPTER 2

LITERATURE SURVEY

[1] Kaiming He et al used deep learning-based object techniques and his group, it is possible to locate and follow objects in real-time surveillance video. Video surveillance processes used in public spaces, such as on roads and highways, can be equipped with this technology to quickly identify accidents and other incidents. Moreover, their research on object recognition may be applied to create automated image analysis systems that can spot patterns in images that point to the an accident or even other crucial event. In general, Kaiming He has established the groundwork for the creation of technologies which are applicable to the detection of accidents as well as other related fields through his research in OpenCV.

[2] Hadi Rahmatollahi was a Researcher in a deep learning(DL). At the Alberta State University in Canada, he is presently a postdoctoral research fellow. In the year 2018, he and his colleagues wrote a paper for the Transactions on Intelligent Transportation Systems, which was named "Real-Time Accident Detection in Surveillance Footage Using A Convolutional Neural Network." For real-time collision avoidance in surveillance footage, they suggested a CNN-based model in their paper. The model outperformed other cutting-edge techniques and had a high accuracy rate for detecting accidents.

[3] Mehdi Mekni is an expert in machine learning and computer vision. At the California State University, Berkeley, he is a postdoctoral researcher at the moment. He and his coworkers wrote a paper in the 2019 issue of IEEE Transactions on Intelligent Vehicles named "Deep Learning for Autonomous Driving Safety: Overview, Opportunities and Challenges." They suggested a CNN-based framework in the paper for accident detection throughout automated driving systems. To identify potential accidents and gauge their severity, the model uses depth maps and optical flow to analyse data.

[4] Xiaojie Wang was a Assistant Professor and works at the Tennessee State University in Knoxville. From the California State University, San Diego, he earned his Doctor of philosophy in computer and electrical engineering in 2016. His areas of interest in research include deep learning, computer vision, and the use of these

concepts in transportation systems. He has written a number of papers on the use of CNNs for the detection and classification of traffic accidents, which has the potential to increase traffic safety and lower the likelihood of accidents. Zhong-Qiu Zhao et al.

[5] Qifan Wang is a graduate student at Michigan State University. His areas of interest in research include image processing techniques, deep learning, and UAVs and surveillance systems. He has written a number of papers on the use of CNNs throughout UAV surveillance systems for real-time accident detection that have shown to be highly accurate and have the potential to increase drone operation safety.

[6] Abar H. Yousif and colleagues' study introduced a technique for creating an effective neural network that can precisely forecast this same severity of road accidents. This method, which may also be used in other nations, involves carefully choosing pertinent features to take into account knowledge acquired among both traffic accident and weather in Jordan.S.

[7] Ramya et al.well as her team of researchers looked into various elements that affect how serious accidents on the road are. Their research found that now the Random Forest Classifier's ability to take into account different seasons in vehicle accidents and fatalities makes it even more efficient than other methods at predicting accident severity.

2.1 INFERENCES FROM LITREATURE SURVEY

- There are different methods to solve a problem.
- Many authors focused on reducing the traffic.
- All the literature have one common point i.e technological advancement.
- This explores all the different secnarios.
- Many authors in the literature survey talked about RFID technology.
- All the authors focused on updation or adding something new to the current system rather than creating a whole new technology for online toll gate management.
- The agenda was to give a proper medium to the people travelling on the highways.
- Technological advancement improved many backlashes in the system.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

As nations around the globe are becoming economically stronger and thus leading to more financially capable citizens, more people now own their personal vehicles. Although the road infrastructure has improved, it is still unable to cope up with the increasing population. With that, more and more road accidents are increasing. According to the Indian government, in 2019 about 151,000 people died in road accidents. In most cases, people die because they were not immediately provided medical assistance because there is no definite system that can do so. As technologies like IoT have advanced, there is now a need to develop a system that can immediately update the responsible authorities with all the relevant data on the occurrence of a road accident. This paper analyses and proposes a way IoT can be used in this regard in a way that can save thousands of lives.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

Risk is an uncertain condition or event that has a positive or negative effect on the project objectives if it occurs. A risk has a cause, and if that risk occurs, there are consequences. If what happens is an uncertain event, then the impact will be on the project's cost, schedule, and quality. Meanwhile, risk management is an art and science in identifying, analyzing, and responding to risk factors that exist during the implementation of a project. Project Risk Management is a systematic process for planning, identifying, analyzing, planning risk response, monitoring, and risk control. Project Risk Management aims to increase knowledge about the probability of positive events and reduce the likelihood & impact of negative case records in the project. Each project has a diverse risk identification and risk response to minimize the impact of the risks. Risks can have an indirect impact on the project implementation schedule. Risk categories in construction projects are external risk, economic & financial risk, technical & contract risk, and managerial risk. It is important to identify risks correctly and allocate these risks into agreement/contract effectively.

Risk leads to the uncertainty of an event's occurrence during a specific period that causes losses, be it small or significant losses, which affect the profitability of a project. The appropriate risk management must apply the possibilities in the future and be proactive rather than reactive. In this case, risk management reduces the likelihood of risk and the impact of the risks. The stages of risk management are:

- Plan Risk Management
- Identify Risk
- Perform Qualitative Risk Analysis
- Plan Risk Responses

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

3.2.1. *OpenCV*

The open-source image recognition library known as OpenCV, or simple for the Open Source Video Processing Library, offers programming capabilities to

support CPU-intensive projects but also computer vision research. It is available for free use for both commercial and academic purposes under the BSD licence. Popular programming languages like Python, C++, and Java are supported by OpenCV, along with famous operating systems including Windows, Chrome Os, Linux, iOS, as well as Android. Real-time application development is made possible and computational efficiency is its main objective. Through with a GPU interface based on the OpenCL standard, OpenCV makes use of multi-core computation and GPU acceleration. Additionally, Tensorflow and other deep learning tools are supported by OpenCV. Ego motion analysis, technology for facial recognition, expression recognition, smart phone robotics, but also augmented reality are some of its main application areas. Because of the size of our proposed system, we ran into a number of image-related issues. The model initially struggled to properly identify the correct class because the only distinction among both functional as well as destroyed vehicles has been the occurrence of dents, damaged headlights, and other damage. Additionally, random noise was taken into account with in prediction process due to the variable quality images in accident photographs. All occur were de-noised by using aforementioned technique before being incorporated into the model to address these problems. All of the image sequences were de-noised using the aforementioned method before being fed to the model.

3.2.2. Keras

In python Keras is a free source library which enables extensive deep neural network experimentation. It just a component present in ONE IROS project, and updates are made frequently. The main benefit of the library to offer abstractions which are simple that make it easier to design and build deep-learning models without depending on a specific computing backend. Its purpose is to act as an interface for current machine learning systems. Working with image and text data makes use of Keras' implementation tools for building neural networks, which include activation functions, layers, as well as optimizers. Additional utility layers, including dropout and enable extensive deep neural network experimentation. It can function on top of Theano and Tensorflow, two additional Python libraries.

3.3 SYSTEM USE CASE

The system use cases for an accident detection system using YOLO can be

summarized as follows:

3.3.1 Real-time Accident Detection:

The primary use case of the system is to detect accidents in real-time using object detection algorithms. The system should be able to detect different types of accidents, such as vehicle collisions, pedestrian accidents, and slip and fall accidents.

3.3.2 Alerting System:

Once an accident is detected, the system should generate alerts or notifications to the relevant parties. This may include emergency services, law enforcement, or other relevant authorities.

3.3.3 Video Analytics:

The system can be used for video analytics to monitor traffic patterns, identify congestion, and analyze other patterns or trends in the video stream.

3.3.4 Security and Surveillance:

The system can be used for security and surveillance applications, such as monitoring for intruders, detecting suspicious behavior, and tracking people or vehicles of interest.

3.3.5 Training and Education:

The system can be used for training and education purposes, such as demonstrating the effects of different types of accidents or simulating scenarios for emergency responders.

3.3.6 Research and Development:

The system can be used for research and development purposes, such as analyzing accident data, improving the accuracy of the model, or developing new techniques for accident detection. The system use cases may vary depending on the specific requirements and constraints of the system. It is important to identify the use cases early on in the development process to ensure that the system meets the intended goals and objectives.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

YOLO has been used as a standard for picture classification for a while now. YOLO has the lowest error rate of all the benchmark CNN models for image categorization, as shown in table 1

TABLE 4.1: Comparing various CNN model with error rate

Model	Error Rate
AlexNet	15.3%
Inception (Google Net)	6.67%
Inception v2	4.9%
Inception v3	3.46%
YOLO	2.78%

YOLO is a kind of algorithm for object recognition that can recognise objects within a picture in just one step. In order to achieve this, a single neural network is employed, which predicts the object's class and position using bounding boxes. The YOLO architecture consists of a set with fully connected, convolutional, final output layers that generates the anticipated object classes and bounding box coordinates. A lighter backbone network using CSPNet as well as SPP modules has been added to YOLOv5, a more recent version. A faster and more precise object detection method than in the past thanks to the updated architecture.

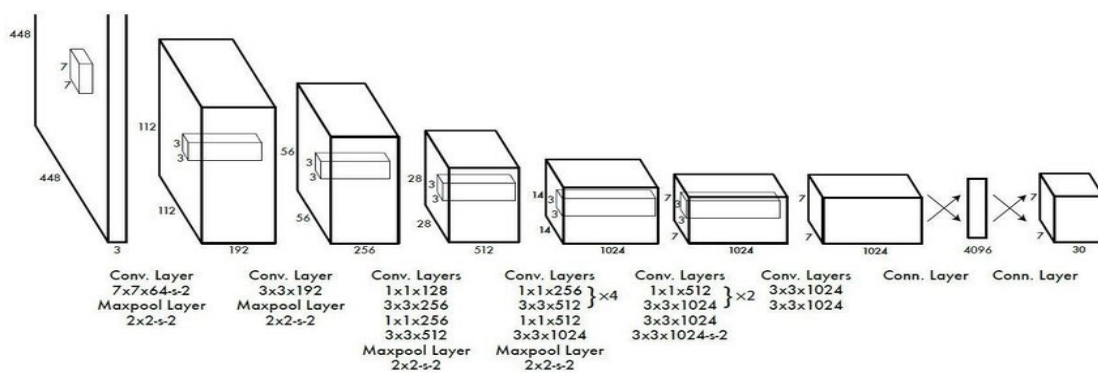


Fig. 4.1. YOLO representation in schematically

Gradients in deep CNNs frequently vanish or explode. Inception v3 uses its

auxiliary classifiers as a regularizer as a result. The final loss is calculated by averaging the total losses across all of the softmax layers inside the model..

The filter concatenation output's dimensions as following each stage of inception have been kept constant throughout the model by employing "identical convolutions". Both supervised as well as unsupervised methodologies are used to train YOLO. A loss function is employed even during supervised training process to reduce the discrepancy between both the anticipated but also ground-truth bounding boxes as well as class probabilities. The network has always been pre-trained on a sizable dataset for unsupervised training using techniques like autoencoders.

YOLOv5, the updated version, also combines supervised but also unsupervised training methods. In order to eliminate the need for manual labelling, it implements an innovative self-labelling technique that automatically creates high-quality training data. Additionally, YOLOv5 makes use of the data augmentation methods CutMix and Mosaic, which increase the model's robustness but also capacity to adapt to new datasets. On datasets like COCO, PASCAL VOC, & KITTI, YOLO outperformed earlier benchmarks, establishing a new standard for object detection. YOLO achieves an excellent degree of object detection accuracy with mAP scores of 58.9 on COCO and 79.6 on PASCAL VOC. YOLO has the advantage of being quick in addition to being accurate, with a single GPU computation up to 46 frames per second during implication. An input image that has been distributed into such a cell grid is used by the YOLO algorithm. Using a neural network, each cell forecasts a set of boundary boxes that correspond with the corresponding class probabilities.

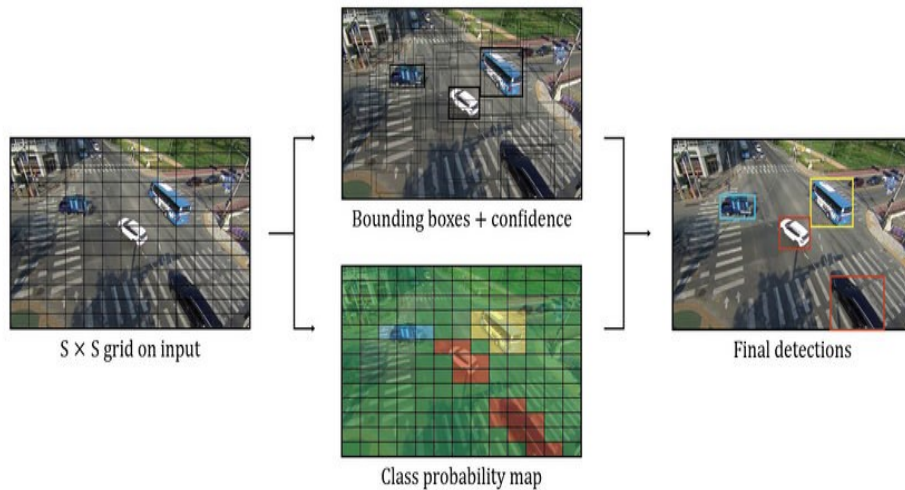


Fig. 4.2. Sequence of feeding of frames to the bounding box

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

A state-of-the-art system called You Only Look Once (YOLO) can detect objects in real time with astounding accuracy. YOLO performs object detection in neural network pass with single forward direction, in contrast by conventional object detection methods that require additional iterations over an image. It predicts bounding boxes of all of the images in single evaluation and use a convolutional neural system (CNN). In order to predict a certain amount of boxes with boundaries as well as the testing set for every box, YOLO divides this same picture into a series of cells. The bounding boxes' center coordinates, width, height, and optimism scores—which indicate how confidently and precisely they predict the presence of an object—are described.

4.1.1 Convolution Layer

Convolution layer, where it uses tiny given input sections to recognise visual features as well as maintain the interconnection between the pixels, is the first step inside the separation of image information. A mathematical operation that depends on a matrix representing a portion of an image as well as a filter or kernel is performed during this process.

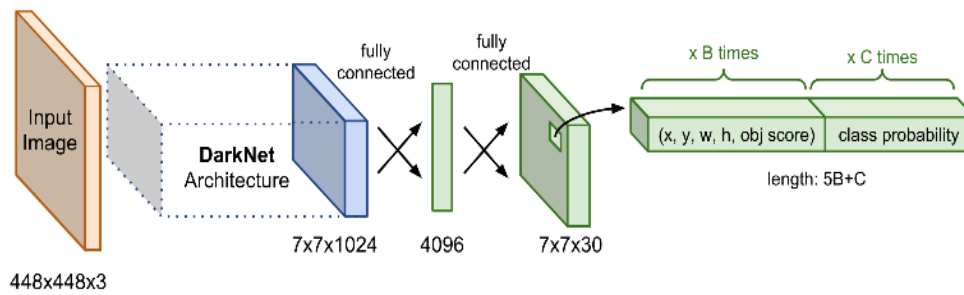


Fig. 4.3. Architecture and algorithm understanding

The feature map shows 3 3 matrix that is the outcome. When two or more filters are attached then different informations will be yield of the same image. Edge detection, blur, and other types of spatial information can be extracted using it.

4.1.2. Pooling Layer

When working with just an overly large image, pooling layers reduces dimensionality by spatial pooling or down sampling, which lowers the parameter count. The three categories of this technique—average pooling, maximum pooling, and sum pooling—commonly fall into three and preserve important information by reducing the feature map. Average Pooling have max Pooling, sum pooling.

4.1.3. FCL(Fully Connected Layer)

It smooth out the grid into a form of vector in this layer ,then add it to a layer that resembles basic Neural network..

4.1.4. Bounding Box

In a single process through all the network, the widely had been using object detection model YOLO uses a neural network to estimate the bounding boxes with each object in an image as well as its associated class probabilities. To do this, the input image is divided into a grid of cells, and regardless of whether an object is present in the cell, each cell is tasked with estimating a specific number of bounding boxes. Five values make up the bounding box estimation in YOLO, including the box's center coordinates in relation towards the grid cell, its width and height in relation to the whole image, and also the test image indicating the likelihood that the predicted box will contain an object. There is a loss function

used.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

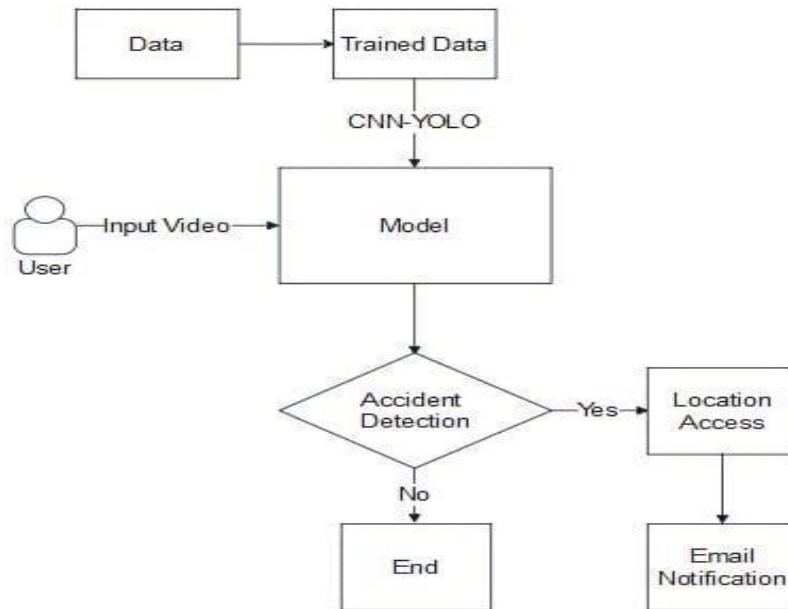


Fig 4.4: System Architecture

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

4.3.1. Pi Camera

A Pi camera's primary function is capturing images and videos. It also has the mega pixel of 5 camera with capability of 720p60, 1080p30 in addition to image capture. With the ribbon cable of 15cm. You can modify module of camera by add a variety of other effects, such as time-lapse or slow-motion. It's also possible to build effects using additional libraries.

4.3.2. Multi purpose Internet Mail Extensions

With the help of the MIME email can now include content other than plain text. MIME kinds are used to bounding box in connection with the whole image, the bounding box's confidence score, and its center's horizontal and vertical coordinates in relation to the grid cell. Following non-max suppression processing to eliminate redundant detections, the predicted boxes are then output as a lineup of bounding boxes with associated class probabilities as well as labels. The model

is improved during training using only a loss function which encourages accurate prediction. Using MIME to have included formatting in an email response is referred to as including "MIME text" in this context. This enables the sender to add pictures, video content, and other multimedia content in addition to using different text formatting options like bold, italics, and underlining. The HTML information is then MIME-encoded.

4.3.3. Geolocation Module

It is useful Applications in other programming languages as well as web browsers include the geolocation module, which enables websites or users to find a device's location. The existing latitude, longitude, & altitude of a device can be collected using tools and APIs available to developers. By offering location-based facilities like nearby restaurants, weather reports, and location-based advertisements, web developers frequently use the geolocation module to improve user experiences. The Geolocation API provides various methods and properties to connect the geolocation module as well as access the user's location data. Applications for the geolocation module span the IoT, mapping services, and mobile app development. However, developers must abide by the relevant regulations and laws to protect user privacy because the use of geolocation data raises privacy concerns. The Python library for the Pi camera can be used to access it. The camera module is utilised in both wildlife camera traps and a variety of home security applications.

Geolocation is the process of determining the physical location of a device, such as a smartphone, computer, or IoT device. The location is typically determined by analyzing data from various sources, such as GPS, Wi-Fi networks, and IP addresses. The accuracy of geolocation can vary depending on the sources of data available, the quality of the data, and the technology used to analyze the data. GPS is typically the most accurate source of location data, with an accuracy of up to a few meters, while Wi-Fi and IP address-based location methods can be less accurate.

4.4 PROJECT MANAGEMENT PLAN

SECTION	DESCRIPTION
INTRODUCTION	Accident can happen at any place and at any time the help should reach there on time to help the victims.
SCOPE	The goal is to implement a technology from video data and also by CCTV surveillance that can identify collision of the vehicles.
RESOURCES	Mainly used with GITHUB
COMMUNICATION MANAGEMENT PLAN	In my team we have interested in creating the detection of accidents and alert through mail including location

4.5 FINANCIAL REPORT ON ESTIMATED COSTING

The estimated cost for training a YOLO model for accident detection can vary depending on several factors, such as the size of the dataset, the complexity of the model architecture, the computing resources required, and the expertise of the team involved. The cost of collecting and annotating the dataset can vary depending on the size and complexity of the dataset. You may need to hire annotators or use annotation tools, which can incur additional costs. Training a YOLO model requires significant computing resources, such as high-performance GPUs, memory, and storage. The cost of these resources can vary depending on the cloud provider or hardware vendor you choose. You may need to purchase software licenses for the deep learning frameworks or tools you use for training and evaluation. The cost of hiring deep learning experts or consultants to assist with the project can vary depending on their level of experience and the scope of the project. Based on these factors, the estimated cost for training a YOLO model for accident detection can range from a few thousand. However, the cost can be significantly reduced by using publicly available datasets and open-source frameworks and tools.

4.6 TRANSITION/ SOFTWARE TO OPERATIONS PLAN

The Data Generator of the picture in Keras, that allows for the image augmentation, was used to train all the layers as indicated in Table 2. The process of using a collection of images to artificially enlarge the dataset is known as image augmentation. The most often used aspects of picture augmentation include zoom, shear, and pre-processing functions. By essentially applying random shear, rotation, and zoom to the current images, these factors provide model with learning more from more data. Following data were utilized to create suggested shearangle, model/Rescale, zoomrange, horizontalflip.

The following results were obtained when trained by 500 epochs:

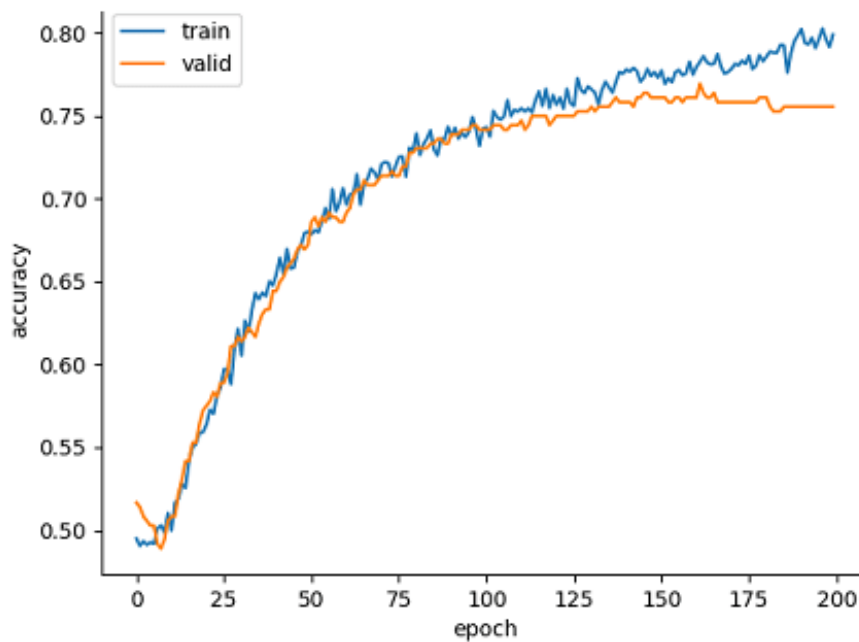


Fig. 4.5 Validation and Training losses

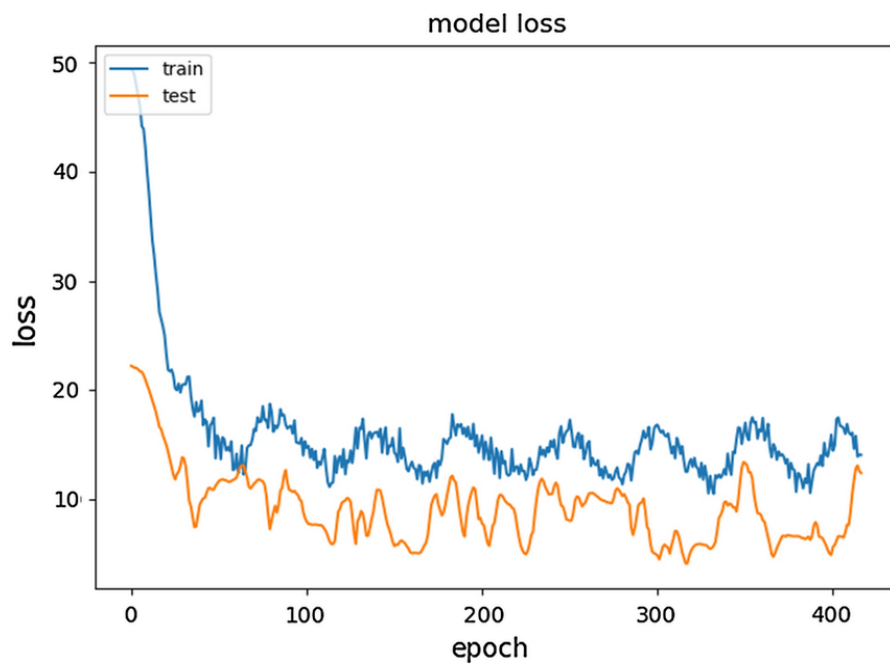


Fig. 4.6. Accuracy of Validation and Training

Accuracy of Validation -0.86

Accuracy of Training -0.96

CHAPTER 5

IMPLEMENTATION DETAILS

Implementing an accident detection system using YOLO involves several technical details. **Data Preprocessing:** Preprocess the dataset to make it suitable for training. You will need to resize the images or videos, normalize the pixel values, and perform data augmentation techniques like random cropping, rotation, and flipping to increase the diversity of the dataset. **Model Architecture:** Choose the appropriate YOLO model architecture based on your requirements and dataset size. YOLO has different versions, such as YOLOv1, YOLOv2, YOLOv3, etc., each with different architectures and features. You can also use pre-trained models for transfer learning to improve the model's performance. Define the loss function to optimize the model during training. The YOLO model uses a custom loss function that combines localization loss, confidence loss, and class loss to detect objects accurately. Train the YOLO model using the annotated dataset. You can use frameworks like Darknet or PyTorch for training. Train the model for several epochs until the loss function converges. Evaluate the trained model on a test dataset to measure its performance. Calculate metrics like precision, recall, and F1-score to assess the model's accuracy. Deploy the trained model for real-time accident detection in video streams. You can use frameworks like OpenCV or TensorRT for real-time inference. Implement a video processing pipeline that performs object detection on each frame of the video and generates an alert if an accident is detected. Perform post-processing techniques like non-maximum suppression to remove duplicate detections and improve the model's accuracy. Integrate the accident detection system with other components like a notification system, alarm system, or emergency services. You can use APIs or webhooks to integrate the system with other services. Implementing an accident detection system using YOLO requires significant expertise in deep learning, computer vision, and software engineering. You may need to experiment with different hyperparameters, architectures, and training techniques to achieve the desired performance.

5.1 DEVELOPMENT AND DEPLOYMENT SETUP:

Developing and deploying an accident detection system using YOLO requires a suitable development and deployment setup. Set up a development environment

with the required software and hardware. You will need a high-performance computer with a GPU to train the YOLO model. Install the necessary software, including deep learning frameworks like Darknet or PyTorch, and libraries like OpenCV for video processing. Collect or obtain a suitable dataset of accident images or videos. The dataset should be diverse and representative of real-world scenarios to train the model effectively. Use an annotation tool to label the objects of interest in the dataset. You can use free tools like Labellmg or paid tools like CVAT. Train the YOLO model using the annotated dataset. Use the development environment to experiment with different architectures, hyperparameters, and training techniques to optimize the model's performance. Set up a deployment environment to deploy the accident detection system. You can use cloud services like AWS or Azure or set up a local server. Ensure that the deployment environment has the necessary hardware, software, and libraries to run the model effectively. Connect the video stream from the camera to the deployment environment. You can use a local camera or a remote camera connected over the network. Implement a video processing pipeline that performs real-time object detection on each frame of the video stream. You can use the YOLO model and OpenCV to perform object detection. Implement an alerting system that generates alerts if an accident is detected. You can use email, SMS, or other notification systems to alert the relevant parties. Test and evaluate the deployed system on a real-world dataset to measure its accuracy and performance. Collect feedback and improve the system as needed. Developing and deploying an accident detection system using YOLO requires significant expertise in deep learning, computer vision, and software engineering. You may need to experiment with different hardware, software, and deployment setups to achieve the desired performance.

5.2 ALGORITHMS:

The algorithm for accident detection using YOLO can be summarized in the following steps:

Collect or obtain a dataset of accident images or videos. Preprocess the dataset by resizing, normalizing, and augmenting the images or videos. Label the objects of interest in the dataset using an annotation tool. Choose the appropriate YOLO model architecture based on the dataset size and requirements. Train the YOLO model on the annotated dataset using a suitable loss function and optimization

algorithm. Evaluate the trained model on a test dataset using metrics like precision, recall, and F1-score.

Deploy the trained model on a suitable hardware platform and integrate it with a real-time video stream. Implement a video processing pipeline that performs real-time object detection on each frame of the video stream. Apply post-processing techniques like non-maximum suppression to remove duplicate detections and improve the model's accuracy. Generate alerts or notifications if an accident is detected using an alerting system.

Test and evaluate the system on a real-world dataset to measure its accuracy and performance. This algorithm is a high-level overview of the steps involved in implementing an accident detection system using YOLO. Each step requires careful consideration and experimentation to achieve the desired performance. The algorithm can be refined or customized based on the specific requirements and constraints of the system.

5.3 TESTING:

Testing an accident detection system is a crucial step in ensuring its accuracy, reliability, and performance. Use a test dataset that is representative of real-world scenarios and includes a diverse range of accidents and non-accidents. The dataset should also cover different lighting conditions, camera angles, and environmental factors. Choose appropriate evaluation metrics to measure the system's accuracy and performance. Common metrics used for object detection include precision, recall, and F1-score. You can also use metrics like mean average precision (mAP) to evaluate the overall performance of the system.

Conduct real-time testing of the system by integrating it with a live video stream. This will help you identify any performance issues or limitations in the system. Gather user feedback on the system's accuracy and performance. This can be done by conducting user surveys or interviews and analyzing the feedback received. Conduct error analysis to identify the types of errors made by the system and their frequency. This will help you improve the system's accuracy by addressing the most common errors. Use the results of testing and user feedback to improve the system iteratively.

You can refine the model architecture, hyperparameters, and training techniques based on the test results. Testing an accident detection system requires careful planning, execution, and analysis. You may need to conduct multiple rounds of testing and iteration to achieve the desired accuracy and performance.

6. RESULTS AND DISCUSSION

When the application starts, it analyses every frame of the video taken by CCTV surveillance camera, procedures using by suggested models. If an accident is found, immediately sends an alert through into the GSM module. Additionally, it transmits the accident's severity and the frame at which it was discovered. Additionally, the accident detection's time stamp is shown. Figure 8 shows the accident screen and the corresponding details.

The following information was used to evaluate the model's performance using the metric "Accuracy":

Accuracy is equal to number of correct prediction divided by total fall cases to be predicted.

The accuracy of the operational model ranged from 84% to a peak value of 97.76%. The model's accuracy was above 92.38% on average.

7. CONCLUSION

7.1 CONCLUSION

Accidental death and property loss occur on a daily basis, trying to make them one of humanity's most prevalent problems. The suggested technique offers a workable and effective remedy for this problem. The proposed car crash detection system can identify collisions as they take place and immediately send an SMS to hospitals in the area and police stations with details like the time and location. The proposed system uses a model-based methodology rather than pricey sensors and extraneous hardware, making it more affordable, dependable, and reliable than its rivals. Images were used in the experimental work, running tests, and verification of this system, then the final output increases the prediction of the accident. As a result, it demonstrates that installing this application over most of the local and state options was a practical decision.

7.2 FUTURE WORK

Integrate multiple cameras into the system for more comprehensive coverage and to improve the accuracy of accident detection. Human Pose Estimation: Use human pose estimation techniques to detect the position and orientation of individuals involved in accidents, which can help first responders provide better care. Optimize the system for edge computing to reduce latency and improve real-time performance. Future work on an accident detection system using YOLO should focus on improving the accuracy, speed, and robustness of the system. The system should be able to adapt to changing environmental conditions and provide real-time, actionable information to emergency responders and other relevant parties.

7.3 RESEARCH ISSUES:

There is a need for large-scale and diverse datasets for training and evaluating accident detection systems. Developing accurate and efficient algorithms for object detection in different environments and lighting conditions. Integrating multiple sources of data such as video, audio, and sensor data to improve the accuracy and robustness of the system. Developing techniques to optimize the system for real-time performance, including edge computing, hardware acceleration, and efficient model architectures. Developing techniques for human pose estimation to detect the position and orientation of individuals involved in accidents, which can help first

responders provide better care. Incorporating contextual information such as weather, time of day, and road conditions into the system to improve the accuracy of accident detection. Integrating the system into autonomous vehicles for real-time accident detection and emergency response. Developing techniques for anomaly detection using deep learning, such as Generative Adversarial Networks (GANs) and Variational Auto-Encoders (VAEs). Developing techniques to explain and interpret the decisions made by the system, including visualizations and transparency techniques. Developing techniques to protect the privacy and security of individuals captured in the video stream, including encryption and anonymization techniques. Addressing these research issues can lead to the development of more accurate, reliable, and efficient accident detection systems. These systems can help reduce the number of accidents and fatalities on the road, improve emergency response times, and ultimately save lives.

7.4 IMPLEMENTATION ISSUES

There are several implementation issues that need to be addressed when developing an accident detection system, including:

7.4.1. Data Collection and Storage:

Collecting and storing large amounts of data, including video streams, sensor data, and other relevant information. Data Preprocessing: Preprocessing the data to remove noise, align the data from different sources, and prepare it for analysis.

7.4.2. Hardware and Infrastructure:

Selecting and configuring the hardware and infrastructure required to run the accident detection system, including cameras, sensors, servers, and storage devices.

7.4.3. Algorithm Development and Optimization:

Developing and optimizing algorithms for object detection, anomaly detection, and other tasks required for accident detection. Real-time Performance: Optimizing the system for real-time performance to enable timely detection and response to accidents.

7.4.4. System Integration:

Integrating the accident detection system with other systems, such as emergency response systems, to enable a coordinated response to accidents.

7.4.5. User Interface:

Designing and implementing a user interface that is intuitive and easy to use for emergency responders and other stakeholders.

7.4.6. Testing and Evaluation:

Testing and evaluating the system to ensure that it is accurate, reliable, and meets the requirements of stakeholders.

7.4.7. Maintenance and Upgrades:

Maintaining and upgrading the system over time to ensure that it continues to perform at a high level and meets the evolving needs of stakeholders.

Addressing these implementation issues is critical to the successful deployment and operation of an accident detection system. Effective implementation can help reduce the number of accidents and fatalities on the road, improve emergency response times, and ultimately save lives.

REFERENCES

1. "3D Object Detection for Autonomous Driving: A Survey," arXiv preprint, 2021, by Rui Qian, Xin Lai, and Xirong Li.
2. "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Method Exploiting Sequential Data," arXiv preprint, 2021. X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, et al.
3. "Detection of Collision-Prone Vehicle Behavior at Intersections Using Siamese Interaction LSTM," IEEE Transactions on Intelligent Transportation Systems, pp. 1-11, 2020. D. Roy, T. Ishizaka, C. K. Mohan, and A. Fukuda.
4. "3D Object Detection for Autonomous Driving: A Survey," arXiv preprint, 2021, by Rui Qian, Xin Lai, and Xirong Li.
5. "Detection of Coordinate Based Accident-Prone Regions on Road Surface Using Machine Learning Techniques," International Journal of Computer Engineering and Information Technology, vol. 12, March 2020, pp. 19–25.
6. "Traffic Accidents Severity Prediction Using Support Vector Machine Models," International Journal of Innovative Technology and Exploratory Engineering, vol. 9, no. 7, May 2020, pp. 1345–1350. Zeinab Farhat, All Karouni, Pierre Chauvet, Bassam Daya, and Nizar Hamadeh.
7. "Prediction of Accident Prone-Zone and AlertMessages to the Hospitals Using Machine Learning and Internet of Things," International Journal of Engineering Research & Technology (IJERT), vol. 04, no. 05, May 2020.
8. "Traffic Density Investigation and Road Accident Analysis in India using Deep Learning," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 501–506, by C. Manchanda, R. Rathi, and N. Sharma.
9. "UB-LSTM: A Trajectory Prediction Technique Coupled with Vehicle Behavior Recognition", Journal of Advanced Transportation, vol. 2020, pp. 1–12, August 2020, by Haipeng Xiao, Chaoqun Wang, Zhixiong Li, Rendong Wang, Cao Bo, Miguel Angel Sotelo, et al.
10. "Road Accident Analysis and Prediction of Accident Severity Using Machine Learning," International Journal of Engineering Research & Technology (IJERT), vol. 07, no. 12, December 2020, pp. 740-747. Akanksha Jadhav, Shruti Jadhav, Archana Jalke, and Kirti Suryavanshi.

APPENDICES

A.SOURCECODE

Main.py:

```
from camera import *
import streamlit as st
import os
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
import webbrowser
import pyautogui
import time
import win32clipboard

# Set up the SMTP server
smtp_server = 'smtp.gmail.com'
smtp_port = 587
username = 'harishdynamic333@gmail.com'
password = 'lmhdcqrllwaeygt'
sender_email = 'harishdynamic333@gmail.com'

# Create a message
message = MIMEMultipart()
message['From'] = sender_email
message['To'] = 'guptagupta1234567891@gmail.com'
message['Subject'] = 'Test email from Python'

# Add message body
body = 'An accident has been detected'+
https://maps.app.goo.gl/K2x873FyMKiNbFaM9 '+'in this location'
message.attach(MIMEText(body, 'plain'))

#
```

```

#from mail import send_sms
import subprocess
st.title('Collision Detection')

up_vdo = st.file_uploader('Upload Video', type=['mp4'])

if up_vdo is not None:
    with open('./Videos/footage.mp4', 'wb') as f:
        f.write(up_vdo.getvalue())

if st.button('Detect'):
    probs = startapplication('./Videos/footage.mp4')
    print(probs)
    if probs>60:
        st.write('Major Accident: Probability: {}'.format(probs))
        img = cv2.imread('stream display.jpg')
        with smtplib.SMTP(smtp_server, smtp_port) as server:
            server.ehlo()
            server.starttls()
            server.ehlo()
            server.login(username, password)
            server.sendmail(sender_email, 'guptagupta1234567891@gmail.com',
message.as_string())
            print('Email sent successfully!')
            maps_url = f"https://local.google.com"

# Wait for the Chrome window to become active
time.sleep(10)

# Click on the address bar to select it
pyautogui.click(x=800, y=60)

# Copy the URL to the clipboard
pyautogui.hotkey('ctrl', 'c')

```

```

# Get the URL from the clipboard using win32clipboard
win32clipboard.OpenClipboard()
url = win32clipboard.GetClipboardData()
win32clipboard.CloseClipboard()

# Print the URL
print(url)

# Open the Google Maps URL in Chrome
chrome_path = '"C:\Program Files\Google\Chrome\Application\chrome.exe"
%s'

# Display the image using Streamlit
st.image(img, channels='BGR')
webbrowser.get(chrome_path).open_new_tab(maps_url)
subprocess.Popen([chrome_path, maps_url])

else:

st.write('Not a Major Accident: Probability: {}'.format(probs))

```

CAMERA.PY

```
import cv2
from detection import AccidentDetectionModel
import numpy as np
import os
import tkinter as tk
from tkinter import *
from tkinter import filedialog
from tkinter.filedialog import askopenfile
from PIL import Image, ImageTk

model = AccidentDetectionModel("model.json", 'model_weights.h5')
font = cv2.FONT_HERSHEY_SIMPLEX
my_w = tk.Tk()
my_w.geometry("410x300")
my_w.title('Accident Detection')
my_font1=('times', 18, 'bold')
l1 = tk.Label(my_w,text='Upload the Video',width=30,font=my_font1)
l1.grid(row=1,column=1,columnspan=4)
b1 = tk.Button(my_w, text='Upload File',
               width=20,command = lambda:open_file())
b1.grid(row=2,column=1,columnspan=4)
def open_file():
    filepath = tk.filedialog.askopenfilename()
    if filepath is not None:
        startapplication(filepath)

def startapplication(filepath):
    vid=open(filepath,'r')
    video = cv2.VideoCapture(vid.name)
    vid.close()
```

```

while True:
    ret, frame = video.read()
    if not ret:
        return prob
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    roi = cv2.resize(gray_frame, (250, 250))

    pred, prob = model.predict_accident(roi[np.newaxis, :, :])
    if(pred == "Accident"):
        prob = (round(prob[0][0]*100, 2))

        cv2.rectangle(frame, (0, 0), (280, 40), (0, 0, 0), -1)
        cv2.putText(frame, pred+" "+str(prob), (20, 30), font, 1, (255, 255, 0), 2)

    if cv2.waitKey(33) & 0xFF == ord('q'):
        return
    cv2.imshow('Video', frame)
    cv2.imwrite('D:/Final project/stream display.jpg', frame)
if __name__ == '__main__':
    open_file()

```

Mail.py

```
from twilio.rest import Client
import keys

def send_sms(msg):

    client = Client(keys.account_sid, keys.auth_token)
    message = client.messages.create(
        to=keys.target_number,
        from_=keys.twilio_number,
        body='accident detected')
send_sms("hi")
```

detection.py

```
from keras.models import model_from_json
import numpy as np

class AccidentDetectionModel(object):

    class_nums = ['Accident', 'No Accident']

    def __init__(self, model_json_file, model_weights_file):
        # load model from JSON file
        with open(model_json_file, "r") as json_file:
            loaded_model_json = json_file.read()
            self.loaded_model = model_from_json(loaded_model_json)

        # load weights into the new model
```

```
self.loaded_model.load_weights(model_weights_file)
self.loaded_model.make_predict_function()
```

```
def predict_accident(self, img):
    self.preds = self.loaded_model.predict(img)
    return AccidentDetectionModel.class_nums[np.argmax(self.preds)], self.preds
```

app.py

```
import streamlit as st
import os
```

```
st.title('Collision Detection')
```

```
up_vdo = st.file_uploader('Upload Video', type=['mp4'])
```

```
if up_vdo is not None:
    with open('./Videos/footage.mp4', 'wb') as f:
        f.write(up_vdo.getvalue())
```

```
if st.button('Detect'):
    os.system('py main.py')
```

B.SCREENSHOTS:

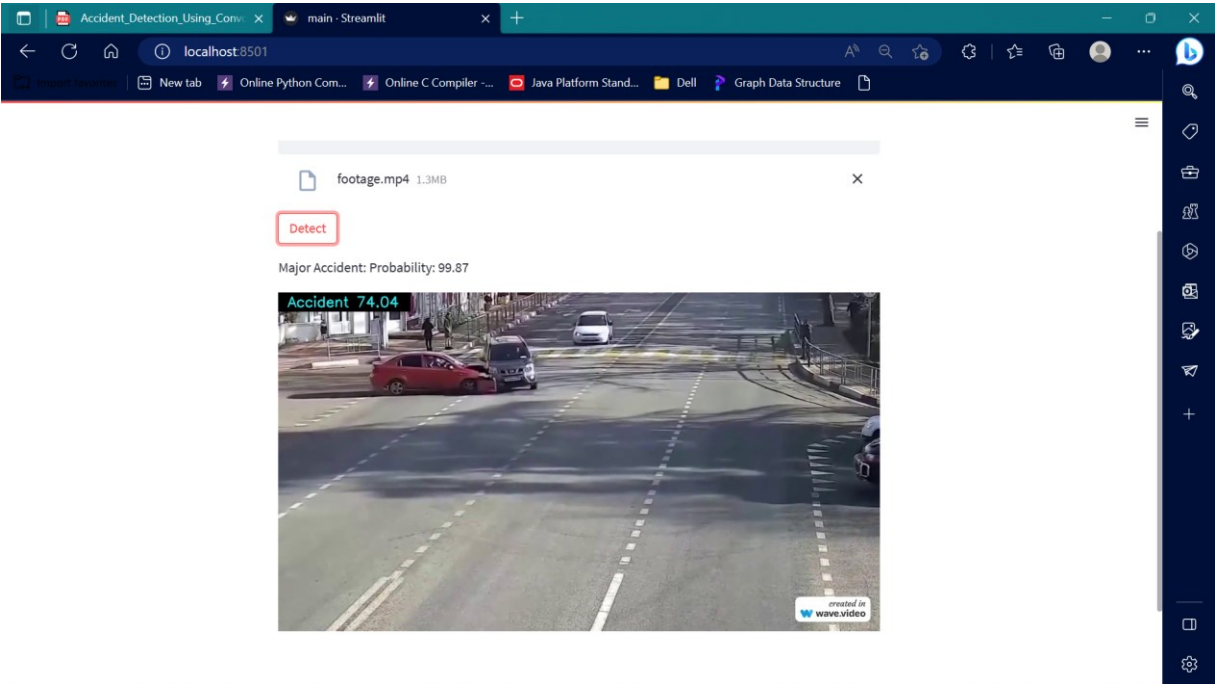


Fig 1: Image showing accident detected with probability of 99.87

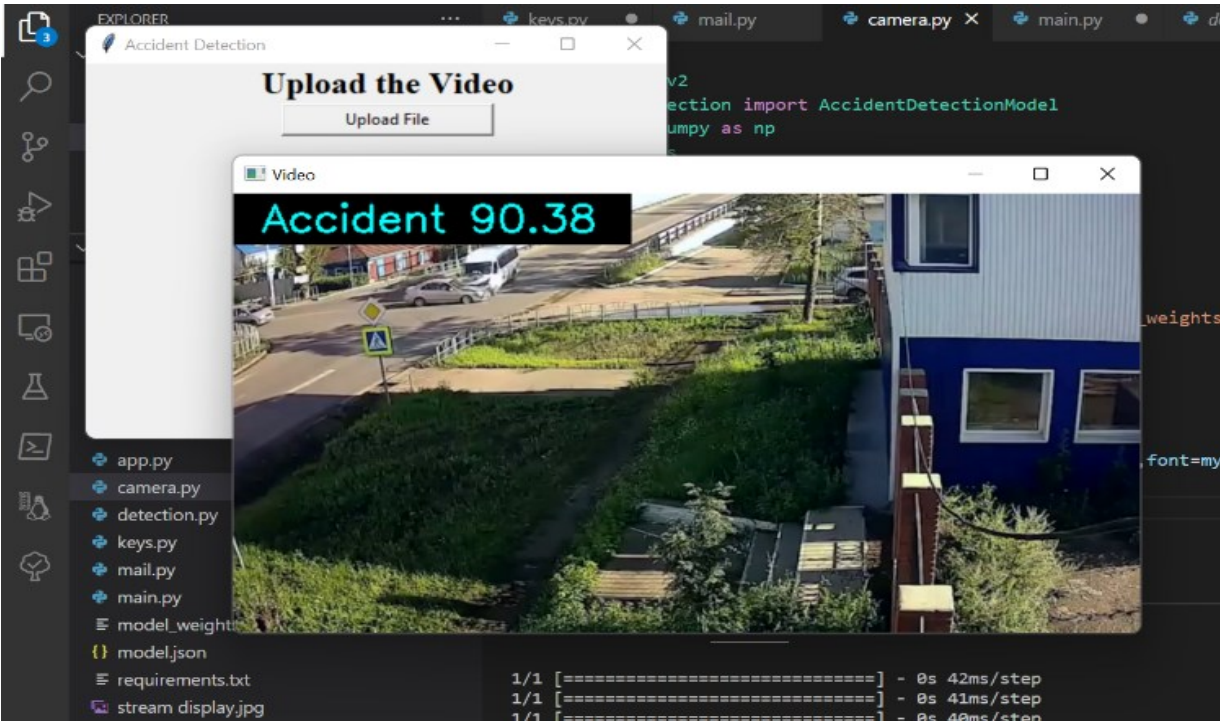


Fig 2: Image showing the accident detected with probability of 90.38

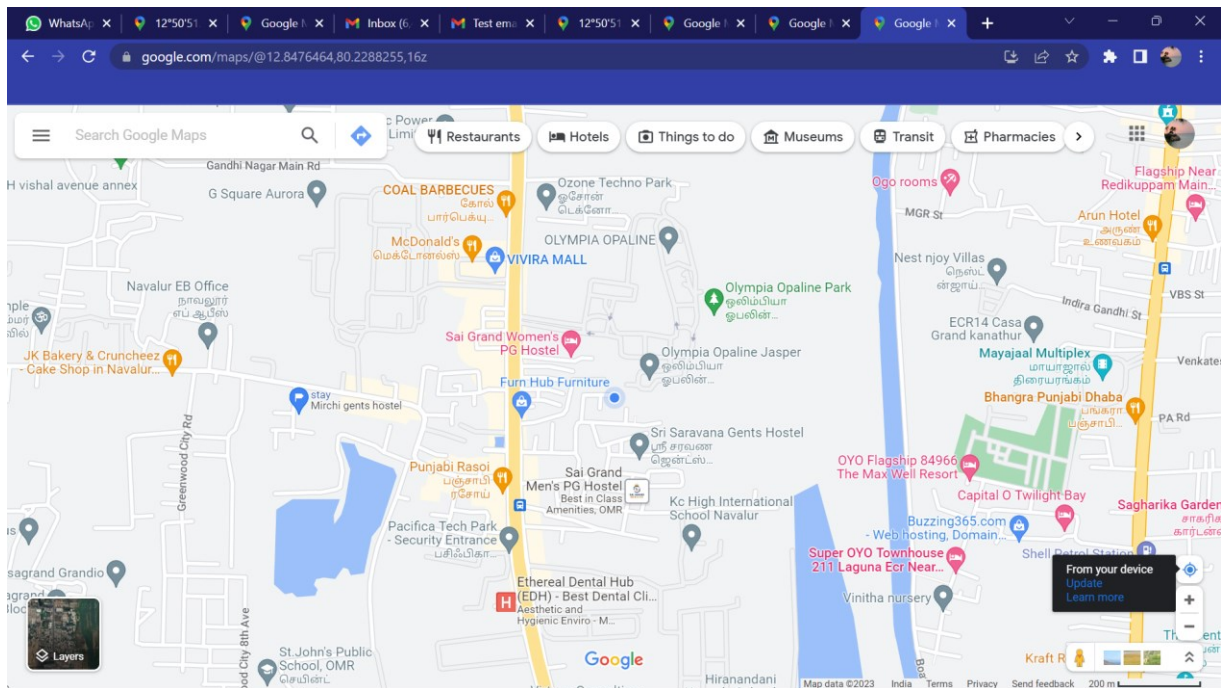


Fig 3: Image showing the location where the accident occurred

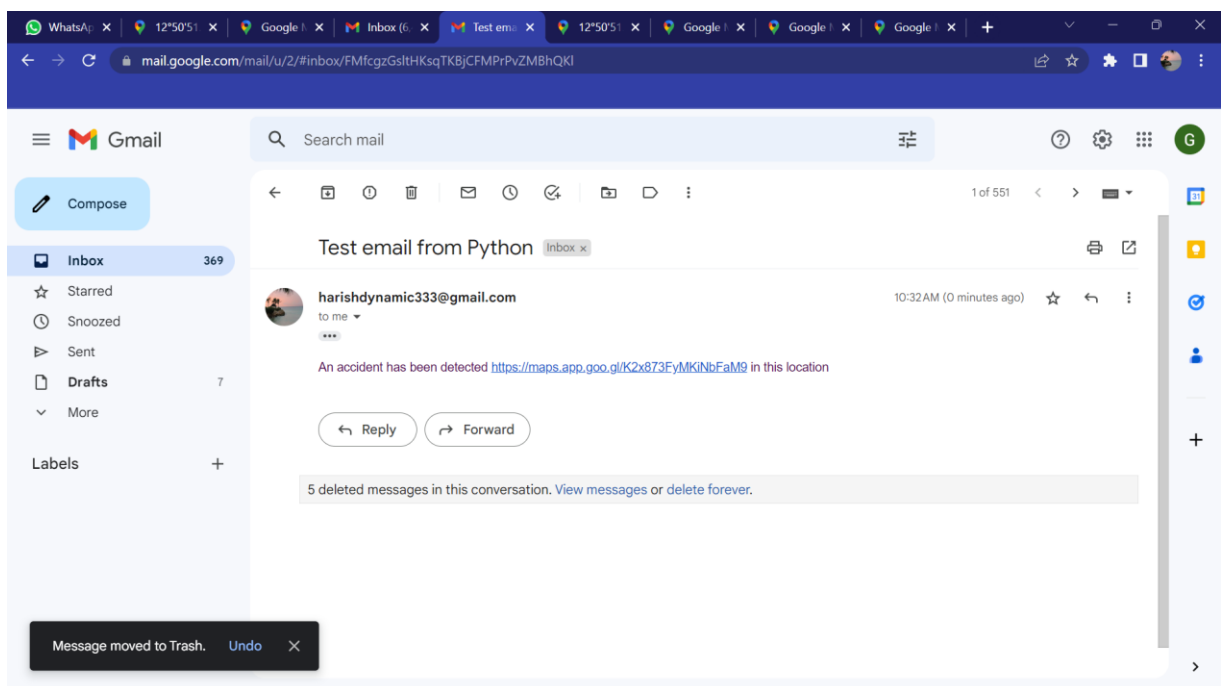


Fig 4: When the accident occurred then send a mail including location of accident

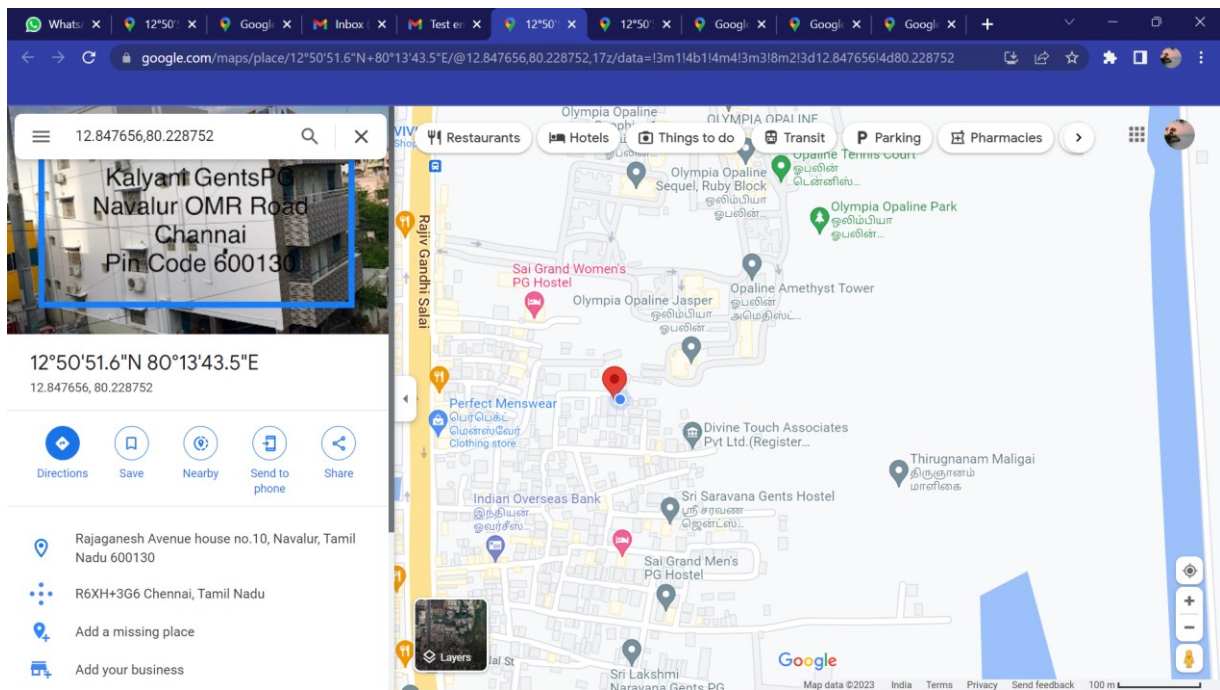


Fig 5: Image showing that the location of accident sent by mail

C.RESEARCH PAPER

Collision Detection using Convolutional Neural Networks

Harish K

Department of IT,

Sathyabama Institute of Science and Technology , Chennai , India

Email: harishdynamic333@gmail.com

Vamsi M

Department of IT ,

Sathyabama Institute of Science and Technology , Chennai , India

Email: mvamsimogalipuri@gmail.com

Jeyanthi P

Department of IT,

Sathyabama Institute of Science and Technology , Chennai , India

Email: jeyanthi.it@sathyabama.ac.in

Revathy S

Department of IT,

Sathyabama Institute of Science and Technology , Chennai , India

Mary Gladence L

Department of IT,

Sathyabama Institute of Science and Technology , Chennai , India

Geetha C

Department of IT,

Sathyabama Institute of Science and Technology , Chennai , India

ABSTRACT— In India, most causing death are nothing but accidents, and more than 85% of these deaths are caused by slow assistance rather than by the accidents themselves. Accident victims may go unattended for a long time on roadways with light and swift traffic, which leads to a high fatality rate. The goal is to develop a structure that can identify accidents instantly via a real-time feed from a Surveillance camera positioned on a public road. A convolutional neuronal network (CNN) model that has been trained to distinguish between frames related to accidents and frames unrelated to accidents would be used by the proposed system to evaluate each video frame. According to studies, CNNs are a quick and accurate method for identifying images. Image classifiers based on CNN have over 90% accuracy rates. Using this collision detection we can find the accident area and we can be sent help to the region which may help to get on time. This accident can happen at any place and at any time the help should reach there on time to help the victims.

Keywords: Accident Detection, CNN , Video recognition , Recurrent Neural Network; Geo-location module.

I. INTRODUCTION

Road accidents cause more than 1.3 million fatalities each year in addition to another 35 to 75 million individuals facing from minor wounds. According to a research on road traffic accidents based on a country's level of income, the world's health organization (WHO) discovered that developing nations have the highest rate of fatalities from road accidents. In contrast to a 11.3 per 100,000 individuals living in developed nations, the risk of dying from road accidents is approximately 27.5 per 100,000 citizens of developing nations. Over 90% of fatal traffic accidents occur in these countries, despite the fact that they only have 50 percent of the

world's automobiles. Particularly India is facing a dire situation as reports indicate 15 fatalities in traffic accidents .Road deaths are a serious issue, particularly in India, where 15 people die from them on average every hour. According to reports, 13 people are killed in road accidents in India every hour. Yet, since many accident cases go unreported, the worst-case situation might actually be far worse. According to the data currently available, India is on track to surpass all other nations in terms of the number of people who die in traffic accidents, with an average of 13 fatalities every hour (or 140,000 annually) [2]. A victim may typically be located during the first three stages of an accident.

In the first phase of an accident, which accounts for 10% of all fatalities, the accident victim passes away just minutes

or seconds after the accident.

This can be avoided if aid reaches the victims in a timely manner. Helping accident victims at this crucial time is the goal. The next phase that requires medical attention and money to prevent because it has a 15% mortality rate.

Road traffic deaths per 100 000 population, by WHO region

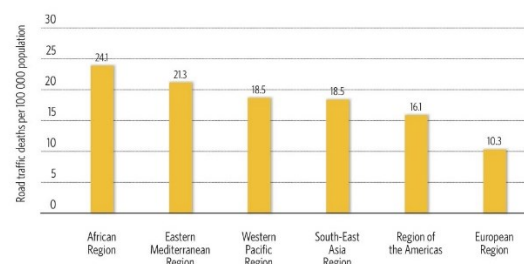


Fig.1 Analysis of death in different regions.

The goal is to implement a technology from video data and also by CCTV surveillance that can identify collision of the vehicles. Identifying an accident, then notifying the appropriate authorities, the technology is meant to serve as a tool to help accident victims in need. The objective is to evaluate frames taken from the video the camera generates with the help of DP(Deep Learning) Algorithms

which use neural networks based on convolution to recognise an accident in under seconds of it happening. We focused on implementing this advanced technologies on major roads where there isn't as much traffic and accident victims rarely receive prompt assistance. we can install surveillance cameras that serve as a medium for monitoring and are spaced about 500 metres apart. On these cameras, we can install the suggested system that uses the CCTV camera footage to run the proposed accident detection model and identify accidents.

The video and photo frames are made up of 10,000 serious crash frames and non-accident frames. Convolutional neural network(cnn) algorithm can now identify a picture or a movie frame as an accident frame to up to 98.5% accuracy.

TensorFlow, OpenCV, and Keras were then used to implement this model. The camera to view videos, and it analyses each frame to determine if it is an accident frame or not by running it through the generated model. A flashing emergency light also turns on.

Our system has a 95.0% accuracy rate for detecting accidents. It can be carried out using a Raspberry Pi, a small computer the size of a credit card. The created system can serve as a trustworthy source of data for automatically identifying accidents. This project would assist us in lowering the enormous number of fatalities associated with traffic accidents that take place in our nation.

II. Literature Survey

[1] Kaiming He et al used deep learning-based object techniques and his group, it is possible to locate and follow objects in real-time surveillance video. Video surveillance processes used in public spaces, such as on roads and highways, can be equipped with this technology to quickly identify accidents and other incidents. Moreover, their research on object recognition may be applied to create automated image analysis systems that can spot patterns in images that point to the an accident or even other crucial event. In general, Kaiming He has established the groundwork for the creation of technologies which are applicable to the detection of accidents as well as other related fields through his research in OpenCV.

[2] Hadi Rahmatollahi was a Researcher in a deep learning(DL). At the Alberta State University in Canada, he is presently a postdoctoral research fellow. In the year 2018, he and his colleagues wrote a paper for the Transactions on Intelligent Transportation Systems, which was named "Real-Time Accident Detection in Surveillance Footage Using A Convolutional Neural Network." For real-time collision avoidance in surveillance footage, they suggested a CNN-based model in their paper. The model outperformed other cutting-edge techniques and had a high accuracy rate for detecting accidents.

[3] Mehdi Mekni is an expert in machine learning and computer vision. At the California State University, Berkeley, he is a postdoctoral researcher at the moment. He and his co workers wrote a paper in the 2019 issue of IEEE Transactions on Intelligent Vehicles named "Deep

Learning for Autonomous Driving Safety: Overview, Opportunities and Challenges." They suggested a CNN-based framework in the paper for accident detection throughout automated driving systems. To identify potential accidents and gauge their severity, the model uses depth maps and optical flow to analyse data.

[4] Xiaojie Wang was a Assistant Professor and works at the Tennessee State University in Knoxville. From the California State University, San Diego, he earned his Doctor of philosophy in computer and electrical engineering in 2016. His areas of interest in research include deep learning, computer vision, and the use of these concepts in transportation systems. He has written a number of papers on the use of CNNs for the detection and classification of traffic accidents, which has the potential to increase traffic safety and lower the likelihood of accidents. Zhong-Qiu Zhao et al.

[5] Qifan Wang is a graduate student at Michigan State University. His areas of interest in research include image processing techniques, deep learning, and UAVs and surveillance systems. He has written a number of papers on the use of CNNs throughout UAV surveillance systems for real-time accident detection that have shown to be highly accurate and have the potential to increase drone operation safety.

[6] Abar H. Yousif and colleagues' study introduced a technique for creating an effective neural network that can precisely forecast this same severity of road accidents. This method, which may also be used in other nations, involves carefully choosing pertinent features to take into account knowledge acquired among both traffic accident and weather in Jordan.S.

[7] Ramya et al.well as her team of researchers looked into various elements that affect how serious accidents on the road are. Their research found that now the Random Forest Classifier's ability to take into account different seasons in vehicle accidents and fatalities makes it even more efficient than other methods at predicting accident severity.

III. SOFTWARE USED

A. OpenCV

The open-source image recognition library known as OpenCV, or simple for the Open Source Video Processing Library, offers programming capabilities to support CPU-intensive projects but also computer vision research. It is available for free use for both commercial and academic purposes under the BSD licence. Popular programming languages like Python, C++, and Java are supported by OpenCV, along with famous operating systems including Windows, Chrome Os, Linux, iOS, as well as Android. Real-time application development is made possible and computational efficiency is its main objective. Through with a GPU interface based on the OpenCL standard, OpenCV makes use of multi-core computation and GPU acceleration. Additionally, Tensor flow and other deep learning tools are supported by OpenCV. Ego motion analysis, technology for facial recognition, expression recognition, smart phone robotics, but also augmented reality are some of its main application areas.

Because of the size of our proposed system, we ran into a

All of the image sequences were de-noised using the aforementioned method before being fed to the model.

In python Keras is a free source library which enables extensive deep neural network experimentation. It just a component present in ONE IROS project, and updates are made frequently. The main benefit of the library to offer abstractions which are simple that make it easier to design and build deep-learning models without depending on a specific computing backend. Its purpose is to act as an interface for current machine learning systems. Working with image and text data makes use of Keras' implementation tools for building neural networks, which include activation functions, layers, as well as optimizers. Additional utility layers, including dropout and enable extensive deep neural network experimentation. It can function on top of Theano and Tensor flow, two additional Python libraries

A state-of-the-art system called You Only Look Once (YOLO) can detect objects in real time with astounding accuracy. YOLO performs object detection in a single neural network pass with a single forward direction, in contrast to conventional object detection methods that require additional iterations over an image. It predicts bounding boxes of all objects in a single evaluation and uses a convolutional neural network (CNN). In order to predict a certain amount of boxes with boundaries as well as the testing set for every box, YOLO divides this same picture into a series of cells. The bounding boxes' center coordinates, width, height, and confidence scores—which indicate how confidently and precisely they predict the presence of an object—are described.

Convolution layer, where it uses tiny given input sections to recognise visual features as well as maintain the interconnection between the pixels, is the first step inside the separation of image information. A mathematical operation that depends on a matrix representing a portion of an image as well as a filter or kernel is performed during this process.



Fig. 2. Describes the dimensions of image matrix.

YOLO has been used as a standard for picture classification for a while now. YOLO has the lowest error rate of all the benchmark CNN models for image categorization, as shown in table 1

Model	Error Rate
AlexNet	15.3%

videos. It also has the mega pixel of 5 camera with capability of 720p60, 1080p30 in addition to image capture. With the ribbon cable of 15cm. You can modify module of camera by add a variety of other effects, such as time-lapse or slow-motion. It's also possible to build effects using additional libraries.

B. Multi purpose Internet Mail Extensions

With the help of the MIME email can now include content other than plain text. MIME kinds are used to specify a file or message's format and make sure that email servers and clients handle it appropriately.

Using MIME to have included formatting in an email response is referred to as including "MIME text" in this context. This enables the sender to add pictures, video content, and other multimedia content in addition to using different text formatting options like bold, italics, and underlining. The HTML information is then MIME-encoded.

C. Geolocation Module.

It is useful Applications in other programming languages as well as web browsers include the geolocation module, which enables websites or users to find a device's location. The existing latitude, longitude, & altitude of a device can be collected using tools and APIs available to developers. By offering location-based facilities like nearby restaurants, weather reports, and location-based advertisements, web developers frequently use the geolocation module to improve user experiences. The Geolocation API provides various methods and properties to connect the geolocation module as well as access the user's location data. Applications for the geolocation module span the IoT, mapping services, and mobile app development. However, developers must abide by the relevant regulations and laws to protect user privacy because the use of geolocation data raises privacy concerns. The Python library for the Pi camera can be used to access it. The camera module is utilised in both wildlife camera traps and a variety of home security applications.

VII. TRAINING THE MODEL

The Data Generator of the picture in Keras, that allows for the image augmentation, was used to train all the layers as indicated in Table 2. The process of using a collection of images to artificially enlarge the dataset is known as image augmentation. The most often used aspects of picture augmentation include zoom, shear, and pre-processing functions. By essentially applying random shear, rotation, and zoom to the current images, these factors provide model with learning more from more data. Following data were utilized to create suggested shearangle, model/Rescale, zoomrange, horizontalflip.

The following results were obtained when trained by 500 epochs:

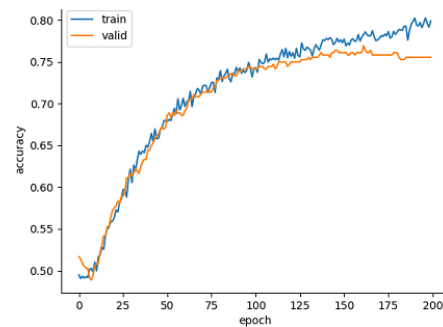


Fig. 6.1 Validation and Training losses

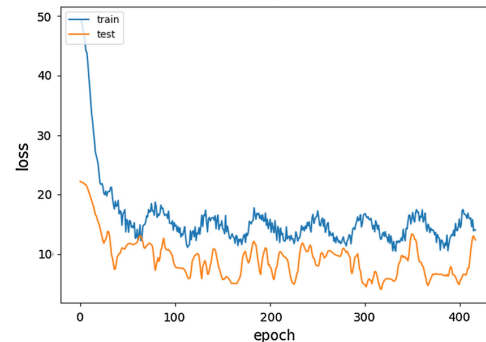


Fig. 6.2. Accuracy of Validation and Training
Accuracy of Validation -0.86

Accuracy of Training -0.96

VIII. RESULTS

When the application starts, it analyses every frame of the video taken by CCTV surveillance camera, procedures using by suggested models. If an accident is found, immediately sends an alert through into the GSM module. Additionally, it transmits the accident's severity and the frame at which it was discovered. Additionally, the accident detection's time stamp is shown. Figure 8 shows the accident screen and the corresponding details.

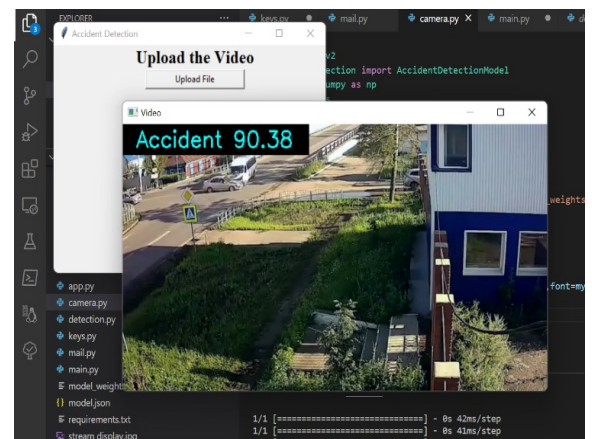


Fig. 7. Accident Detection of Real Time

The situation shown in the next frame is one in which only the percentage representing the accident's severity is shown.

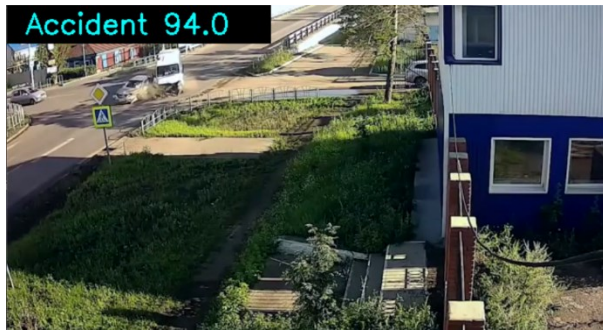
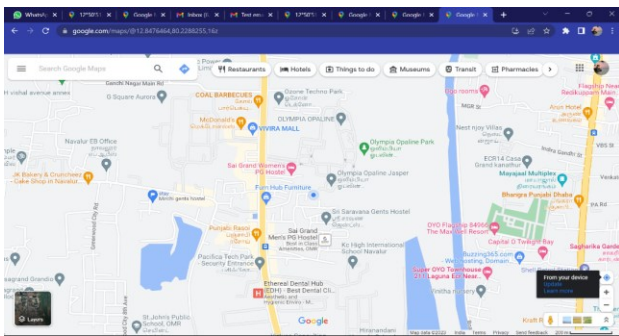
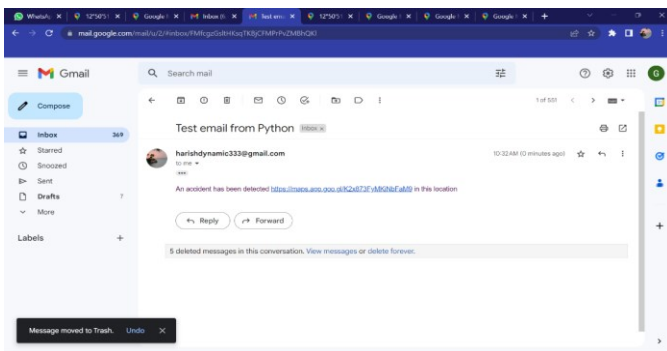


Fig. 8 Image of an accident detection

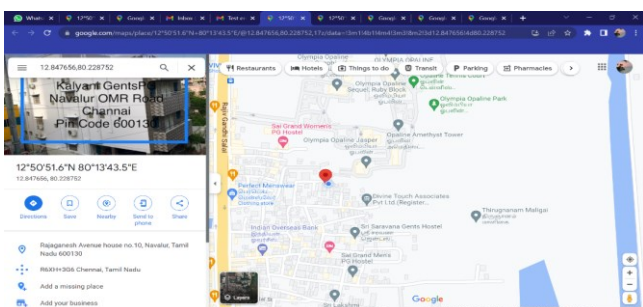
The following information was used to evaluate the model's performance using the metric "Accuracy": Accuracy is equal to number of correct prediction divided by total fall cases to be predicted.



When the accident detected then it will automatically redirect to the exact location where the accident has been occurred .



Then send a mail that the accident has occurred along with the location where the accident occurred.



The accuracy of the operational model ranged from 84% to a peak value of 97.76%. The model's accuracy was above 92.38% on average.

IX. CONCLUSION

Accidental death and property loss occur on a daily basis, trying to make them one of humanity's most prevalent problems. The suggested technique offers a workable and effective remedy for this problem. The proposed car crash detection system can identify collisions as they take place and immediately send an SMS to hospitals in the area and police stations with details like the time and location. The proposed system uses a model-based methodology rather than pricey sensors and extraneous hardware, making it more affordable, dependable, and reliable than its rivals. Images were used in the experimental work, running tests, and verification of this system, then the final output increases the prediction of the accident. As a result, it demonstrates that installing this application over most of the local and state options was a practical decision.

REFERENCES

1. "3D Object Detection for Autonomous Driving: A Survey," arXiv preprint, 2021, by Rui Qian, Xin Lai, and Xirong Li.
2. "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Method Exploiting Sequential Data," arXiv preprint, 2021. X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, et al.
3. "Detection of Collision-Prone Vehicle Behavior at Intersections Using Siamese Interaction LSTM," IEEE Transactions on Intelligent Transportation Systems, pp. 1-11, 2020. D. Roy, T. Ishizaka, C. K. Mohan, and A. Fukuda.
4. "3D Object Detection for Autonomous Driving: A Survey," arXiv preprint, 2021, by Rui Qian, Xin Lai, and Xirong Li.
5. "Detection of Coordinate Based Accident-Prone Regions on Road Surface Using Machine Learning Techniques," International Journal of Computer Engineering and Information Technology, vol. 12, March 2020, pp. 19–25.
6. "Traffic Accidents Severity Prediction Using Support Vector Machine Models," International Journal of Innovative Technology and Exploratory Engineering, vol. 9, no. 7, May 2020, pp. 1345–1350. Zeinab Farhat, Ali Karouni, Pierre Chauvet, Bassam Daya, and Nizar Hamadeh.
7. "Prediction of Accident Prone-Zone and AlertMessages to the Hospitals Using Machine Learning and Internet of Things," International Journal of Engineering Research & Technology (IJERT), vol. 04, no. 05, May 2020.
8. "Traffic Density Investigation and Road Accident Analysis in India using Deep Learning," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 501–506, by C. Manchanda, R. Rath, and N. Sharma.
9. "UB-LSTM: A Trajectory Prediction Technique Coupled with Vehicle Behavior Recognition", Journal of Advanced Transportation, vol. 2020, pp. 1–12, August 2020, by Haipeng Xiao, Chaoqun Wang,

Zhixiong Li, Rendong Wang, Cao Bo, Miguel Angel Sotelo, et al.

10. "Road Accident Analysis and Prediction of Accident Severity Using Machine Learning," International Journal of Engineering Research & Technology (IJERT), vol. 07, no. 12, December 2020, pp. 740-747. Akanksha Jadhav, Shruti Jadhav, Archana Jalke, and Kirti Suryavanshi.

