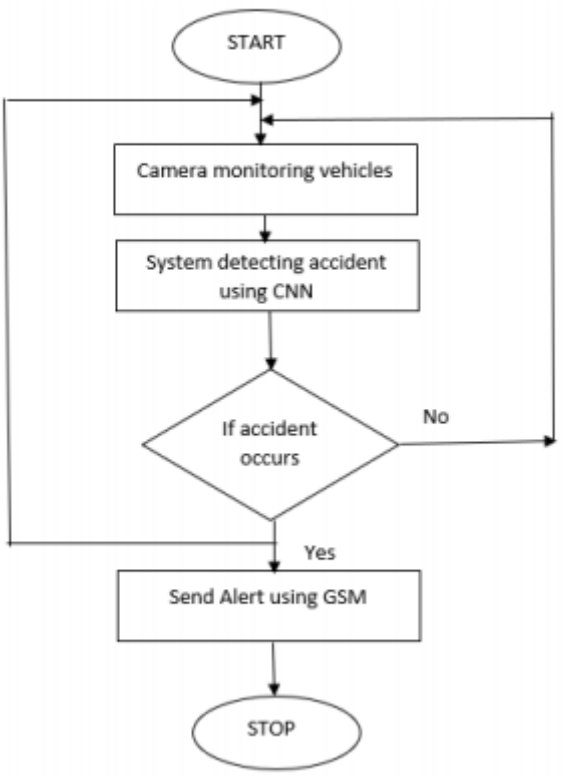
ACCIDENT DETECTION SYSTEM USING MACHINE LEARNING

1.INTRODUCTION

Road accidents in India are a major cause of decreasing life expectancy with road accidents contributing to over 148,000 deaths out of 467,000 deaths in 2016. Indian Economy has a hit of 3 percent of GDP growth due to road accidents as per the United Nations with an estimated loss of $58,000 in terms of value every year. The metropolitan cities such as Chennai, Mumbai and New Delhi have been increasingly highlighted for lack of road safety and rash driving cases. The recent trends show that there has been an increase in the global number of road accidents even in developed countries. However, under-developed and developing countries suffer a more significant impact due to life and economic losses. These accidents occur due to violation of traffic safety rules, careless rash driving, driver drowsiness and lack of good quality roads. The problem becomes more adverse for highways and hilly areas where accidents are unavoidable. Road accidents are characterized by high death rates due to delay in arrival of help and inefficient systems of mitigation to alert the concerned authorities. Road accidents on the highways are typically caused by natural reasons such as extreme weather conditions such as fog and consecutive collision of vehicles are common on Indian highways due to lack of visibility. The states of Maharashtra, Tamil Nadu and Uttar Pradesh account for the highest number of road accidents in India.The problem can be handled by making use of computer vision and low-cost sensor networks. The current solutions involved heavy dependency on sensor networks and area coverage. This can be substantially replaced by making use of object detection and image segmentation for accident classification. The system identifies the accident-prone areas which are the target stakeholders for the deployment and sets it apart from other implementations since it provides a feasibility factor associated with it. Furthermore, the system provides enhanced mitigation alert to the concerned authorities which helps in preventing any consecutive collisions that could possibly lead to greater loss of lives .

1.1 Structure of Project



2 .LITERATURE SURVEY

Various literature papers were studied and analyzed to understand their work and techniques. Thus, we studied the demerits and merits of various ideas related to accident detection. In one paper , two phases are used, an accident detection phase and an accident prevention phase. The authors have mentioned that they used IR sensors and Arduino Uno technology. But in , they do not provide an accurate result and also the sensors are costly. In another paper, an accident detection system using Inertial Measurement Unit (IMU) and 3G cellular module using an accident detection method, but the IMU suffer accumulated error and it is complex. Another technology discussed in the paper was a vision-based accident detection system for detecting, recording, and reporting accidents at intersections. They extract features of moving vehicles using cameras in order to detect accidents.

Machine Learning Classifiers:

These are used to predict the class/target/labels/categories of a given data points. Classification belongs to the category of supervised learning in which the targets are provided with input data. They are used in many applications like medical diagnosis, spam detection, target marketing etc. They use a mapping function (f) from input variables (X) to discrete output variables(Y).

Opencv:

OpenCV is an open-source library which is primarily used for Computer Vision Applications. This contains many functions and algorithms for Motion tracking, Facial recognition, Object Detection, Segmentation and recognition and many other applications. Images and real time video streams can be manipulated to suit different needs using this library.

Tensorflow:

It is an open-source machine learning framework to build and train neural networks. It has a collection of tools, libraries and community resources which helps in easy building of deployment of ML powered applications. This is developed and maintained by Google and was released in 2015.

3.PROBLEM ANALYSIS

3.1 EXISTING SYSTEM

Two phases are used, an accident detection phase and an accident prevention phase. The authors have mentioned that they used IR sensors and Arduino Uno technology. But in they do not provide an accurate result and also the sensors are costly. In another paper an accident detection system using Inertial Measurement Unit (IMU) and 3G cellular module using an accident detection method, but the IMU suffer accumulated error and it is complex.

3.1.1 Disadvantages

- The rapid growth of technology has made everything more facile and this advancement in technology additionally increased accidents. Due to this delayed medical attention, the accident victims might die as well.

3.2 PROPOSED SYSTEM

Since there was no dataset available, a dataset was created that includes accident and non-accident images. If an accident occurs, an alert message will be sent to the nearby control unit. We trained the system with the created dataset. The trained system is then incorporated with the cameras so as to capture the video of the vehicles on the road. By calculating the probability, the system predicts whether an accident happened or not. In case of an accident, an alert is sent to the control rooms using the GSM module. Fig. 1 is the flowchart depicting working of the system. The camera module records the video of vehicles in the road. The camera is placed at fixed locations, mostly in accident-prone areas. Whenever an accident occurs, it is predicted using our deep learning model and followed by sending alert message to the nearby control rooms.

3.2.1 Advantages:

- The working of the system is based on deep learning techniques that use convolutional neural networks. By utilizing this system, many people can be saved from death.

3.3 Software Requirements

For developing the application the following are the Software Requirements:

1. Python
2. Django
3. Mysql
4. Wampserver

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor: Pentium IV or higher
- RAM: 256 MB
- Space on Hard Disk: minimum 512MB

3.4 ALGORITHM

Algorithm Used : Convolutional nueral network.

There are two main parts to a CNN architecture

- A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction
- A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

Convolution Layers

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

1. Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

2. Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer

3. Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

4. Dropout

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

5. Activation Functions

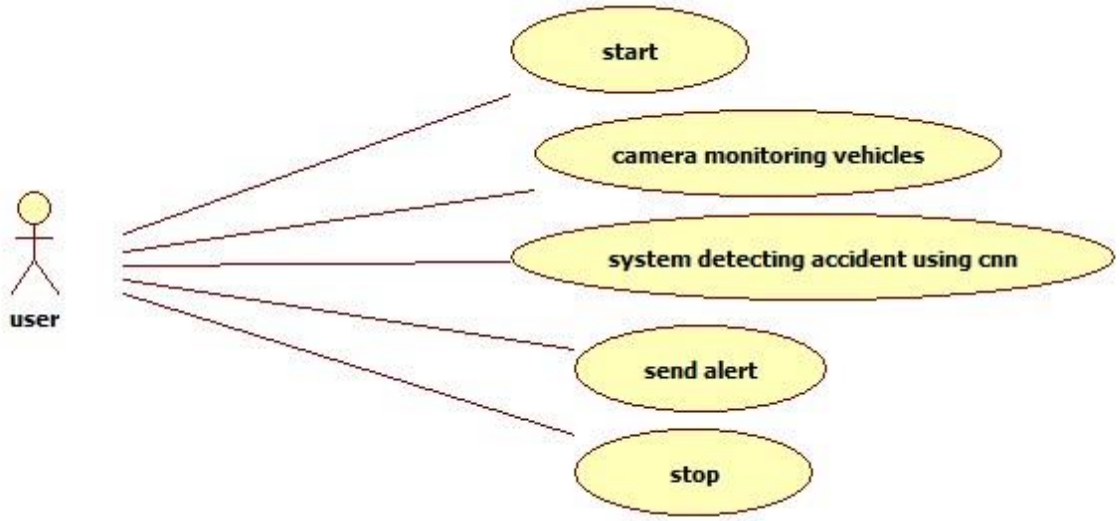
Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred an for a multi-class classification, generally softmax us used.

4.UML DIAGRAMS

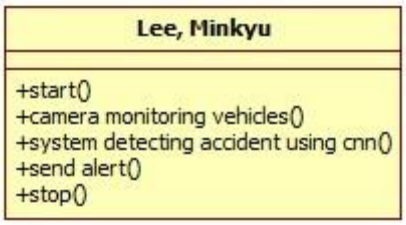
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



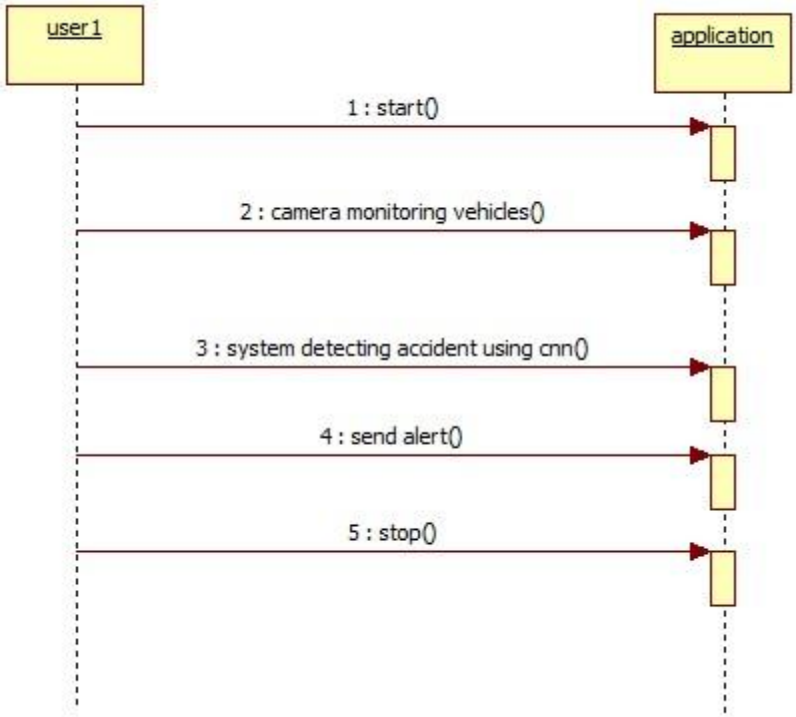
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



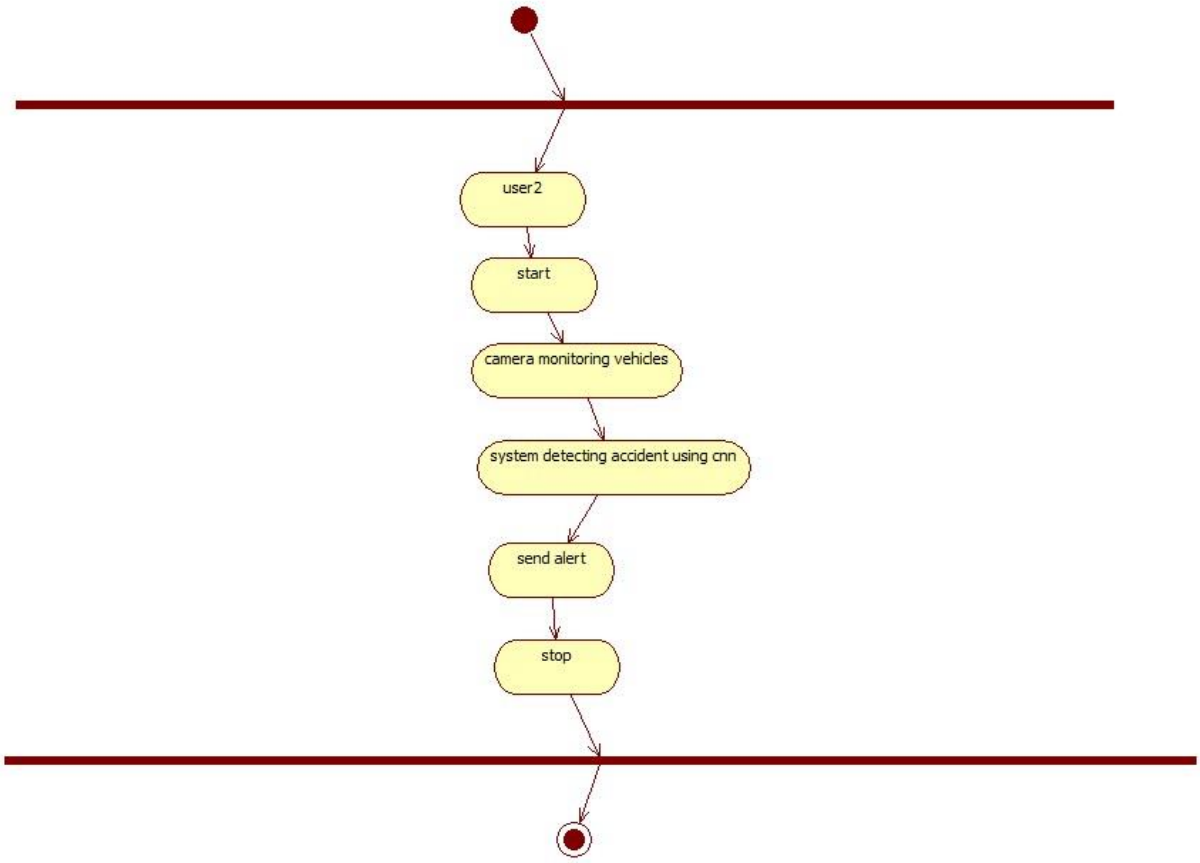
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



5.IMPLEMENTATION ON (PYTHON):

Python is a general purpose, dynamic, high level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is *easy to learn* yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and *dynamic typing* with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports *multiple programming pattern*, including object-oriented, imperative, and functional or procedural programming styles.

Python is not intended to work in a particular area, such as web programming. That is why it is known as *multipurpose* programming language because it can be used with web, enterprise, 3D CAD, etc.

We don't need to use data types to declare variable because it is *dynamically typed* so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging *fast* because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

Python 2 vs. Python 3

In most of the programming languages, whenever a new version releases, it supports the features and syntax of the existing version of the language, therefore, it is easier for the projects to switch in the newer version. However, in the case of Python, the two versions Python 2 and Python 3 are very much different from each other.

A list of differences between Python 2 and Python 3 are given below:

1. Python 2 uses **print** as a statement and used as print "something" to print some string on the console. On the other hand, Python 3 uses **print** as a function and used as print("something") to print something on the console.

2. Python 2 uses the function raw_input() to accept the user's input. It returns the string representing the value, which is typed by the user. To convert it into the integer, we need to use the int() function in Python. On the other hand, Python 3 uses input() function which automatically interpreted the type of input entered by the user. However, we can cast this value to any type by using primitive functions (int(), str(), etc.).
3. In Python 2, the implicit string type is ASCII, whereas, in Python 3, the implicit string type is Unicode.
4. Python 3 doesn't contain the xrange() function of Python 2. The xrange() is the variant of range() function which returns a xrange object that works similar to Java iterator. The range() returns a list for example the function range(0,3) contains 0, 1, 2.
5. There is also a small change made in Exception handling in Python 3. It defines a keyword **as** which is necessary to be used. We will discuss it in Exception handling section of Python programming tutorial.

Python Applications

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

Here, we are specifing applications areas where python can be applied.

1) Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser etc. It also provides Frameworks such as Django, Pyramid, Flask etc to design and delelop web based applications. Some important developments are: PythonWikiEngines, Pocoo, PythonBlogSoftware etc.

2) Desktop GUI Applications

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

3) Software Development

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

4) Scientific and Numeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

5) Business Applications

Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform.

6) Console Based Application

We can use Python to develop console based applications. For example: **IPython**.

7) Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

8) 3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

9) Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

10) Applications for Images

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.

5.1 CODE

```
from tkinter import messagebox

from tkinter import *

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

from tkinter.filedialog import askopenfilename

import time

import cv2

import tensorflow as tf

from collections import namedtuple

from collections import defaultdict

from io import StringIO

from PIL import Image

import numpy as np

import winsound


main = tkinter.Tk()

main.title("Accident Detection")

main.geometry("1300x1200")


global filename

global detectionGraph
```

```
global msg


def loadModel():
    global detectionGraph
    detectionGraph = tf.Graph()
    with detectionGraph.as_default():
        od_graphDef = tf.GraphDef()
        with tf.gfile.GFile('model/frozen_inference_graph.pb', 'rb') as file:
            serializedGraph = file.read()
            od_graphDef.ParseFromString(serializedGraph)
            tf.import_graph_def(od_graphDef, name='')

    messagebox.showinfo("Training model loaded","Training model loaded")



def beep():
    frequency = 2500  # Set Frequency To 2500 Hertz
    duration = 1000  # Set Duration To 1000 ms == 1 second
    winsound.Beep(frequency, duration)


def uploadVideo():
    global filename
    filename = filedialog.askopenfilename(initialdir="videos")
```

```
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");


def calculateCollision(boxes,classes,scores,image_np):
    global msg
    #cv2.putText(image_np, "NORMAL!", (230, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.0,
(255, 255, 255), 2, cv2.LINE_AA)
    for i, b in enumerate(boxes[0]):
        if classes[0][i] == 3 or classes[0][i] == 6 or classes[0][i] == 8:
            if scores[0][i] > 0.5:
                for j, c in enumerate(boxes[0]):
                    if (i != j) and (classes[0][j] == 3 or classes[0][j] == 6 or classes[0][j] == 8) and
scores[0][j]> 0.5:
                        Rectangle = namedtuple('Rectangle', 'xmin ymin xmax ymax')
                        ra = Rectangle(boxes[0][i][3], boxes[0][i][2], boxes[0][i][1], boxes[0][i][3])
                        rb = Rectangle(boxes[0][j][3], boxes[0][j][2], boxes[0][j][1], boxes[0][j][3])
                        ar = rectArea(boxes[0][i][3], boxes[0][i][1],boxes[0][i][2],boxes[0][i][3])
                        col_threshold = 0.6*np.sqrt(ar)
                        area(ra, rb)
                        if (area(ra,rb)<col_threshold) :
                            print('accident')
                            msg = 'ACCIDENT!'
```

```
                    beep()
                    return True
                else:
                    return False



def rectArea(xmax, ymax, xmin, ymin):
    x = np.abs(xmax-xmin)
    y = np.abs(ymax-ymin)
    return x*y

def load_image_into_numpy_array(image):
  (im_width, im_height) = image.size
  return np.array(image.getdata()).reshape((im_height, im_width, 3)).astype(np.uint8)

def area(a, b):  # returns None if rectangles don't intersect
    dx = min(a.xmax, b.xmax) - max(a.xmin, b.xmin)
    dy = min(a.ymax, b.ymax) - max(a.ymin, b.ymin)
    return dx*dy

def detector():
    global msg
```

```
    msg = ''
    cap = cv2.VideoCapture(filename)
    with detectionGraph.as_default():
        with tf.Session(graph=detectionGraph) as sess:
            while True:
                ret, image_np = cap.read()
                image_np_expanded = np.expand_dims(image_np, axis=0)
                image_tensor = detectionGraph.get_tensor_by_name('image_tensor:0')
                boxes = detectionGraph.get_tensor_by_name('detection_boxes:0')
                scores = detectionGraph.get_tensor_by_name('detection_scores:0')
                classes = detectionGraph.get_tensor_by_name('detection_classes:0')
                num_detections = detectionGraph.get_tensor_by_name('num_detections:0')
                (boxes, scores, classes, num_detections) = sess.run([boxes, scores, classes, num_detections], feed_dict={image_tensor: image_np_expanded})
                calculateCollision(boxes, classes, scores, image_np)
                cv2.putText(image_np, msg, (230, 50), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (255, 0, 0), 2, cv2.LINE_AA)
                cv2.imshow('Accident Detection', image_np)
                if cv2.waitKey(25) & 0xFF == ord('q'):
                    cv2.destroyAllWindows()
                    break
```

```
def exit():
    main.destroy()



font = ('times', 16, 'bold')
title = Label(main, text='Accident Detection')
title.config(bg='light cyan', fg='pale violet red')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Load & Generate CNN Model", command=loadModel)
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='light cyan', fg='pale violet red')
pathlabel.config(font=font1)
pathlabel.place(x=460,y=100)

webcamButton = Button(main, text="Browse System Videos", command=uploadVideo)
```

```
webcamButton.place(x=50,y=150)

webcamButton.config(font=font1)


webcamButton = Button(main, text="Start Accident Detector", command=detector)

webcamButton.place(x=50,y=200)

webcamButton.config(font=font1)


exitButton = Button(main, text="Exit", command=exit)

exitButton.place(x=330,y=250)

exitButton.config(font=font1)



font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=250)

text.config(font=font1)



main.config(bg='snow3')

main.mainloop()
```

6.TESTING

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

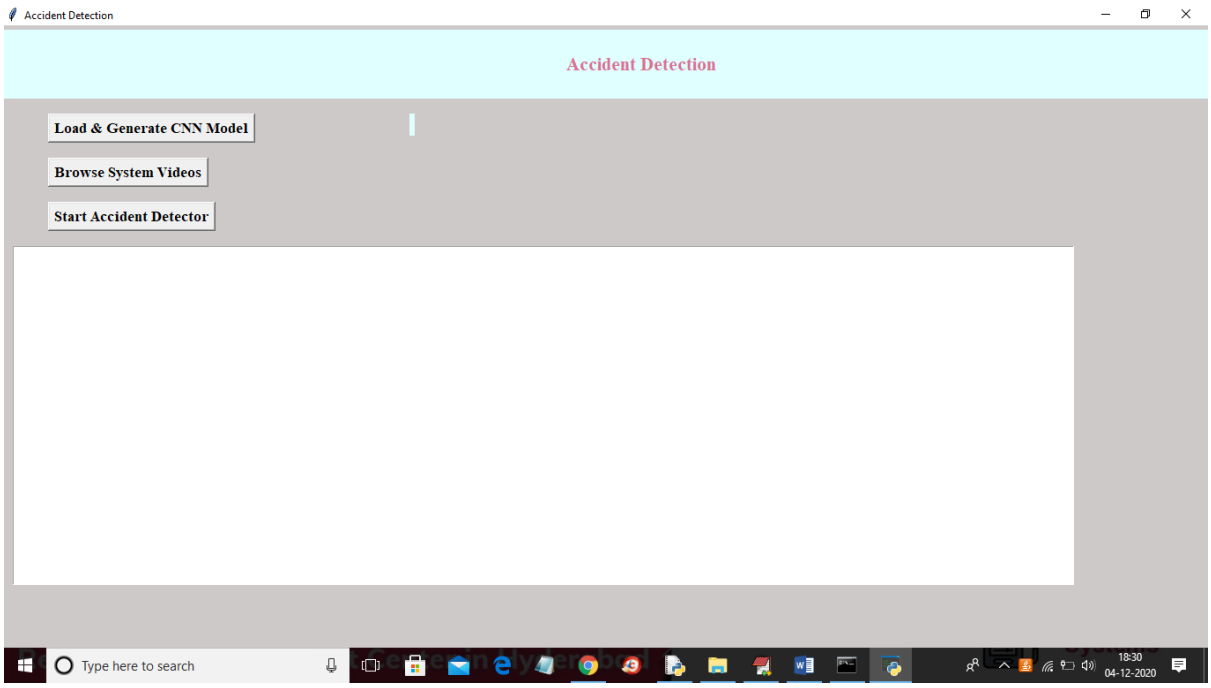
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

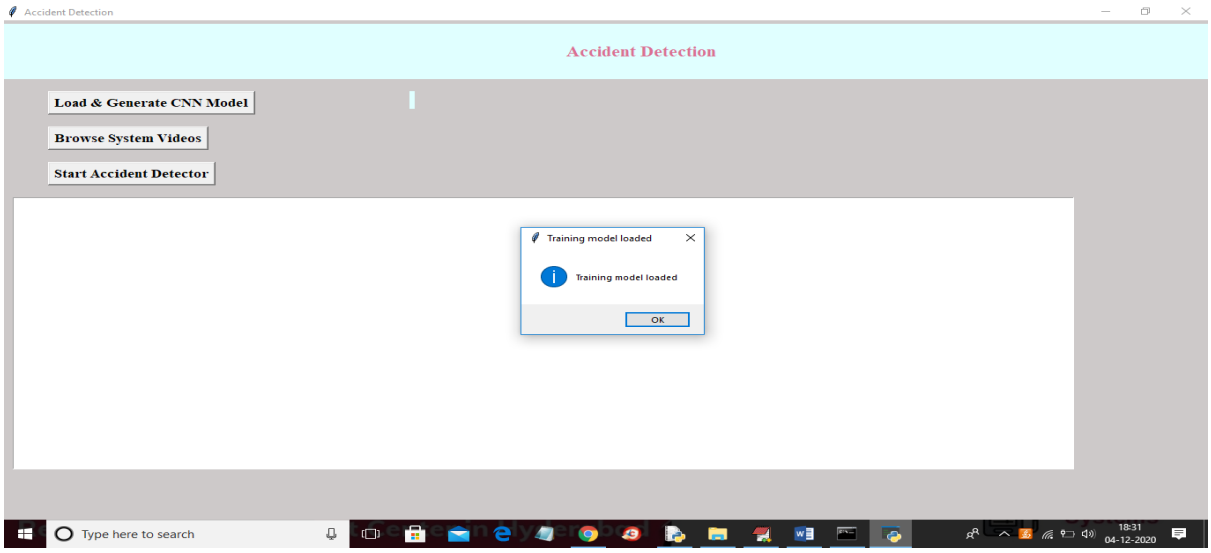
7.RESULTS

This project is trained with images where vehicles collided and accident occurred and in test video if anything such collision happens between vehicles then application detect as accident. Training is done with tensorflow and CNN Algorithm.

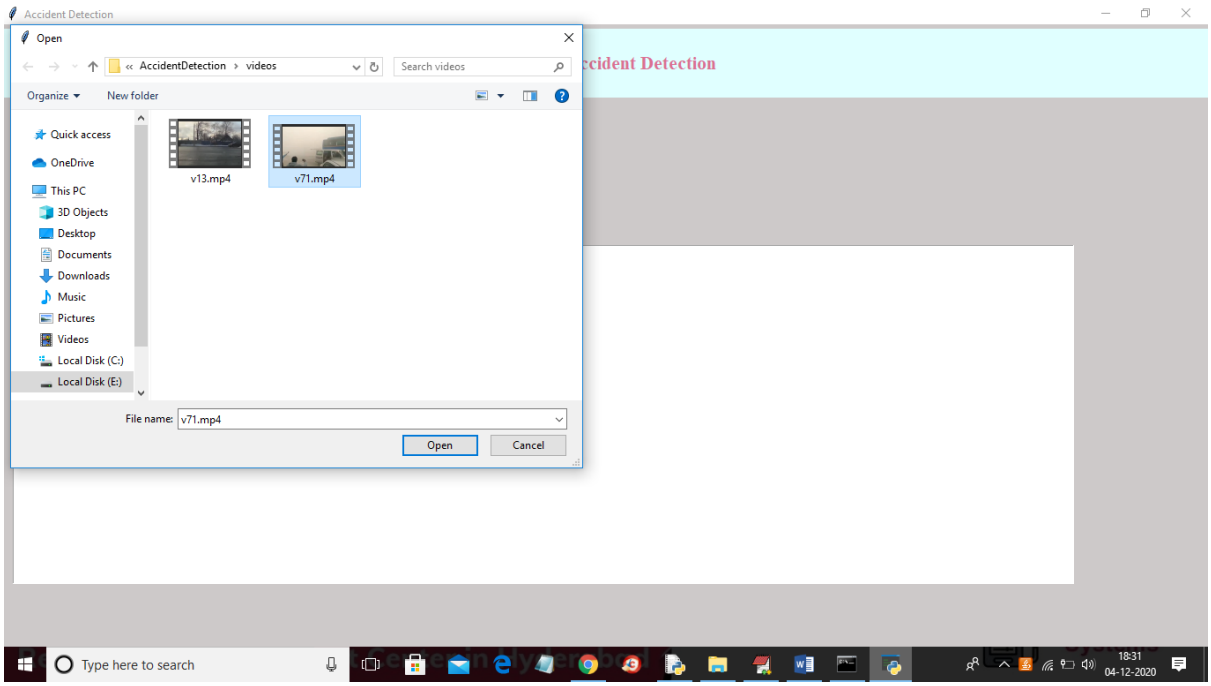
To run project double click on run.bat file to get below screen



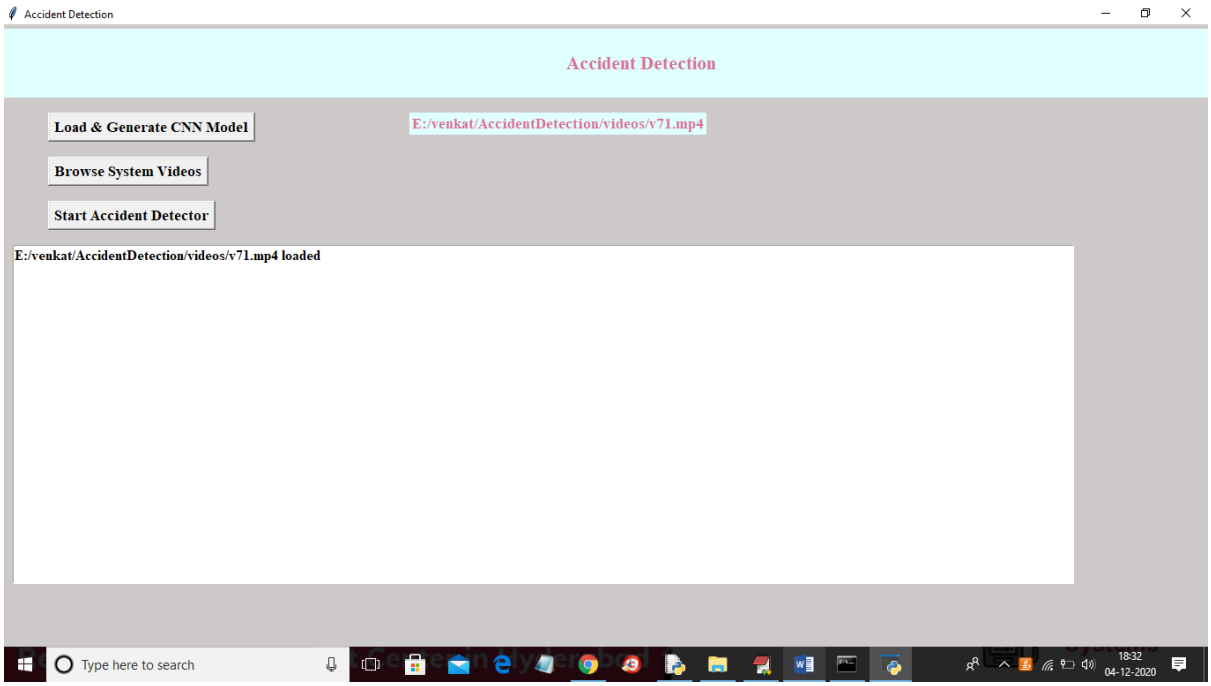
In above screen click on 'Load & Generate CNN Model' button to trained CNN with dataset and to load CNN model using tensorflow



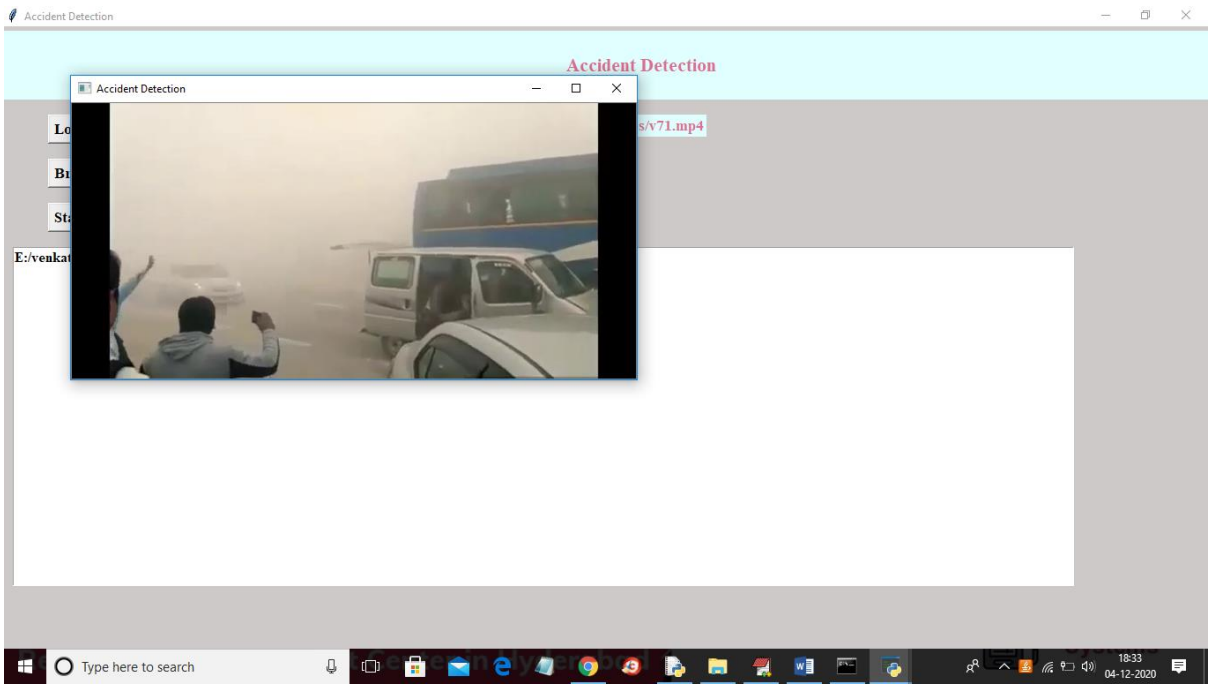
In above screen tensorflow model is loaded and now click on ‘Browse System Video’ button to upload video



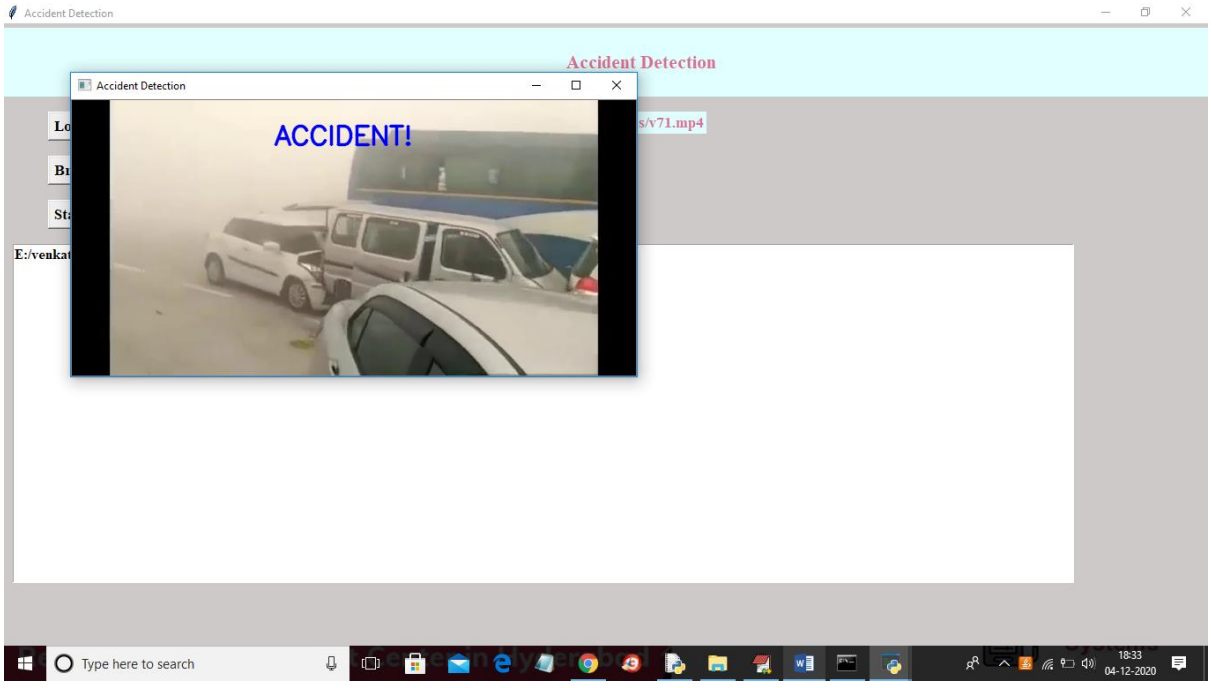
In above screen selecting and uploading video and then click on ‘Open’ button to load video



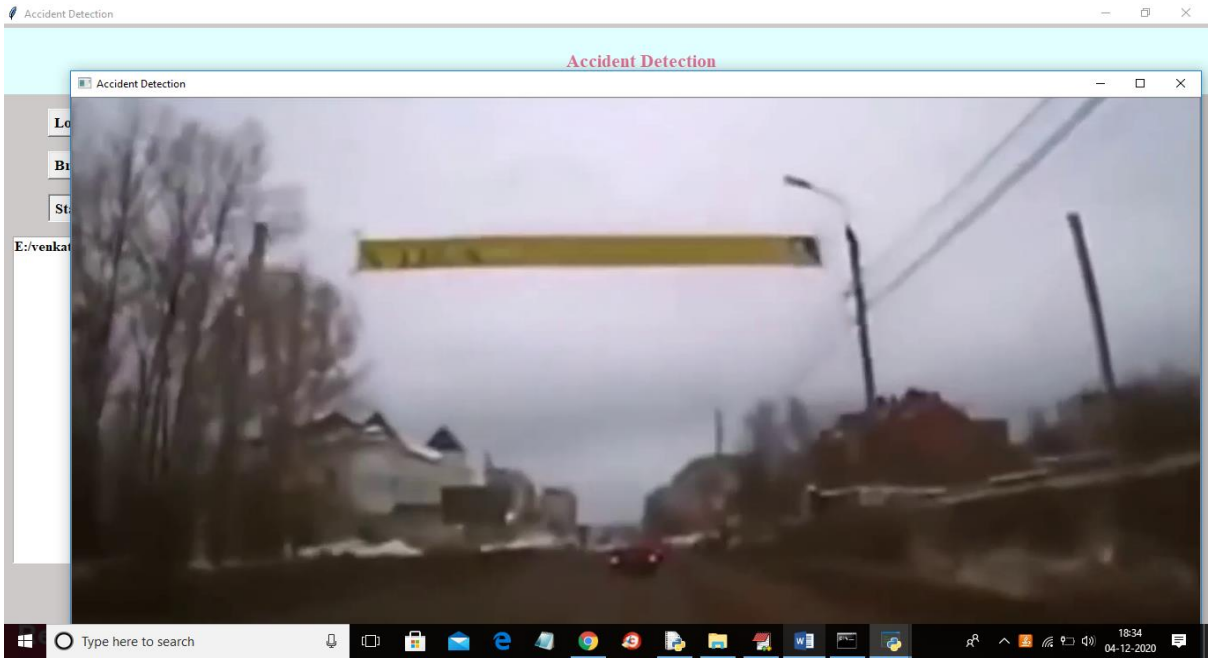
In above screen video is loaded and now click on ‘Start Accident Detector’ button to play video and detect accident

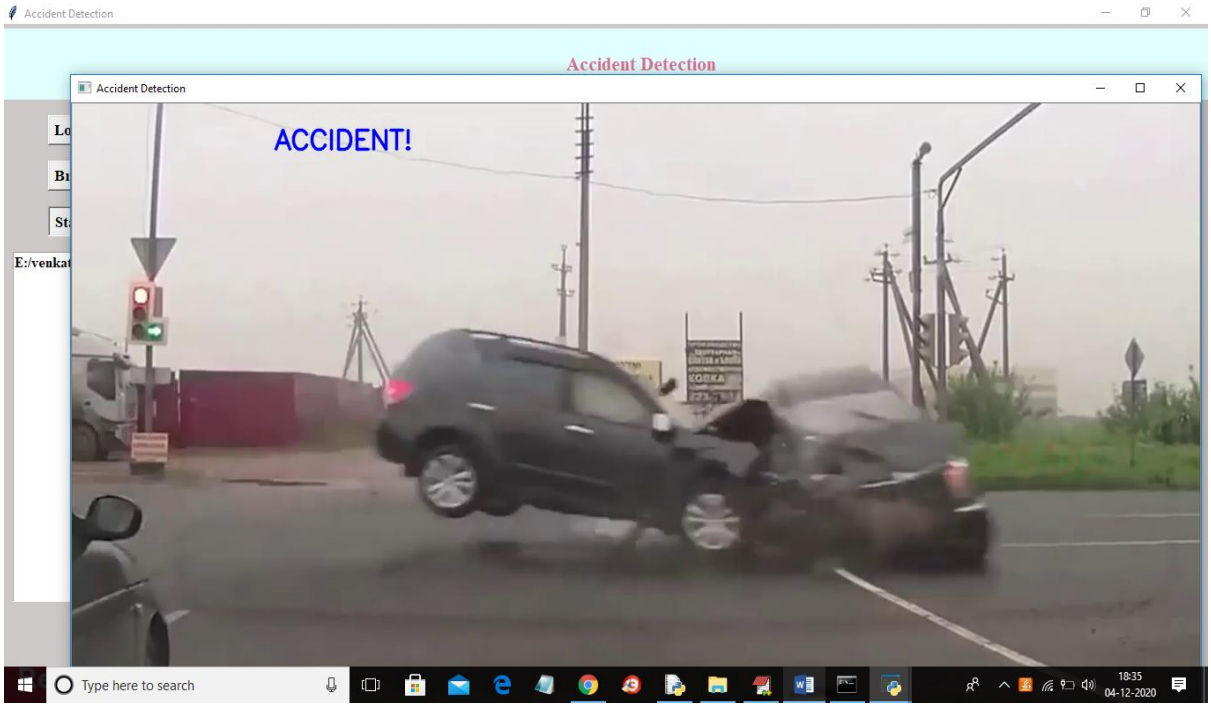


In above screen video start playing and upon accident detection will get below screen with beep sound



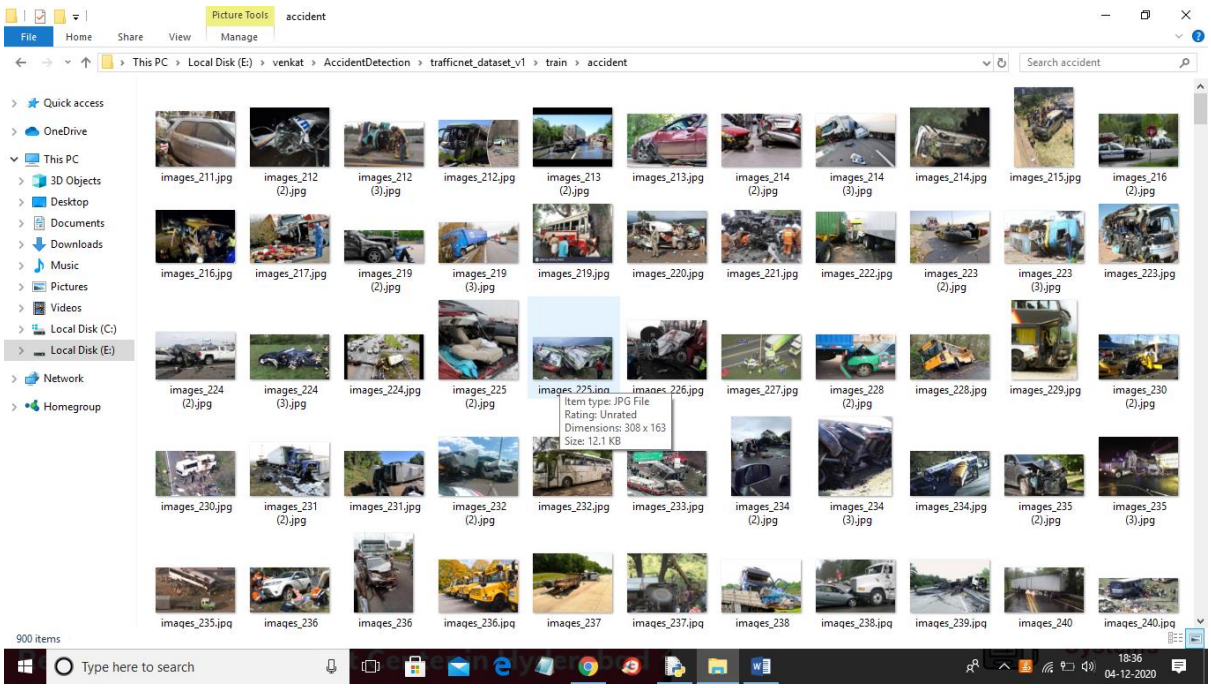
In below screen playing another video without message if normal driving appear





In above screen upon collision then accident display message will appear with beep sound

In below screen application is trained with below images


Picture Tools
accident
File
Home
Share
View
Manage
This PC › Local Disk (E:) › venkat › AccidentDetection › trafficnet_dataset_v1 › train › accident
Search accident
Quick access
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Network
Homegroup
images_211.jpg
images_212 (2).jpg
images_212 (3).jpg
images_212.jpg
images_213 (2).jpg
images_213.jpg
images_214 (2).jpg
images_214 (3).jpg
images_214.jpg
images_215.jpg
images_216 (2).jpg
images_216.jpg
images_217.jpg
images_219 (2).jpg
images_219 (3).jpg
images_219.jpg
images_220.jpg
images_221.jpg
images_222.jpg
images_223 (2).jpg
images_223 (3).jpg
images_223.jpg
images_224 (2).jpg
images_224 (3).jpg
images_224.jpg
images_225 (2).jpg
images_225.jpg
images_226.jpg
images_227.jpg
images_228 (2).jpg
images_228.jpg
images_229.jpg
images_230 (2).jpg
Item type: JPG File
Rating: Unrated
Dimensions: 308 x 163
Size: 12.1 KB
images_230.jpg
images_231 (2).jpg
images_231.jpg
images_232 (2).jpg
images_232.jpg
images_233.jpg
images_234 (2).jpg
images_234 (3).jpg
images_234.jpg
images_235 (2).jpg
images_235 (3).jpg
images_235.jpg
images_236
images_236
images_236.jpg
images_237
images_237.jpg
images_238
images_238.jpg
images_239.jpg
images_240
images_240.jpg
900 items
Type here to search
18:36
04-12-2020

CONCLUSION

The proposed system is used to detect road accidents. When an accident is detected, an alert message is sent to nearby control rooms using the GSM module. This system is more reliable and economical when compared to existing systems. It can detect accidents with high level of accuracy as the model architecture is trained using the created dataset. Our preliminary evaluation shows that the system works in a perfect manner and can be deployed over a large area. With the help of this system, immediate action can be taken by sending alert to the officials and will help the medical teams to reach the accident spot in time and save the valuable human lives. Thus, the proposed system will play an important role in the society where road accidents have nowadays become a major threat.

REFERENCES

[1] "Global status report on road safety 2015", World Health Organization, 2019. [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/20 15/en/. [Accessed: 07- Mar- 2019].

[2] Prabakar, S., et al. "An enhanced accident detection and victim status indicating system: Prototype." India Conference (INDICON), 2012 Annual IEEE. IEEE, 2012.

[3] "Lexus Enform", Lexus, 2019. [Online]. Available: https://www.lexus.com/enform. [Accessed: 07- Mar- 2019].

[4] "OnStar Safety and Security Services", Onstar.com, 2019. [Online]. Available: https://www.onstar.com/us/en/services/safety-security/. [Accessed: 07- Mar- 2019].

[5] "SOSmart automatic car crash detection and notification app", SOSmart automatic car crash detection app, 2019. [Online]. Available: http://www.sosmartapp.com. [Accessed: 07- Mar- 2019].

[6] C. Kockan, “Communication between vehicles” PhD thesis, Istanbul Technical University, 2008

[7] Zeng, Yuanyuan, Deshi Li, and Athanasios V. Vasilakos. "Opportunistic fleets for road event detection in vehicular sensor networks." Wireless Networks 22.2 (2016): 503-521.