PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

Project Report (Phase-1)

on

Accident Detection and Reporting.

Submitted by

C Shashank (PES1PG22CA051)

Oct 2023 – Jan 2024

under the guidance of

Guide Details

Dr. Premalatha H M

Associate Professor

Department of Computer Applications,

PESU, Bengaluru – 560085

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER APPLICATIONS
PROGRAM – MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled

Accident Detection and Reporting.

is a bonafide work carried out by

C Shashank – PES1PG22CA051

in partial fulfillment for the completion of Capstone Project, Phase-1 work in the Program of Study MCA under rules and regulations of PES University, Bengaluru during the period Oct. 2023 – Jan 2024. The project report has been approved as it satisfies the academic requirements of 3rd semester MCA.

Internal Guide
Dr. Premalatha H M
Associate Professor.
Department of Computer Applications,
PES University, Bengaluru - 560085

Chairperson
Dr. Veena S
Department of Computer Applications,
PES University, Bengaluru - 560085

DECLARATION

I, **C Shashank,** bearing **PES1PG22CA051** hereby declare that the Capstone project phase-1 entitled, **Accident Detection and Reporting***,* is an original work done by me under the guidance of **Dr. Premalatha H M,** Associate Professor, PES University, and is being submitted in partial fulfillment of the requirements for completion of 3rd Semester course in the Program of Study **MCA**. All corrections/suggestions indicated for internal assessment have been incorporated in the report.

PLACE:

DATE:

C Shashank

ABSTRACT

Automobile has a great importance in our daily life. We utilize it to go to our work place, keep in touch with our friends and family, and deliver our goods. But it can also bring disaster to us and even can kill us through accidents. Speed is one of the most important and basic risk factors in driving.

It not only affects the severity of a crash, but also increases risk of being involved in a crash.

Despite many efforts taken by different governmental and non-governmental organizations all around the world by various programs to aware against careless driving, yet accidents are taking place every now and then. However, many lives could have been saved if the emergency service could get the crash information in time. As such, efficient automatic accident detection with an automatic notification to the emergency service with the accident location is a prime need to save the precious human life.

Contents

Page no.

1.INTRODUCTION **1-3**

1.1 PROJECT DESCRIPTION

Problem Definition 1

Proposed Solution 2

Purpose 3

Scope 3

2. LITERATURE SURVEY **4-7**

2.1 DOMAIN SURVEY 4

2.2 RELATED WORK 4-6

2.3 EXISTING SYSTEMS 7

3. HARDWARE AND SOFTWARE REQUIREMENTS **8**

3.1 HARDWARE REQUIREMENTS

3.2 SOFTWARE REQUIREMENTS

4. SOFTWARE REQUIREMENTS SPECIFICATION **9-10**

4.1 USERS 9

4.2 FUNCTIONAL REQUIREMENTS 9

4.3 NON-FUNCTIONAL REQUIREMENTS 10

5. SYSTEM DESIGN **11-12**

5.1 ARCHITECTURE DIAGRAM 11

5.2 BLOCK DIAGRAM / CONTEXT FLOW DIAGRAM 12

6. DETAILED DESIGN **13-14**

6.1 PROCESS FLOW DIAGRAM 13

6.2 USE CASE DIAGRAM 14

7. IMPLEMENTATION 15-21

7.1 SCREEN SHOTS

Appendix A BIBLIOGRAPHY (References) 22

Chapter 1

Introduction

1.1 Project Description

1.1.1 Problem Scenario

The problem of deaths and injuries as a result of accidents is to be a global phenomenon. Traffic safety has been a serious concern since the start of the automobile age, almost one hundred years ago. It has been estimated that over 300,000 persons die and 10 to 15 million persons are injured every year in road accidents throughout the world. Statistics have also shown that mortality in road accidents is very high among young adults that constitute the major part of the work force. In order to overcome this problem, there is need of various road safety strategies and measure. Losses in road accidents are unbearable, to the society as well as a developing country like us. According to the world Health organization 2022 records accidents are the most hazardous events and one of the leading causes of death, disability and injury. Keeping in mind these events can't be dissolved completely but we can take steps to help and reduce the response time to the accidents using emerging technology deep learning.

.

1.1.2 Proposed Solution

In this project the main purpose is to develop an accurate accident detection and reporting system utilizing Convolutional Neural Networks (CNN) within the domain of deep learning. The project revolves around the analysis of accident and non -accident datasets for detection of accident from the video footage.

CNN model developed here can

1.detect live accident

2.Provide an alert message which includes exact time date and location of the incident to the most proximate control room so the rescue becomes very fast

3.Alert messages will be sent to selected contacts to inform about accident

Successful implementation of this CNN-based approach has the potential to accurately detect the accident. Future work may involve refining the model architecture and exploring additional avenues for improving predictive accuracy.

1.1.3 Project Purpose:

The purpose of the project is to develop a robust and accurate accident detection and Reporting system utilizing deep learning algorithms and computed accident and non-accident dataset as input. This initiative aims to enhance detection of road accidents and reduce the response time for the rescue.

Through the utilization of state-of-the-art deep learning techniques, the project seeks to analyze accident and non-accident images with a high level of precision, identifying subtle patterns and anomalies that exist. By leveraging the power of artificial intelligence, we aim to establish a reliable predictive model capable of assisting emergency services to reach out to the accident location and timely rescue of causalities on road.

1.1.4 Project Scope:

- The system's accuracy may vary in different environmental conditions, such as adverse weather (heavy rain, snow, fog) or poor lighting conditions. These factors can impact the effectiveness of sensors like cameras
- Deploying accident detection systems involves collecting and transmitting data, raising privacy concerns.
- System detects accidents which occurs on road
- The accuracy of the system heavily relies on the capabilities of the camera used. camera may have limitations in terms of range, resolution, and sensitivity.

Chapter 2

Literature Survey

2.1 Domain Study

The domain associated with the project is machine learning and computer vision. These two fields are advantageous for the system to process the images and identify the patterns in order to compute the distance and classify the images into accident and non-accident classes. Machine learning helps in building various modules which can learn itself from the datasets consisting of labels. The computer vision on the other hand helps for making the accident prediction on the video footages.

2.2 Related work

Title: Accident Detection and Alert System

Author: Mahendra Vucha

Year of publication: 2019 | IJITEE

Summary:

In this paper, they designed a system to control accidents by sending SMS to registered numbers and medical centers without human intervention. Soon after the accident, the vibration sensor gets activated it intimates to the microcontroller which is connected with GPS and GSM(Global System for Mobile Communications.) modules, where GPS finds the location and information is intimated through GSM by sending a notification message to the medical shops and registered numbers automatically.

Title: IoT Based Automatic Accident Detection & Rescue Management In Vanet

Author: Koushik S

Year of publication: 2019 | SSRG International Journal of Computer Science and Engineering

Summary:

This system is designed to detect an accident at non- network places using VANET. VANET is nothing but a vehicle Ad hoc network.VANETs are a type of mobile ad hoc network that specifically focuses on communication between vehicles.It works effectively and efficiently in non-network areas, Once an accident is detected, the notification is transferred to nearby traveling vehicles using the RF module.Nearby traveling vehicles within the range of the RF module acknowledge the accident notification.The acknowledgment continues until the notification reaches a vehicle within a network area.when a vehicle carrying the notification enters an area with network coverage. then dispatched to a base station. Which is then forwarded to relevant authorities.

Title: Vehicle Accident Detection System by using GSM AND GPS

Author: Mithun Haridas

Year of publication:2020 | International Research Journal of Engineering and Technology (IRJET)

Summary:

Vibration sensors are employed to detect signs of an accident. These sensors can pick up sudden and abnormal vibrations that may occur during a collision. The vibration sensors relay the detected information to a microcontroller. The microcontroller serves as the central processing unit, analyzing the data and determining if it corresponds to an accident .Upon recognizing an accident, the microcontroller initiates an alert. The cancel switch provided for driver to take decision Depending on the decision made by the driver (pressing cancel switch or not), the system may activate or deactivate the process of sending alert.The system utilizes a GSM (Global System for Mobile Communications) module for wireless communication module allows the system to send SMS messages to predefined services or contacts.

Title:Vehicle Accident Detection and Alerting System

Author: Hemangi S.Ahire

Year of publication: 2018 | International Journal for Research in Applied Science & Engineering Technology

Summary:

Vibration sensors are employed to detect signs of an accident. These sensors can pick up sudden and abnormal vibrations that may occur during a collision. A microcontroller or similar processing unit is responsible for processing data from the vibration sensor and coordinating the communication with the RF transmitter and GPS tracking system.When an accident is detected, the system creates a data packet containing relevant information. The RF transmitter sends the data packet to predefined services, which could include home, hospital, and police. A GPS (Global Positioning System) tracking system is integrated into the system. This allows accurate location information to be included in the accident report.

Title: Automatic Accident Detection and Alerting System Based on IoT

Author: Jithin P Babu

Year:2020 | International Journal of Innovative - Research in Computer

Summary:

The accelerometer sensor detects sudden changes in acceleration, which can be indicative of an accident. This serves as the primary sensor for accident detection Arduino UNO Microcontroller acts as the brain of the system, making decisions based on the input from sensors . A cancel switch provides a way for the victim to signal that the accident is not severe.If the cancel switch is not pushed within 15 seconds, the system considers the accident as severe.The Arduino UNO sends notifications to the, possibly emergency services or predefined contacts.Arduino UNO uses GPS to determine the accident spot accurately

2.3 EXISTING SYSTEMS

The existing system for accident detection and reporting relies on eyewitnesses or involved parties to call emergency services. This can result in delays, especially in remote areas or during low-traffic periods when accidents may go unnoticed for extended periods.

Human reporting can be subjective and prone to errors. Eyewitnesses might not provide accurate details, leading to challenges for emergency responders in assessing the severity of the accident and dispatching appropriate resources. Existing system might not cover all the areas equally. Remote or less populated areas may lack proper monitoring infrastructure, leaving them vulnerable to delayed response time.

Chapter 3

Hardware and Software Requirements

3.1 Hardware Requirements

RAM : 4 GB RAM(Minimum)

HDD : 80 GB HDD or above

Processor : Dual Core processor –intel i3 or higher.

3.2 Software Requirements

Operating System: windows 7 or higher

Libraries: Numpy,Cv2,Tensorflow

Image processing libraries: PIL

Visualization tool: Matplotlib

IDE: Jupyter notebook,

PycharmLanguage: Python

3.11.5

Front end :Tkinter

API : Twillio

Chapter 4

Software Requirements Specification

4.1 Different users with explanation

Emergency Services : can be used by relevant authorities for detecting accidents and provide assistance.

Fleet management companies: can use real-time accident detection systems to monitor their vehicles and drivers.

Insurance companies: for monitor driving behavior.

4.2 FUNCTIONAL REQUIREMENTS

Data collection: This involves collection of two classes of data being accident and non-accident using which the CNN model will be trained.

Pre-processing: involves processing the collected data which includes

1) counting the dataset :counting the number of accident and non-accident dataset.
2) RGB To Grayscale: converting RGB image to grayscale image.
3) scaling images: Scaling the pixel values between 0 to 1.
4) resizing the image: Resizing the image to 224 X 224 format
5) resizing the shape: Resizing the Boundary of the images
6) creating matrices to store processed image and assign labels

Training Model: IN this process processed images along with labels are given as the input for CNN model for training purpose .Train size being 80 % and test size being 20 %.

Testing Model : Here the fully trained CNN model is given the testing images for prediction.

Prediction :It makes predictions on new and unseen image data.

Reporting Model: After the Prediction Phase if the CNN model labels the input as accident then the messages is sent to the Emergency services Using Twilio.

4.3 NON-FUNCTIONAL REQUIREMENTS

Performance: The system shall detect accidents accurately and promptly, with a response time of [specified time].

Reliability : The system shall have a high reliability rate, minimizing false alarms and ensuring accurate accident detection.

Security: The system shall ensure the confidentiality and integrity of the transmitted data, employing appropriate security measures.

Usability: The system shall have a user-friendly interface, making it easy for users to interact with the system

Chapter 5

System Design

5.1 Architecture Diagram

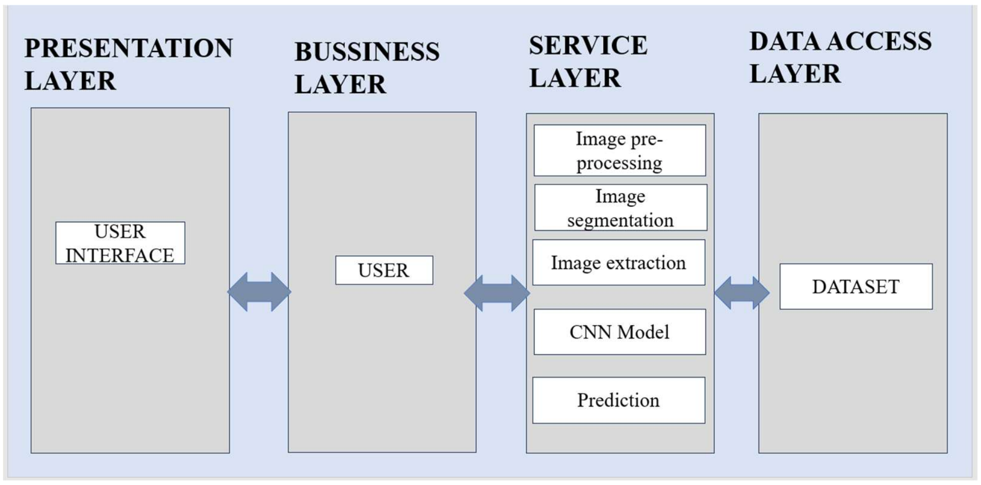


Figure 5.1: Architecture Diagram

The Frontend section is composed of the User Interface, which is responsible for providing an intuitive and user-friendly interface for the user to interact with the Accident detection and prediction system. The User is a main actor in the system. The Services section is responsible for handling Image preprocessing, image segmentation, image extraction and it uses CNN model for predicting the result.

5.2 CONTEXT DIAGRAM

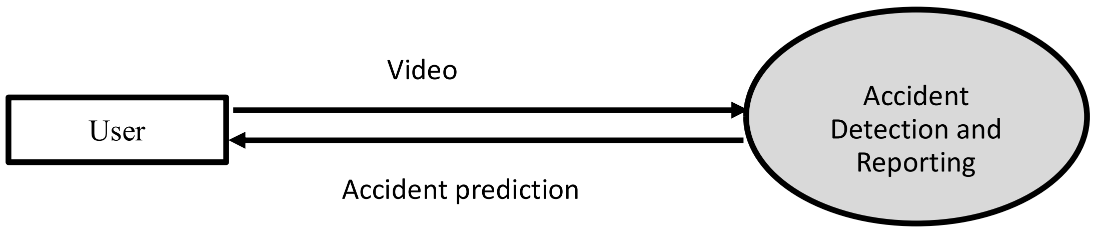


Figure 5.2: Context Diagram

The Context Diagram, also known as a Level 0 Data Flow Diagram (DFD), illustrates the interactions between the main components of the User and Accident Detection and Reporting System. In this diagram, the user uploads video footage to the Accident Detection and Reporting System. The model predicts whether the accident has taken place or not and gives the result back to the user.

Chapter 6

DETAILED DESIGN

6.1 PROCESS FLOW

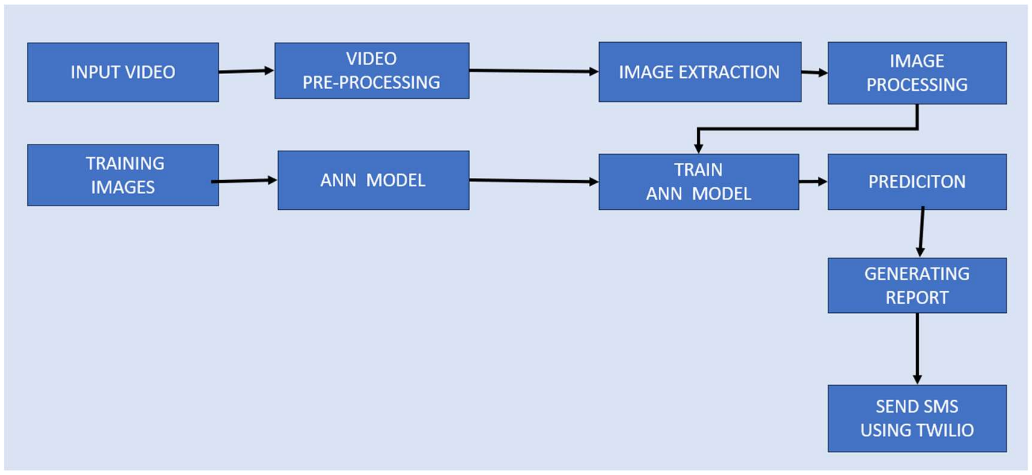


Figure 6.1: Process flow Diagram

The above Accident Detection and Reporting system model process flow explains the collection of the training images and then processing them for training the CNN model. Video given as the input to model is processed and image is extracted which is given to CNN Model for prediction of accident.

6.2 USE CASE DIAGRAM

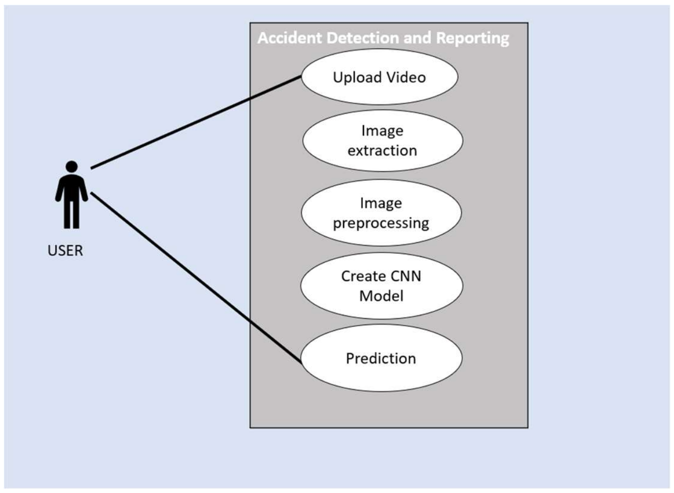


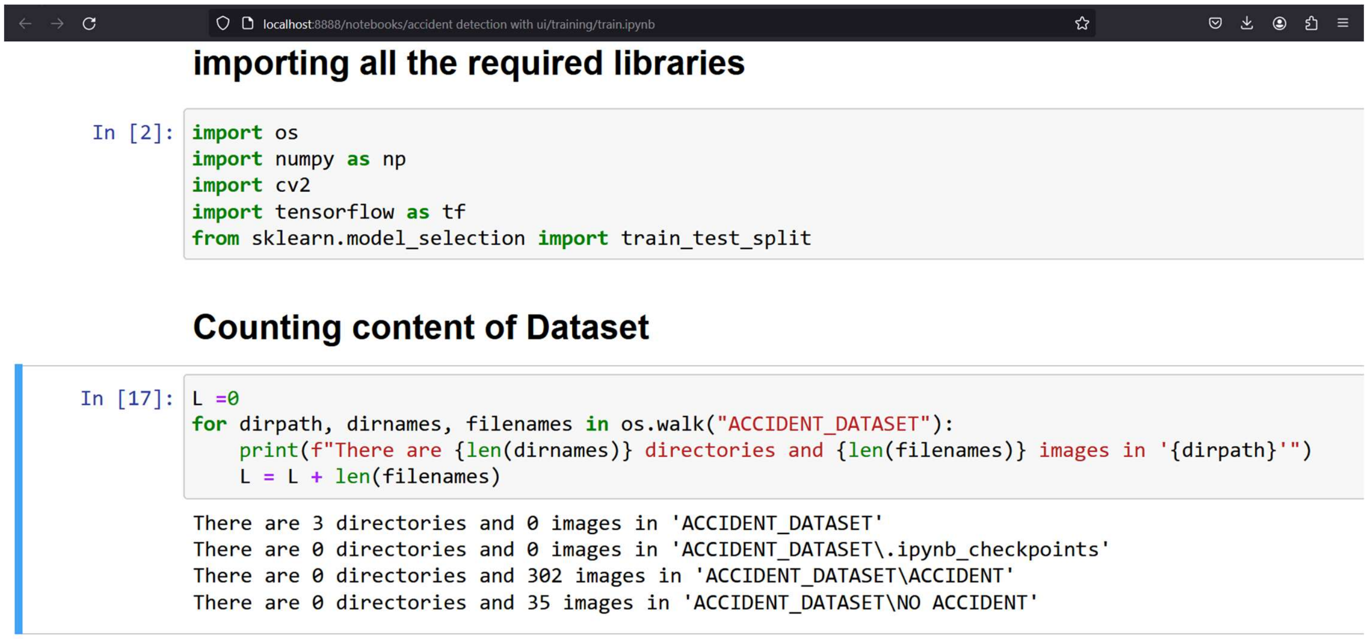
Figure 6.2: Use case Diagram

The use case diagram provides a visual representation of the interactions between the different actors, in this case, there is only one user and the services provided by the Accident detection and reporting model. It helps to predict whether the accident has taken place in the input video.

Chapter 7

Implementation

7.1 Screenshots



localhost:8888/notebooks/accident detection with ui/training/train.ipynb

importing all the required libraries

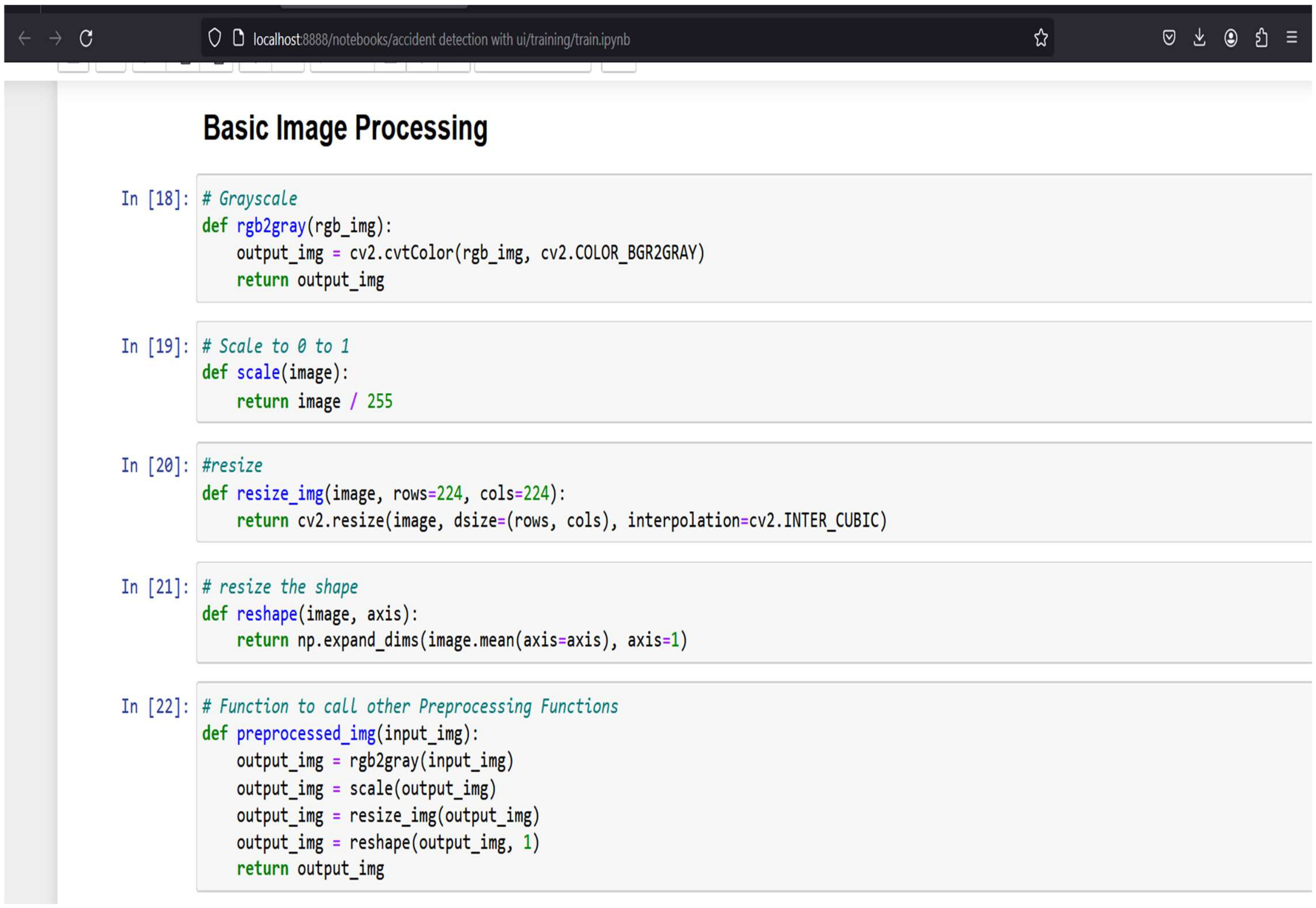
In [2]:
```
import os
import numpy as np
import cv2
import tensorflow as tf
from sklearn.model_selection import train_test_split
```

Counting content of Dataset

In [17]:
```
L =0
for dirpath, dirnames, filenames in os.walk("ACCIDENT_DATASET"):
    print(f"There are {len(dirnames)} directories and {len(filenames)} images in '{dirpath}'")
    L = L + len(filenames)
```

```
There are 3 directories and 0 images in 'ACCIDENT_DATASET'
There are 0 directories and 0 images in 'ACCIDENT_DATASET\.ipynb_checkpoints'
There are 0 directories and 302 images in 'ACCIDENT_DATASET\ACCIDENT'
There are 0 directories and 35 images in 'ACCIDENT_DATASET\NO ACCIDENT'
```

- Os : provides a way to interact with the file system
- Numpy: provides support for large multi-dimesional arrays and matrices
- Cv2 : libraray used for computer vision tasks such as image and video processing .provides features such as image recognition, feature detection
- Matplotlib.pyplot: data visualization in form of graph
- Tenserflow: used for machine learning and deeplearing tasks, provides ways to create a neural network
- Train -test-split : used to split a database into training and testing sets
- Os.walk ()function to traverse the directory given

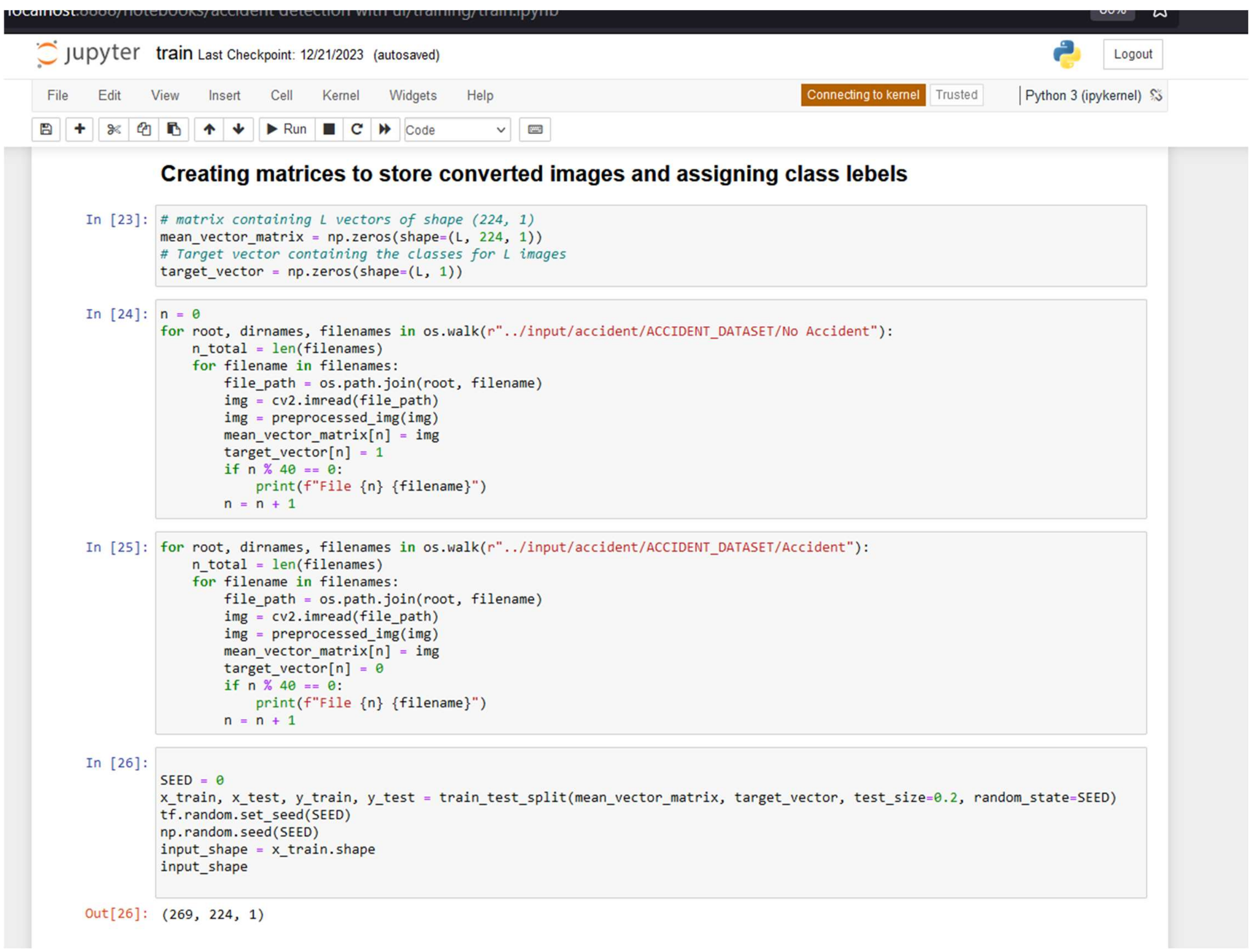


localhost:8888/notebooks/accident detection with ui/training/train.ipynb

Basic Image Processing

```
In [18]: # Grayscale
         def rgb2gray(rgb_img):
             output_img = cv2.cvtColor(rgb_img, cv2.COLOR_BGR2GRAY)
             return output_img

In [19]: # Scale to 0 to 1
         def scale(image):
             return image / 255

In [20]: #resize
         def resize_img(image, rows=224, cols=224):
             return cv2.resize(image, dsize=(rows, cols), interpolation=cv2.INTER_CUBIC)

In [21]: # resize the shape
         def reshape(image, axis):
             return np.expand_dims(image.mean(axis=axis), axis=1)

In [22]: # Function to call other Preprocessing Functions
         def preprocessed_img(input_img):
             output_img = rgb2gray(input_img)
             output_img = scale(output_img)
             output_img = resize_img(output_img)
             output_img = reshape(output_img, 1)
             return output_img
```

- RGB To Grayscale: converting RGB image to grayscale image.
- scaling images: Scaling the pixel values between 0 to 1.
- resizing the image: Resizing the image to 224 X 224 format
- resizing the shape: Resizing the Boundary of the images


jupyter train Last Checkpoint: 12/21/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Connecting to kernel
Trusted
Python 3 (ipykernel)
Run
Code

Creating matrices to store converted images and assigning class lebels

```
In [23]: # matrix containing L vectors of shape (224, 1)
         mean_vector_matrix = np.zeros(shape=(L, 224, 1))
         # Target vector containing the classes for L images
         target_vector = np.zeros(shape=(L, 1))
```

```
In [24]: n = 0
         for root, dirnames, filenames in os.walk(r"../input/accident/ACCIDENT_DATASET/No Accident"):
             n_total = len(filenames)
             for filename in filenames:
                 file_path = os.path.join(root, filename)
                 img = cv2.imread(file_path)
                 img = preprocessed_img(img)
                 mean_vector_matrix[n] = img
                 target_vector[n] = 1
                 if n % 40 == 0:
                     print(f"File {n} {filename}")
                 n = n + 1
```

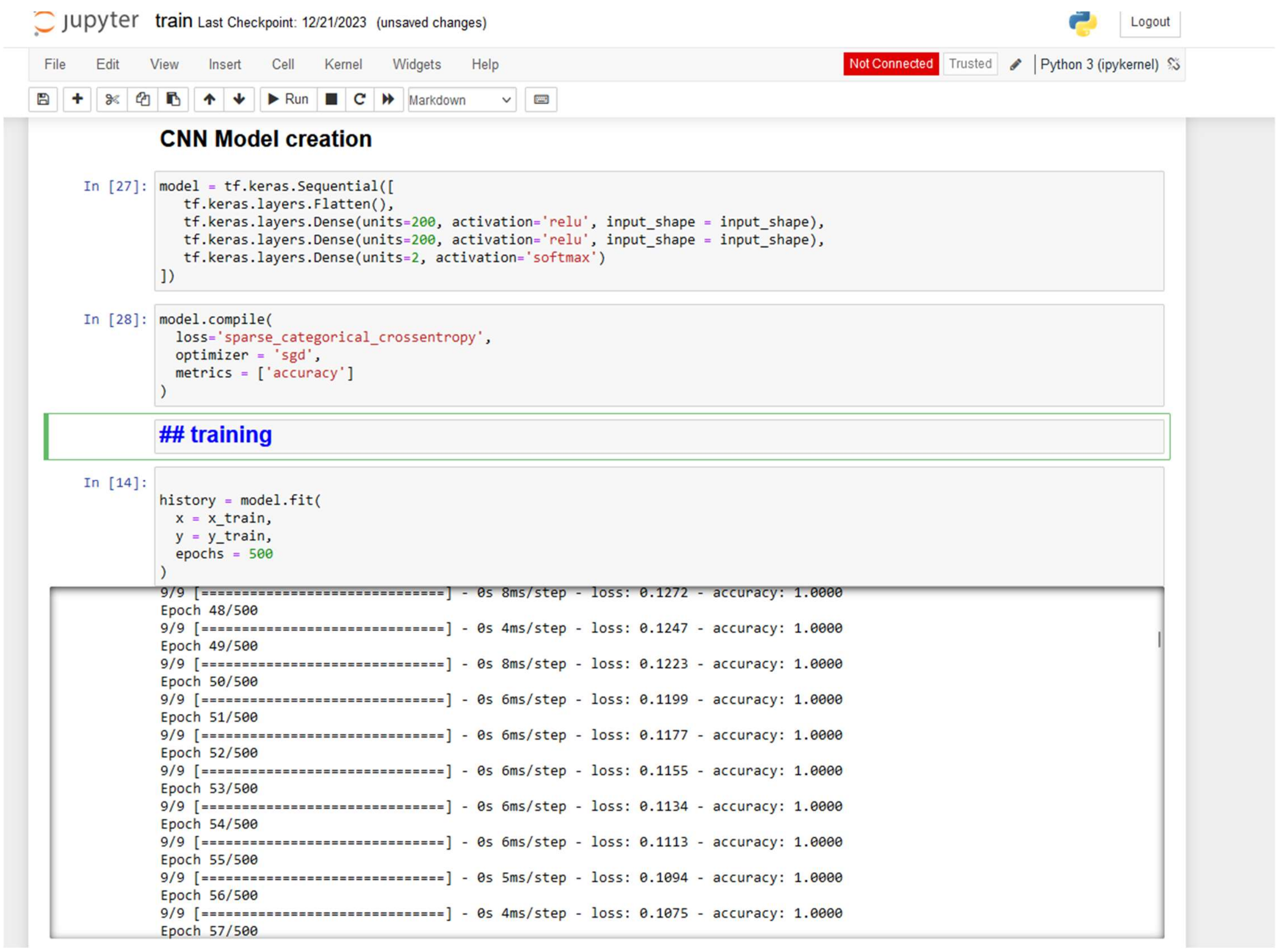
```
In [25]: for root, dirnames, filenames in os.walk(r"../input/accident/ACCIDENT_DATASET/Accident"):
             n_total = len(filenames)
             for filename in filenames:
                 file_path = os.path.join(root, filename)
                 img = cv2.imread(file_path)
                 img = preprocessed_img(img)
                 mean_vector_matrix[n] = img
                 target_vector[n] = 0
                 if n % 40 == 0:
                     print(f"File {n} {filename}")
                 n = n + 1
```

```
In [26]:
         SEED = 0
         x_train, x_test, y_train, y_test = train_test_split(mean_vector_matrix, target_vector, test_size=0.2, random_state=SEED)
         tf.random.set_seed(SEED)
         np.random.seed(SEED)
         input_shape = x_train.shape
         input_shape

Out[26]: (269, 224, 1)
```

creating matrices to store processed image and assign labels

Above code stores the preprocessed image in mean_vector_matrix array and stores class labels for image in target_vector array.

Train_test_split function used to split Mean_vector_matrix and target _vector arrays into training and testing sets.Stores the training set in input_shape varaiable.



jupyter train Last Checkpoint: 12/21/2023 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected Trusted Python 3 (ipykernel)
Run Markdown

CNN Model creation

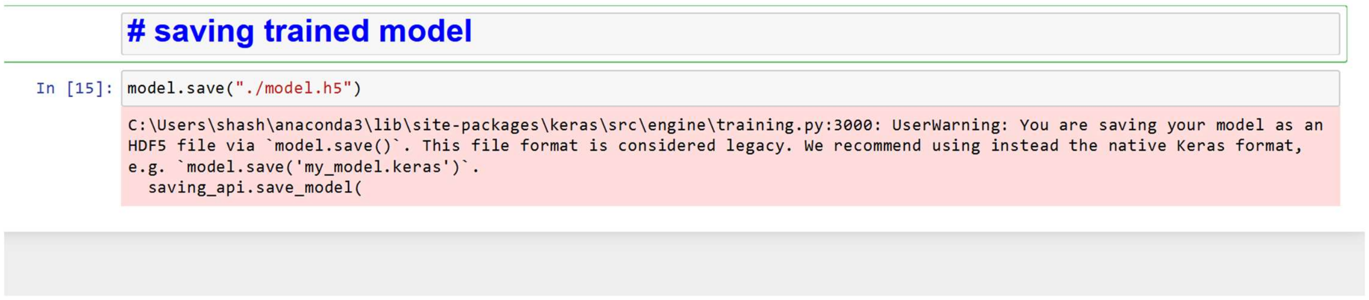
```
In [27]: model = tf.keras.Sequential([
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(units=200, activation='relu', input_shape = input_shape),
            tf.keras.layers.Dense(units=200, activation='relu', input_shape = input_shape),
            tf.keras.layers.Dense(units=2, activation='softmax')
         ])
```

```
In [28]: model.compile(
           loss='sparse_categorical_crossentropy',
           optimizer = 'sgd',
           metrics = ['accuracy']
         )
```

training

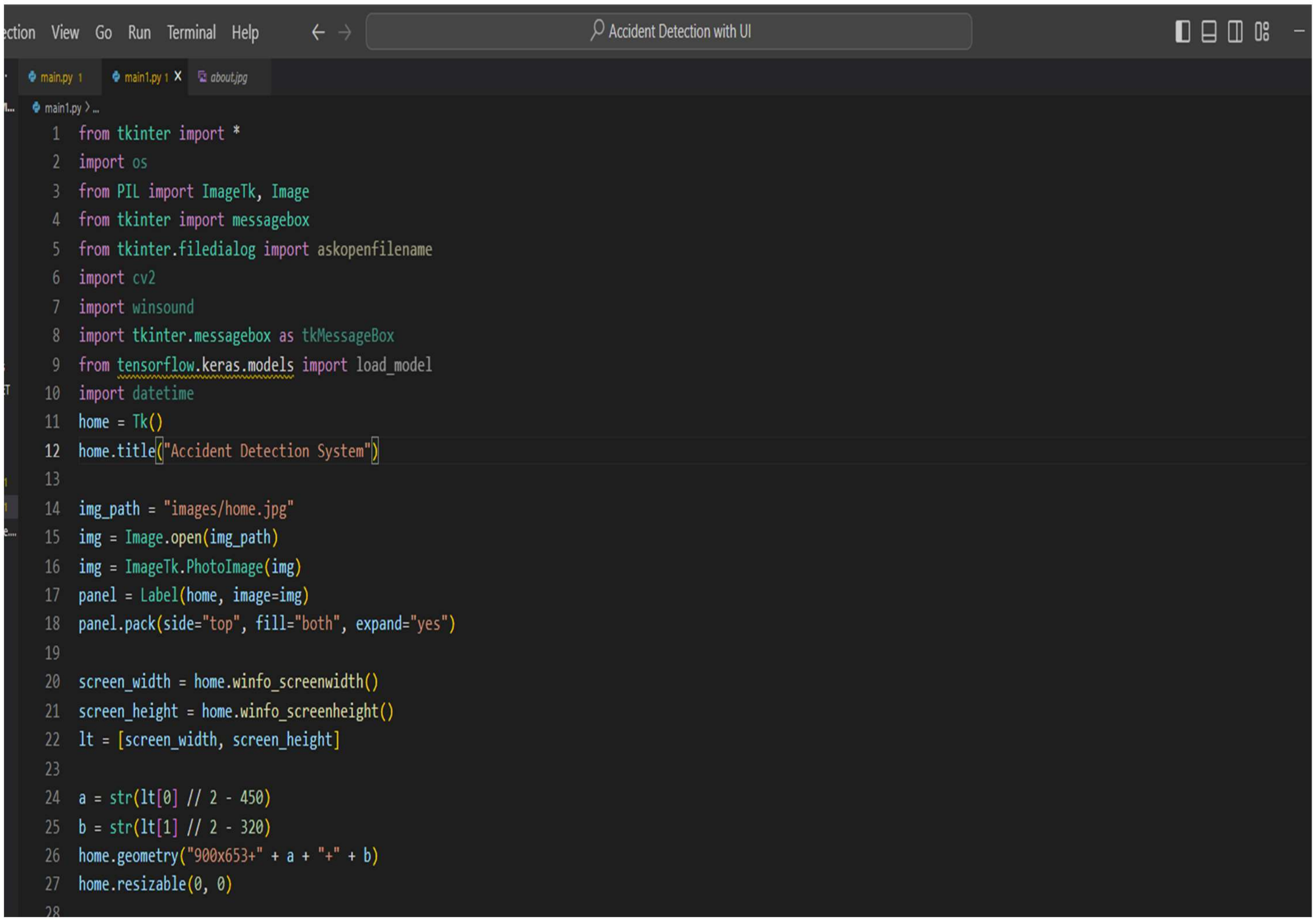
```
In [14]:
         history = model.fit(
           x = x_train,
           y = y_train,
           epochs = 500
         )
```

```
9/9 [==============================] - 0s 8ms/step - loss: 0.1272 - accuracy: 1.0000
Epoch 48/500
9/9 [==============================] - 0s 4ms/step - loss: 0.1247 - accuracy: 1.0000
Epoch 49/500
9/9 [==============================] - 0s 8ms/step - loss: 0.1223 - accuracy: 1.0000
Epoch 50/500
9/9 [==============================] - 0s 6ms/step - loss: 0.1199 - accuracy: 1.0000
Epoch 51/500
9/9 [==============================] - 0s 6ms/step - loss: 0.1177 - accuracy: 1.0000
Epoch 52/500
9/9 [==============================] - 0s 6ms/step - loss: 0.1155 - accuracy: 1.0000
Epoch 53/500
9/9 [==============================] - 0s 6ms/step - loss: 0.1134 - accuracy: 1.0000
Epoch 54/500
9/9 [==============================] - 0s 6ms/step - loss: 0.1113 - accuracy: 1.0000
Epoch 55/500
9/9 [==============================] - 0s 5ms/step - loss: 0.1094 - accuracy: 1.0000
Epoch 56/500
9/9 [==============================] - 0s 4ms/step - loss: 0.1075 - accuracy: 1.0000
Epoch 57/500
```

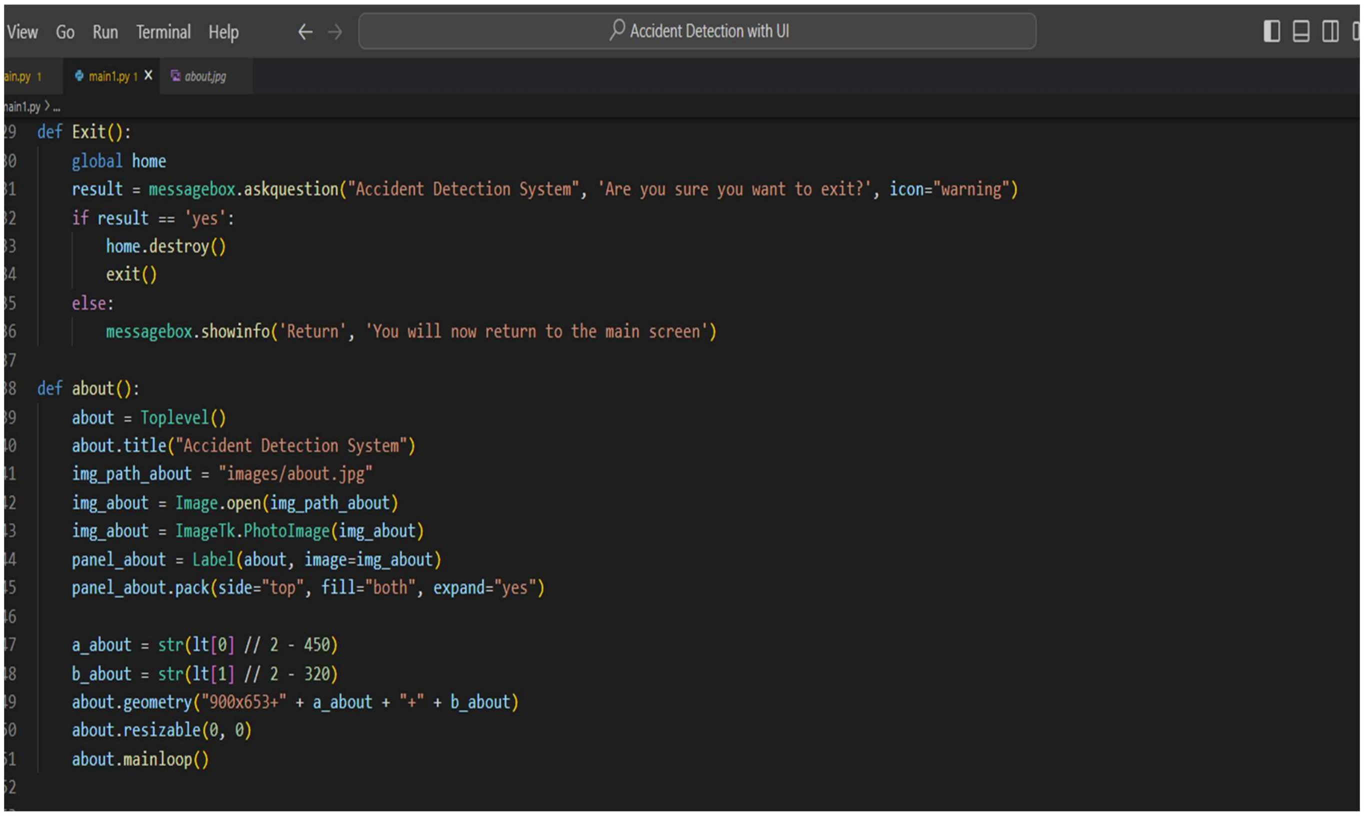


saving trained model

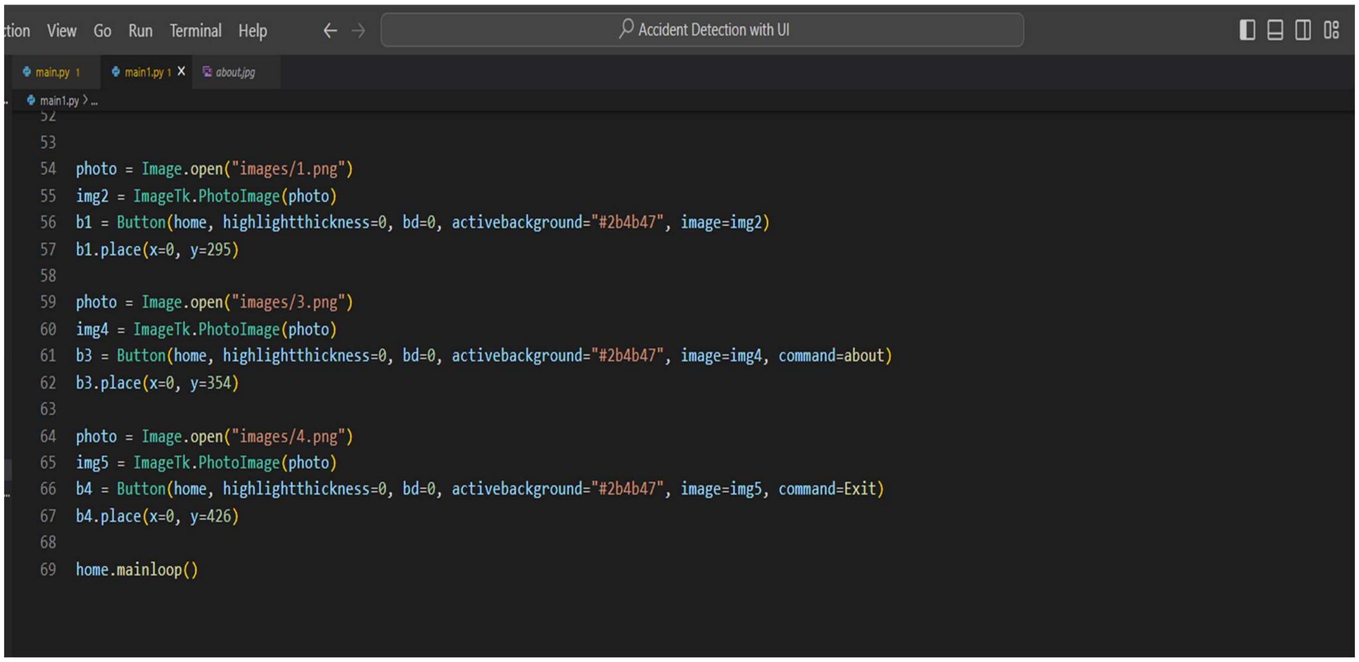
```
In [15]: model.save("./model.h5")

C:\Users\shash\anaconda3\lib\site-packages\keras\src\engine\training.py:3000: UserWarning: You are saving your model as an
HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format,
e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```



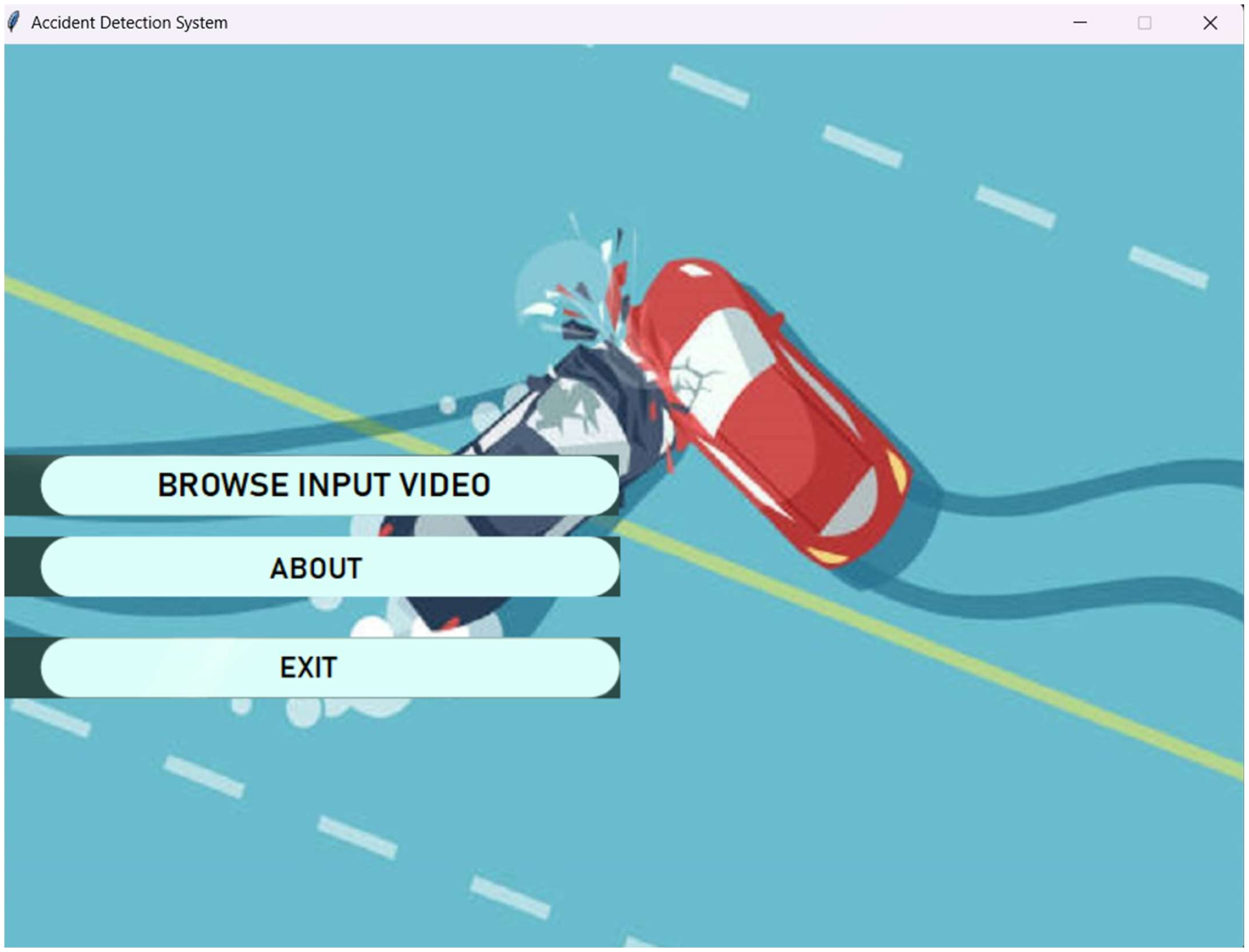
```
from tkinter import *
import os
from PIL import ImageTk, Image
from tkinter import messagebox
from tkinter.filedialog import askopenfilename
import cv2
import winsound
import tkinter.messagebox as tkMessageBox
from tensorflow.keras.models import load_model
import datetime
home = Tk()
home.title("Accident Detection System")

img_path = "images/home.jpg"
img = Image.open(img_path)
img = ImageTk.PhotoImage(img)
panel = Label(home, image=img)
panel.pack(side="top", fill="both", expand="yes")

screen_width = home.winfo_screenwidth()
screen_height = home.winfo_screenheight()
lt = [screen_width, screen_height]

a = str(lt[0] // 2 - 450)
b = str(lt[1] // 2 - 320)
home.geometry("900x653+" + a + "+" + b)
home.resizable(0, 0)
```



```python
def Exit():
    global home
    result = messagebox.askquestion("Accident Detection System", 'Are you sure you want to exit?', icon="warning")
    if result == 'yes':
        home.destroy()
        exit()
    else:
        messagebox.showinfo('Return', 'You will now return to the main screen')

def about():
    about = Toplevel()
    about.title("Accident Detection System")
    img_path_about = "images/about.jpg"
    img_about = Image.open(img_path_about)
    img_about = ImageTk.PhotoImage(img_about)
    panel_about = Label(about, image=img_about)
    panel_about.pack(side="top", fill="both", expand="yes")

    a_about = str(lt[0] // 2 - 450)
    b_about = str(lt[1] // 2 - 320)
    about.geometry("900x653+" + a_about + "+" + b_about)
    about.resizable(0, 0)
    about.mainloop()
```



```python

photo = Image.open("images/1.png")
img2 = ImageTk.PhotoImage(photo)
b1 = Button(home, highlightthickness=0, bd=0, activebackground="#2b4b47", image=img2)
b1.place(x=0, y=295)

photo = Image.open("images/3.png")
img4 = ImageTk.PhotoImage(photo)
b3 = Button(home, highlightthickness=0, bd=0, activebackground="#2b4b47", image=img4, command=about)
b3.place(x=0, y=354)

photo = Image.open("images/4.png")
img5 = ImageTk.PhotoImage(photo)
b4 = Button(home, highlightthickness=0, bd=0, activebackground="#2b4b47", image=img5, command=Exit)
b4.place(x=0, y=426)

home.mainloop()
```


Accident Detection System
BROWSE INPUT VIDEO
ABOUT
EXIT

APPENDIX A
BIBLIOGRAPHY

- Mahendra Vucha, et.al, [1] have proposed "Accident Detection and Alert System" and was published in the International Journal of Innovative Technology and Exploring Engineering (IJITEE) (March2019).
- Koushik S et.al, [2] has proposed "IoT Based Automatic Accident Detection & Rescue Management In Vanet" and was published in the SSRG International Journal of Computer Science and Engineering (SSRG - IJCSE) - - Special Issue ICFTESH (Feb2019).
- Madhu mitha et.al, [3] has proposed a "Vehicle Accident Detection System by using GSM AND GPS" and was published in the International Research Journal of Engineering and Technology (IRJET) (Jan2020).
- Hemangi S.Ahire et.al, [4] have proposed a "Vehicle Accident Detection and Alerting System" and was published in the International Journal for Research in Applied Science & Engineering Technology(Jan2018).
- Automatic Accident Detection and Alerting System Based on IoT was published in the International Journal of Innovative - Research in Computer (May2020).
- www.youtube.com
- www.github.com
- https://www.geeksforgeeks.org/python-gui-tkinter