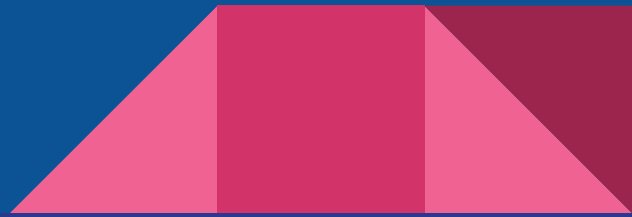
Driver Drowsiness Detection System

Martin Berberi DSIR-222

May, 13 2022

Agenda:

- Introduction
- Objectives
- Working Procedure
- ML Models Performance
- Real World Application
- Conclusion
- Live Demonstration



Introduction

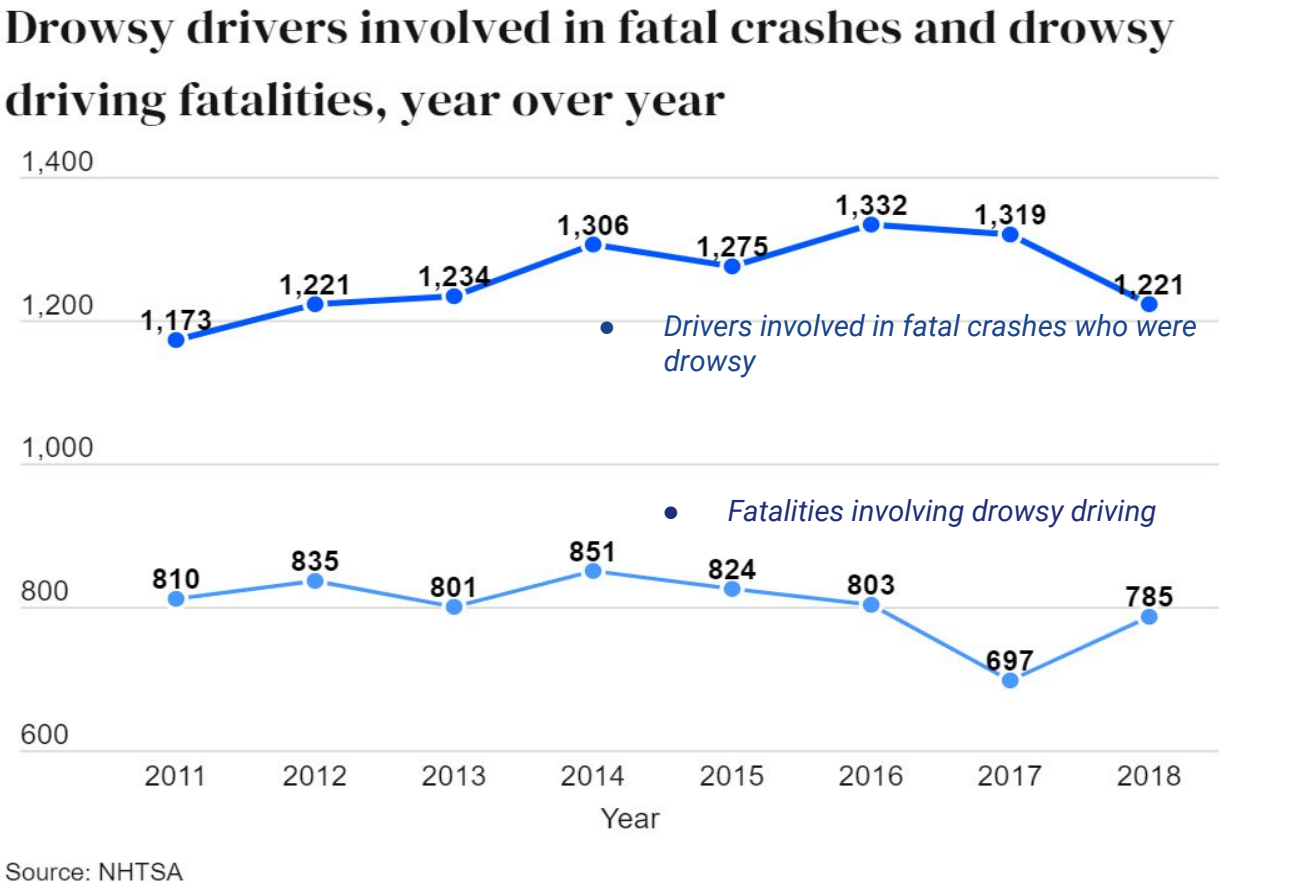
Problem Statement:

As a data scientist hired from a limo company, my job is to build a Drowsiness Detection System to avoid accidents that may happen from their drowsy drivers during long time of driving.

"Drowsiness is one of the biggest reasons for road accidents and Data Science is the best remedy for it"

- *Drowsiness System is a safety technology that can prevent accidents caused by drivers who fell asleep while driving.*

- *According to CDC's study, from 150,000 adults who were surveyed 1 of 25 drivers admitted have fallen asleep.*



Objectives:

- The main objective of this project is to prepare a prototype of a driver drowsiness detection system that trigger an alarm sound when the driver fell asleep.

- This is Achieved by using a camera which send multiple frames in second to my machine learning model, which get the eyes status and generate predictions based on the state of the driver’s eyes.

Working Procedure:

Step 1: Take image as input from a Camera

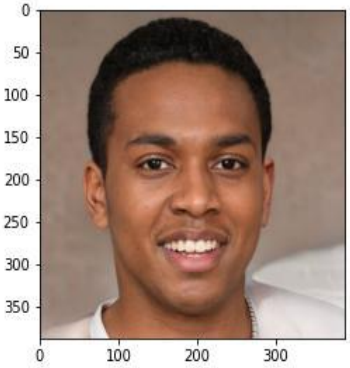
- *From OpenCV library used cv2.VideoCapture(0) to access the camera and set the capture object (video).*
- *video.read() will read each frame and stored the frame in a image variable*
- *This image will be used as input for my model*

Working Procedure:

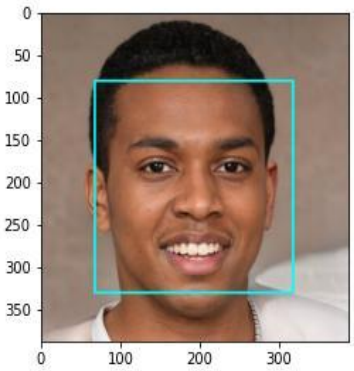
Step 2: Detect Face & Create The New Image (crop just the face)

- First convert the image to grayscale because of the OpenCV detect object algorithm requirements
- Used the cascade classifier to detect faces

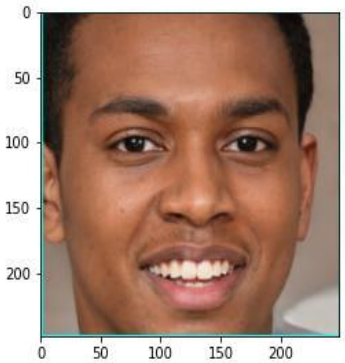
Original Image



Detect Face



Crop Face

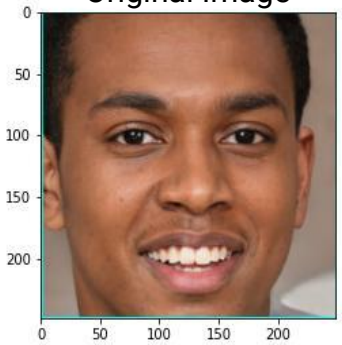


Working Procedure:

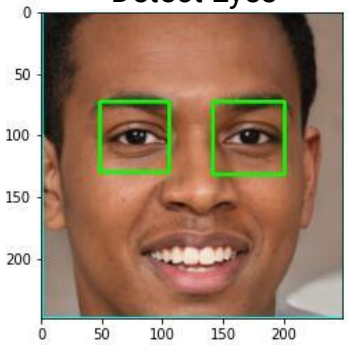
Step 3: Detect the Eyes from Cropped Image

- *The same procedure as detecting faces was used for eyes now.*
- *Used the cascade classifier to detect eyes*
- *Cropping those eyes as final image*

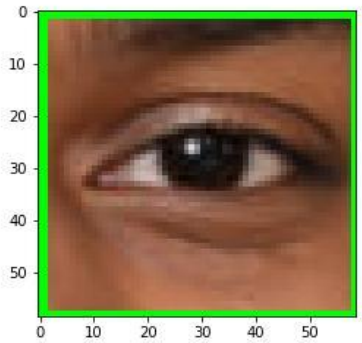
Original Image



Detect Eyes



Crop Eye



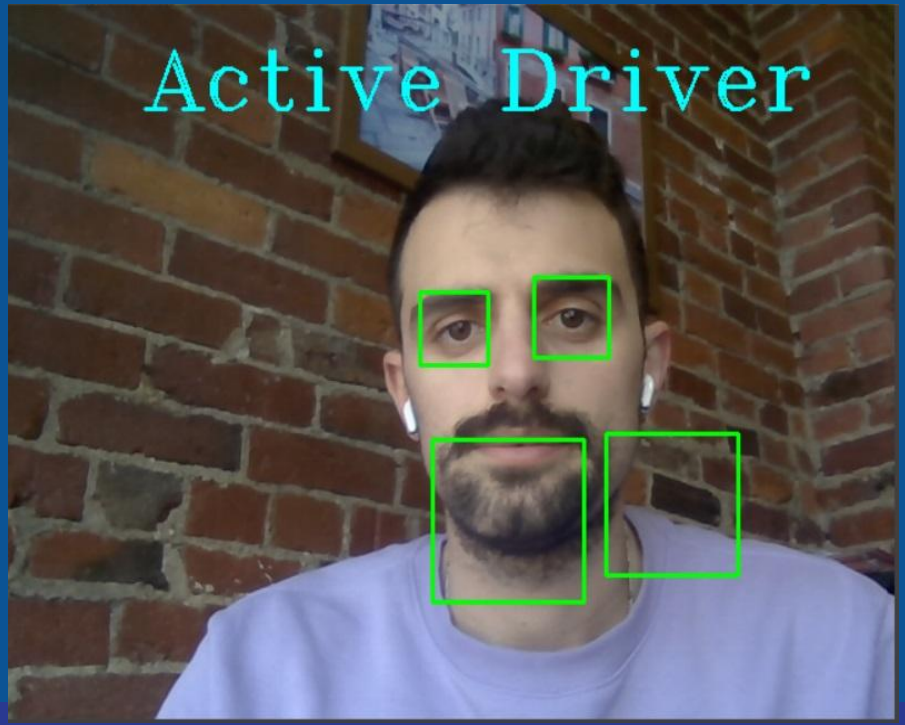
Working Procedure:

Step 4: Model will classify either a close eye or open eye

- *Image was converted to grayscale: cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)*
- *Image was resized to size 224*224 pixels & normalized by dividing by 255.0*
- *Then prediction was generated from classifier model*
- *If that prediction is equal or greater than 0.5 assigned to 1 (close eyes) otherwise 0 (open eyes)*

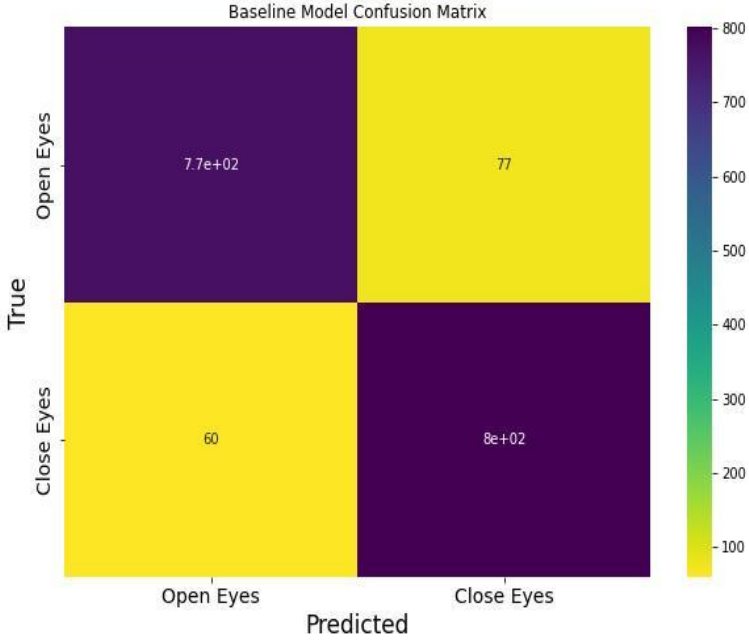
Working Procedure:

Step 5: Result

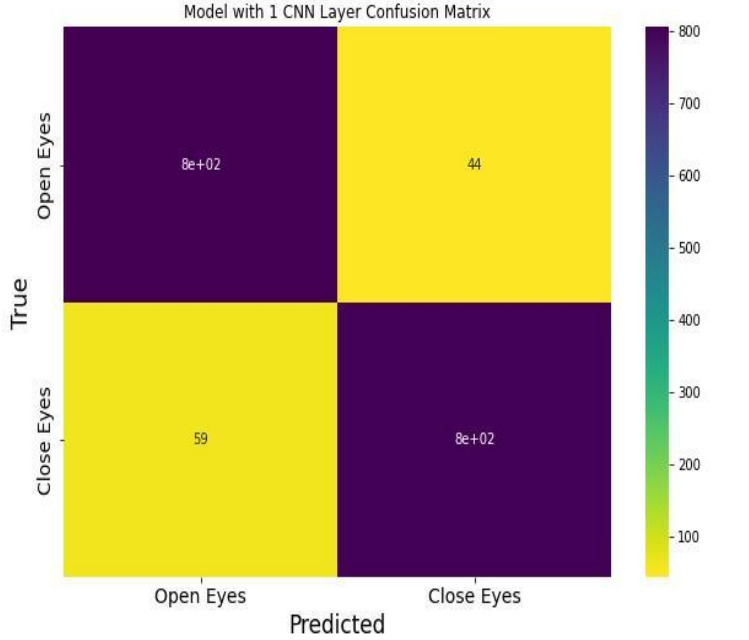


Machine Learning Models Performance

Sensitivity: 93%
F1 Score: 92%

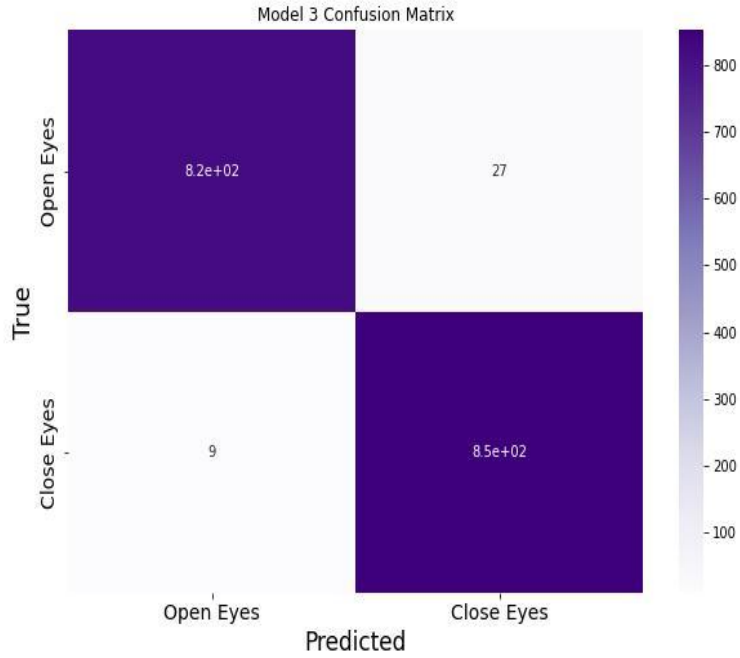


Sensitivity: 93%
F1 Score: 94%

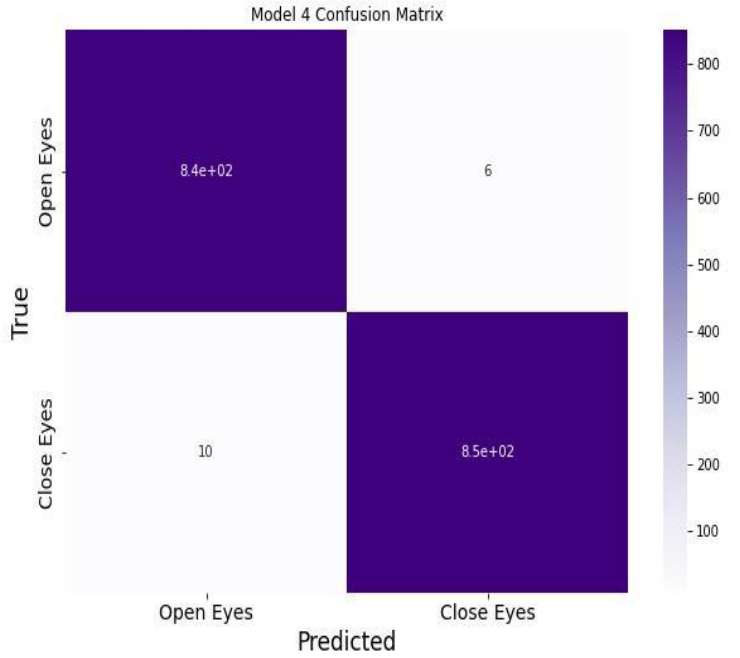


Machine Learning Models Performance

Sensitivity: 98.95%
F1 Score: 97.93%

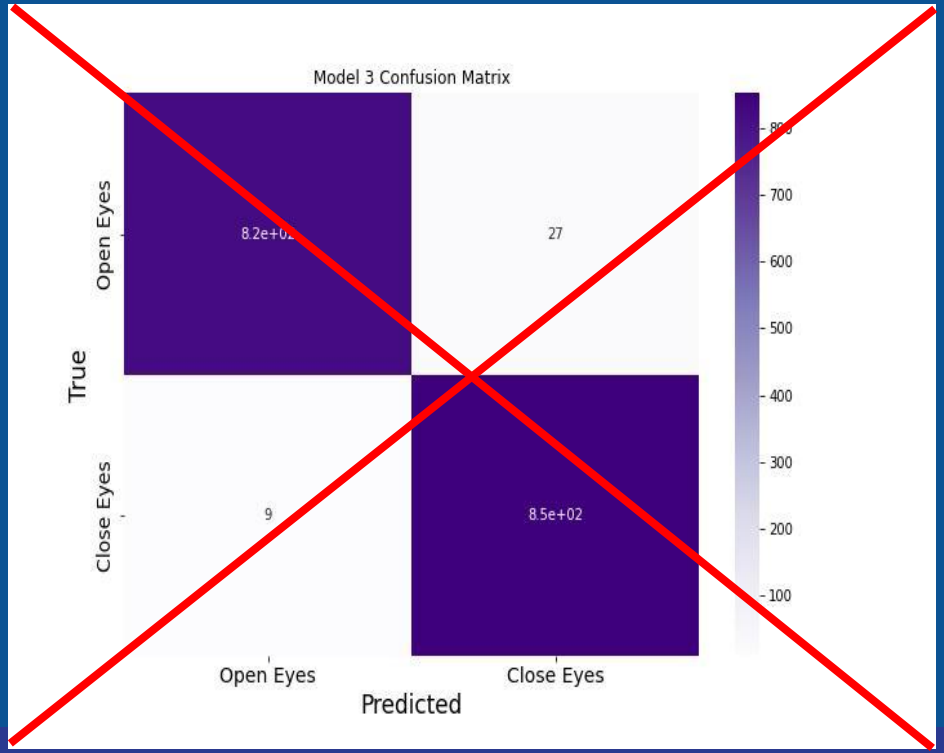


Sensitivity: 98.83%
F1 Score: 99.1%

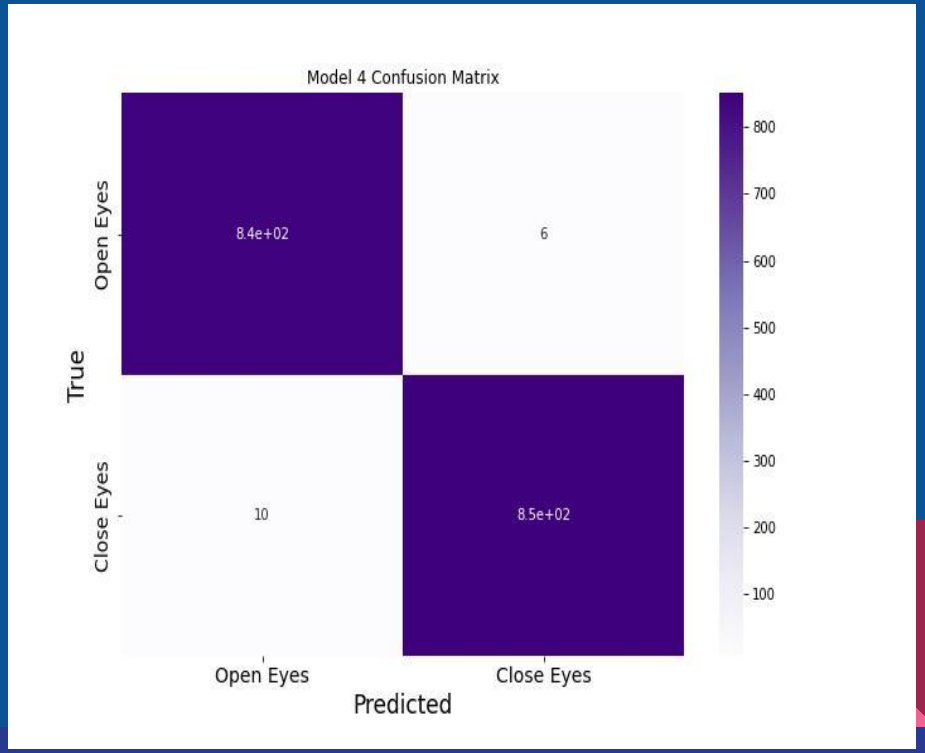


Machine Learning Models Performance

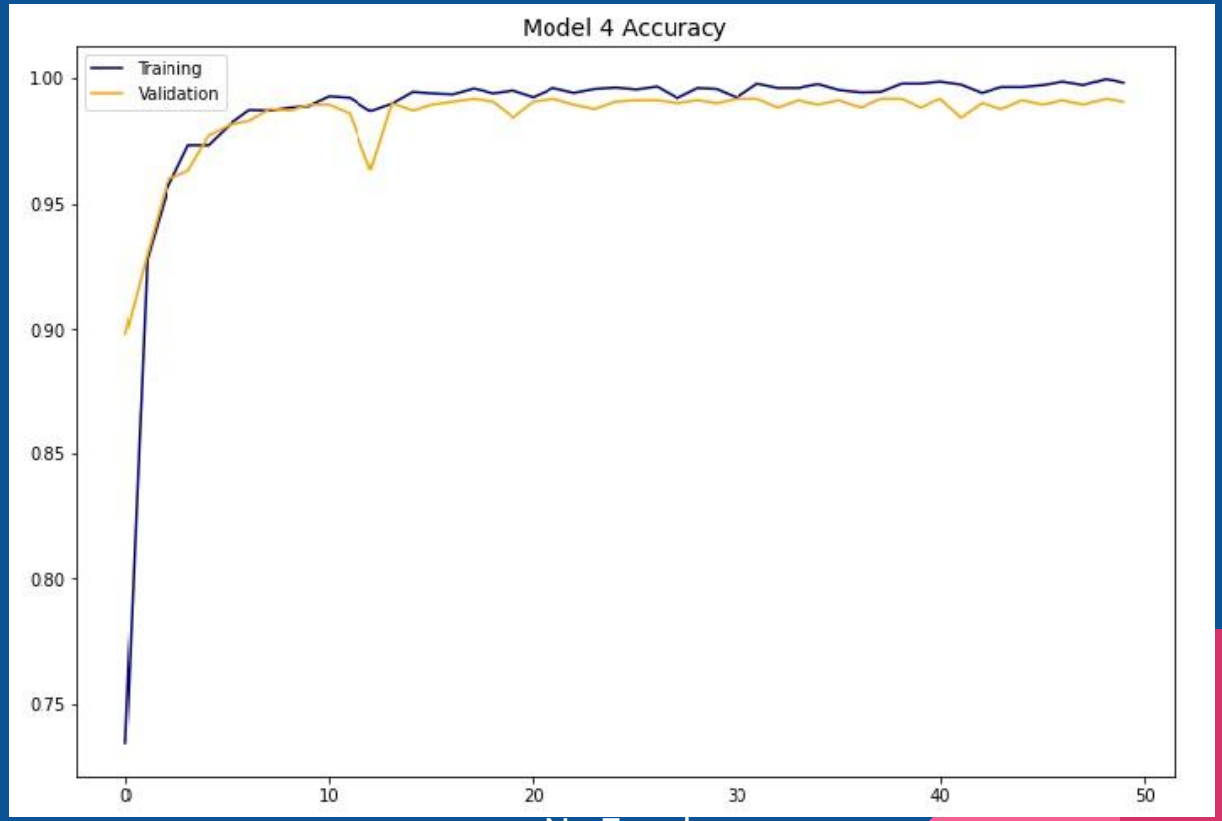
Sensitivity: 98.95%
F1 Score: 97.93%



Sensitivity: 98.83%
F1 Score: 99.1%



Final Model Accuracy Performance



Real World Application:

- ***This system can be used to any automobile***
- ***This can be implemented with a help of a webcam, Raspberry pi & a car speaker for the alarm sound***

Web Camera

Raspberry pi

Car Speaker

Conclusions:

- *This system can reduce the road accidents that happens from drowsy drivers*
- *This system can save lives which is the main motive of this system*
- *This system is not very complex and works effectively*

Thank you!!

Presented by:

Martin Berberi