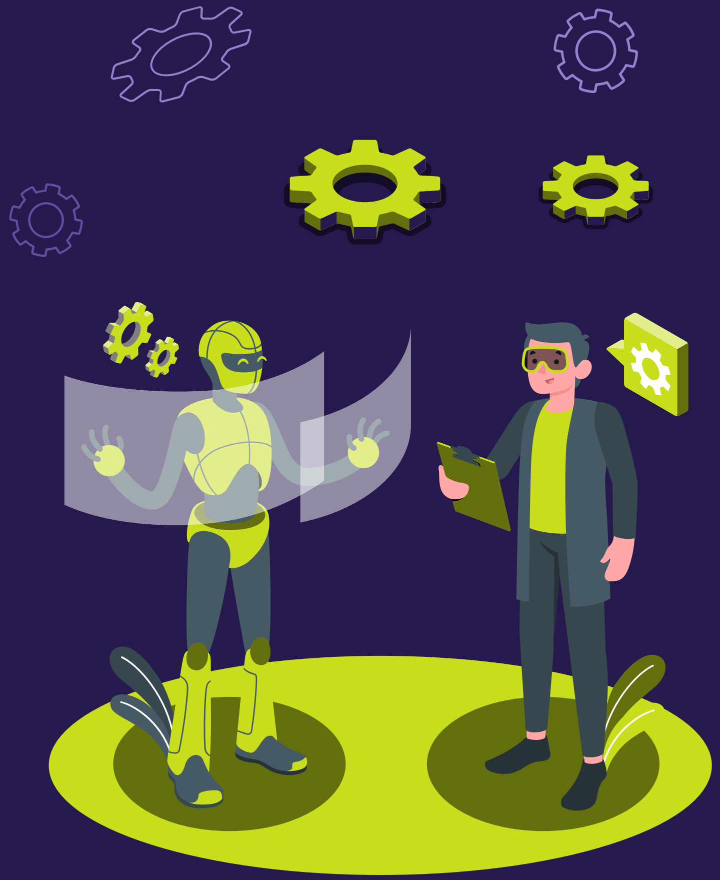
DISTRACTED DRIVER RECOGNITION

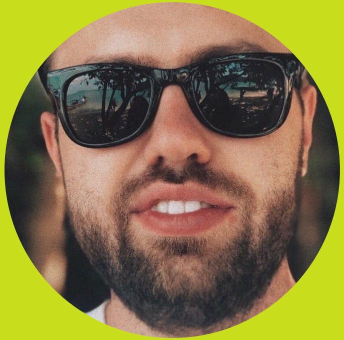
FOR INSURANCE PURPOSES



OUR TEAM

MERT BEKTAS

Project Leader
-
Image Recognition



TARAS IVANTSIV

Team Member
-
Server-Side

ABOUT OUR PROJECT

Our goal is to create a system that detects distracted drivers using Deep Learning, by a phone that is placed in car.

And, storing the information in our server for future needs.

DISTRACTED DRIVING KILLS

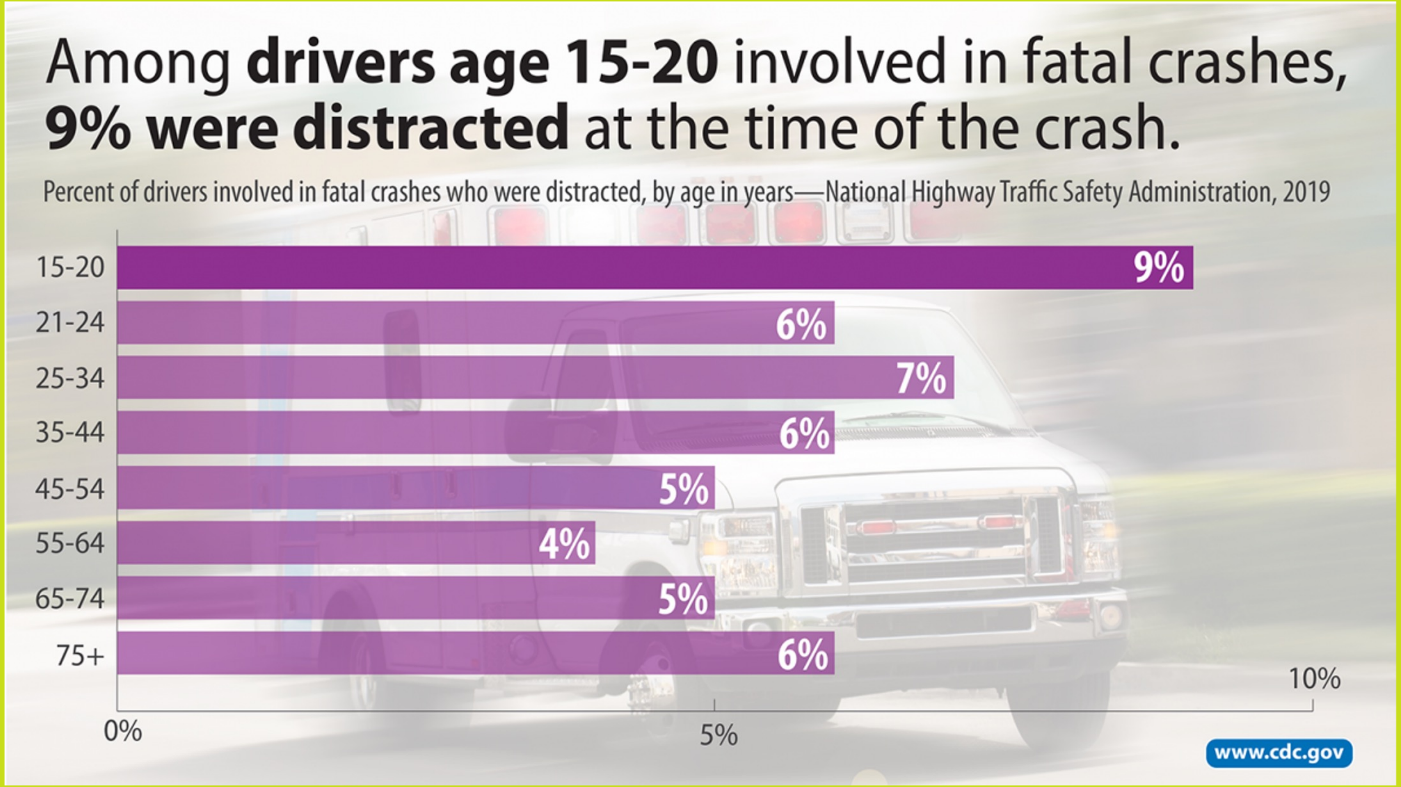
9

people in the United States are

killed every day

in crashes that are reported to involve a **distracted driver**

DISTRACTED DRIVING KILLS



Among **drivers age 15-20** involved in fatal crashes, **9% were distracted** at the time of the crash.

Percent of drivers involved in fatal crashes who were distracted, by age in years—National Highway Traffic Safety Administration, 2019

OBJECTIVES

Goals of this Project is to:

- Make accidents more transparent
- Be sure that driver is focused on the road
- Decrease the numbers of accidents caused by distraction

PROS

- Encourage people to drive more carefully.
- Insurance companies can get more knowledge about who is guilty from the accident.
- Make roads safer.

CONS

- It needs internet connection. And, the connection may be weak.
- Being watched in their car, may make some people uncomfortable or even illegal.

TOOLS

- Phone (Web App) to access the camera and take pictures
- Server (Prediction and Communication Between Car and Insurance Company)

DATASET

- We will use ”**State Farm Distracted Driver Detection**” dataset to train our model.
- This dataset is shared on **Kaggle** by the insurance company ”**State Farm**”.
- It contains over 100.000 images. We used 4806 of them.

HOW IT WORKS

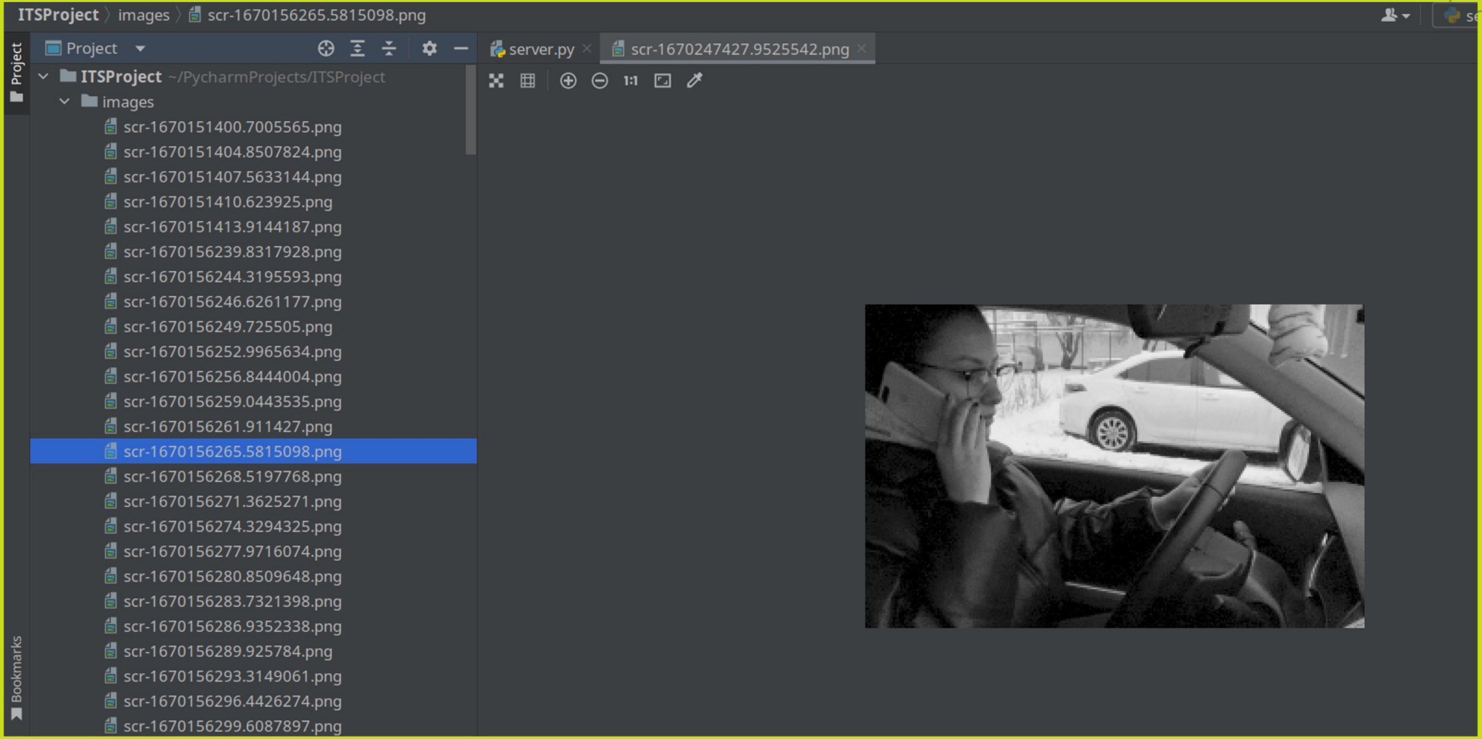
PHONE (WEB APP)

- Captures snapshots from web-camera
- Sends them to Server
- Notifies driver if distracted

SERVER

- Preprocessing snapshots
- Classifies them with CNN
- Stores results
- Sends classification results to Web App

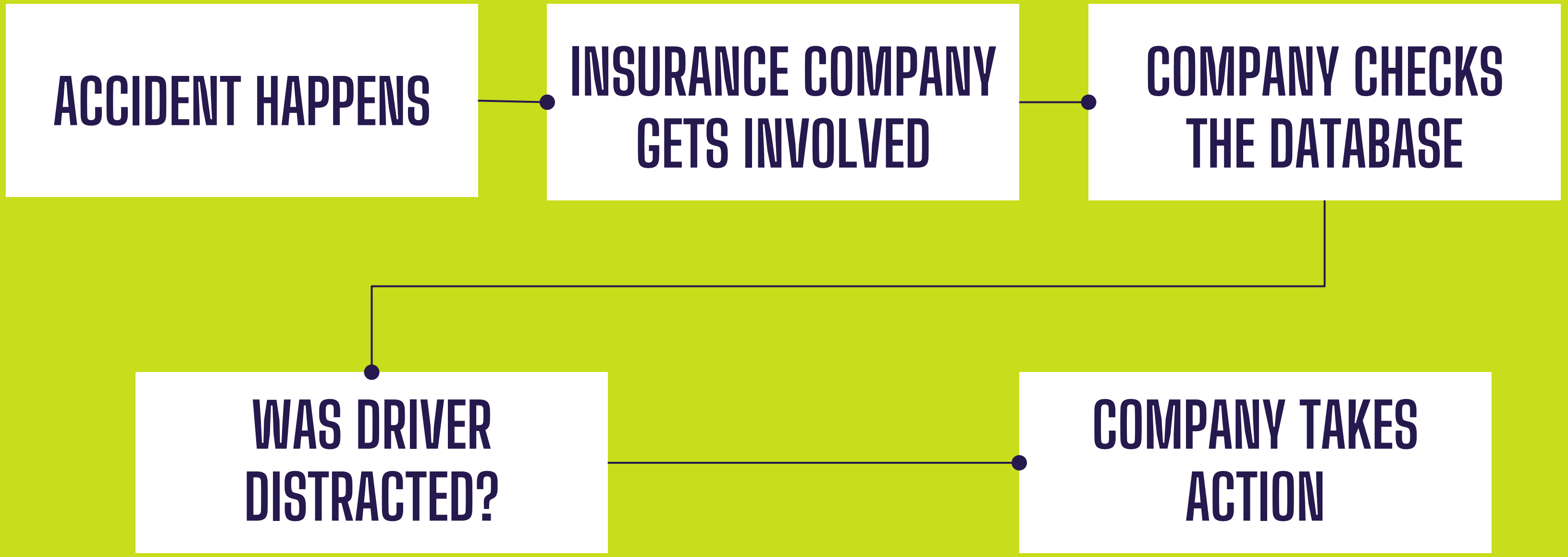
HOW IT WORKS



ITSProject 〉 images 〉 scr-1670156265.5815098.png

ITSProject ~/PycharmProjects/ITSProject
images
scr-1670151400.7005565.png
scr-1670151404.8507824.png
scr-1670151407.5633144.png
scr-1670151410.623925.png
scr-1670151413.9144187.png
scr-1670156239.8317928.png
scr-1670156244.3195593.png
scr-1670156246.6261177.png
scr-1670156249.725505.png
scr-1670156252.9965634.png
scr-1670156256.8444004.png
scr-1670156259.0443535.png
scr-1670156261.911427.png
scr-1670156265.5815098.png
scr-1670156268.5197768.png
scr-1670156271.3625271.png
scr-1670156274.3294325.png
scr-1670156277.9716074.png
scr-1670156280.8509648.png
scr-1670156283.7321398.png
scr-1670156286.9352338.png
scr-1670156289.925784.png
scr-1670156293.3149061.png
scr-1670156296.4426274.png
scr-1670156299.6087897.png

WHAT INSURANCE COMPANY DOES ?



ACCIDENT HAPPENS

INSURANCE COMPANY GETS INVOLVED

COMPANY CHECKS THE DATABASE

WAS DRIVER DISTRACTED?

COMPANY TAKES ACTION

IMAGE RECOGNITION

- Filtering the dataset
- Preprocessing
- Training (Convolutional neural network)
- Evaluation

FILTERING the DATASET

There were 10 labels in the dataset. I used 2 of them to train the model.

These are:

1-) Driving

2-) Talking on the Phone

PREPROCESSING

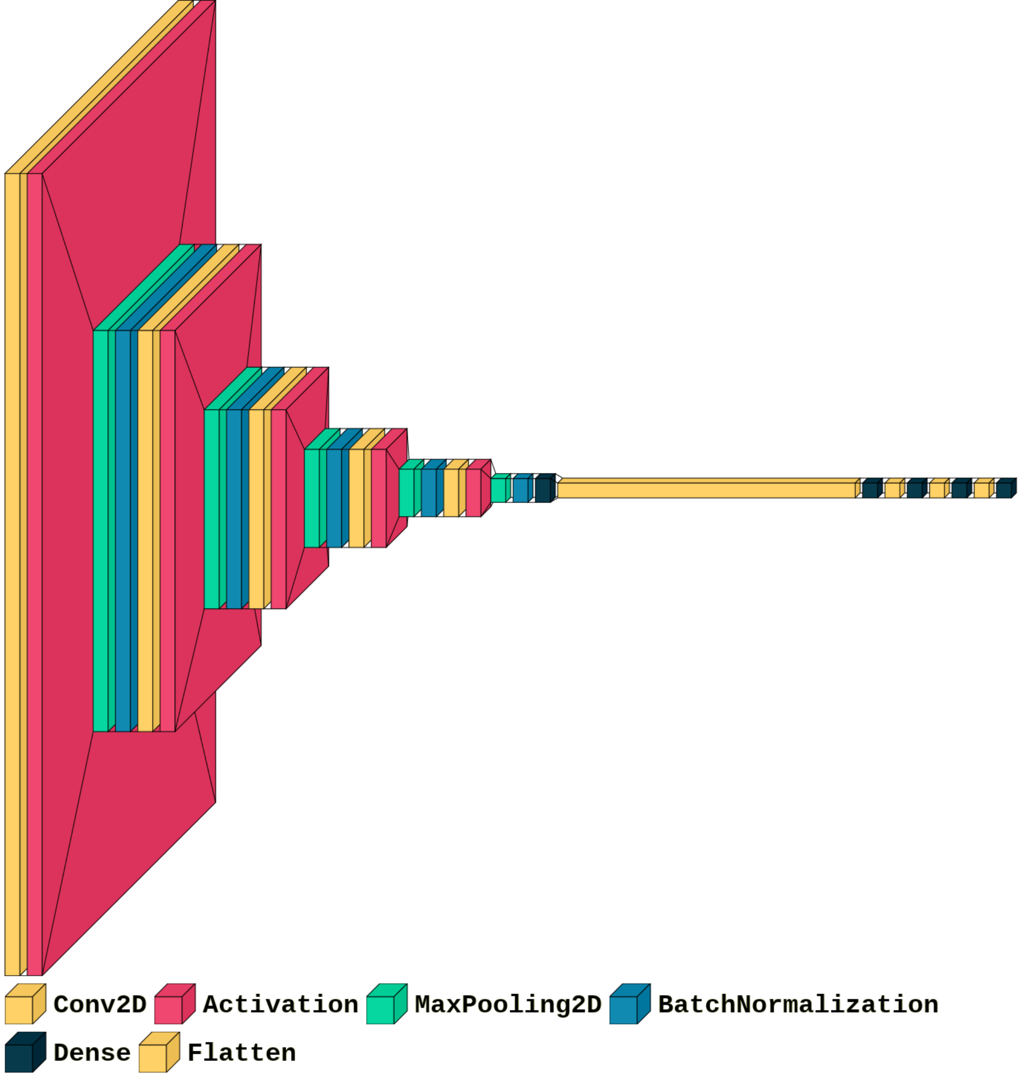
- Cropping to the most important part.
- Resizing the image to (270, 175) from (480, 640).

PREPROCESSING

One hot encoding to labels, using LabelBinarizer from sklearn

” Driving ” → 0

” Talking ” → 1



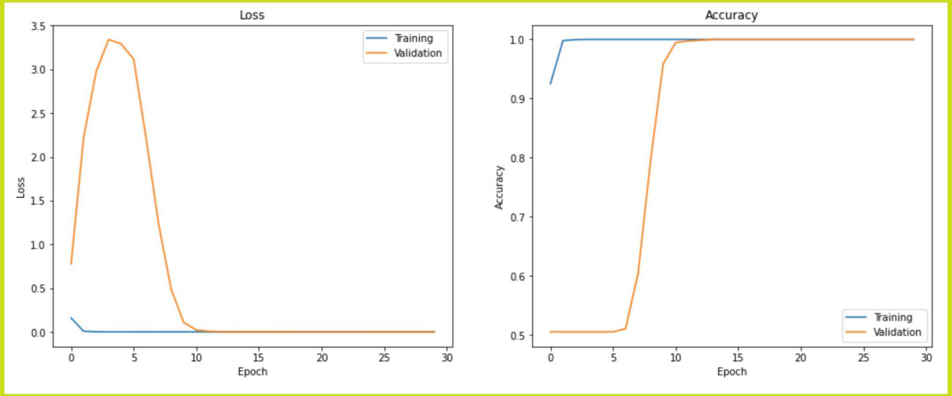
FIRST CNN MODEL

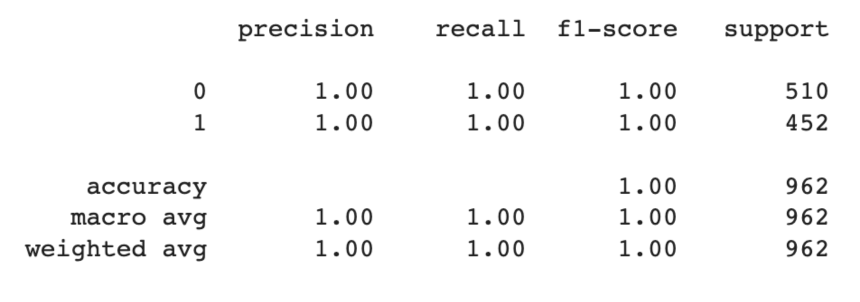
- 5 * (Conv2d layer, ReLU activation layer, MaxPooling 2d layer, Batch Normalization)

- 4 Dense layers with ReLU activation (with parameters of 128, 64, 32, 16)

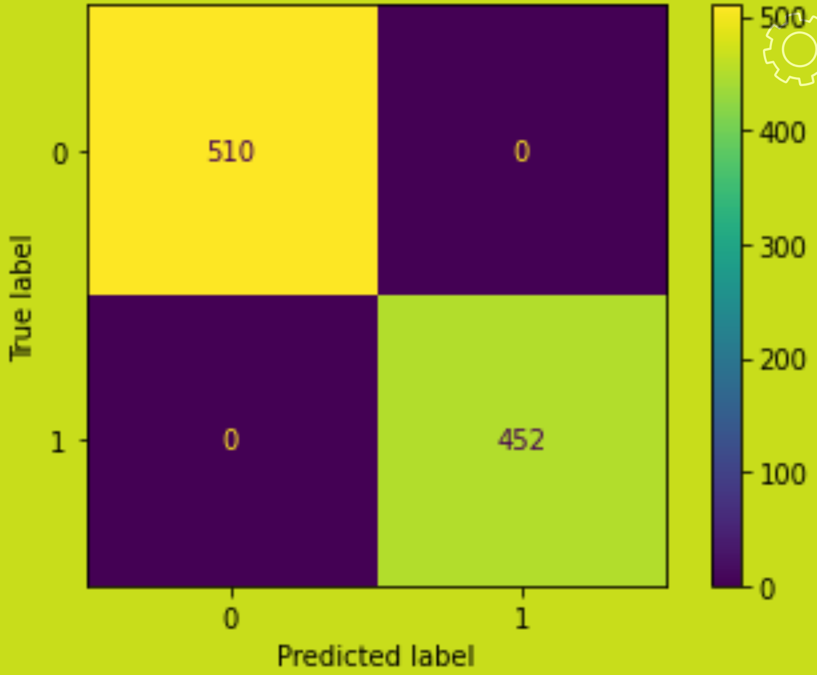
- Fully connected layer with sigmoid activation and 1 output (Binary output)

FIRST MODEL’S PERFORMANCE



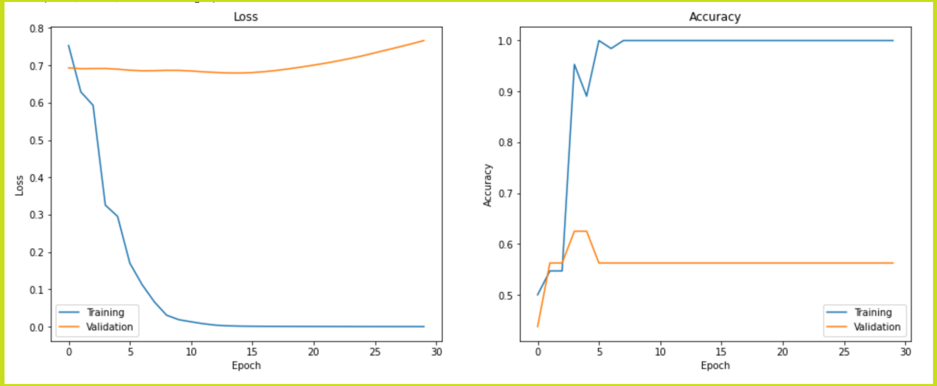


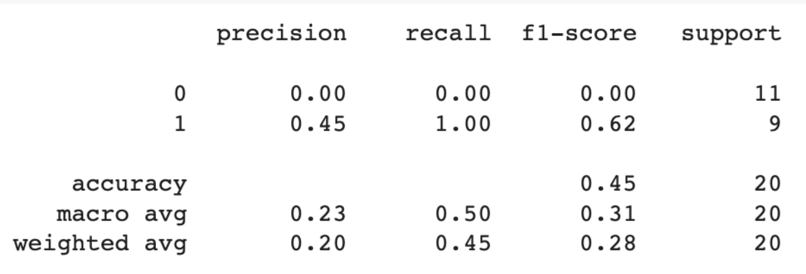
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       510
           1       1.00      1.00      1.00       452

    accuracy                           1.00       962
   macro avg       1.00      1.00      1.00       962
weighted avg       1.00      1.00      1.00       962
```



ONE MODEL to RULE THEM ALL

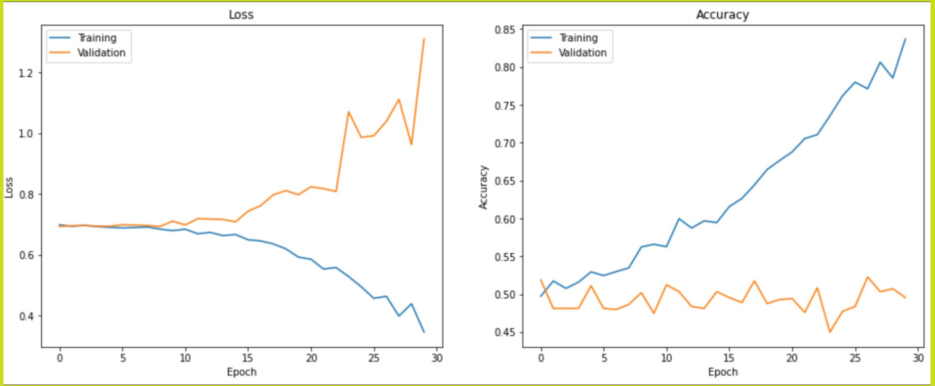
Training with only 50 images

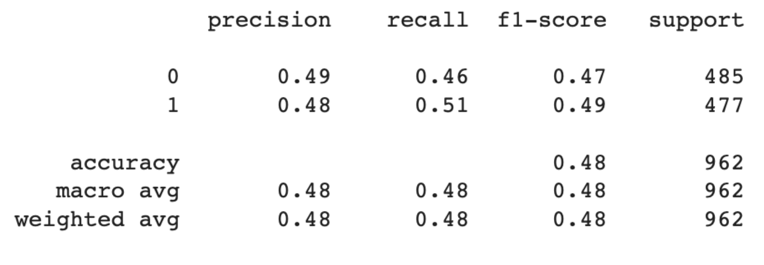




```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        11
           1       0.45      1.00      0.62         9

    accuracy                           0.45        20
   macro avg       0.23      0.50      0.31        20
weighted avg       0.20      0.45      0.28        20
```

Assigning random labels to images



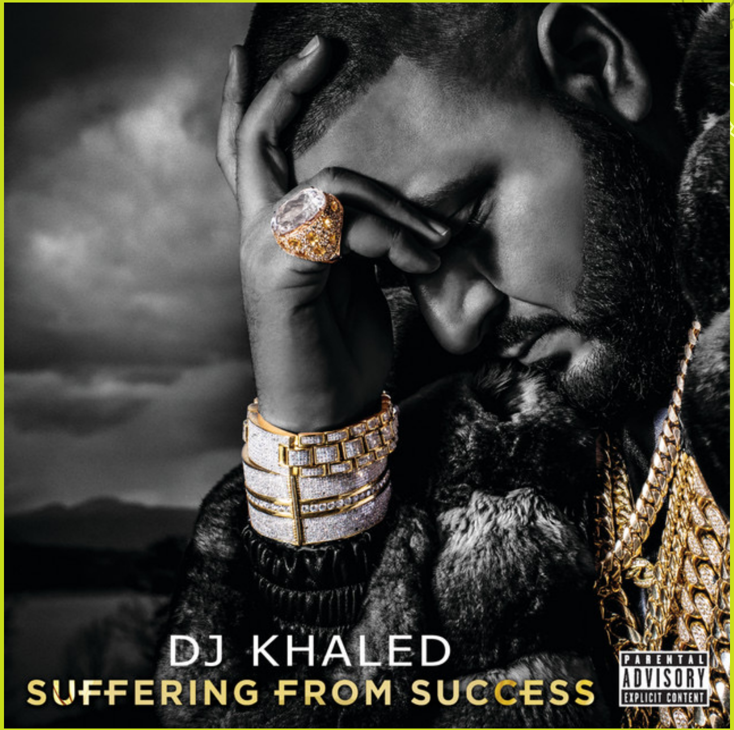


```
              precision    recall  f1-score   support

           0       0.49      0.46      0.47       485
           1       0.48      0.51      0.49       477

    accuracy                           0.48       962
   macro avg       0.48      0.48      0.48       962
weighted avg       0.48      0.48      0.48       962
```

TESTING

After our real life tests, we realized the model wasn't working properly.

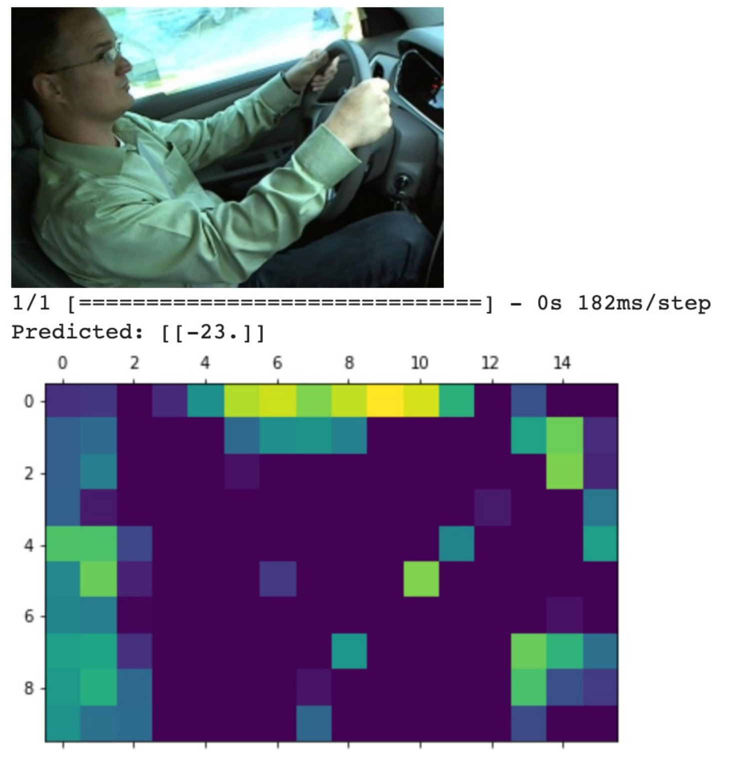
It was predicting every image as driving.



SOLVING THIS PROBLEM

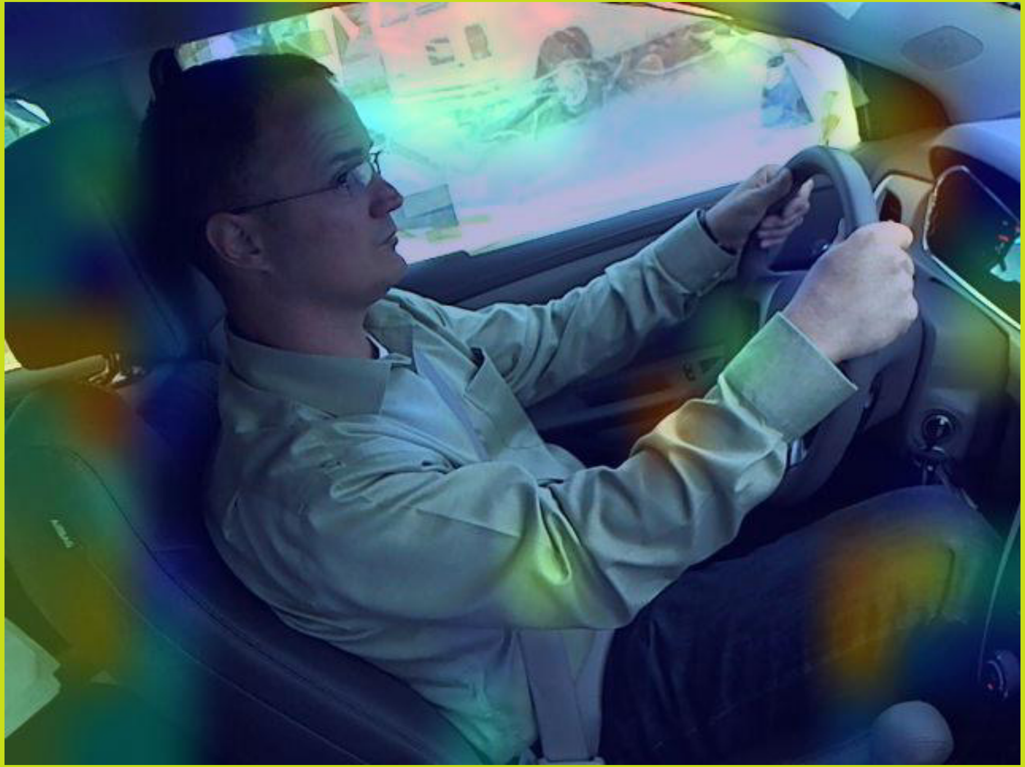
Following to this, I used Grad-CAM to understand what is wrong with my model.

Grad-Cam shows where your model focuses on.



```
1/1 [==============================] - 0s 182ms/step
Predicted: [[-23.]]
```

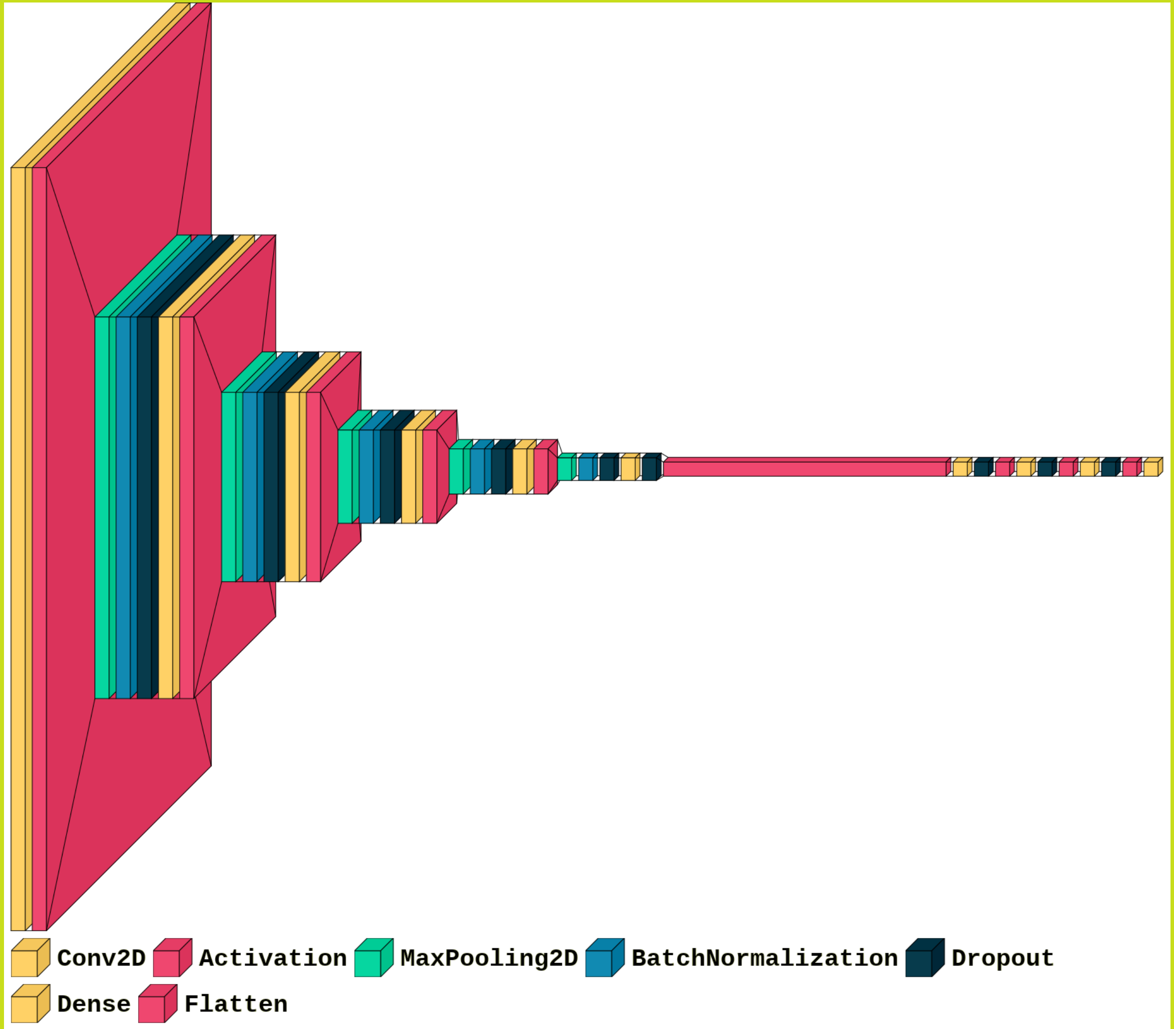
SOLVING THIS PROBLEM



SOLVING THIS PROBLEM

Then, I came up with solution ideas. Such as:

- Removing background from images to make model focusing only on arms
- Adding dropout layers to prevent model from being perfect
- Adding image augmentation like changing perspective of images, rotating images etc.
- Converting images to gray scale to make model less perfectionist about background



FINAL CNN MODEL

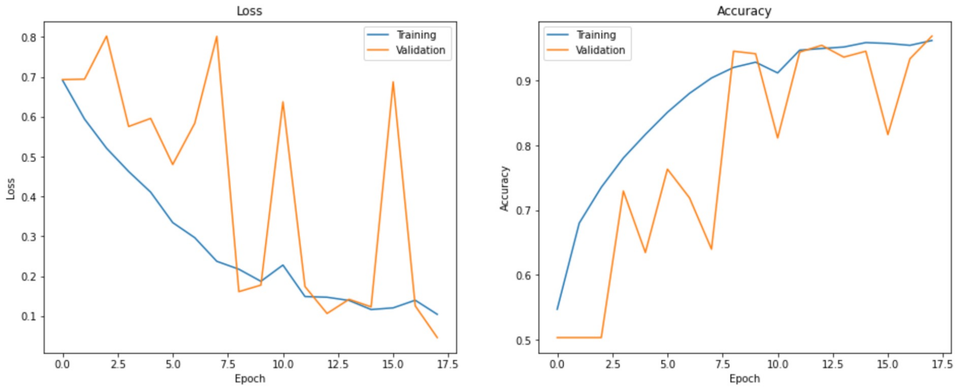
- 5 * (Conv2d layer,
  ReLU activation layer,
  MaxPooling 2d layer,
  Batch Normalization,
  Dropout(0.25))

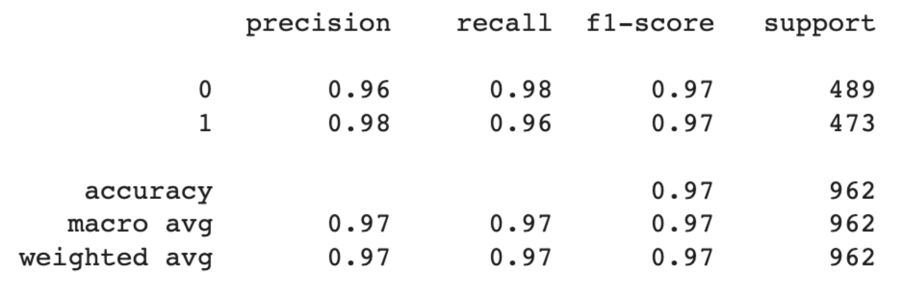
- 4 Dense layers with ReLU activation (with parameters of 128, 64, 32, 16)

- **Dropout(0.25) after every Dense layer**

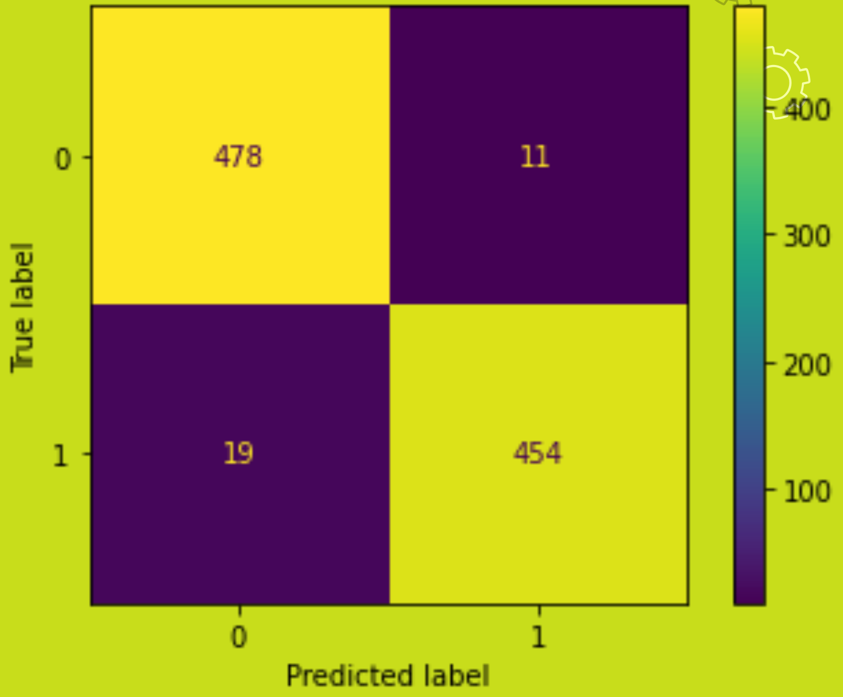
- Fully connected layer with sigmoid activation and 1 output (Binary output)

FINAL MODEL'S PERFORMANCE

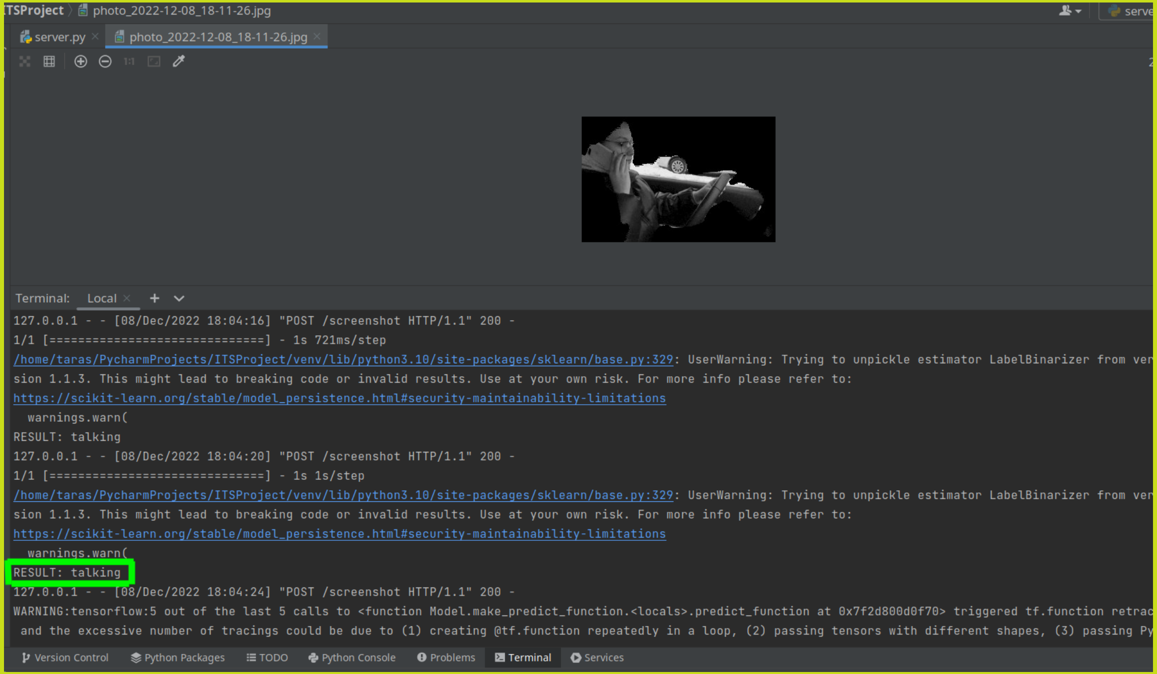




```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       489
           1       0.98      0.96      0.97       473

    accuracy                           0.97       962
   macro avg       0.97      0.97      0.97       962
weighted avg       0.97      0.97      0.97       962
```



A MODEL THAT DOESN'T RULE THEM ALL BUT AT LEAST WORKS IN REAL LIFE



```
127.0.0.1 - - [08/Dec/2022 18:04:16] "POST /screenshot HTTP/1.1" 200 -
1/1 [==============================] - 1s 721ms/step
/home/taras/PycharmProjects/ITSProject/venv/lib/python3.10/site-packages/sklearn/base.py:329: UserWarning: Trying to unpickle estimator LabelBinarizer from ver
sion 1.1.3. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
RESULT: talking
127.0.0.1 - - [08/Dec/2022 18:04:20] "POST /screenshot HTTP/1.1" 200 -
1/1 [==============================] - 1s 1s/step
/home/taras/PycharmProjects/ITSProject/venv/lib/python3.10/site-packages/sklearn/base.py:329: UserWarning: Trying to unpickle estimator LabelBinarizer from ver
sion 1.1.3. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
RESULT: talking
127.0.0.1 - - [08/Dec/2022 18:04:24] "POST /screenshot HTTP/1.1" 200 -
WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f2d800d0f70> triggered tf.function retrac
 and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Py
```

DEMO

CONCLUSION

In conclusion, this project presented a deep learning model that can minimize accidents caused by distracted drivers. Therefore, it is possible to prevent accidents by using deep learning Technologies.

THANK YOU!

DO YOU HAVE ANY QUESTIONS?