Traffic Sign Detection Project

Supervisor Dr . Ahmed Abdel-Reheem
Advisor Eng . Suhaila Mostafa

Outlines

1. Work Team
2. Introduction
3. Importance of Traffic Sign Detection
4. Technology
5. Problems statement
6. Solution
7. Background
8. Proposal
9. Methodology
10. User journey
11. Prototype
12. Demo Video
13. conclusion

Outlines

14. Future Work

15. Model & components

16. Hardware components

17. Code Snapshot

18. Demo Video

19. Team Members

Work Team

01 Ahmed Hamdy Gaber

02 Alhassan Mohamed Hassan

03 Aya Zain Elabdeen

04 Haneen Hamdy Rabie

05 Hlaa Osama Mohamed

06 Yasmeen Ehab Badr

Introduction

- **Auto Vision** :

- Enhancing Autonomous Driving with Traffic Sign Detection

- Brief overview of the importance of traffic sign detection in autonomous vehicles.

Importance of Traffic Sign Detection

- Safety enhancement: Real-time recognition of traffic signs for proactive vehicle response.

- Regulatory compliance: Adherence to traffic rules for safe and legal operation.

Technology

Python

Google Colab

Dart

Flutter

☹ Problems statement

➢ the speed

- There is a direct relationship between the increase in average speed, the probability of accidents occurring, and the severity of their consequences. As an example, every 1 km/h increase in average vehicle speed results in a 4% increase in the risk of a fatal accident and a 3% increase in the risk of a serious accident.

➢ Driving under the influence of alcohol and other psychoactive substances

- Driving under the influence of alcohol or any psychoactive substance or drug increases the risk of accidents that lead to death or serious injury.

➢ Not wearing motorcycle helmets, seat belts, or child restraints

- Wearing a helmet correctly can reduce the risk of a fatal accident by more than six times and reduce the risk of a traumatic brain injury by approximately 74%.

proposal

- We will make a model to avoid this problems that can avoid many accident from occur via camera and good application we can see that in the following slides

- advanced computer vision with a powerful dashcams camera to instantly recognize and display traffic signs

Background

➢ Artificial intelligence :

- Definition: Artificial Intelligence (AI) refers to the simulation of human intelligence in machines, enabling them to learn, reason, and perform tasks autonomously.

- Increased Efficiency: AI automates repetitive tasks, reducing errors and freeing up human resources for higher-value work.

Background

➢ Machine learning:

- Definition: Machine learning is a branch of artificial intelligence that enables computer systems to learn and improve from experience without being explicitly programmed.

- It is based on the idea that systems can automatically analyse data, detect patterns, and make intelligent predictions or decision.

- Data: Machine learning algorithms learn from large datasets to uncover patterns and extract relevant information

Background

➢ Deep learning:

- Definition: Deep learning is a subset of machine learning that utilizes artificial neural networks with multiple layers to process and analyse complex data.

- Mimicking the Human Brain: Deep learning algorithms aim to replicate the structure and functionality of the human brain's neural networks.

- Neural Networks: Deep learning models consist of interconnected layers of artificial neurons, enabling the network to learn and make predictions.

- Deep Layers: Deep learning models typically have multiple hidden layers that allow for the extraction of increasingly abstract features from the input data

Background

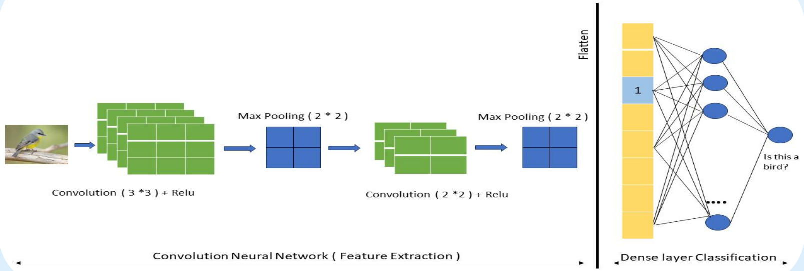
- ➢ **Two-dimensional Convolutional Neural Network:**

  ❑ **Introduction:**

  - Conv2D, or two-dimensional convolution, is a fundamental operation in deep learning, especially in the field of computer vision. It is widely used for processing and analyzing visual data such as images.

  - Conv2D is a type of convolutional layer commonly employed in convolutional neural networks (CNNs), a class of deep learning models designed for tasks like image recognition, object detection, and segmentation.

Background


Max Pooling (2 * 2)
Max Pooling (2 * 2)
Flatten
1
Is this a bird?
Convolution (3 *3) + Relu
Convolution (2 *2) + Relu
....
Convolution Neural Network (Feature Extraction)
Dense layer Classification

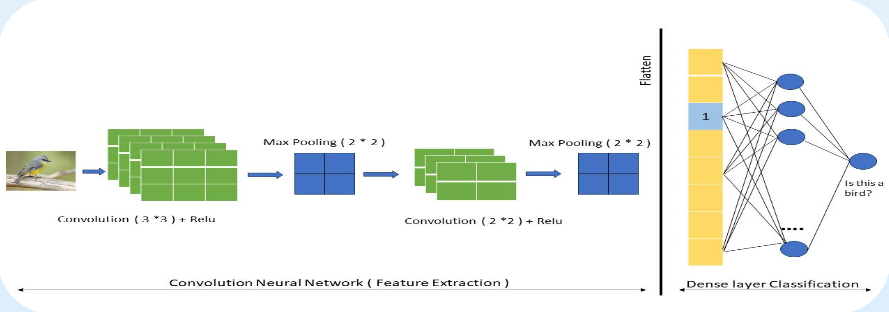
➢ Conv2d layers:

- The Conv2D layer in a neural network serves the purpose of learning spatial hierarchies of features within input data , particularly in the context of image data for tasks such as image recognition, object detection, and image segmentation.

➢ The "MaxPooling2D" layer:

- is a down-sampling operation commonly used in convolutional neural networks (CNNs).Its primary function is to reduce the spatial dimensions (width and height) of the input volume while retaining the most important information

Background


Max Pooling (2 * 2)
Max Pooling (2 * 2)
Flatten
1
Convolution (3 *3) + Relu
Convolution (2 *2) + Relu
Is this a bird?
Convolution Neural Network (Feature Extraction)
Dense layer Classification

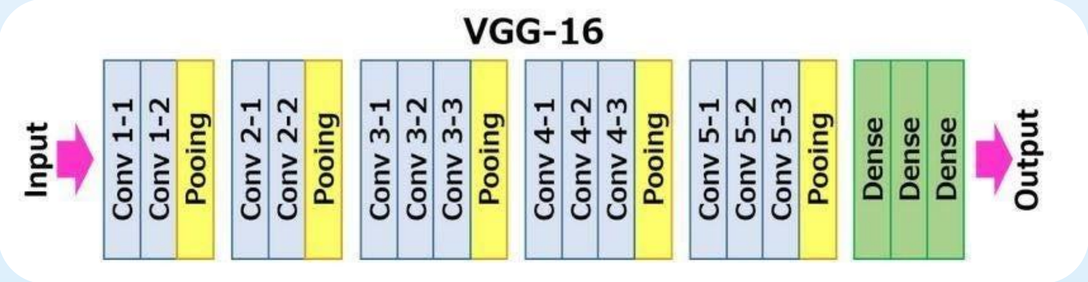
➢ Flatten layers:

- This is necessary when transitioning from convolutional or recurrent layers to fully connected layers, as fully connected layers require one-dimensional input.

➢ The Dense layer:

- Function This Layer Performs The Final Classification Or Prediction And Producing A Single Output Value Between 0 And 1

Background



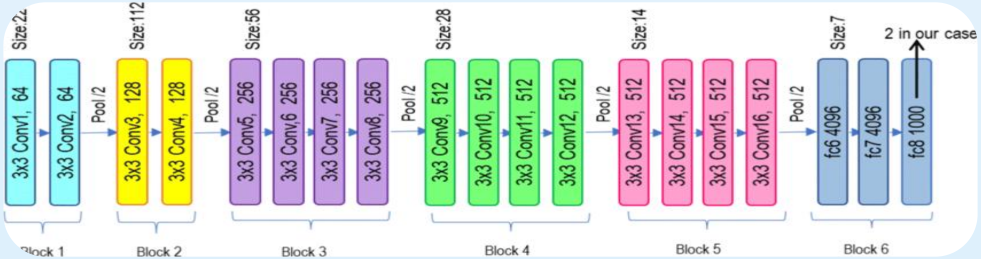
➢ Visual Geometry Group 16 network (VGG16) :

❖ VGG-16, short for Visual Geometry Group 16, is a convolutional neural network.

❖ VGG-16 is characterized by its depth and simplicity. It consists of :

- 16 layers, including:
- 13 convolutional layers: These layers perform feature extraction.
- 3 fully connected

Background

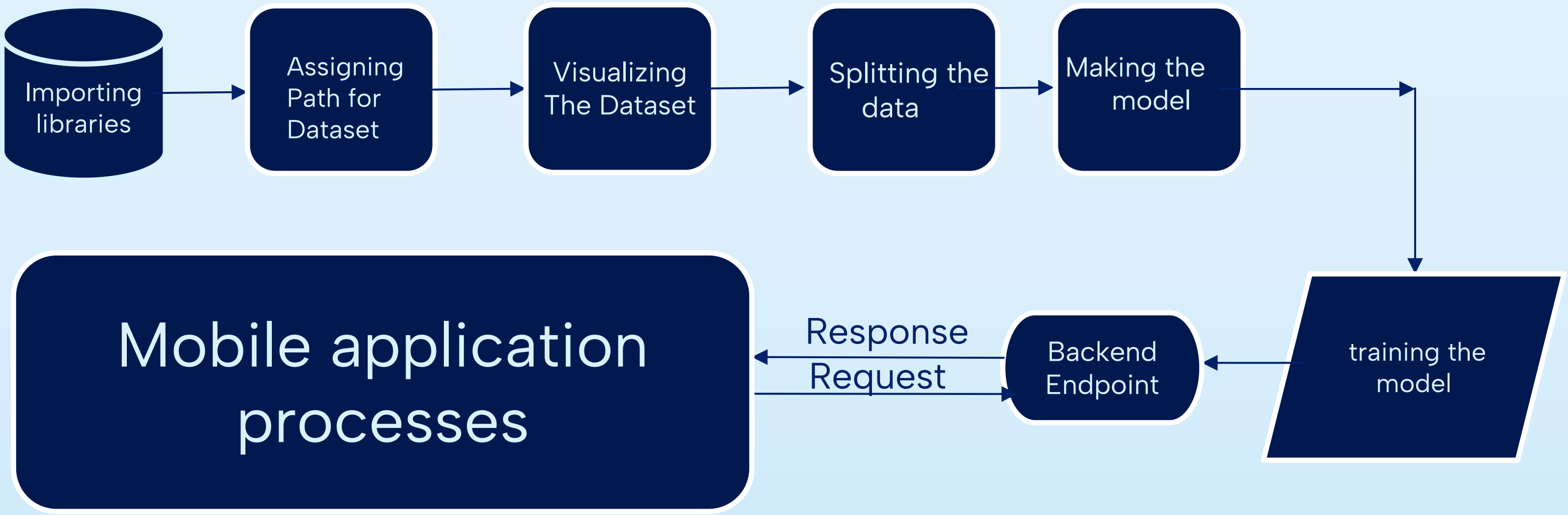


➢ Visual Geometry Group19 network (VGG19):

❖ VGG-19, short for Visual Geometry Group19, is a convolutional neural network. (CNN) model developed by theVisual Geometry Group It is an extension of the earlier VGG16 model and is known for its simplicity and effectiveness in image classification tasks.

❖ VGG-19 is an extension of the earlier VGG-16 model.
It consists of 19 layers, including 16 convolutional layers and 3 fully connected layers.
The architecture is characterized by its depth and simplicity.

Traffic Sign Detection Project Flowchart:



Methodology

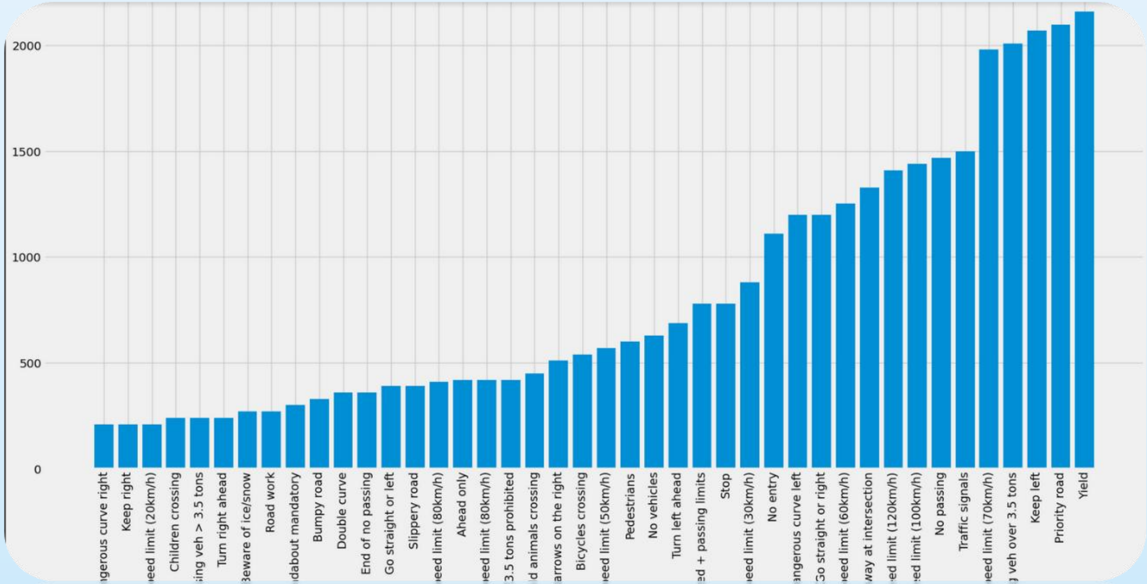
1. Collecting the dataset
2. Visualizing the dataset
3. Training the model
4. Evaluating the model

1.Collecting the dataset

- Originally, the dataset comes from Kaggle
- It have :

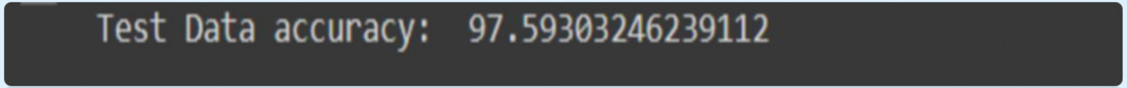
Number of classes	43 classes
Types of images	Single image
Number of images	More than 50.000

2.Visualization the dataset



3. Training the model

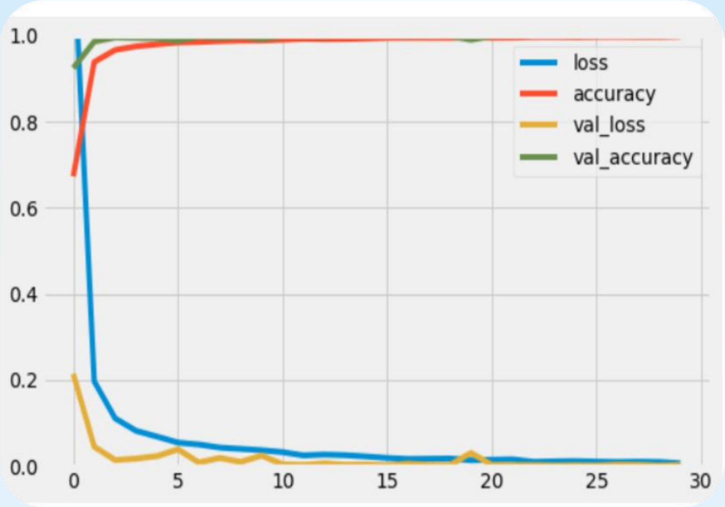
- After Training And Testing These Algorithms On The Dataset, We've Came To These Result:



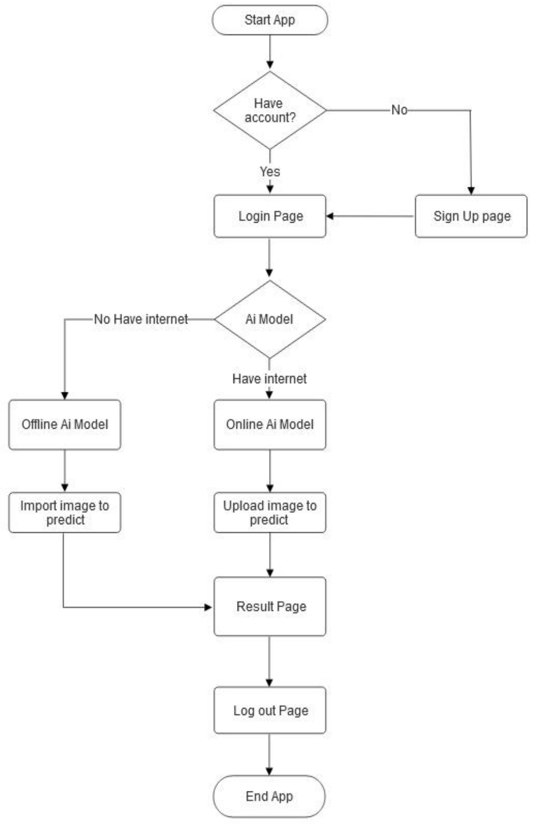
```
Test Data accuracy:  97.59303246239112
```

4. Evaluating the model

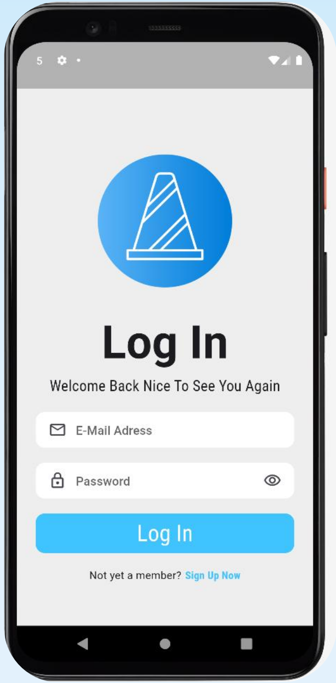
- Our model boasts an exceptional accuracy rate of 97% :



User Journey

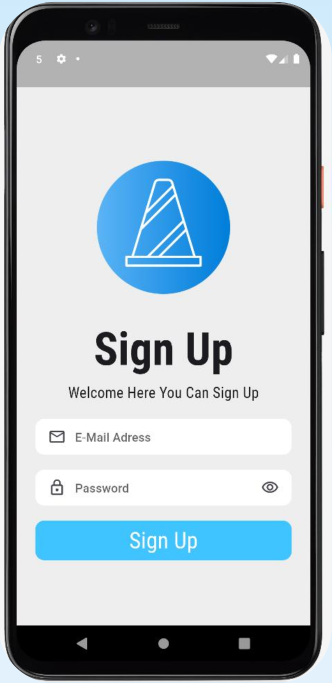


Prototype



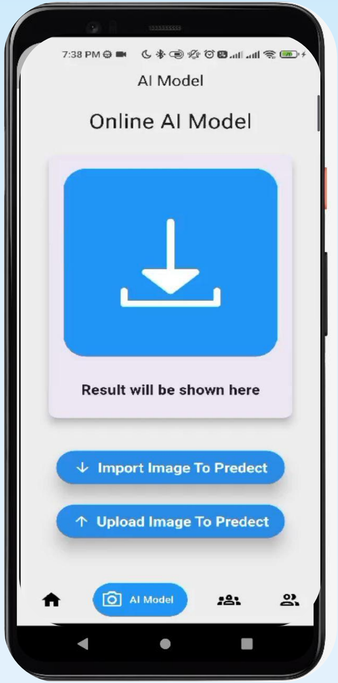
Log In

Welcome Back Nice To See You Again



Sign Up

Welcome Here You Can Sign Up

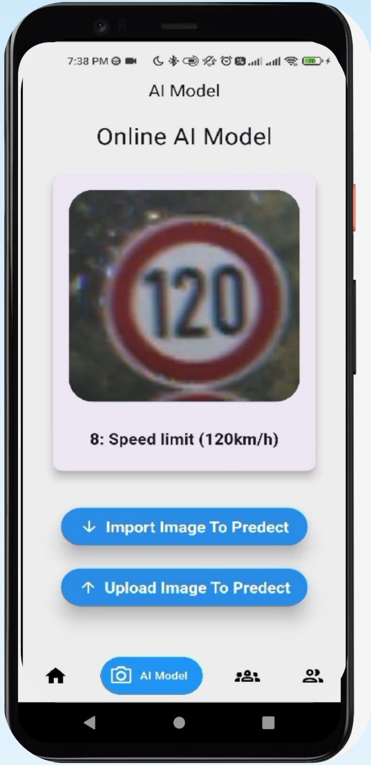


AI Model

Online AI Model

Result will be shown here

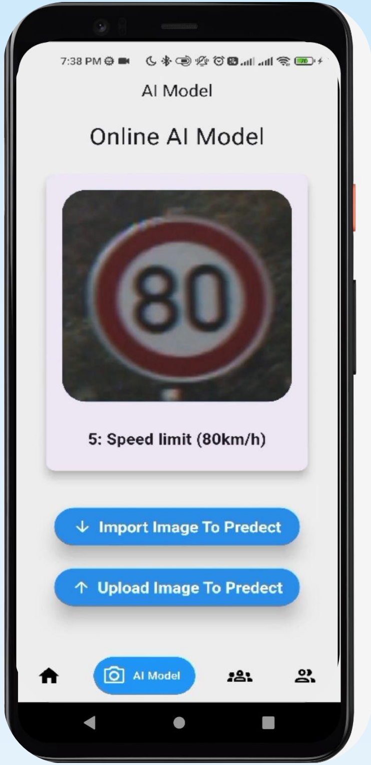
Prototype



AI Model

Online AI Model

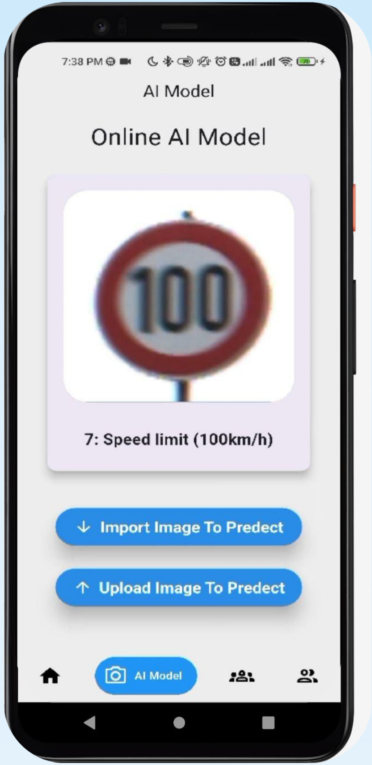
8: Speed limit (120km/h)



AI Model

Online AI Model

5: Speed limit (80km/h)

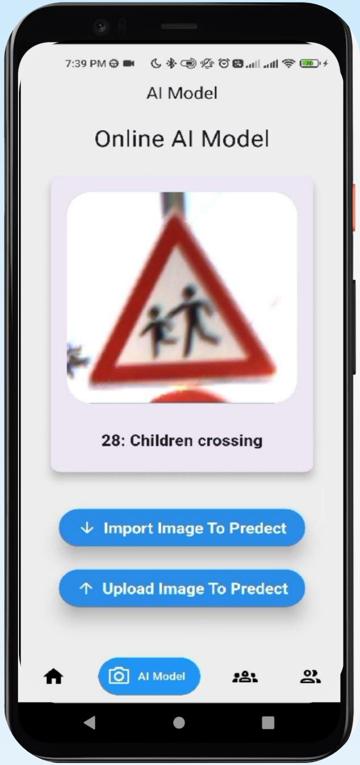


AI Model

Online AI Model

7: Speed limit (100km/h)

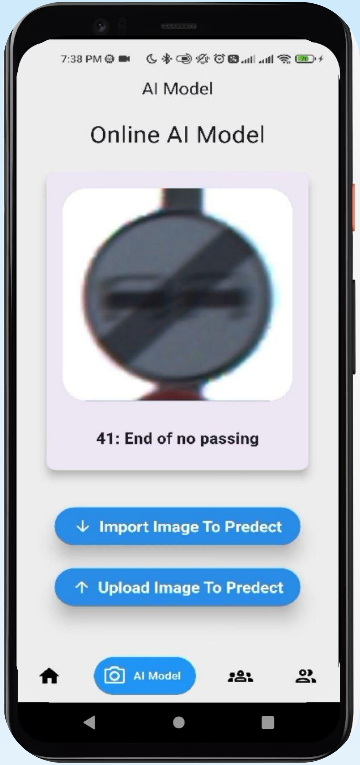
Prototype



AI Model

Online AI Model

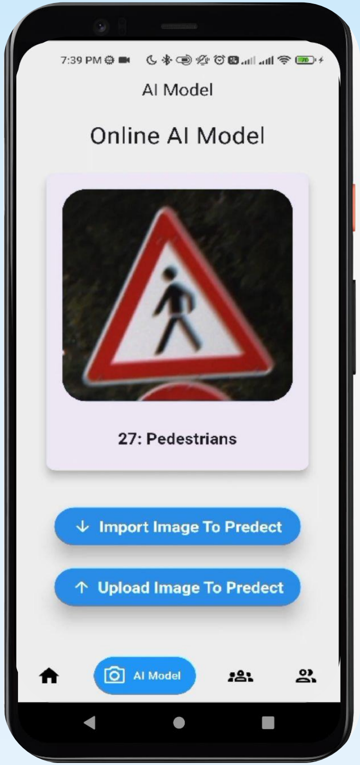
28: Children crossing



AI Model

Online AI Model

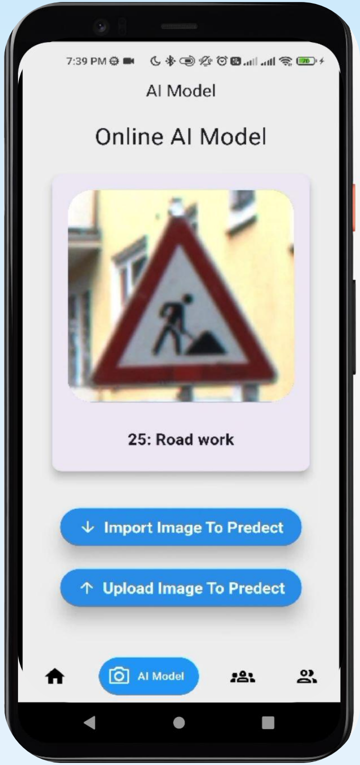
41: End of no passing



AI Model

Online AI Model

27: Pedestrians

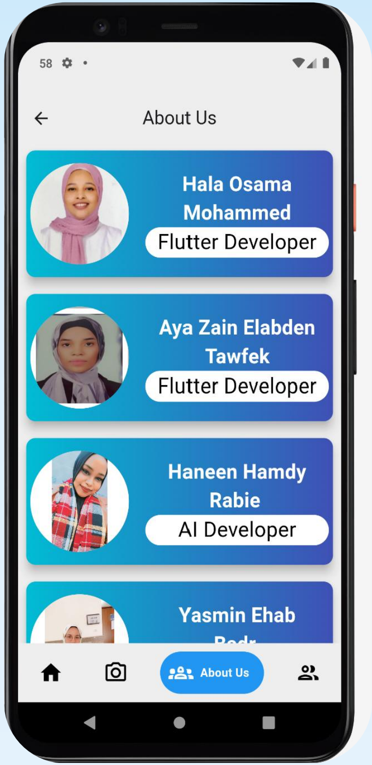


AI Model

Online AI Model

25: Road work

Prototype



Hala Osama Mohammed

Flutter Developer

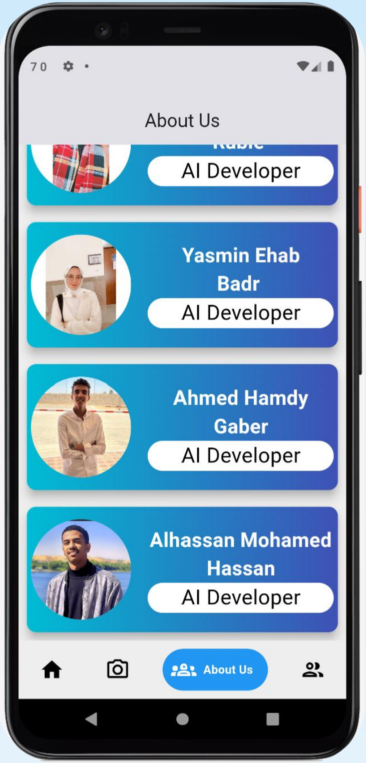
Aya Zain Elabden Tawfek

Flutter Developer

Haneen Hamdy Rabie

AI Developer

Yasmin Ehab Badr



Rabie

AI Developer

Yasmin Ehab Badr

AI Developer

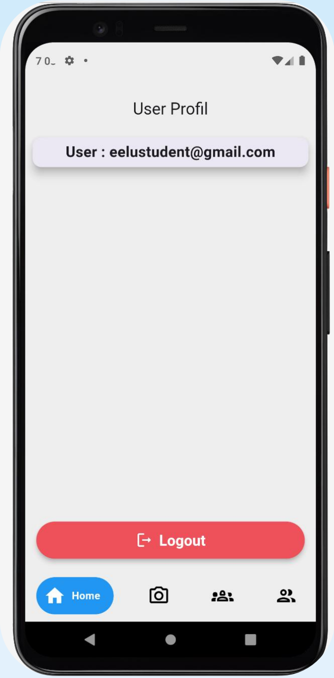
Ahmed Hamdy Gaber

AI Developer

Alhassan Mohamed Hassan

AI Developer

Prototype

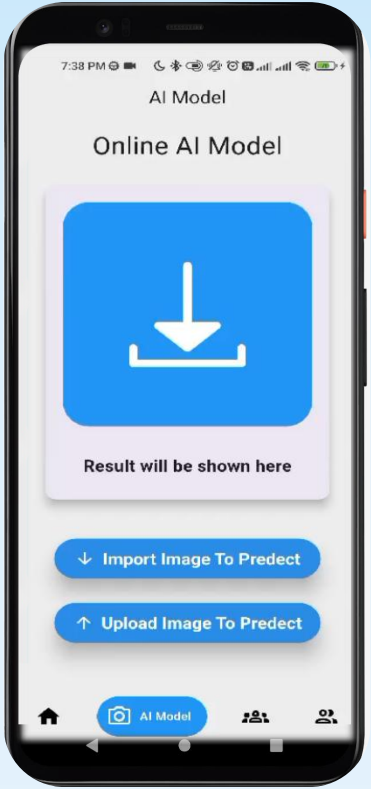


7 0_

User Profil

User : eelustudent@gmail.com

Demo Video



AI Model

Online AI Model

Result will be shown here

❑ conclusion

- The main objective of this study is to categorize the prime approaches into sections to make the concept easily understandable and to give a comprehensive review of different methods available for traffic sign detection and recognition; including along with database and research trends associated with TSDR systems [6]. In this paper, the detection of traffic signs using color and shape based on different techniques was explored.
- This study suggests that support vector machine and convolutional neural network based on methods were found most efficient and used because they have a high rate of detection, are very flexible, and are simple to adopt. [2] With the aid of new datasets from other nations, the total performance might be enhanced and tailored.
- After we used CNN model achieving an accuracy score of 97%

Model & Components

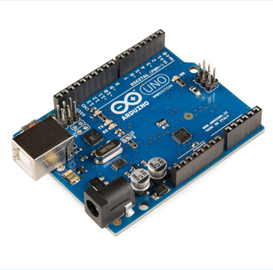
- Esp32 Camera
- Esp32 CAM.USB
- Aurduino
- Breadboard
- Jumper Wires

A prototype car

Hardware Components

Esp32 Camera

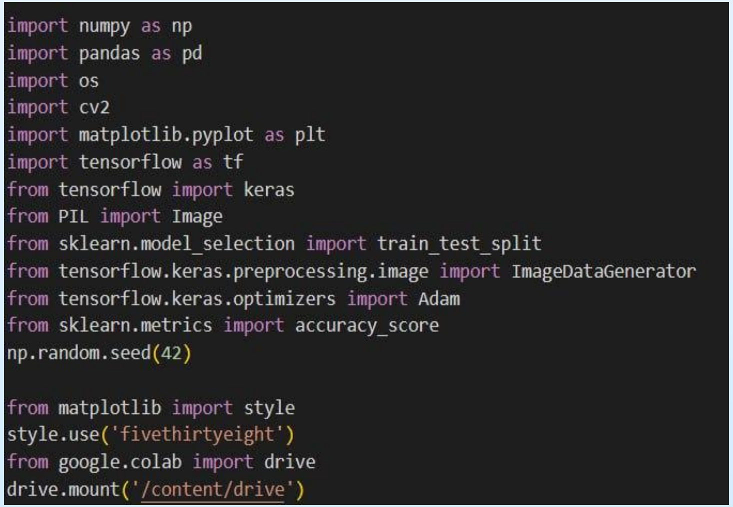
Aurduino



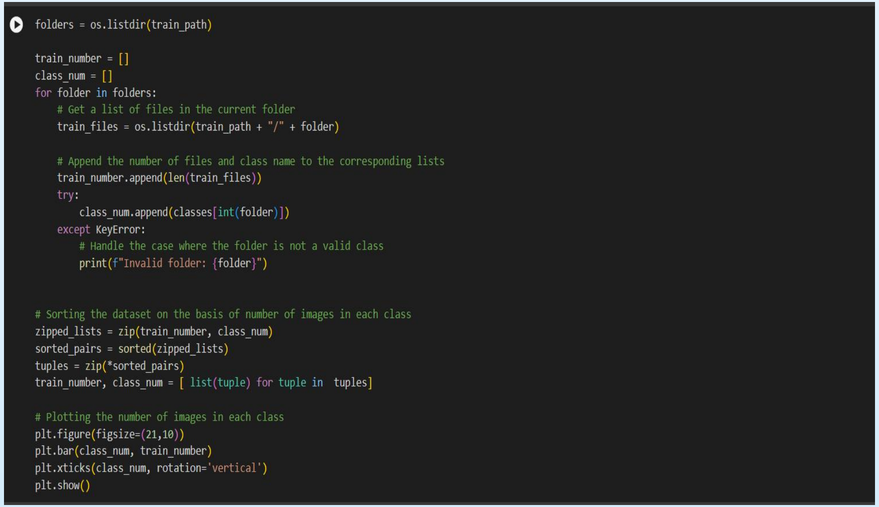
Breadboard

Jumper Wires

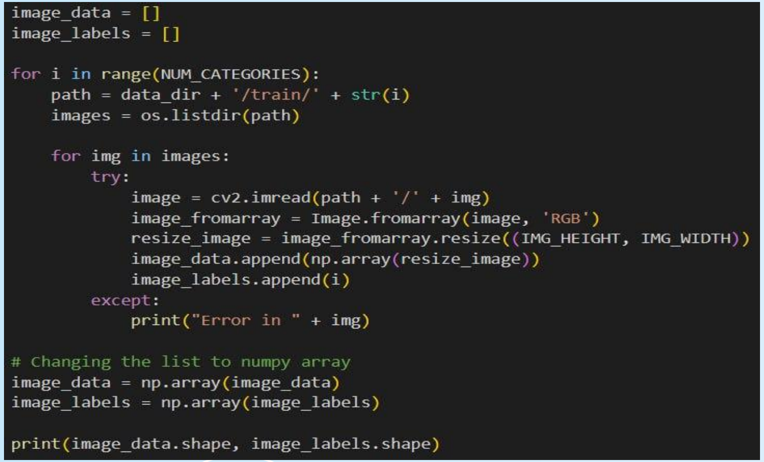
Code Snapshot



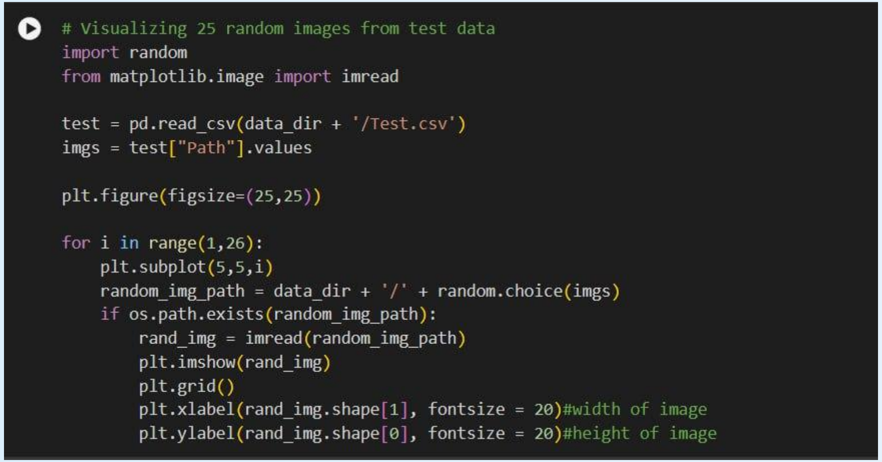
```
import numpy as np
import pandas as pd
import os
import cv2
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from PIL import Image
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import accuracy_score
np.random.seed(42)

from matplotlib import style
style.use('fivethirtyeight')
from google.colab import drive
drive.mount('/content/drive')
```



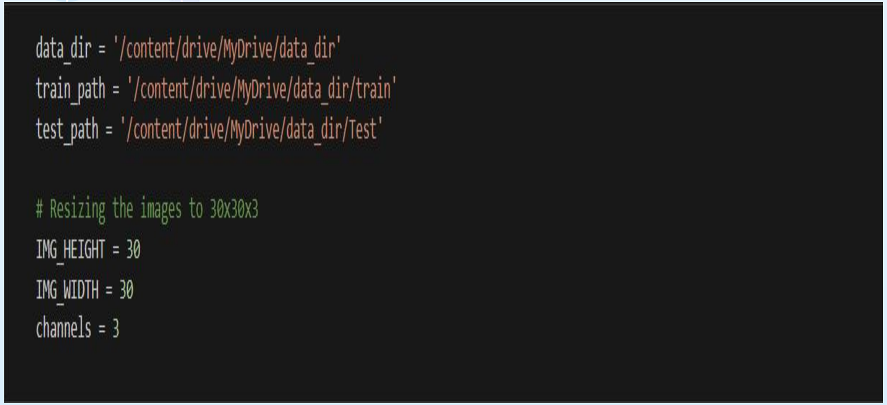
```
folders = os.listdir(train_path)

train_number = []
class_num = []
for folder in folders:
    # Get a list of files in the current folder
    train_files = os.listdir(train_path + "/" + folder)

    # Append the number of files and class name to the corresponding lists
    train_number.append(len(train_files))
    try:
        class_num.append(classes[int(folder)])
    except KeyError:
        # Handle the case where the folder is not a valid class
        print(f"Invalid folder: {folder}")


# Sorting the dataset on the basis of number of images in each class
zipped_lists = zip(train_number, class_num)
sorted_pairs = sorted(zipped_lists)
tuples = zip(*sorted_pairs)
train_number, class_num = [ list(tuple) for tuple in  tuples]

# Plotting the number of images in each class
plt.figure(figsize=(21,10))
plt.bar(class_num, train_number)
plt.xticks(class_num, rotation='vertical')
plt.show()
```



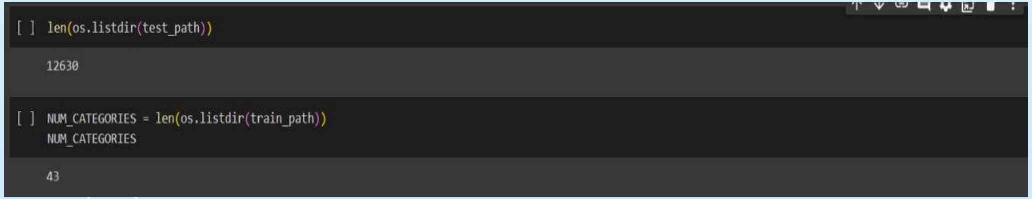
```python
image_data = []
image_labels = []

for i in range(NUM_CATEGORIES):
    path = data_dir + '/train/' + str(i)
    images = os.listdir(path)

    for img in images:
        try:
            image = cv2.imread(path + '/' + img)
            image_fromarray = Image.fromarray(image, 'RGB')
            resize_image = image_fromarray.resize((IMG_HEIGHT, IMG_WIDTH))
            image_data.append(np.array(resize_image))
            image_labels.append(i)
        except:
            print("Error in " + img)

# Changing the list to numpy array
image_data = np.array(image_data)
image_labels = np.array(image_labels)

print(image_data.shape, image_labels.shape)
```



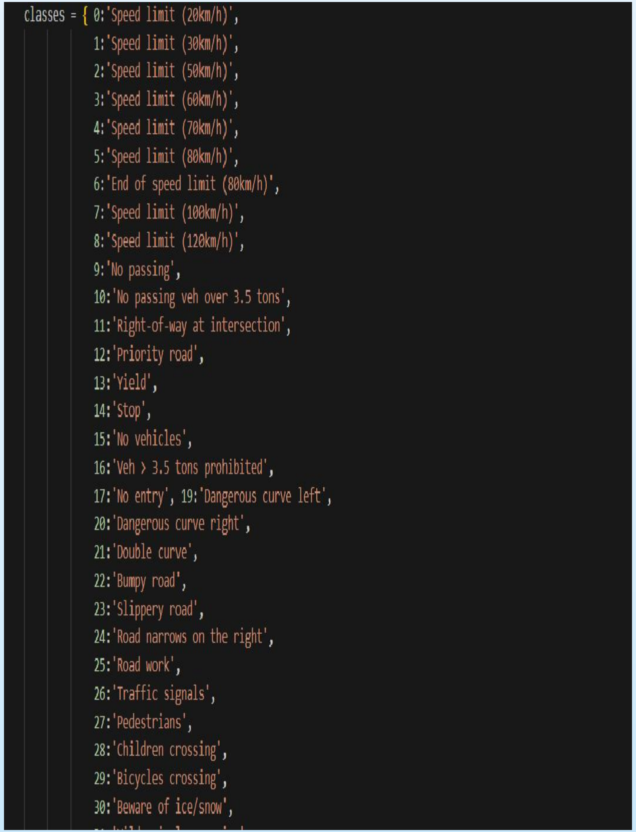
```python
# Visualizing 25 random images from test data
import random
from matplotlib.image import imread

test = pd.read_csv(data_dir + '/Test.csv')
imgs = test["Path"].values

plt.figure(figsize=(25,25))

for i in range(1,26):
    plt.subplot(5,5,i)
    random_img_path = data_dir + '/' + random.choice(imgs)
    if os.path.exists(random_img_path):
        rand_img = imread(random_img_path)
        plt.imshow(rand_img)
        plt.grid()
        plt.xlabel(rand_img.shape[1], fontsize = 20)#width of image
        plt.ylabel(rand_img.shape[0], fontsize = 20)#height of image
```



```
data_dir = '/content/drive/MyDrive/data_dir'
train_path = '/content/drive/MyDrive/data_dir/train'
test_path = '/content/drive/MyDrive/data_dir/Test'

# Resizing the images to 30x30x3
IMG_HEIGHT = 30
IMG_WIDTH = 30
channels = 3
```

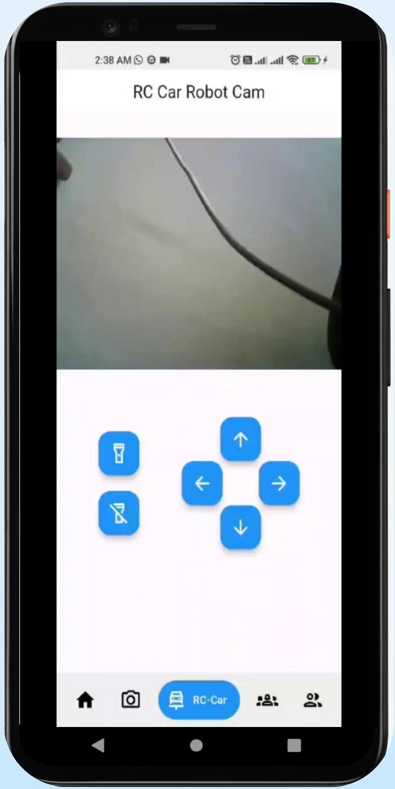


```
[ ] len(os.listdir(test_path))

    12630

[ ] NUM_CATEGORIES = len(os.listdir(train_path))
    NUM_CATEGORIES

    43
```



```
classes = { 0:'Speed limit (20km/h)',
            1:'Speed limit (30km/h)',
            2:'Speed limit (50km/h)',
            3:'Speed limit (60km/h)',
            4:'Speed limit (70km/h)',
            5:'Speed limit (80km/h)',
            6:'End of speed limit (80km/h)',
            7:'Speed limit (100km/h)',
            8:'Speed limit (120km/h)',
            9:'No passing',
            10:'No passing veh over 3.5 tons',
            11:'Right-of-way at intersection',
            12:'Priority road',
            13:'Yield',
            14:'Stop',
            15:'No vehicles',
            16:'Veh > 3.5 tons prohibited',
            17:'No entry', 19:'Dangerous curve left',
            20:'Dangerous curve right',
            21:'Double curve',
            22:'Bumpy road',
            23:'Slippery road',
            24:'Road narrows on the right',
            25:'Road work',
            26:'Traffic signals',
            27:'Pedestrians',
            28:'Children crossing',
            29:'Bicycles crossing',
            30:'Beware of ice/snow',
```

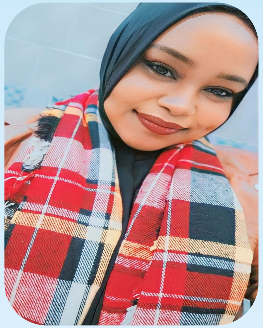
Demo Video



Team Member

Ai Team

Ai Team



Ai Team

Ai Team

Flutter Team

Flutter Team

Thanks!

Any questions?