VEHICLE DETECTION

Presented by: Suliman alotaibi

أكاديمية طويق
Tuwaiq Academy

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

Introduction:

The aim of this project is to predict the presence of cars and motorcycles moving on the dedicated bus lane, which disrupts the speed of buses. These bus lanes are specifically designed to provide people with a fast and efficient means of transportation between stations, allowing them to reach their destinations quickly while avoiding the traffic congestion caused by cars and motorcycles.

Data Set:

This is a vehicle image classification dataset containing images of four different types of vehicles: Car, Bus, and Motorcycle. The dataset is curated to help learners to develop and evaluate image classification models for identifying various vehicle types from images.

https://www.kaggle.com/datasets/kaggleashwin/vehicle-type-recognition

The Data is spilt in three folder images for the BUS ,CAR,MOTOR
and i applied manually spilt of the data and classess for car is 0 and bus 1 motor 2

Libaries imports:

```
# Import libraries
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,Activation
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from sklearn.metrics import confusion_matrix
import shutil
import glob
import itertools
import os
import random
import numpy as np
import matplotlib.pyplot as plt
```

Handling the data:

First think that what i do here is my data is one folder that have the images of cars in buss aloe the motors
every images have in first name belong to, so i spilt my data here two three folder path for train and test and valid

```
os.chdir('/content/my_data')

if not os.path.exists('train/car'):
    os.makedirs('train/car')
    os.makedirs('train/bus')
    os.makedirs('train/mot')
    os.makedirs('valid/car')
    os.makedirs('valid/bus')
    os.makedirs('valid/mot')
    os.makedirs('test/car')
    os.makedirs('test/bus')
    os.makedirs('test/mot')


    for c in random.sample(glob.glob('car*'), 69):
        shutil.move(c, 'train/car' )
    for c in random.sample(glob.glob('bus*'), 69):
        shutil.move(c, 'train/bus')
    for c in random.sample(glob.glob('mot*'), 69):
        shutil.move(c, 'train/mot')
    for c in random.sample(glob.glob('car*'), 20):
        shutil.move(c, 'valid/car')
    for c in random.sample(glob.glob('bus*'), 20):
        shutil.move(c, 'valid/bus')
    for c in random.sample(glob.glob('mot*'), 20):
        shutil.move(c, 'valid/mot')
    for c in random.sample(glob.glob('car*'), 10):
        shutil.move(c, 'test/car')
    for c in random.sample(glob.glob('bus*'), 10):
        shutil.move(c, 'test/bus')
    for c in random.sample(glob.glob('mot*'), 10):
        shutil.move(c, 'test/mot')
```

Data Collection & Preprocessing :

i prepare the data from the path of train and valid and test in same time i a fiting the data(image) in size of (224*224) also assing a label of all three classes.

```
#assign an valrbale that have the path of train and valid and test
train_path = '/content/my_data/train'
valid_path = '/content/my_data/valid'
test_path = '/content/my_data/test'
```

```
#to be sure all the images in the same size before statring my model i use ImageDataGenerator with vgg16
train_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input)\
    .flow_from_directory(directory=train_path, target_size=(224,224), classes = ['car','bus','mot'], batch_size=10)
valid_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input)\
    .flow_from_directory(directory=valid_path, target_size=(224,224), classes = ['car','bus','mot'], batch_size=10)
test_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.vgg16.preprocess_input)\
    .flow_from_directory(directory=test_path, target_size=(224,224), classes = ['car','bus','mot'], batch_size=10, shuffle=False)
```

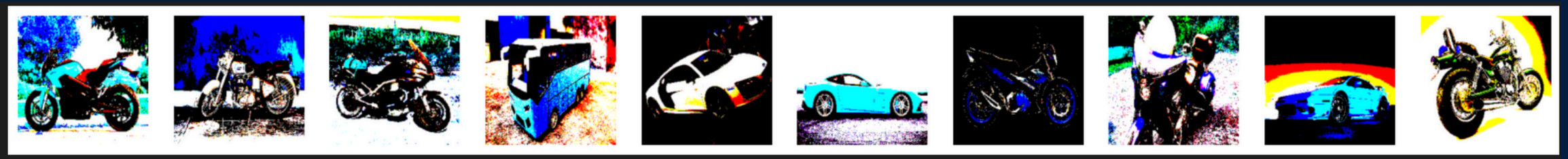
```
Found 207 images belonging to 3 classes.
Found 60 images belonging to 3 classes.
Found 30 images belonging to 3 classes.
```

Model Selection & Implementation:

I chose CNN model to clasefy the three chose thre chose the classes

```
'''
bulding the model, with start the filters 32 for the kerinal 3*3 metrix activatin relu for the hidden layer
and padding for if is more pixel is take the same and don't take care for road input size 224*224*3 for the image
3 is for the color rbg
'''
model = Sequential([
    Conv2D(filters=32, kernel_size=(3,3) ,activation = 'relu', padding = 'same' , input_shape=(224,224,3)),
    MaxPooling2D(pool_size=(2,2), strides=2),
    Conv2D(filters=64, kernel_size=(3,3),activation = 'relu', padding = 'same'),
    MaxPooling2D(pool_size=(2,2), strides=2),
    Flatten(),
    Dense(units=3, activation='softmax')
])
```

Showing our three clasess:

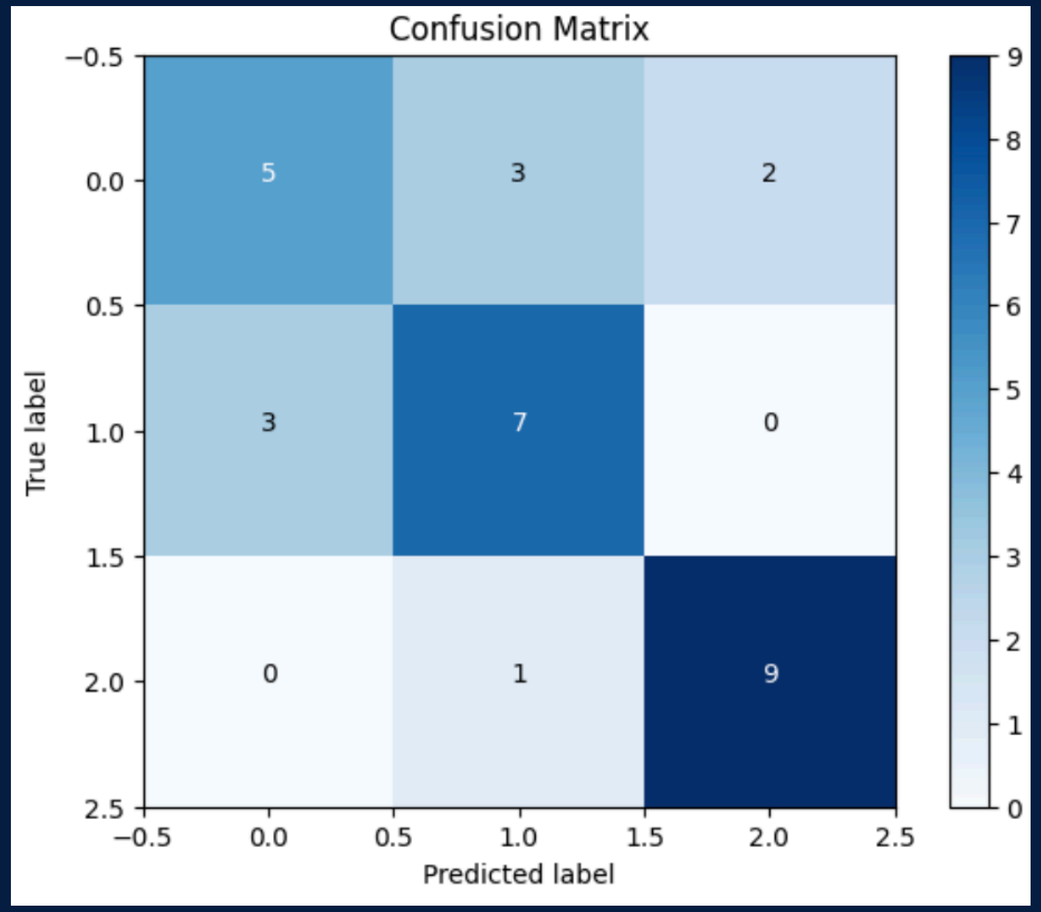


```
[[0. 0. 1.]
 [0. 0. 1.]
 [0. 0. 1.]
 [0. 1. 0.]
 [1. 0. 0.]
 [1. 0. 0.]
 [0. 0. 1.]
 [0. 0. 1.]
 [1. 0. 0.]
 [0. 0. 1.]]
```

The arthucture:

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 224, 224, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_4 (Conv2D)	(None, 112, 112, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 56, 56, 64)	0
flatten (Flatten)	(None, 200704)	0
dense (Dense)	(None, 3)	602,115

Evaluation:



Validation Loss: 5.703274726867676
Validation Accuracy: 0.7666666507720947

Conclusion:

#1. What challenges did you face during data collection and how did you overcome them?

The data was seperatd in multiple failes so i gathered them into files for train and validation and test and label them in the same time, but the big challenge was the data is small

#2. Why did you choose your specific model architecture?

because i want the high probability of one class to my three clasess

#3. How does the performance of your chosen model compare to alternative approaches, and what factors contributed to the differences?

the performance was good but is hve an overfting

#4. What did you learn from storing the data in a database, and how did it impact your workflow?
i didn’t use database and i need go back of the database lecture in my on,(beabause no one give us the class of it)

#5. If you were to extend this project, what additional features or improvements would you consider?
explore more data and use Technique to solve overrating problems