CV Report 2

Teacher: Prof.Mehul Raval

Date: 16/02/2022

- Team : The Weekenders
- Members :
  - Aditya Raj(AU1920177)
  - Dhruvanshu Parmar(AU1940166)

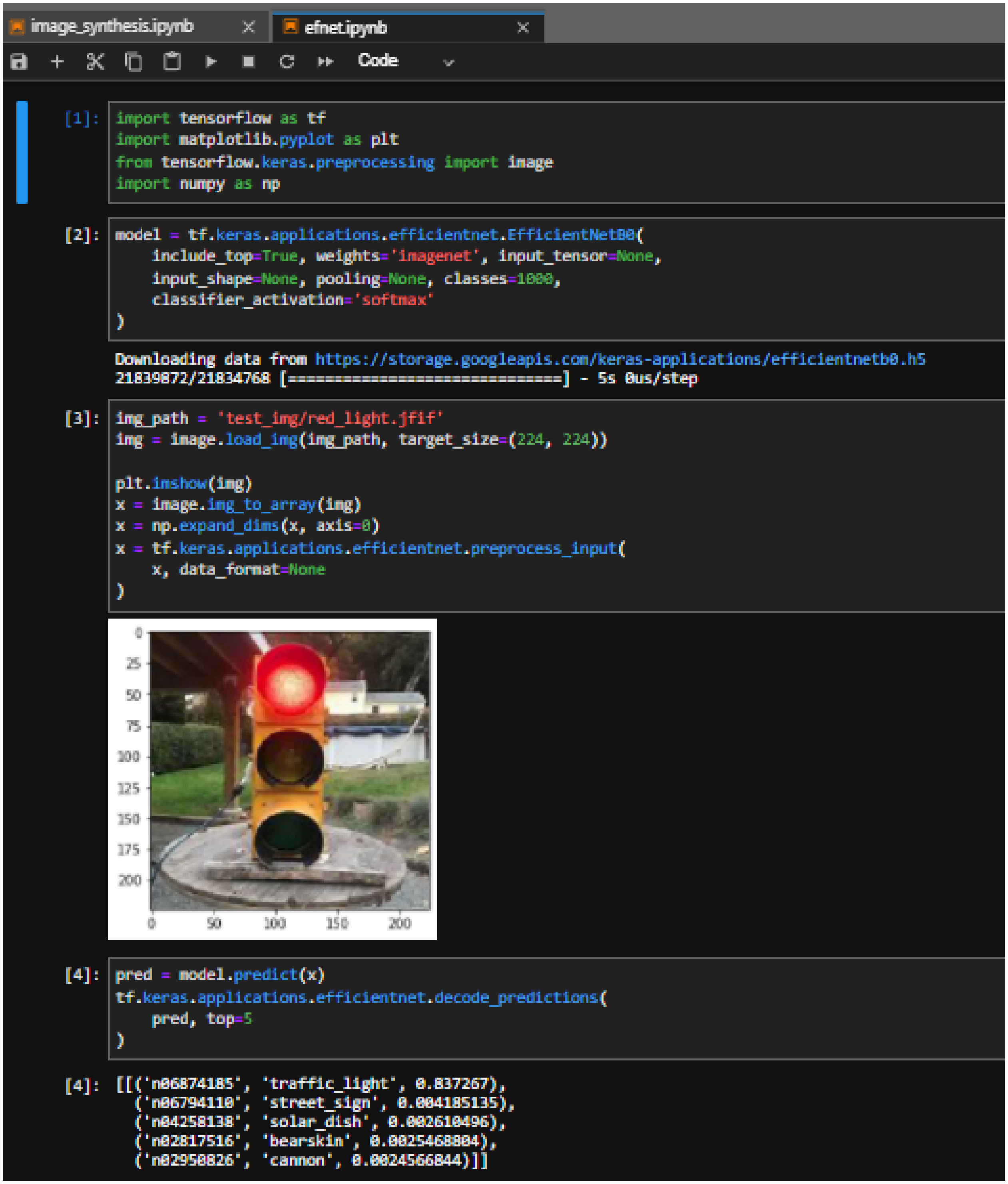
Task Performed in this week :

This week we are able to explore efficientNet neural network architecture developed by google AI. After Reading the Official Paper on *EfficientNet* [1], we are convinced that we are going to use this architecture in our Project and proceed further with our new milestone.

Then we searched for the Implementation of this architecture and we came across Kearas application where there are two versions of EffecientNet implementation available and we decided to use the 2nd version as it is better than the first version. And then we played around with the pretrained model available for prediction.

Outcomes :

In this test we experimented with EffecientNetB0 variant. This variant is the base variant for another variant and this goes from B1 - B7. The base variant requires less computation and as expected it has less accuracy than other variants but this is best for the embedded system. Since we have to do this under fog condition for sign detection, we need higher accuracy so we will use higher accuracy variant like B5, B6 or B7.



```
[1]: import tensorflow as tf
     import matplotlib.pyplot as plt
     from tensorflow.keras.preprocessing import image
     import numpy as np

[2]: model = tf.keras.applications.efficientnet.EfficientNetB0(
         include_top=True, weights='imagenet', input_tensor=None,
         input_shape=None, pooling=None, classes=1000,
         classifier_activation='softmax'
     )

     Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0.h5
     21839872/21834768 [==============================] - 5s 0us/step

[3]: img_path = 'test_img/red_light.jfif'
     img = image.load_img(img_path, target_size=(224, 224))

     plt.imshow(img)
     x = image.img_to_array(img)
     x = np.expand_dims(x, axis=0)
     x = tf.keras.applications.efficientnet.preprocess_input(
         x, data_format=None
     )

[4]: pred = model.predict(x)
     tf.keras.applications.efficientnet.decode_predictions(
         pred, top=5
     )

[4]: [[('n06874185', 'traffic_light', 0.837267),
       ('n06794110', 'street_sign', 0.004185135),
       ('n04258138', 'solar_dish', 0.002610496),
       ('n02817516', 'bearskin', 0.0025468804),
       ('n02950826', 'cannon', 0.0024566844)]]
```

Tasks to be performed next week:

Our next milestone is to preprocess the data for model input for transfer learning. Preprocessing data constitutes many parts, labeling data or acquiring labeled data, resizing the data, and converting in proper format for model.

Reference:

[1]. arXiv:1905.11946 **[cs.LG]**