Project Report

© Shubham Kondekar @ Intern at CloudCounselage

Github link : https://github.com/kondekarshubham123/ (https://github.com/kondekarshubham123/)

PROBLEM STATEMENT:

Interns are expected to create a TensorFlow model for identification of color. Your code will be given an image as an input and the code is supposed to categorize which RGB color it matches to. This code is to be done in Python.

On completion, create a Github account and create a repository and commit your code in the same.

Input:

Solution

Generated data

I have generated dataset to train model

code for dataset generation is located in

```
color-recognition/Dataset/DatasetGenerator/
```

Run `python Generate.py`

it will generate datset for our project

This model can only recognise the following colours

```
1. Red
2. Blue
3. Orange
4. Green
5. Yellow
6. Indigo
7. Voilet
```

Requirement for this model are

```
python3.X
tensorflow
pillow library
matplotlib library
```

Trained model

Trained Model is located in

```
src/keras_model.h5
```

Locate Model

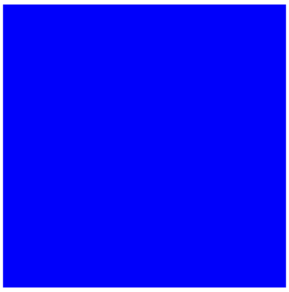
In [2]:

```
import os
os.chdir('src/')
print(os.getcwd())
```

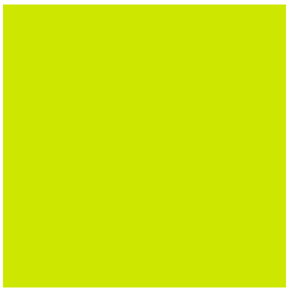
```
/home/shubham/Documents/color-recognition/src
```

Input Image file IP.png

Here are few Examples of Generated dataset give it as input to program



```
Image Name : IP.png
```



```
Image Name : IP2.png
```

```
Image Name : IP3.png
```

In [3]:

```python
#Import Library
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np
```

In [4]:

```python
# Disable scientific notation for clarity
np.set_printoptions(suppress=True)
```

In [5]:

```python
# Load the model
model = tensorflow.keras.models.load_model('keras_model.h5')
```

```
WARNING:tensorflow:No training configuration found in the save file, so t
he model was *not* compiled. Compile it manually.
```

In [6]:

```python
# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
```

In [39]:

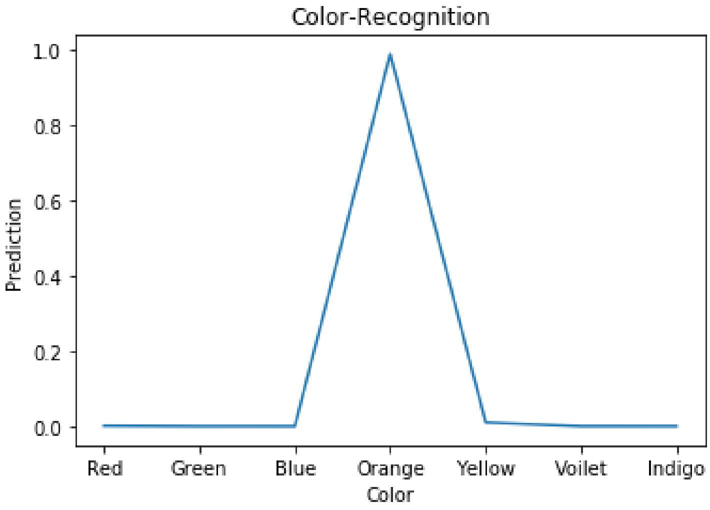
```python
# Replace this with the path to your image
image = Image.open('IP3.png')
```

In [40]:

```python
#resize the image to a 224x224 with the same strategy as in TM2:
#resizing the image to be at least 224x224 and then cropping from the
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)

#turn the image into a numpy array
image_array = np.asarray(image)

# display the resized image
image.show()

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

# Load the image into the array
data[0] = normalized_image_array

# run the inference
prediction = model.predict(data)

# Label
Label = ["Red","Green","Blue","Orange","Yellow","Voilet","Indigo"]
```

In [41]:

```python
# Prediction Graph
import matplotlib.pyplot as plt


y,x = prediction[0],Label
plt.plot(x,y)
plt.xlabel('Color')
plt.ylabel('Prediction')
plt.title("Color-Recognition")
plt.show()
```



In [42]:

```python
ans = max(list(zip(prediction[0],Label)))
print("Given input file is of {} color with prediction percentage {:2f
```

```
Given input file is of Orange color with prediction percentage 98.792726%
```

Thank you

All done !!

Predicted output will be shown to you

Email : kondekarshubham123@gmail.com (mailto:kondekarshubham123@gmail.com)

© Shubham Kondekar