

Importing the libraries openCV and Matplotlib for reading and plotting images

```
In [2]: import tensorflow as tf
```

```
In [4]: ! pip install opencv-python
```

```
Collecting opencv-python
  Downloading opencv_python-4.3.0.36-cp36-cp36m-win_amd64.whl (33.4 MB)
Requirement already satisfied: numpy>=1.11.3 in c:\users\jaya\anaconda3\lib\site-packages (from opencv-python) (1.19.1)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.3.0.36
```

```
In [5]: from keras.datasets import mnist
import numpy as np
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
```

Downloading the MNIST data

```
In [6]: (x_train, _), (x_test, _) = mnist.load_data()
print("The shape of x_train dataset is", x_train.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [=====] - 2s 0us/step
The shape of x_train dataset is (60000, 28, 28)
```

Reading greyscale image

Loading first sample from MNIST dataset. Resizing the image to 18x18.

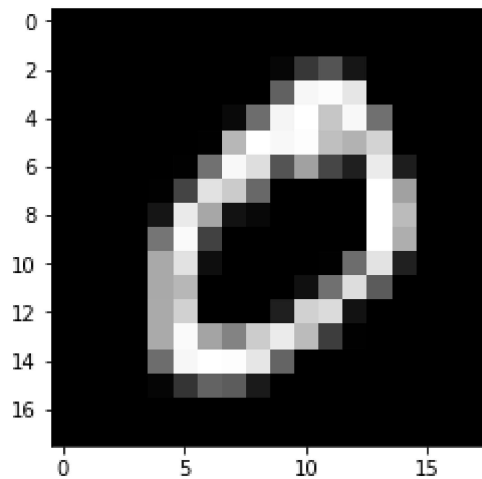
```
In [7]: # selecting the first sample
x = x_train[1]
print("The dimension of x is 2D matrix as ", x.shape)
# Resizing the image
x = cv2.resize(x, (18,18))
```

```
The dimension of x is 2D matrix as (28, 28)
```

Plotting the image using Matplotlib

```
In [8]: plt.imshow(x, cmap='gray')
```

```
Out[8]: <matplotlib.image.AxesImage at 0x26effd258d0>
```



You can see that height and width of the matrix is 18x18, same as height and width of above image. So, each pixel is represented by number.

```
In [9]: print("The range of pixel varies between 0 to 255")
print("The pixel having black is more close to 0 and pixel which is white is more close to 255")
print(x)
```

The range of pixel varies between 0 to 255

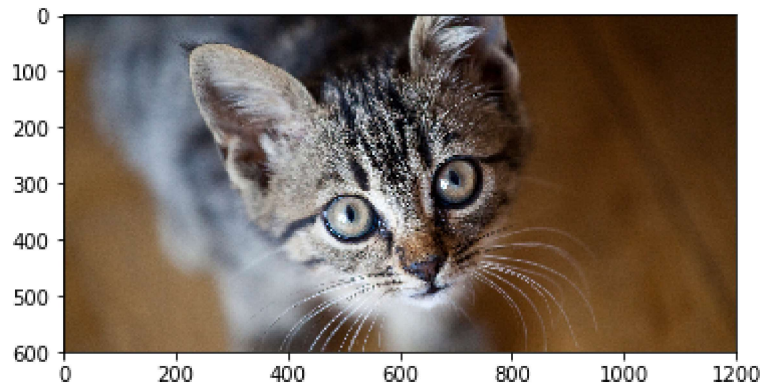
The pixel having black is more close to 0 and pixel which is white is more close to 255

```
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  6 55 84 22  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  9 108 243 252 196 247 110  0  0]
 [ 0  0  0  0  0  0  2 181 252 247 251 189 178 210  0  0  0]
 [ 0  0  0  0  0  2 112 247 220 84 159 69 30 234 29  0  0]
 [ 0  0  0  0  1 68 223 201 103  0  0  0  0 252 160  0  0]
 [ 0  0  0  0 21 232 166 17  7  0  0  0  0 252 184  0  0]
 [ 0  0  0  0 116 248 65  0  0  0  0  0  0 253 172  0  0]
 [ 0  0  0  0 167 223 15  0  0  0  0  2 107 225 33  0  0]
 [ 0  0  0  0 168 182  0  0  0  0 16 111 219 90  0  0  0]
 [ 0  0  0  0 169 208  0  0  0 30 207 217 18  0  0  0  0]
 [ 0  0  0  0 169 248 162 130 202 234 184 62  2  0  0  0  0]
 [ 0  0  0  0 108 245 253 251 229 99  0  0  0  0  0  0  0]
 [ 0  0  0  0  5 52 98 91 26  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]]
```

Reading colour image

```
In [13]: # Reading color image
cat = cv2.imread('cat.jpg')
plt.imshow(cv2.cvtColor(cat, cv2.COLOR_BGR2RGB))
```

Out[13]: <matplotlib.image.AxesImage at 0x26effde8588>



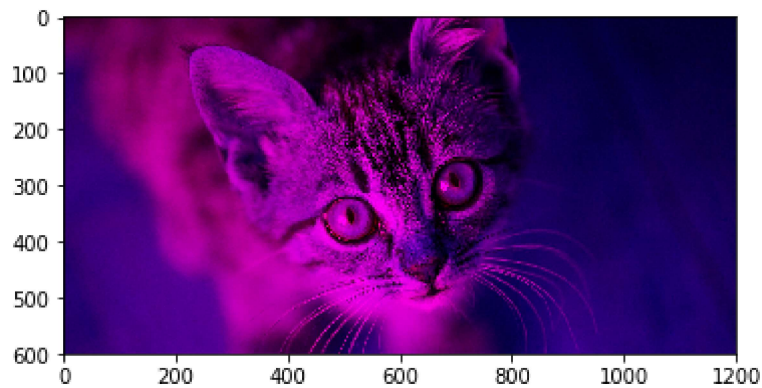
```
In [14]: print('The shape of image is ', cat.shape)

The shape of image is (600, 1200, 3)
```

Plotting the RGB channels of the image.

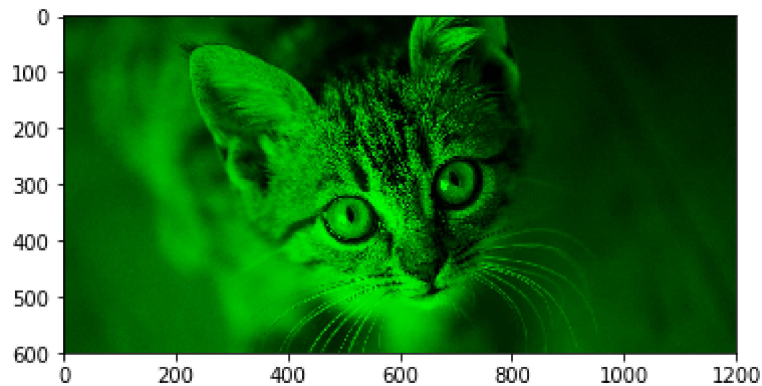
```
In [15]: cat_r = cv2.imread('cat.jpg')
cat_r[:, :, 1:2] = 0
plt.imshow(cat_r)
```

Out[15]: <matplotlib.image.AxesImage at 0x26e82d2e748>



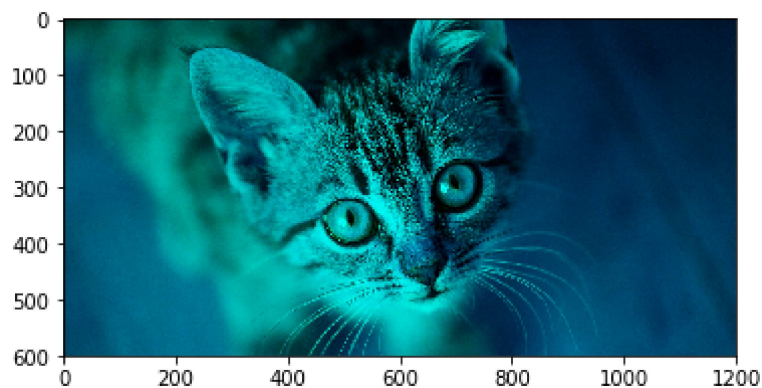
```
In [16]: cat_g = cv2.imread('cat.jpg')
cat_g[:, :, (0,2)] = 0
plt.imshow(cat_g)
```

Out[16]: <matplotlib.image.AxesImage at 0x26e82d81ba8>



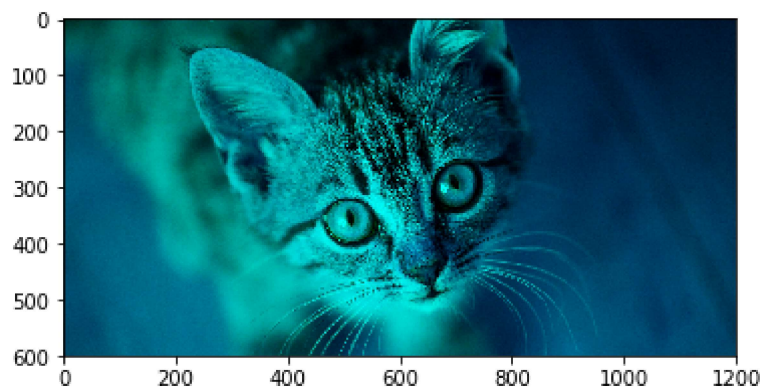
```
In [22]: cat_b = cv2.imread('cat.jpg')
cat_b[:, :, 0:1] = 0
plt.imshow(cat_b)
```

Out[22]: <matplotlib.image.AxesImage at 0x26e846d9828>



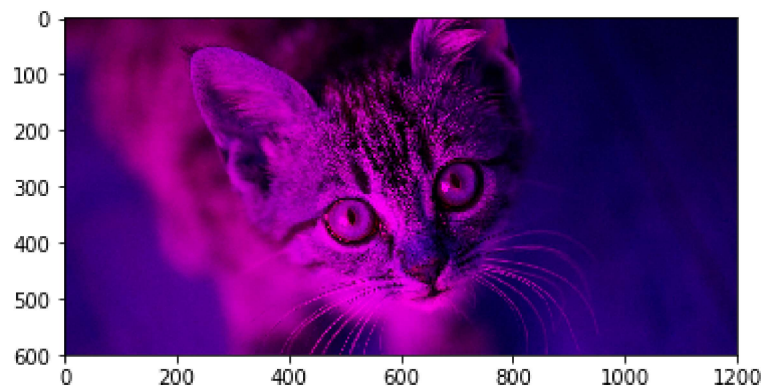
```
In [25]: cat_bg = cv2.imread('cat.jpg')
cat_bg[:, :, 0] = 0
plt.imshow(cat_bg)
```

Out[25]: <matplotlib.image.AxesImage at 0x26e847dde10>



```
In [21]: cat_rg = cv2.imread('cat.jpg')
cat_rg[:, :, 1] = 0
plt.imshow(cat_rg)
```

Out[21]: <matplotlib.image.AxesImage at 0x26e846825c0>



```
In [24]: cat_rb = cv2.imread('cat.jpg')
cat_rb[:, :, 2] = 0
plt.imshow(cat_rb)
```

Out[24]: <matplotlib.image.AxesImage at 0x26e84785c18>

