Image Processing

Praveen Kumar R

September 7, 2019

1 Computer Vision

Computer Vision is the branch of Computer science that lays protocol for computer and other machines to how to see the ouside world. Object detection, Rendering Segmented mask, Pattern recognition, Image transfromation and preprocessring,etc., are some of the major goals of this branch of study. Recently CV has evolved to visualise more mathematically complicated concepts like fuzzy logic, laplace transformation, Abstract Algebra, etc., thus giving these concepts a better scope of understanding.

2 What is an Image?

An image in its fundementel level is conceptualised as a collection of meaningful pixels. On the basis of how an image is constructed from it atomic stage, images can be classified into many categories

- RGB images

- HSV images

- CMYK images

- HWB images

- HEX images

etc,. Each of these categories differ in syntax of how the color of each pixel is represented. A same image represented in these different formats will differ in quality and color. This is because some formats dosent give the liberty to choose many colors. RGB format is generally adapted for any kind of image processing technique.

3 RGB Image

Every pixel in a RGB image is defined by 3 values Red, Green, and Blue. Each of these values uses 8 bit standard(IEEE) representation. 7 bits to represent value and the most significant bit to represent sign. Thus each of the values for R,G and B will run from 0 t0 255. As the value of ceratain channel reaches 0 the brightness of the color imparted by

that channel tend to become low, meaning the channel will become deeper in color and eventually become black. Similarly as the value of a certain channel approches to 255 the color imparted by that channel becomes more brighter. From the above statements one may arrive at the following hypothesis,

- (0,0,0) : will be black color

- (255,255,255): will be white color

- (255,0,0) : will be bright red color

- (0,255,0) : will be bright green color

- (0,0,255) : will be bright blue colors

the representation is based on the assumption that pixel representation is in RGB format. By fundemental principle of counting the number of different colors that can be formed is 256 x 256 x 256 = 16,777,216
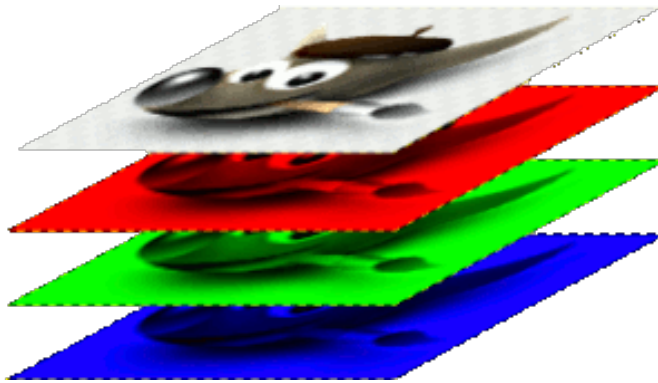

Figure 1: RGB image - The top layer shown is the actual image you see. But it actually represented using 3 different channels

Mathematically a 3 channel RGB image is modeled as a 4 dimensinol matrix. The inner most dimension will provide information about the pixel value. The next dimension will be an vector of these pixels forming rows of the image channel matrix. These rows combine in next dimension vertically to form a channel. Three such channels will combine to form a RGB image. In case of Grayscale image there will be only one channel, and each pixel is defined by one single value ranging from 0 to 255(extremes inclusive).

4   Installation

OpenCV is a open source community the keep constantly updating its libraries based on input from programmers all around the world. OpenCV is available in two different distribution Python and C++. Though the functionality of both the distribution are maintained to be the same, the syntax and the parameters differ in each of these language distributions. In this guide you will be able to install the python distribution of opencv. The commands in this installation steps were tried and tested in Ubuntu 18.04.

5 Pre-requisite

It is recommended that the programmer uses a linux or unix based operating system like Ubuntu, Kali, or MacOS. Standard distribution for python comes pre installed with these opearting system. To check the version of python run the following command.

```
$ python --version
```

If python package is not found in your system install the package by running the following commands.

```
$ sudo add-apt-repository ppa:jonathonf/python-3.6
$ sudo apt-get update
$ sudo apt-get install python3.6
```

6 Installation Steps

To install addtional libraries for python you need either one of the installation tools

- pip
- pip3 (recommended for python3)
- conda

pip

To install pip use the following command.

```
sudo apt-get install python3-pip
```

pip3

You can upgrade your pip to pip3 by running the following command.

```
sudo -H pip3 install --upgrade pip
```

conda

To install conda download the conda bash .sh file from [[https://repo.anaconda.com/archive/Anaconda3-2019.03-Linux-x86_64.sh]]. Once the download is complete go to the directory where the extracted .sh file is present and run the following command.

```
bash Anaconda3-2019.03-Linux-x86_64.sh
```

NOTE: The bash command will vary with name of the .sh script installed. So remember to give the correct file name in the command.

I would recommend to install all these tools because some of the libraries are easily installed using each of these tools.

Dependency packages

The program requires the following packages for execution

```
opencv
numpy
pandas
theano
tensorflow
keras
pillow
```

Before installing any of these libraries I would recommend you to create a new environment in your your conda to avoid libraries version clashes. To create a new environment run the following command in your terminal.

```
conda create --name myenv
```

By running the above command you will create a new environment called 'myenv'. In most of the terminal the command prompt message is preceeded with the environment where you are currently in. The default conda environment is called 'base' to move to another environment run the following command.

```
conda activate myenv
```

or

```
source activate myenv
```

Once you land your newly created environment you are ready to install the packages. Run the following commands to install the packages mentioned above. Using conda is useful because conda will take care of the libraries verions to be installed.

```
$ conda install -c anaconda numpy
$ conda install -c anaconda pandas
$ conda install -c conda-forge opencv
$ conda install -c conda-forge theano
$ conda install -c conda-forge tensorflow
$ conda install -c conda-forge keras
$ conda install -c anaconda pillow
```

   Note: If you are willing to install the tensorflow and keras for GPU, do it only if you have good Graphical processing unit in you machine, because it is tedious process. For GPU installation follow the instruction in the [[https://www.tensorflow.org/install/gpu]] **GOOD LUCK**

7 Introduction to OpenCV python3

Importing libraries

To start with import the necessary libraries for the for the programmer

```
import cv2
import numpy
```

if any of the above import fails it means that you made mistake in you installation process. Then go back to the previous section and reinstall all the libraries after removing them.

Reading images

To read a image and store it in a variable use the imread function.

```
img = cv2.imread('sampleImage.jpg')
```

the above line will read a image named 'sampleImage.jpg' from the present working directory. The image read is by default in BGR format , to read the image in gray scale replace the above line with

```
img = cv2.imread('sampleImage.jpg',0)
```

The image read is a numpy array. Thus you can create you own image by creating an numpy array with a element with 8 bit representation

```
img = np.ones((300,400,3),dtype = np.uint8)*255
```

the above line will create a 3 channel image with height 300 and 400 pixels fill all pixel values to (255,255,255), meaning the image will be a white image.

Displaying images

To display an image cv2 provides a function called 'imshow'

```
cv2.imshow('window_name',img)
```

If you run your code now your program might probably collapse and the window will stop responding. This is because once the imshow fuction is called the program will create a thread (child thread) to display the image but as soon as the child thread is created the parent process terminates leaving the child thread in orphan state. Thus the window will not repond to any clicks. So to avoid it always use the imshow function with a function called 'waitKey()' to make the parent process to wait till the child terminates and close all the windows before you exit the program.

```
cv2.imshow('window_name',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

NOTE: 'cv2.waitKey(0)' will make the program to halt and wait until a keyboard stroke is recieved. In case of video where the frame are continuously read, waiting for key board stroke will not be a efficient job to do, so we we give waitKey(1) where the program don't halt but continue executing and scans for keyboard stroke in interval of 1 sec.

So a Simple program will look like

```python
import cv2
import numpy as np

img = cv2.imread('sampleImage.jpg')

cv2.imshow('Image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

So to display a live video use this program

```python
import cv2
import numpy as np

cap = cv2.VideoCapture(0)
while True:
    ret,frame = cap.read()
    cv2.imshow('video',frame)
    if(cv2.waitKey(1)==13):
        break
cap.release()
cv2.destroyAllWindows()
```
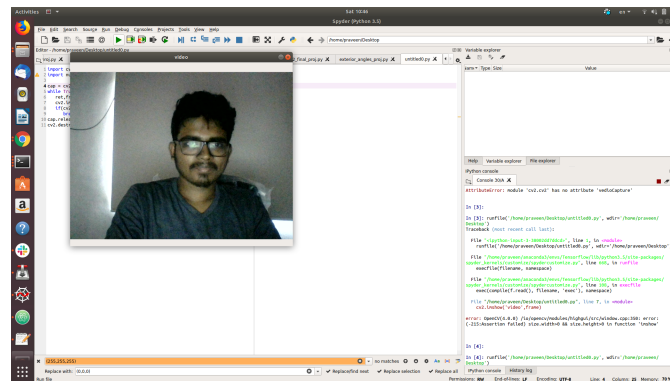

Figure 2: Output