

# OpenCV using Python



Introduction to Image processing and computer vision

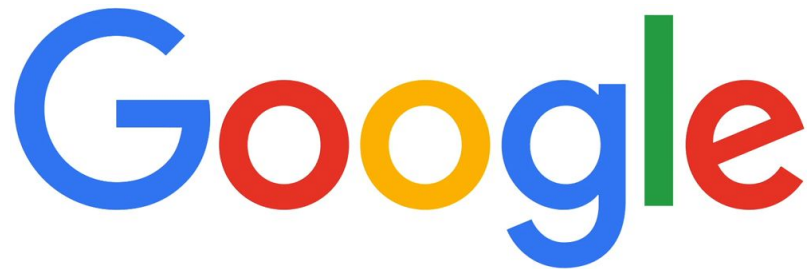
<https://github.com/human-divanshu/talks/tree/master/opencv>

# What is OpenCV ?

- Image processing library aimed at real time computer vision
- Available for
  - Python
  - C
  - C++
  - Java
  - and many others
- For more details
  - Read <https://en.wikipedia.org/wiki/OpenCV>
  - Official website <http://opencv.org>

# Who uses openCV ?

- Most famous image processing library
- About 73% open source image processing projects use it



# Where it is used ?

- Medical imaging
  - Ultrasound, MRI, CT Scan
- Astronomical imaging
- Security
  - Face recognition
- Image search
- Social apps
  - Instagram, Facebook
- Robotics

Anywhere you want to use it.

# Installation on Linux

```
sudo apt-get install python-numpy
```

```
sudo apt-get install libopencv-*
```

```
sudo apt-get install python-opencv
```

# Installation on Windows - 1

For OpenCV

- Download from official website

OR

- Download from here  
<http://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv>
- And install using pip

# Installation on Windows - 2

- `pip install numpy`
- `pip install matplotlib`

Run above commands in cmd by changing to bin folder of your Python installation.

# Installation on MacOS

- Download from official website
- <http://opencv.org/downloads.html>
- And install numpy and matplotlib using pip just as in case of windows



# **Image basics**

# What is image ?

- Just a collection of number where each number represents some color

# What is color ?

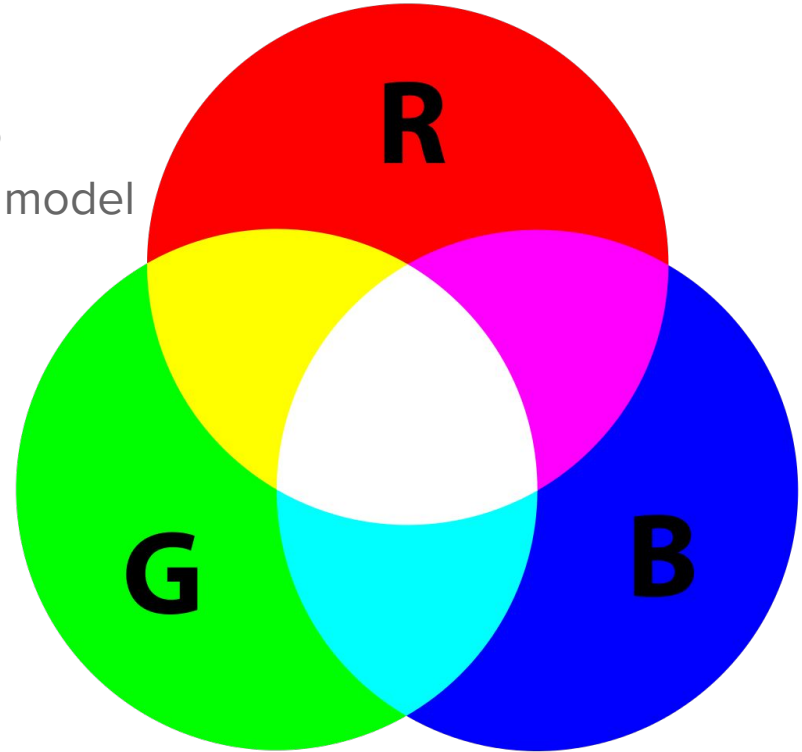
Colors is represented via some color family / model :

- RGB
- BGR
- HSV
- Grayscale

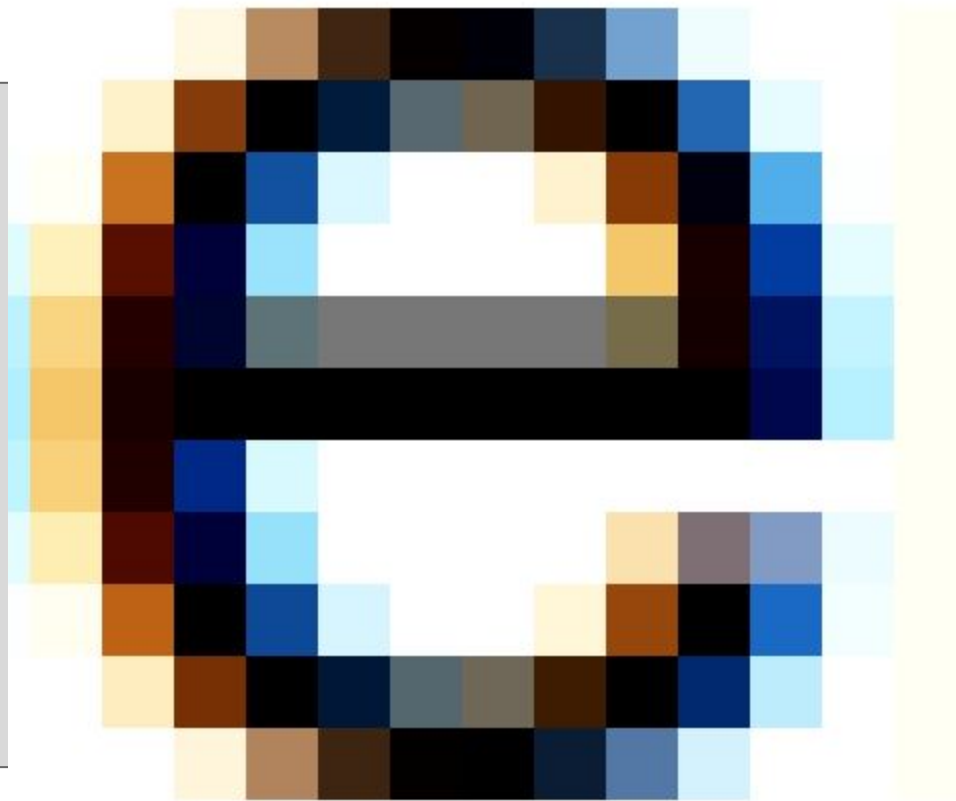
and many more are there

# RGB - Red, Green and Blue

- Color is value of RGB taken together
- Each color has value between 0 to 255
- Total colors are about 1.6 million in this model



# RGB Demo



# Some common colors in RGB

- Black
  - 0, 0, 0
- White
  - 255, 255, 255
- Red
  - 255, 0, 0
- Green
  - 0, 255, 0
- Blue
  - 0, 0, 255

# Grayscale Image ( $R = G = B$ )

- White is 255, 255, 255
- Any value  $X, X, X$  is a gray shade
- Black is 0, 0, 0
- Pixel is represented by single number because  $R = G = B$

So how many gray shades should be there in total ?

# Some common terminologies

- 3 channel image
  - RBG, BGR, HSV images are called 3 channel images
  - Because you need 3 numbers to represent each pixel
- 1 channel image
  - Gray scale image
  - Because you need 1 number to represent each image
- Binary image
  - Image that just has two colors
  - Generally black and white
  - Can be represented using bits 0 and 1 alone



# How are numbers converted to image ?



**Read/Write  
Image**

# Read/Write Image (demo1.py)

```
import cv2
```

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

```
cv2.imwrite('newpic.png', image)
```

# Documentation

<http://docs.opencv.org/>

**Convert image  
to  
Gray scale**

# Convert image to Grayscale (demo2.py)

```
import cv2  
  
image = cv2.imread('img.jpg', cv2.CV_LOAD_IMAGE_GRAYSCALE)  
  
cv2.imwrite('grayimg.jpg', image)
```

Default is CV\_LOAD\_IMAGE\_COLOR

**Showing image on screen (demo3.py)**

**Showing multiple windows and `cvtColor`  
method (demo4.py)**



# **Resizing image (demo5.py)**

**How are movies made ?**

**Simple Green Screen Example  
(demo6.py)**

**How are movies made ?**

**Improved Green Screen Example  
(demo7.py)**

# Thresholding

**Means allows everything after**

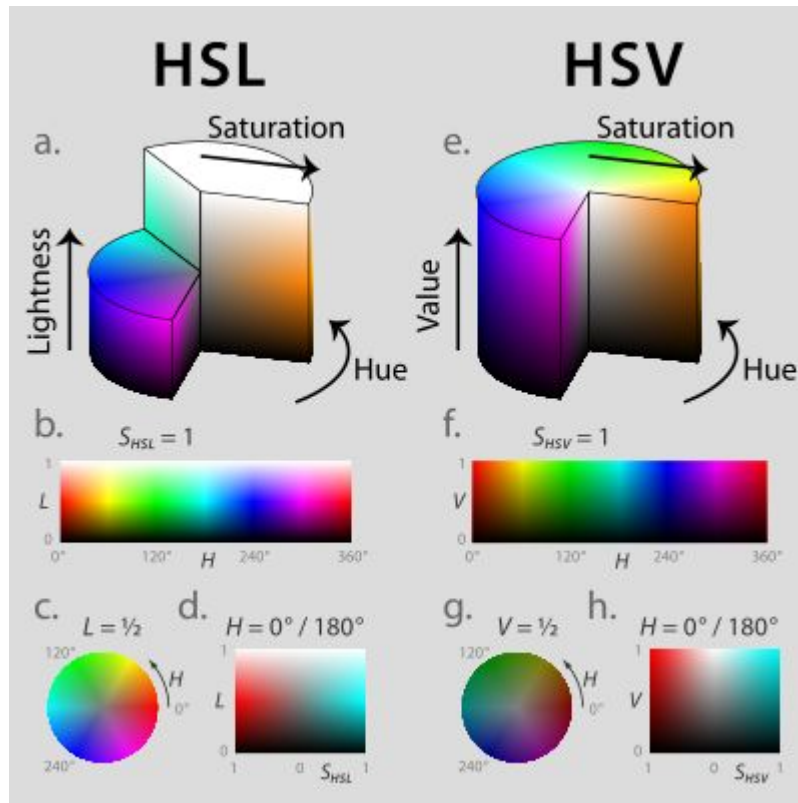
# **Simple threshold (demo8.py)**

# **Grayscale threshold (demo9.py)**

# **Adaptive threshold (demo10.py)**

# Understanding HSV (Cylindrical model)

- Hue - Which color ?
- Saturation - How pure ?
  - Decreasing saturation will fade off the color
- Value - How bright ?
  - How much light falls on the color





# **Color filtering (demo11.py)**

# How to generate filters ?

- Hit and trial
- Statistical techniques

# **Image smoothing (demo12.py)**

**Image smoothing used anywhere ?**

**Hint : Ever watched a 3D movie ?**

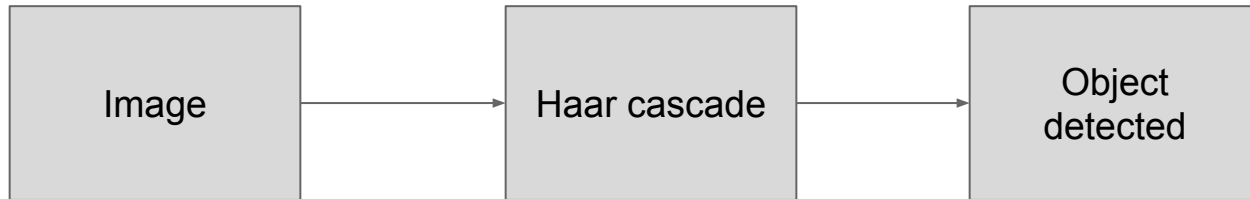
# Object detection

# How do we detect thing ?

- By feature extraction
  - Size
  - Shape
  - Edge
  - Color
  - And many more

# Haar cascade

- Is basically a feature list for any object



# How we make it ?

- By providing few hundred thousand positive and negative images to haar algorithm
- Normally generating good haar cascade will take about weeks or even months of continuous processing
- We will use haar generated by INTEL
- <https://github.com/opencv/opencv/tree/master/data/haarcascades>



# **Face detection (demo13.py)**

# **Eyes detection (demo14.py)**

# **Video Processing**

**Because images are not cool**

**Showing simple video feed  
(demo15.py)**

# **Processing video feed (demo16.py)**

# **Motion detection (demo17.py)**

# Doubts ?

Learn by doing is the best way to learn.

Pick up a small fun project.