



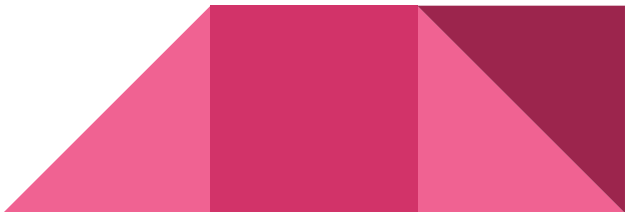
# Image Processing - The Other Way

# Some general questions

To get to know the demography here.

1. Branch?
2. Year?
3. Background in C++, Java, Python?
4. Linear Algebra?
5. Webcam?

## Prerequisites

1. Ubuntu
  2. Wifi Drivers
  3. IITG\_WIFI
  4. Proxy Settings
  5. OpenCV
- 

# What do I mean by other way?

1. Well, you install some software first
2. Then you write some code, and run it.
3. You learn what the code did, at code level, then at mathematical level.
4. Rinse and Repeat.

Why is this way better? Because there is lesser chance that you get bored and drop learning Image Processing.



# MATLAB vs OpenCV

## Matlab is

1. Easier to and use
2. Easier to learn
3. Slower than openCV
4. Difficult(maybe even impossible) to integrate with various hardware and software - like android, python, PHP etc.

## OpenCV is

1. Comparatively, More difficult to use
2. Comparatively, More difficult to learn
3. When it comes to speed - If Matlab was a cheetah, opencv is a rocket.
4. It has support for C++, Python, Java, Android, IOS.

# Why am I still going with openCV, even though MATLAB is easier

We are not developing products. So why do it the hard way.

Well, the answer is there in the question. Doing it the hard way, you learn something that's worth professionally.

I got the internships that I did, because I knew openCV, not because I know Image Processing.



# IITG Wifi Settings in Linux



# Proxy Settings in Linux

There are three places where you need to set the proxy.

1. Network Settings - You all must have done this
2. `/etc/apt/apt.conf` file - Some of you must have done this too
3. `~/.bashrc` - Very few would have done this

Link for setting up and learning about proxy in linux :

<https://raghuramkrishnaswami.wordpress.com/2016/01/18/setting-up-proxyies-in-ubuntu/>



# Installing Sublime Text and Terminator

1. Follow this to install Sublime Text 3 :  
<http://askubuntu.com/questions/172698/how-do-i-install-sublime-text-2-3>
2. Follow this to install Terminator : <http://technicalworldforyou.blogspot.in/2012/11/install-terminator-terminal-emulator-in.html>





# Installing OpenCV

The instructions to install opencv can be found here :

<http://rodrigoberriel.com/2014/10/installing-opencv-3-0-0-on-ubuntu-14-04/>

Please follow them exactly.



# While it's installing,

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

<http://stackoverflow.com/>

These are the places from where I learn opencv. I am not kidding.

The best way to learn any programming language or library is to just start doing it.



# How to check if opencv is installed?

```
pkg-config --cflags --libs opencv
```

Run the above command in you terminal. If opencv was not installed properly, ubuntu will say something like “no such package found”

Else, you will get a list of files.



# Tutorial 1 : Load and Display an Image

Link : [http://docs.opencv.org/3.0-alpha/doc/tutorials/introduction/display\\_image/display\\_image.html#display-image](http://docs.opencv.org/3.0-alpha/doc/tutorials/introduction/display_image/display_image.html#display-image)

## OpenCV functions and classes involved

1. Mat
2. namedWindow
3. imread
4. imshow
5. waitKey

## Image Processing Concepts

1. Representation of Digital Image in Computers



# Tutorial 2 : Load, Modify and Save an Image

Link : [http://docs.opencv.org/2.4/doc/tutorials/introduction/load\\_save\\_image/load\\_save\\_image.html#load-save-image](http://docs.opencv.org/2.4/doc/tutorials/introduction/load_save_image/load_save_image.html#load-save-image)

## OpenCV functions involved

1. `imwrite`
2. `cvtColor`

## Image Processing Concepts

1. Colors and Colorspaces



# Tutorial 3 : Streaming a Video

File : videoStreaming.cpp

OpenCV class involved

1. VideoCapture



# Exercise 1 : Channels of a Video Stream

Instead of streaming the video directly from the camera, we are going to do a little processing now.

From the stream of BGR images, get the following streams,  
RGB, HSV, R, G, B, H, S, V

To get the components or “channels” of RGB or HSV image, use split :

[http://docs.opencv.org/2.4/modules/core/doc/operations\\_on\\_arrays.html#split](http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#split)

The output should be 9 video streams, including the original.



# Tutorial 4 : Thresholding

Link : <http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/threshold/threshold.html#basic-threshold>

## OpenCV functions involved

1. threshold

## Image Processing Concepts

1. Color, Grayscale and Binary Images
  - a. Color : Usually 8-bit 3-channel images
  - b. Grayscale : Usually 8-bit 1-channel images
  - c. Binary : Images in which there are only 2 colors : white and black
2. Thresholding





## Exercise 2 : Rudimentary Blob Detection

Take the result of Exercise 1. Using thresholding and logical operations like AND, OR, NOT, create a stream that only shows your face as white, and the background as black.

Note : The result of this exercise doesn't have to be perfect. As you can see, this is just a rudimentary attempt.

The documentation for applying logical operations on "Mat" can be found here :  
[http://docs.opencv.org/3.0-alpha/modules/core/doc/operations\\_on\\_arrays.html](http://docs.opencv.org/3.0-alpha/modules/core/doc/operations_on_arrays.html)



# Tutorial 5 : Basic Drawing

Link : [http://docs.opencv.org/3.0-alpha/doc/tutorials/core/basic\\_geometric\\_drawing/basic\\_geometric\\_drawing.html](http://docs.opencv.org/3.0-alpha/doc/tutorials/core/basic_geometric_drawing/basic_geometric_drawing.html)

OpenCV functions and classes involved

1. Point
2. Scalar
3. line
4. ellipse
5. rectangle
6. circle



## Exercise 3 : Draw a stickman



We know how to draw circles and lines. Create a completely white image. Draw black lines and circles on it to form a stick figure.

For creating a completely white image, you need to know how to “Construct” a Mat object. So far we have used `imread` to create Mat objects. But we can create mat object manually as well.

For more info, look at the documentation of Mat object :

[http://docs.opencv.org/3.0-alpha/modules/core/doc/basic\\_structures.html#mat-mat](http://docs.opencv.org/3.0-alpha/modules/core/doc/basic_structures.html#mat-mat)



# Tutorial 6 : Affine Transformations

Link : [http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/imgtrans/warp\\_affine/warp\\_affine.html](http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/imgtrans/warp_affine/warp_affine.html)

## OpenCV functions involved

1. `getRotationMatrix2D`
2. `getAffineTransform`
3. `warpAffine`

## Image Processing Concepts

1. Geometric Transformations

## To Explore

1. `findHomography`



# Tutorial 7 : Contours

Link : [http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/shapedescriptors/find\\_contours/find\\_contours.html](http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html)

## OpenCV functions involved

1. findContours
2. drawContours

## To Explore

1. arcLength
2. contourArea
3. moments



# Tutorial 8 : Smoothing Images

## Links :

[http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/gaussian\\_median\\_blur\\_bilateral\\_filter/gaussian\\_median\\_blur\\_bilateral\\_filter.html](http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html)

## OpenCV functions involved

1. blur
2. gaussianBlur
3. medianBlur
4. bilateralFilter

## Image Processing Concepts

1. Linear and Nonlinear Filters



# Tutorial 9 : Morphological Operations

## Links :

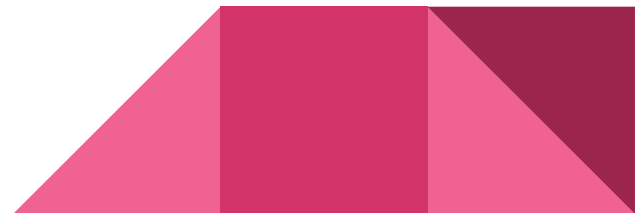
1. [http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/erosion\\_dilatation/erosion\\_dilatation.html](http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html)
2. [http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/opening\\_closing\\_hats/opening\\_closing\\_hats.html](http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html)

## OpenCV functions involved

1. erode
2. dilate
3. morphologyEx

## Image Processing Concepts

1. Morphological Operations on Images



# Tutorial 9 : User Defined Linear Filters

Links : [http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/imgtrans/filter\\_2d/filter\\_2d.html](http://docs.opencv.org/3.0-alpha/doc/tutorials/imgproc/imgtrans/filter_2d/filter_2d.html)

OpenCV functions involved

1. filter2D

Image Processing Concepts

1. Linear Filters
2. Convolution





# Do we have time left?

1. Sobel Filters
2. Laplace Operator
3. Canny Edge Detector
4. Hough Lines and Hough Circles
5. Histogram Calculation
6. Template Matching
7. Convex Hull
8. Trackbars in opencv



# Where now?

1. Mathematical approach to image processing : <https://www.youtube.com/watch?v=bxhJEe38bhY&list=PLZ9qNFMHZ-A79y1StvUUqgyL-00fZh2rs>
2. <http://www.docs.opencv.org>
3. stackoverflow
4. DIY Projects
5. Academic Projects





