BLEED-AI

Bringing Computer Vision to the masses

Computer Vision/Image Processing with Python and OpenCV

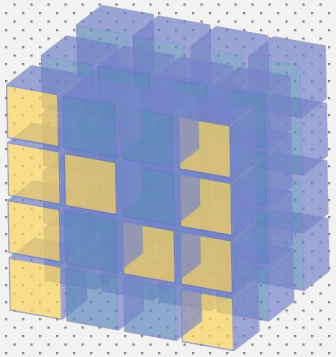
This is an Applied classical computer vision course which is highly practical and is mostly focused on building Vision powered Applications and grasping all the fundamentals. This course consists of 2 months. There are a total of 16 classes with 2 classes each week. The only prerequisite for this course is that you have a basic programming knowledge.

Day 1: Getting started with Python

1.1 Introduction to Computer Vision

1.2 Installation of Anaconda with required libs

1.3 One day Python Crash Course



NumPy

Day 2: Getting started with NumPy

2.1 One day NumPy Crash Course

2.2 Installation/resolving issues of Anaconda

Day 3: Computer Vision & Image Processing Basics

3.1 Loading & Displaying image from disk.

3.2 Resized Window

3.3 Writing/saving image

3.4 Show image using Matplotlib

3.5 Drawing shapes & write text on image

Day 3: Computer Vision & Image Processing Basics (Cont.)

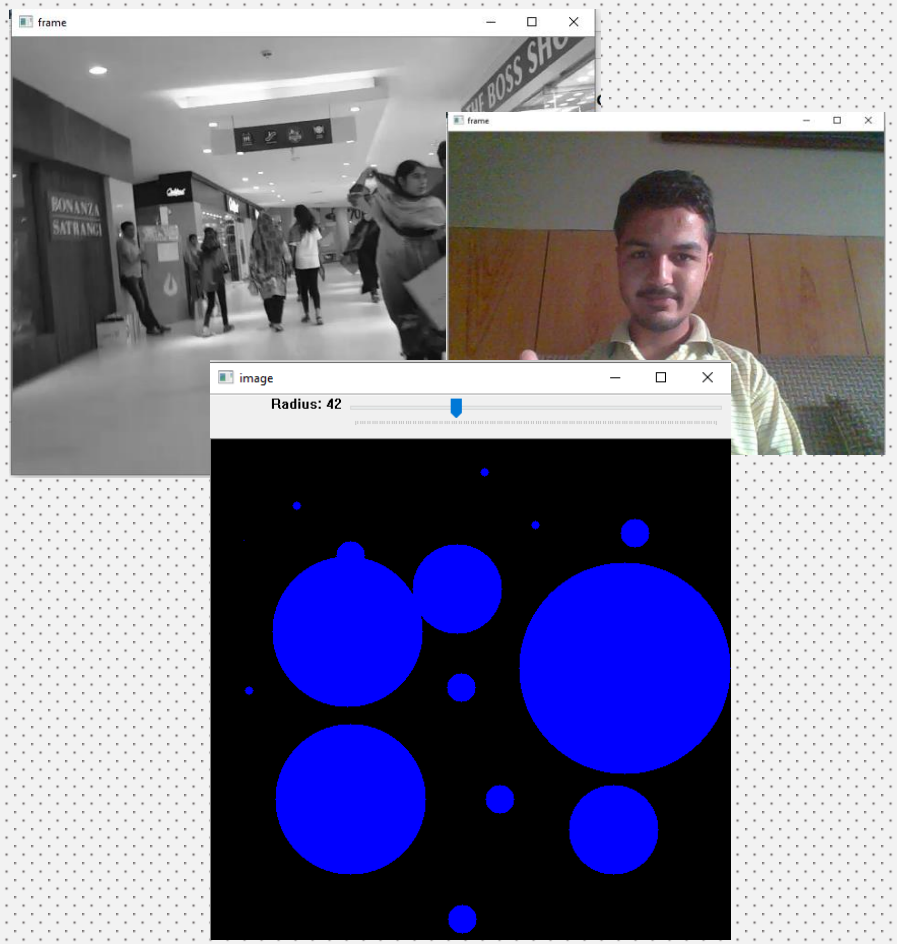
3.6 Playing Video from disk in grayscale and playing in Slow Motion/fast Forward.

3.7 Taking live streams.

3.8 Saving live stream & perform operations.

3.9 Getting/Setting video frame size.

3.10 Using Mouse Events & Trackbar.



Day 4: Basic Image Manipulations & Operations

4.1 Modify ROI & getting Image Properties

4.2 Image Channels

4.3 Manipulating Image with ROI

4.4 Code optimization check

4.5 Image Addition, Blending & Resizing

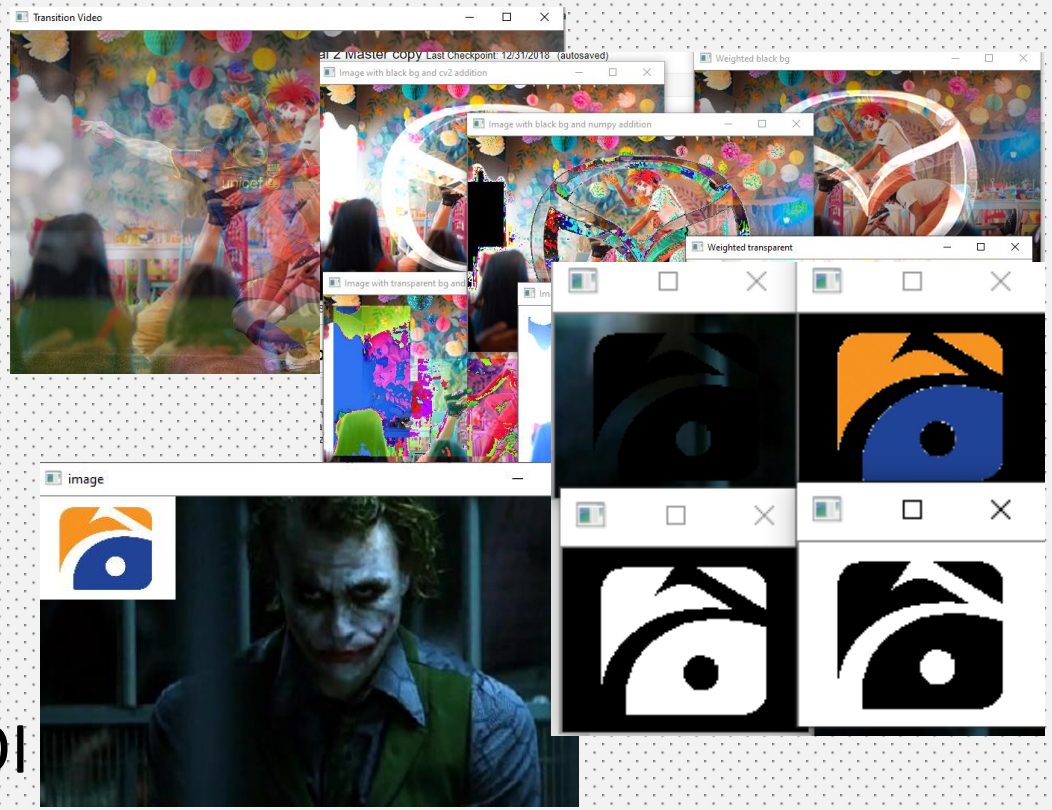
Day 4: Basic Image Manipulations & Operations(Cont.)

4.6 Image Transition Application

4.7 Changing ROI with other Image

4.8 Working with Transparent & Black background Image

4.9 Bitwise Operations, Thresholding & ROI manipulation



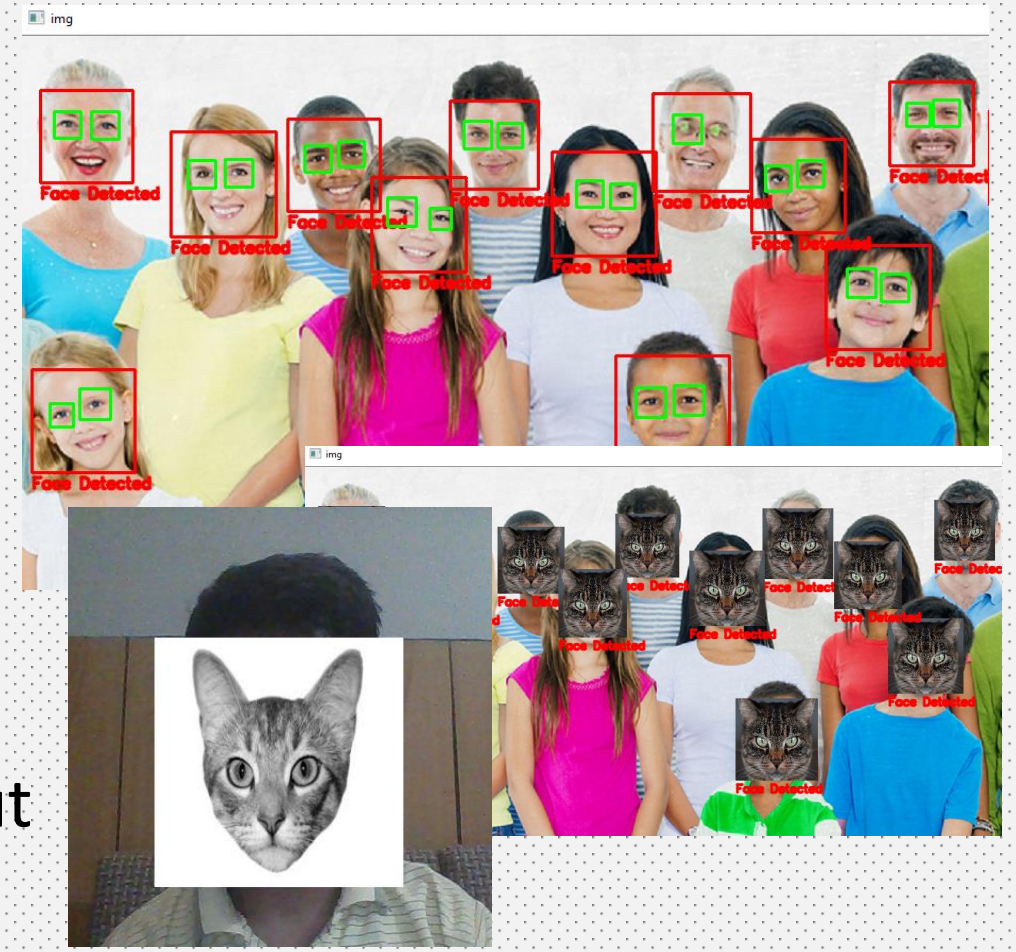
Day 5: Face/eye masking and Capturing objects with color.

5.1 Face & Eye Detection on Images

5.2 Face & Eye Detection on live video

5.3 Face replacement with Image (without Background Removal)

5.4 Face replacement in Live video (without Background Removal)



Day 5: Face/eye masking and Capturing objects with color. (Cont.)

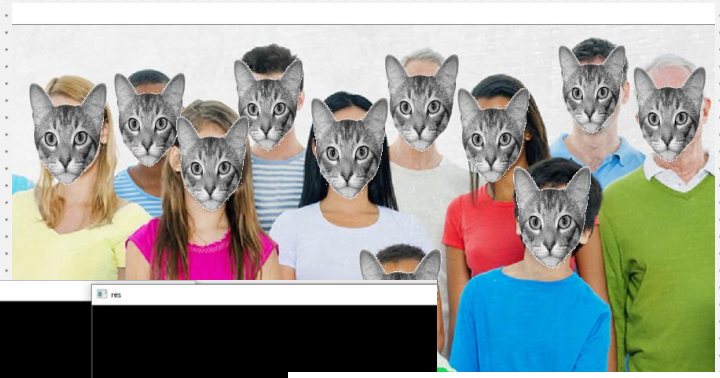
5.5 Face replacement in Image (with Background removal)

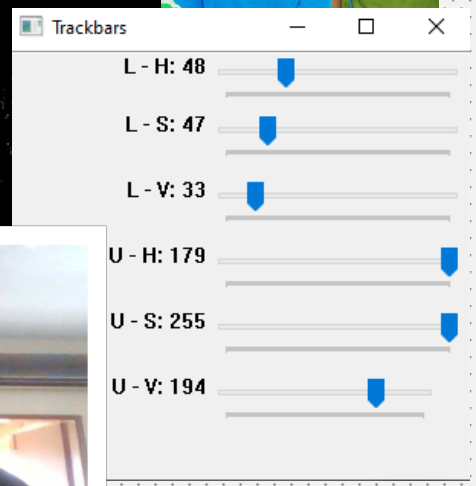
5.6 Face replacement in Live video (with Background removal)

5.7 Finding desired colored objects

5.8 Capturing Red color

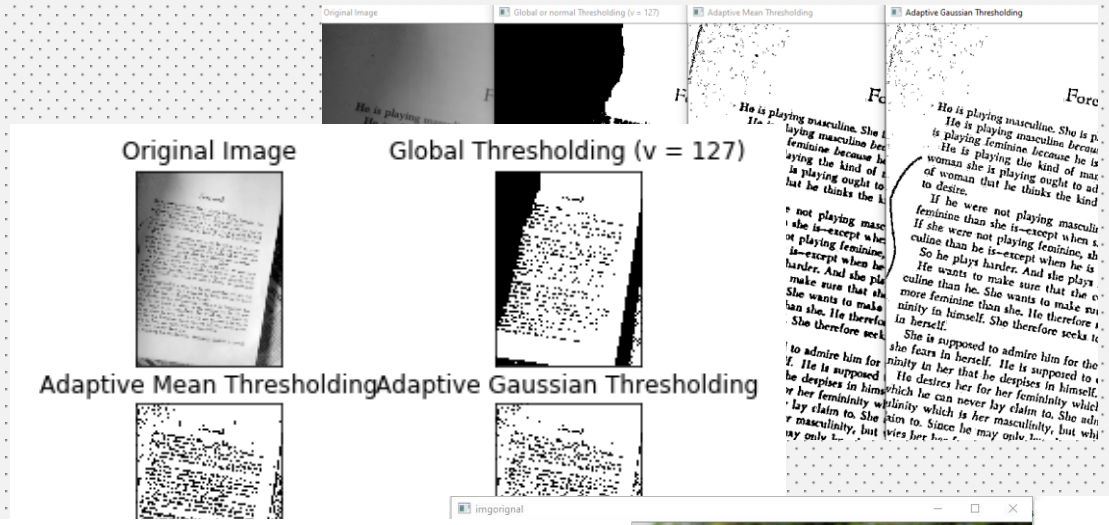
5.9 Color spaces





Day 5: Face/eye masking and Capturing objects with color. (Cont.)

- 5.10 Multiple types of Thresholding
- 5.11 Matplotlib subplots
- 5.12 Manual Blurring or 2D convolution
- 5.13 Average Blur
- 5.14 Gaussian Blur
- 5.15 Median Blur
- 5.16 Bilateral Blur



Day 6: Image Transformations, Morphological Operations, Edge detections, Image Pyramids

6.1 Translation

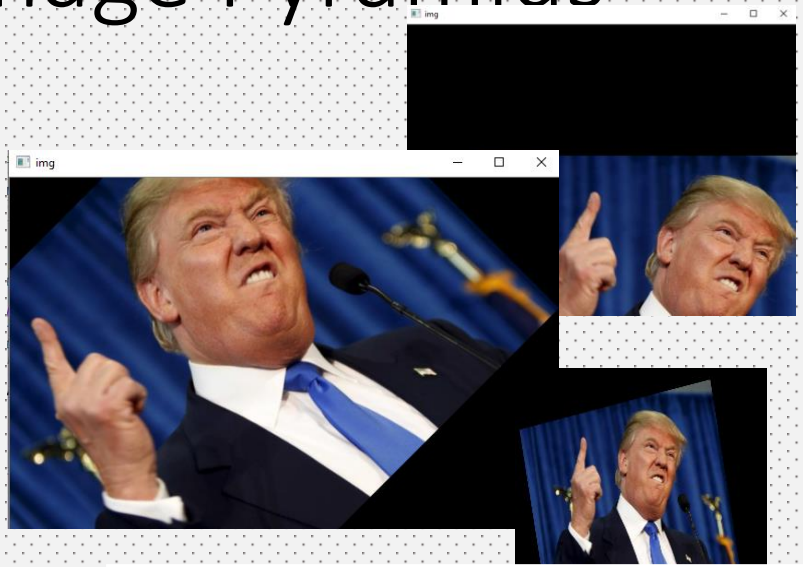
6.2 Rotation

6.3 Affine Transformation

6.4 Perspective Transformation

6.5 Points Finder

6.6 Perspective Transformation Application



Day 6 Image Transformations, Morphological Operations, Edge detection, Image Pyramids (Cont.)

6.7 Image Cropping Application

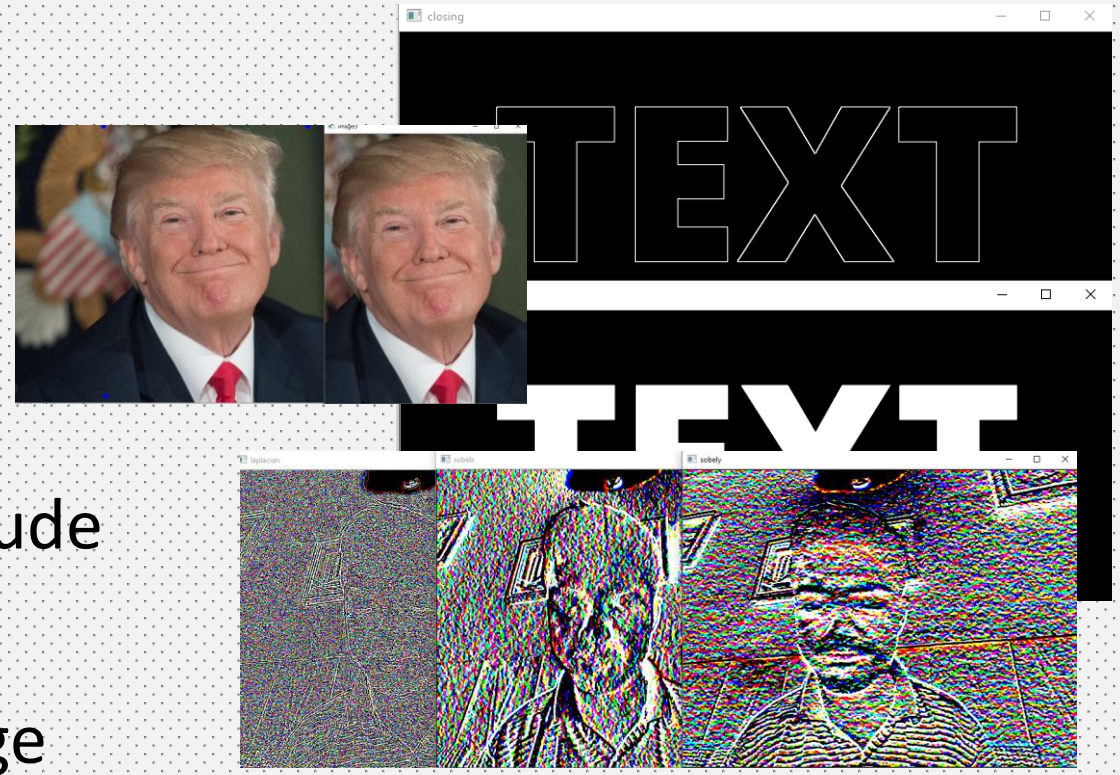
6.8 Erosion, Dilation

6.9 Opening, Closing

6.10 Morphological Gradient

6.11 Sobel & Laplacian Derivatives Or Crude Edge Detection

6.12 Canny edge Detection (A classic edge detection algorithm)



Day 6: Image Transformations, Morphological Operations, Edge detections, Image Pyramids (Cont.)

6.13 Image Pyramids Changing Image Resolution & Laplacian pyramid

6.14 Pyramid Image Merging Application (step by step)

6.15 Pyramid Image Merging Application (Full Code)

6.16 Pyramid Image Merging Application 2 Live Merging of Faces

Day 7: Histograms, Template Matching, Hough Transforms

7.1 Histogram Calculation in OpenCV

7.2 Histogram Equalization

7.3 CLAHE (Contrast Limited Adaptive Histogram Equalization)

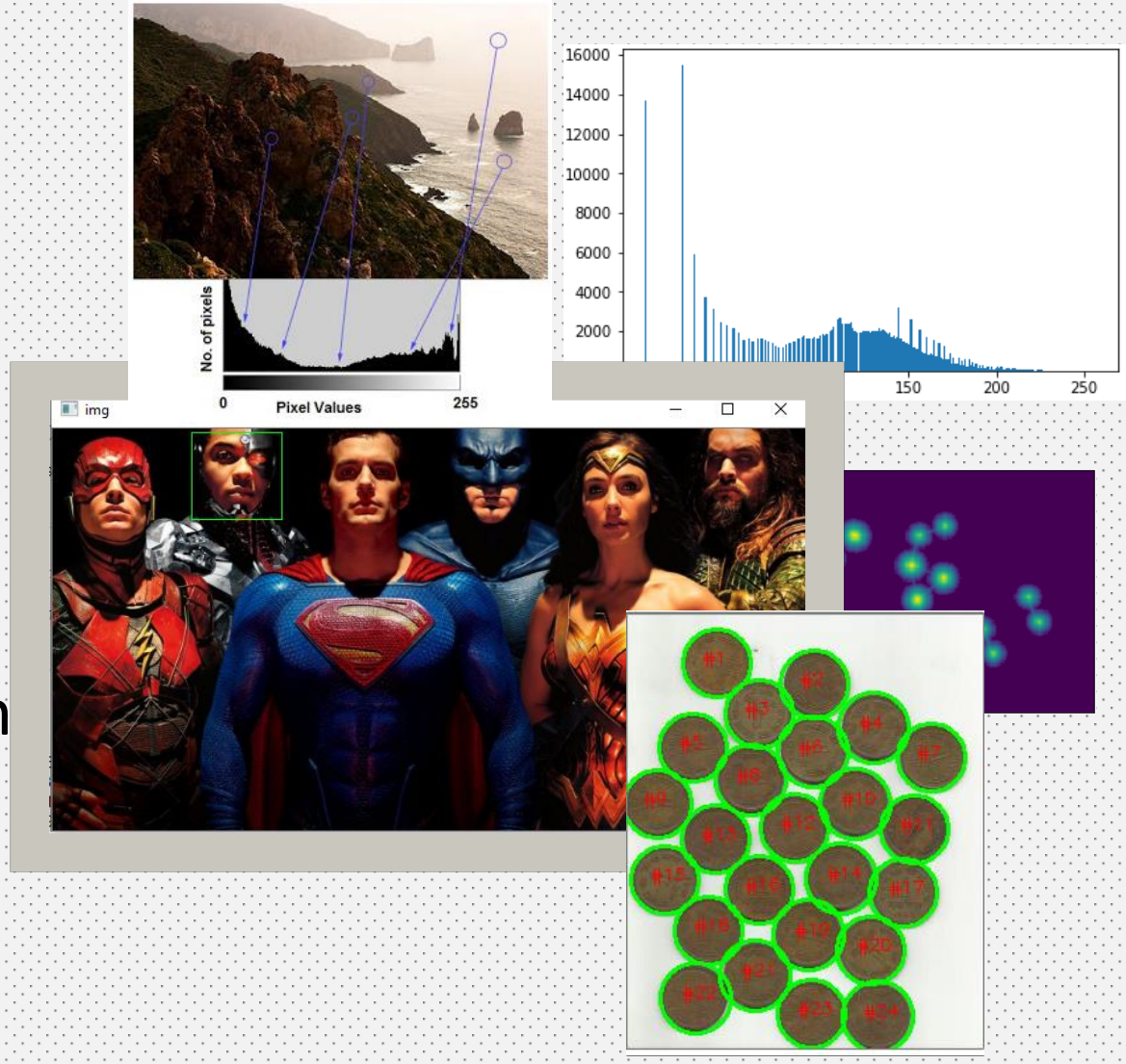
7.4 Template Matching

7.5 Probabilistic Hough Line Transform

7.6 HOUGH CIRCLE

7.7 Watershed Algorithm

7.8 Histogram Back projection



Day 8: 8.1 Feature Detection

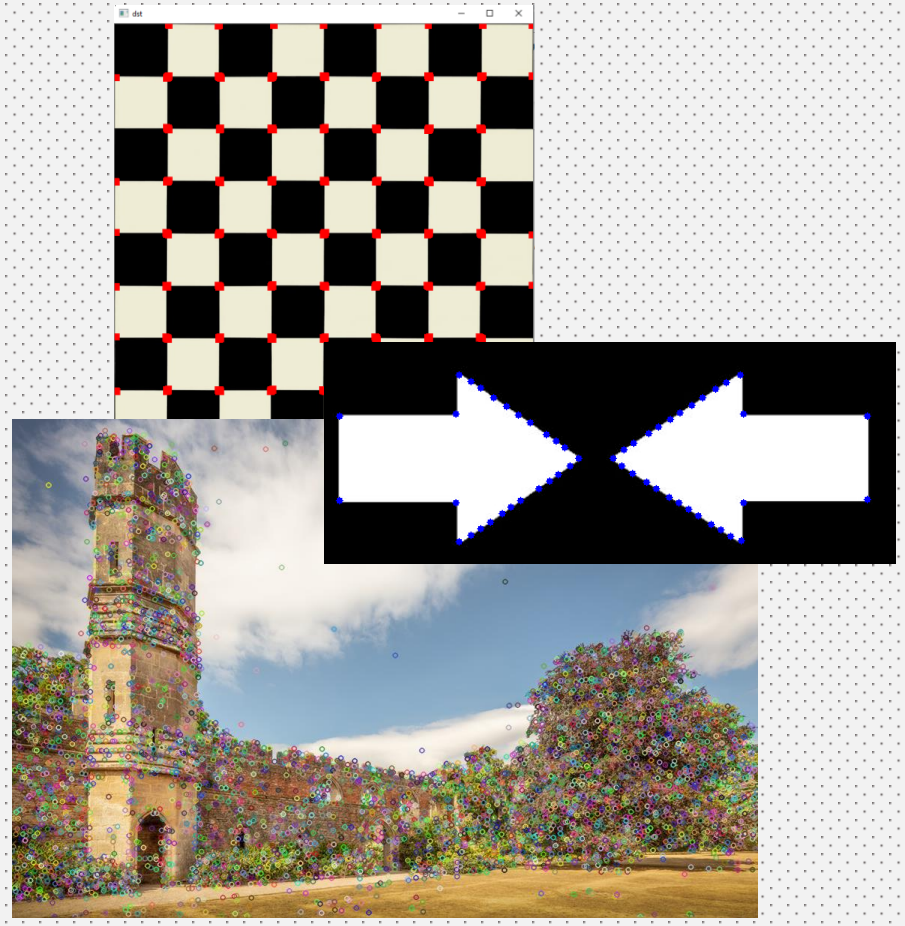
8.1.1 Harris Corner Detection

8.1.2 Shi-Tomasi Corner Detector & Good Features to Track

8.1.3 Introduction to SIFT (Scale-Invariant Feature Transform)

8.1.4 Introduction to SURF (Speeded-Up Robust Features)

8.1.5 FAST Algorithm for Feature Detection



Day 8: 8.1 Feature Detection (Cont.)

8.1.6 BRIEF (Binary Robust Independent Elementary Features)

8.1.7 ORB (Oriented FAST and Rotated BRIEF)

8.1.8 BRISK (Binary Robust Invariant Scalable Key points)

8.1.9 KAZE & AKAZE Feature detector and descriptors

Day 8: 8.2 Feature Matching & Object Detection

8.2.1 BF Matcher

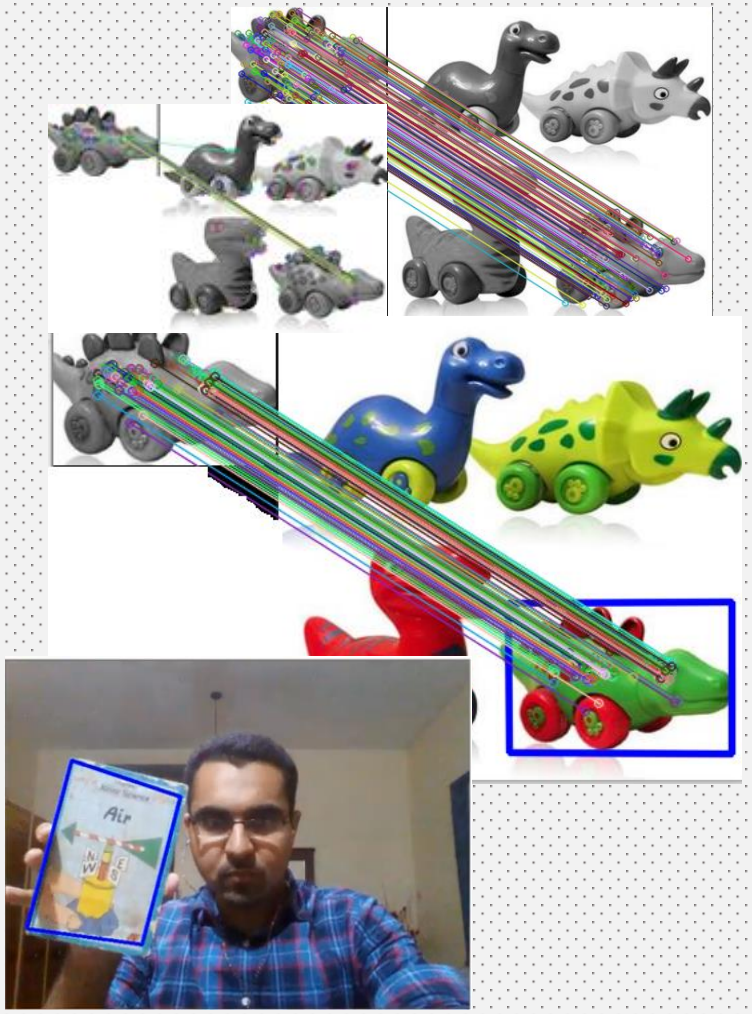
8.2.2 BF KNN Matcher

8.2.3 Flann based Matcher

8.2.4 Feature Matching + Homography to find Objects AKA Robust Object detection

8.2.5 Real Time Object Detection And Tracking with Sift

8.2.6 Real Time Object Detection And Tracking with Orb



Day 9: Object Tracking & Background Subtraction

9.1 Meanshift & Camshaft in OpenCV

9.2 Interactive Camshift with Live feed

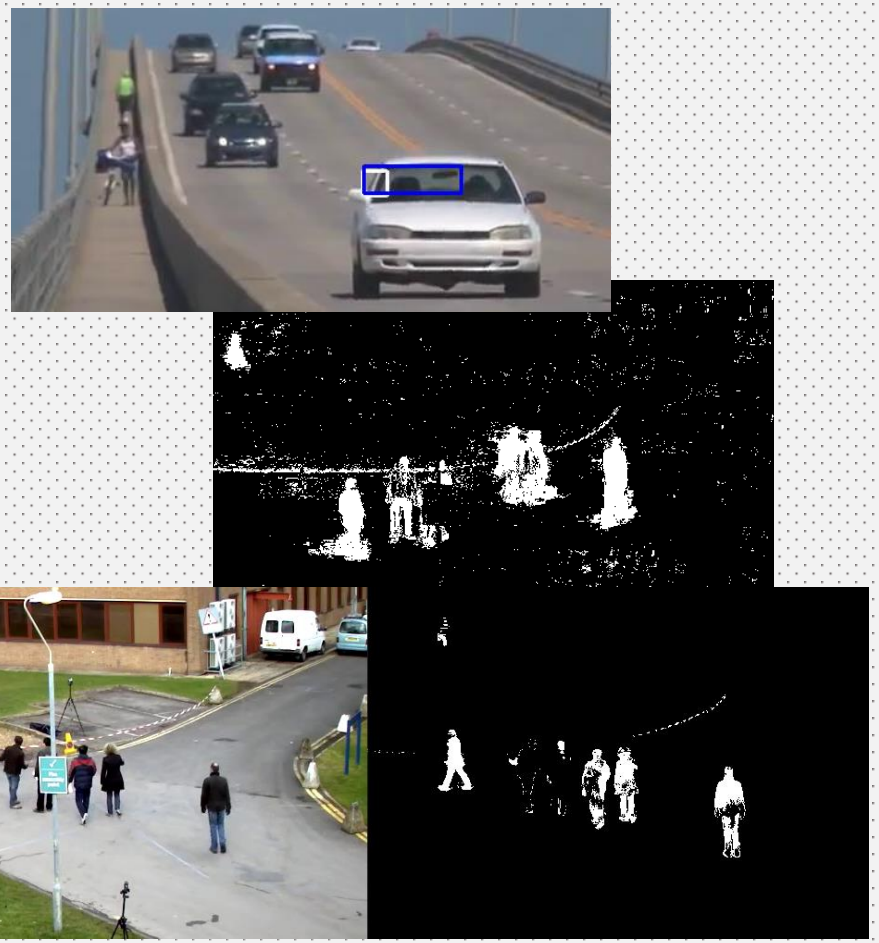
9.3 Optical Flow

9.4 Lucas-Kanade Optical Flow in OpenCV with multiple points

9.5 Background Subtracting Basics

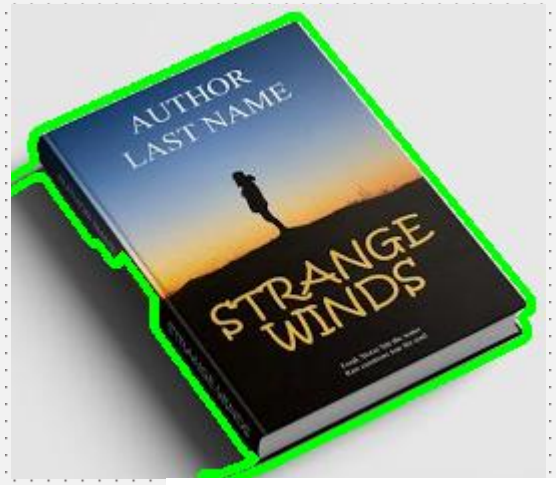
9.6 Background Subtractor MOG

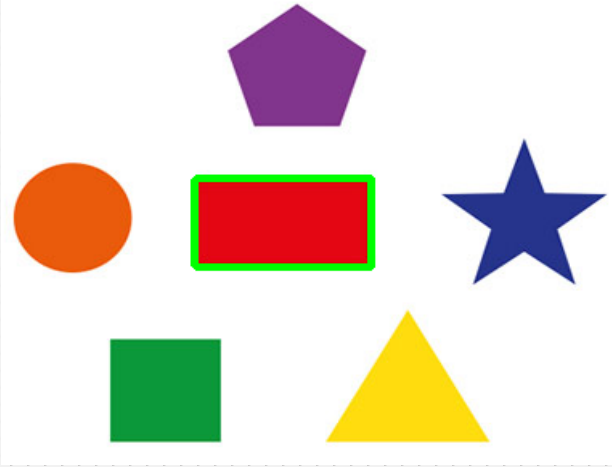
9.7 Background Subtractor MOG2



Day 10: Contours

10.1 image moments
10.2 Contour Approximation
10.3 Convex Hull
10.4 Straight Bounding Rectangle
10.5 Rotated Rectangle
10.6 Minimum Enclosing Circle
10.7 Fitting an Ellipse
10.8 Hu- moments



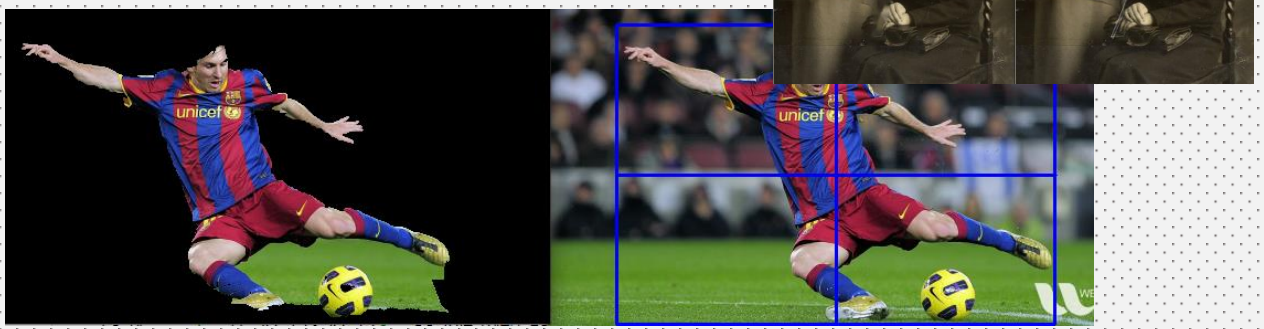


Day 10: Contours (Cont.)

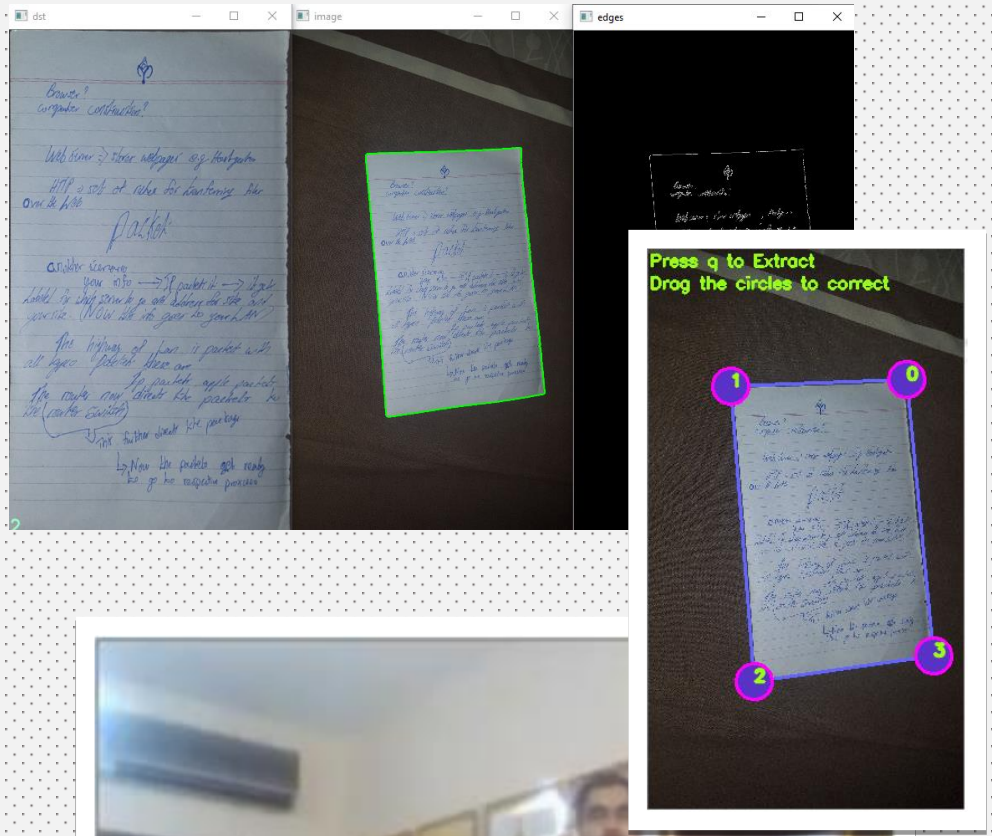
10.9 Convexity Defects
10.10 Point Polygon Test
10.11 Match Shapes
10.12 Getting Max Contour or Nth large contours

Day 10 part 2: Image Restoration

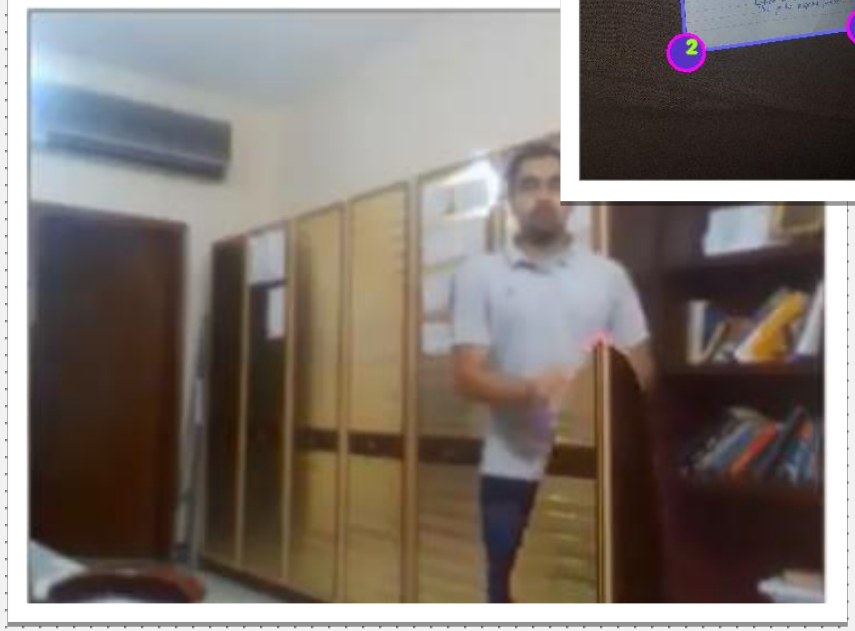
10.13 Image Inpainting Basics
10.14 Grab-cut Algorithm
10.15 Basics Grab-cut application.



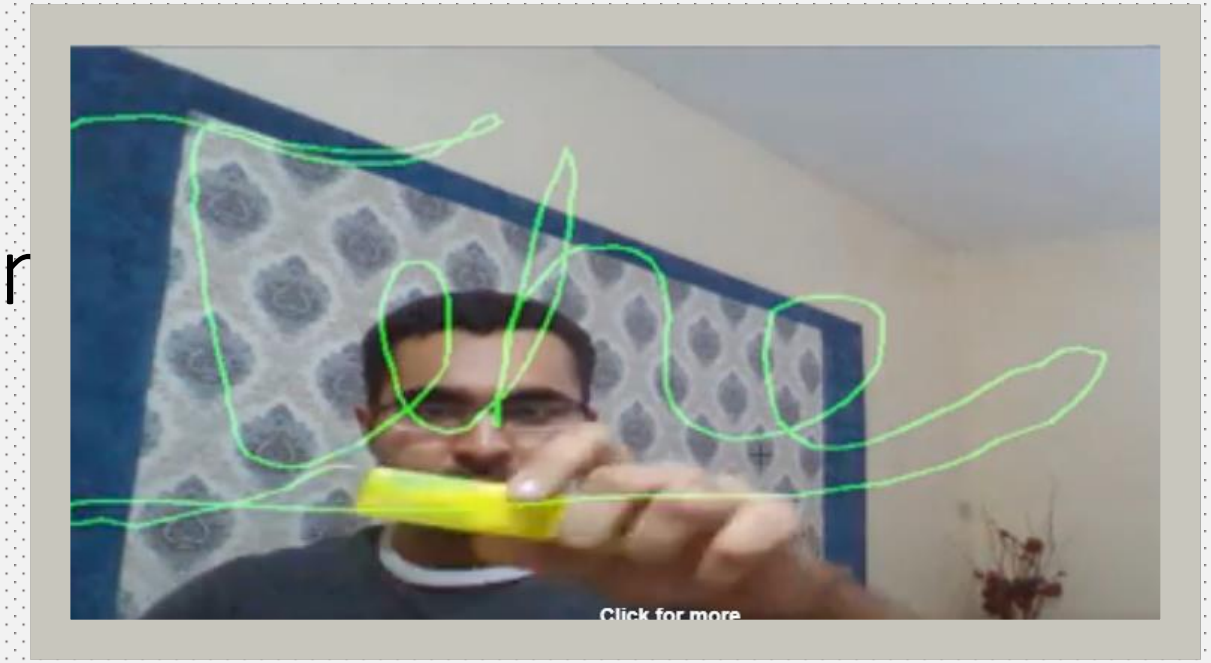
Day 11: Building Document Scanner Application step by step (Major Project)



Day 12: Building Harry Potter Invisibility Cloak (Major Project)

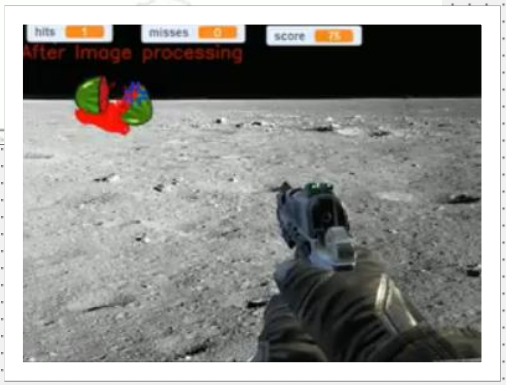


Day 13: Building the Virtual Per (Major Project)

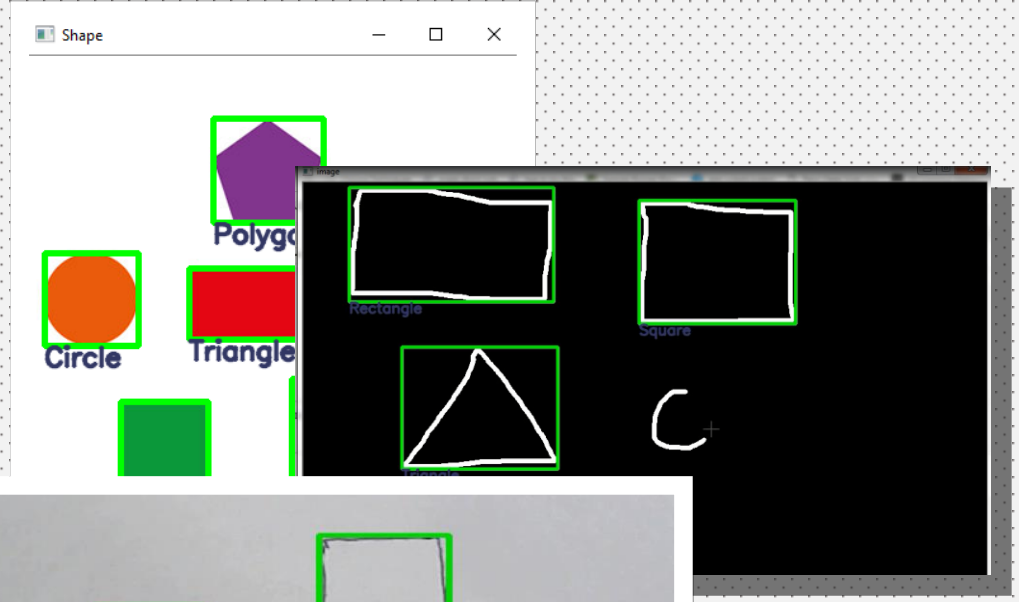


Day 14: Building the Virtual Drum (Major Project)

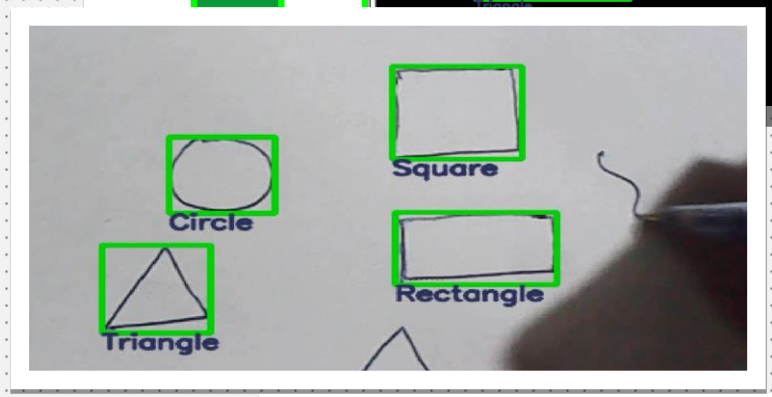
Day 14: Making AI video game Bots (Major Project)



Day 15: Building Shape Detector Application (Major Project)



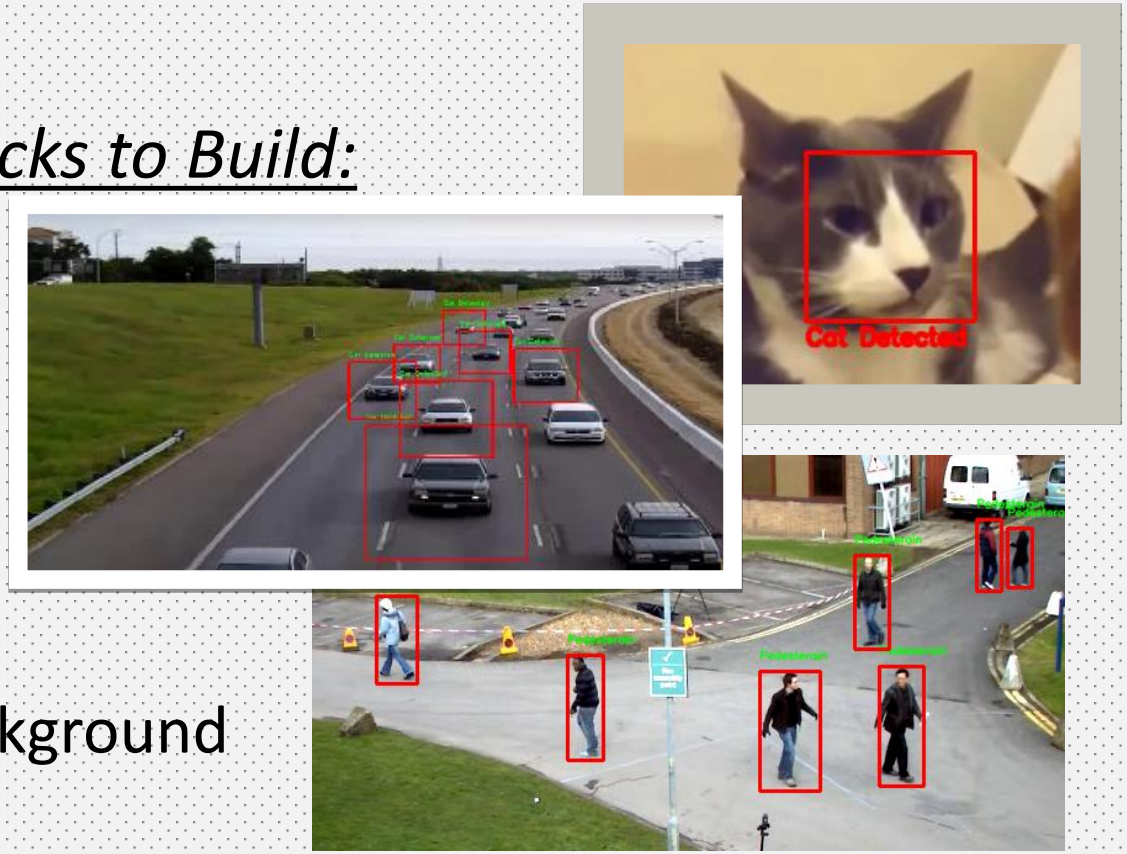
Day 16: Building a Real time Hand Gesture Recognition Calculator with 3 different CV techniques (Major Project)



Day 17: Bonus Class

Utilizing Haar Cascades and some tricks to Build:

- Car detectors
- Cat detectors
- Pedestrian detectors.
- Building Pedestrian detector with Background subtraction method



Why take this course ?

I’m biased but still I can Confidently say this is probably the most practical comprehensive and applied Classical Computer Vision course out there. I’ve taken a little above 6 months to design the Jupyter notebooks for this course making sure that any student with just a descent programming experience can understand all the concepts and build powerful applications.

“YOU DON’T HAVE TO BE GREAT TO GET STARTED BUT YOU HAVE TO GET STARTED TO BE GREAT” –Les Brown

Ready To Jump in and Begin the journey;

Click here to Sign Up for the Course

(Remember after enough registrations we won’t be accepting more students until the next batch)

You can Contact me by messaging on the page **bleedai.com** Or Mail me at: info@bleedai.com