# OpenCV Fundamental

Practice Project:

<https://github.com/Bangbiu/OpenCVPracticing.git>

Tutorial:

<https://www.youtube.com/watch?v=WQeoO7MI0Bs&t=61s>

### Data Reading

1. Read Image

cv2.imread(<PATH>)

1. Show image

cv2.imshow(<WindowName>, <IMAGE>)

1. Pause until Key Press

cv2.waitKey(<MILISEC>)

1. Read Video

cv2.VideoCapture(<PATH>)

1. Read Webcam

cv2.VideoCapture(<Channel>) #1

### Image Processing

1. Convert Color Mode 2 Gray

cv2.cvtColor(<IMAGE>, cv2.COLOR\_BGR2GRAY)

1. Blur

cv2.GaussianBlur(<IMAGE>,<Kernel Size 2D>,<SigmaX>)

1. Canny

cv2.Canny(<IMAGE>,<threshold1>, <threshold2>)

1. Dilate

cv2.dilate(<IMAGE>, <kernel>, <iterations>)

1. Erode

cv2.erode(<IMAGE>, <kernel>, <iterations>)

1. Resize

cv2.resize(<IMAGE>, <SIZE 2D>)

### Sketching

1. Line

cv2.line(<IMAGE>,<StartPT>, <EndPT>,<Color>, <Thickness>)

1. Rectangle

cv2.rectangle(<IMAGE>,<PT1>,<PT2>,<Color>,<Thickness>)

1. Circle

cv2.circle(<IMAGE>, (300,300),100,(255,255,0),3)

1. Text

cv2.putText(<IMAGE>,<TEXT>,<Origin>,<Font>,<Size>,<Color>,<Thickness>)

### Transformative Perspective:

cv2.getPerspectiveTransform (<OriginalPTs4>,<TargetPTs4>)

cv2.warpPerspective(<IMAGE>, <MATRIX>,<SIZE 2D>)

### Color Detection:

1. HSV Model

cv2.cvtColor(<IMAGE>, cv2.COLOR\_BGR2HSV)

1. Get Mask

cv2.inRange(<IMAGE>,<Lower Bound 3D>,<Upper Bound 3D>)

1. Masking

cv2.bitwise\_and(<IMAGE>, <IMAGE>, mask = <MASK>)

### Shape Detection:

1. Canny

See Fundamental

1. getContour

contours, hierarchy =

cv2.findContours(<CannyImage>, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)

1. Calculate Contour Area

cv2.contourArea(<Contour>)

1. Draw Contour

cv2.drawContours(<IMAGE>,<Contour>,<Index,-1==all>,<Color3D>,<Thickness>)

1. Approximate Polymer Vertices

cv2.approxPolyDP(<Contour>,<Epsilon>,<IsClose>)

1. Draw Polymer Vertices

cv2.polylines(<Image>,<ApproxPTs>,<IsClose>,<Color3D>,<Thickness>)

1. Bounding Box

x, y, w, h = cv2.boundingRect(<ApproxPTs>)

### Face Detection:

1. Face Cascade Classifier

cv2.CascadeClassifier(<PathToCascadeFile>)

1. Detect

<Cascade>.detectMultiScale(<Image>,<Scale>,<Neighbor>)



# Pose estimation

Tutorial:

<https://github.com/cmu-perceptual-computing-lab/openpose>

<https://www.youtube.com/watch?v=4FZrE3cmTPA>

https://www.youtube.com/watch?v=9jQGsUidKHs