

# Using OpenCV with Python

cs512

## Introduction

### OpenCV

- C/C++ library with Python wrappers
- Open source
- Implementations of standard computer vision algorithms
- Tutorial: [http://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_tutorials.html](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html)

### Functionality

- Basic Graphical User Interface (GUI)
- Read/write/capture images and videos in various formats
- Image Processing - filtering, edge detection, corner detection, histograms, ...
- Machine Learning - SVM, Decision Trees,
- Object Detection, tracking

### Python wrapper versions

- cv (older, non python standard image storage)
- cv2(contains cv, uses numpy nd-arrays)

```
import cv2
import numpy as np
```

## Image I/O

### Reading an image

```
import sys

# check if image name is specified in command line
if len(sys.argv) == 2:
    filename = sys.argv[1]
elif len(sys.argv) < 2:
    filename = "myfile.jpg"

image = cv2.imread(filename)
print image.shape
```

### Capturing an image

```
if len(argv)<2 :                # no parameters
    cap = cv2.VideoCapture(0)   # set capture device
    retval,image = cap.read()   # capture the original image
else:
    image = cv2.imread("myfile.jpg")
```

### Processing a video

```
movie=cv2.VideoCapture(filename)
while True:
    valid_frame, image = movie.read()
    if not valid_frame:
        break
    cv2.imshow('Video Example',image)
    cv2.waitKey(5)             # 5ms

cv2.destroyAllWindows()
```

### Writing an image

```
cv2.imwrite("out.jpg",image)
```

## Basic image processing

### Resizing an image

```

b = cv2.imread("myfile.jpg")

bs = cv2.resize(b, (800,600))          # absolute resize
print bs.shape

bs = cv2.resize(b, (b.shape[1]/4, b.shape[0]/4)) # relative resize
print bs.shape

```

## Converting an image to grayscale

```

image      = cv2.imread("myfile.jpg")
image_bw   = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY )

```

## Traversing an image

```

image = cv2.imread("myfile.jpg")
outImg = np.ones((img.shape[0],img.shape[1],3),np.uint8)
for i in range(0,img.shape[0],1):
    for j in range(0,img.shape[1],1):
        outImg[i][j][0] = 255-img[i][j][0]
        outImg[i][j][1] = 255-img[i][j][1]
        outImg[i][j][2] = 255-img[i][j][2]

imageBw = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
outBwImg = np.ones((img.shape[0],img.shape[1]),np.uint8)
for i in range(0,imageBw.shape[0],1):
    for j in range(0,imageBw.shape[1],1):
        outBwImg[i][j] = imageBw[i][j] * 0.5

```

## Rotating an image

```

img      = cv2.imread("myfile.jpg")
angle    = 45*np.pi/180
rows     = img.shape[0]
cols     = img.shape[1]
M        = cv2.getRotationMatrix2D((cols/2,rows/2),angle,1)
dst      = cv2.warpAffine(img,M,(cols,rows)) # affine transformation

```

## GUI

### Displaying image

```

a = np.random.rand(256,256)
cv2.imshow('Random Image', a)
cv2.waitKey()           # wait for key press
cv2.destroyWindow('Random Image')

b = cv2.imread("myfile.jpg")
print b.shape
bs = cv2.resize(b,(800,600))
cv2.imshow('Hello World', bs)
cv2.waitKey()
cv2.destroyAllWindows()

```

### Annotating an image

```

image = cv2.imread(filename)
textColor = (0, 0, 255) # red
cv2.putText(image, "My annotation", (200 ,200),
             cv2.FONT_HERSHEY_PLAIN, 3.0, textColor,
             thickness=4, linetype=cv2.CV_AA)

```

### Using a keyboard loop

```

while(True):
    key = cv2.waitKey(10)
    if key ==ord('i'):
        doInversion(img)
        cv2.imshow(winName,img)

    if key == ord('w'):
        cv2.imwrite("out.jpg",img)

    if key==ord('h'):
        print "<ESC>: quit"
        print "'i' - reload image"

    if key==27:
        cv2.destroyWindow(winName)
        break

```

#### Notes:

1. In Linux use: `key = cv2.waitKey(10) & 255`
2. To wait for a key (instead of having empty loop events) use: `key = cv2.waitKey()`

## Using a track bar (slider bar) event handler

```
def sliderHandler():
    global img
    n=5
    kernel = np.ones((n,n),np.float32)/(n*n)    # make a smoothing filter
    dst = cv2.filter2D(src,-1,kernel)          # convolve the image
    cv2.imshow(winName,dst)

winName = 'Hello world'
img = cv2.imread(filename)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow(winName,img)
cv2.createTrackbar('s',winName,0,255,sliderHandler) # slider bar [0,255]
```

## Examples

### Feature detection example

```
image = cv2.imread(filename)
image_small = cv2.resize(image, (image.shape[1]/4, image.shape[0]/4))
image_gray = cv2.cvtColor(image_small, cv2.cv.CV_RGB2GRAY)

# find 50 points using a 0.1 threshold and a minimum distance of 50
features = cv2.goodFeaturesToTrack(image_gray, 50, .01 , 50)

# reshape the 3D shape array (50,1,2) to 2D array (50,2)
features = features.reshape((-1 , 2))

for x, y in features:
    cv2.circle(image_small, (x, y), 10, (0, 0, 255))

cv2.imshow('Features Example', image_small)
cv2.waitKey()
cv2.destroyAllWindows()
```