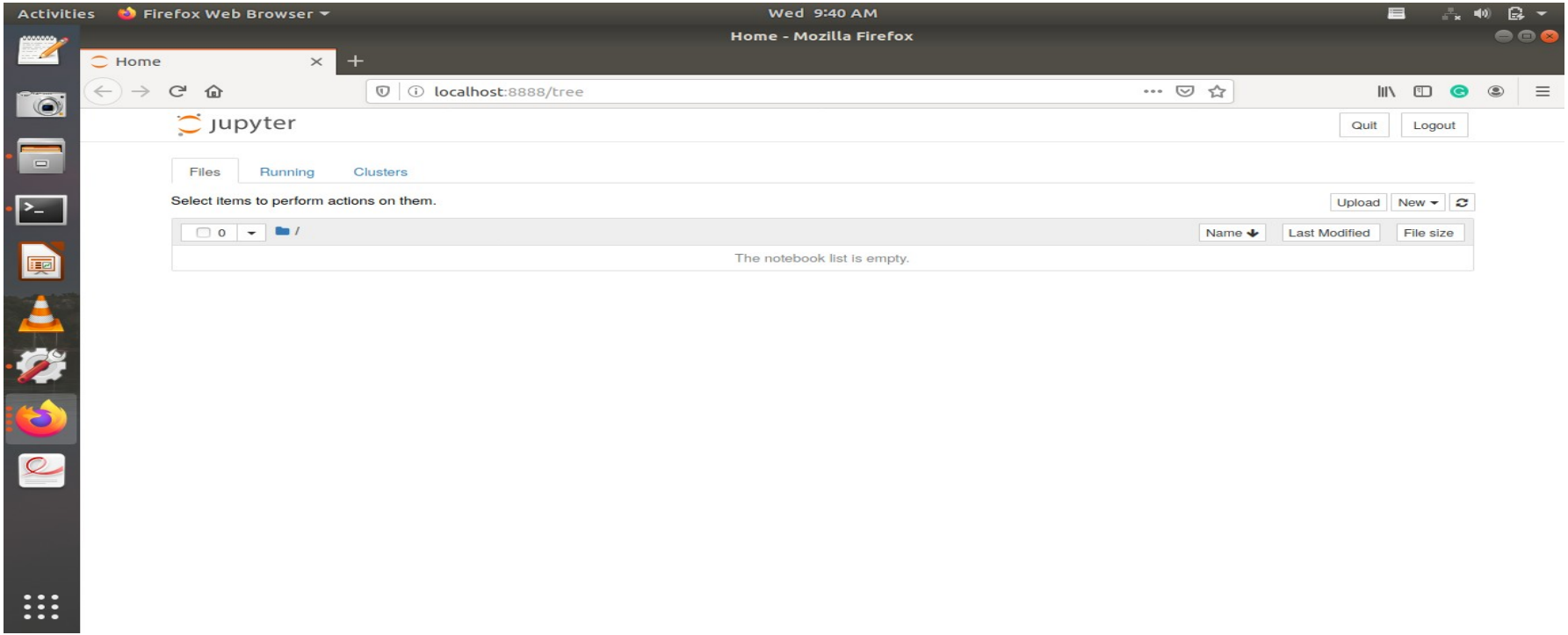
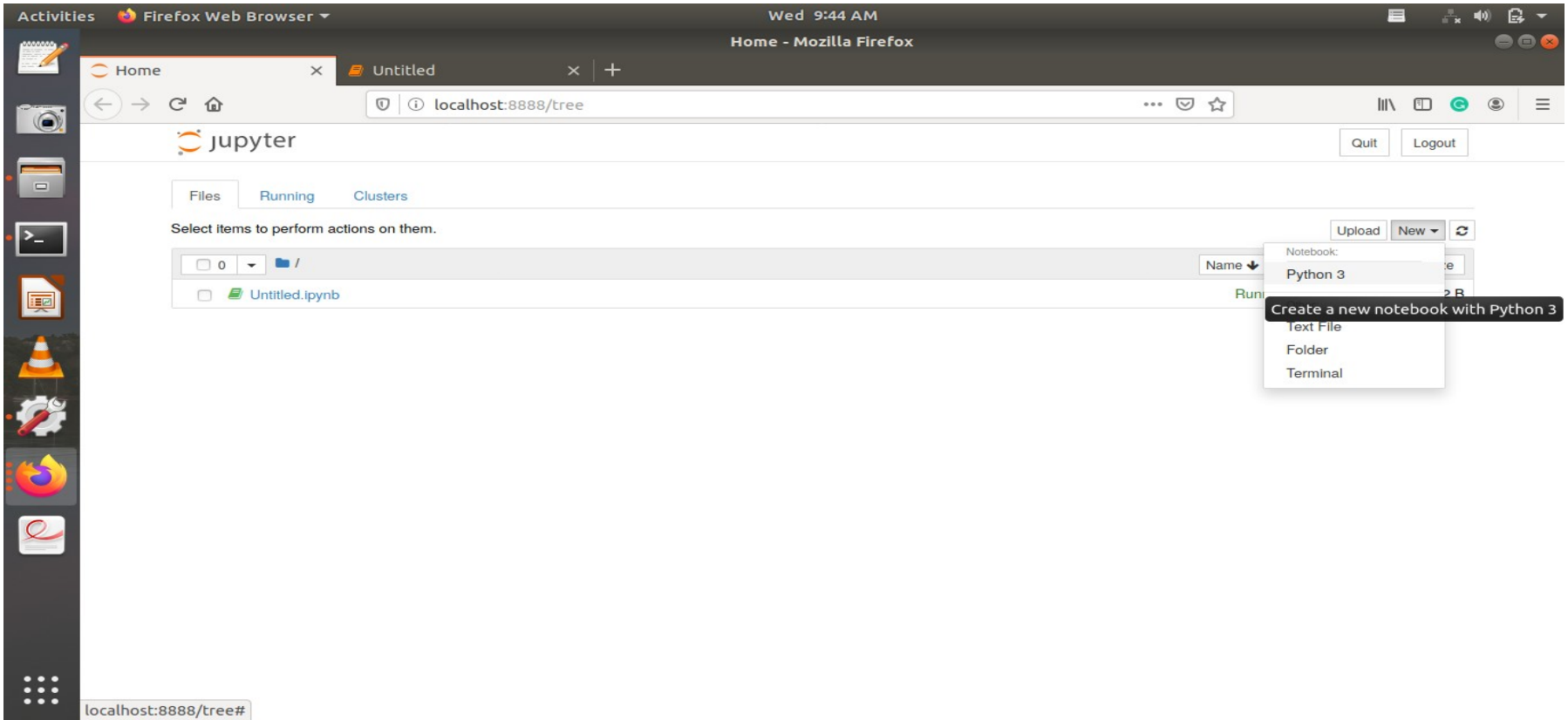
OpenCV for DIP

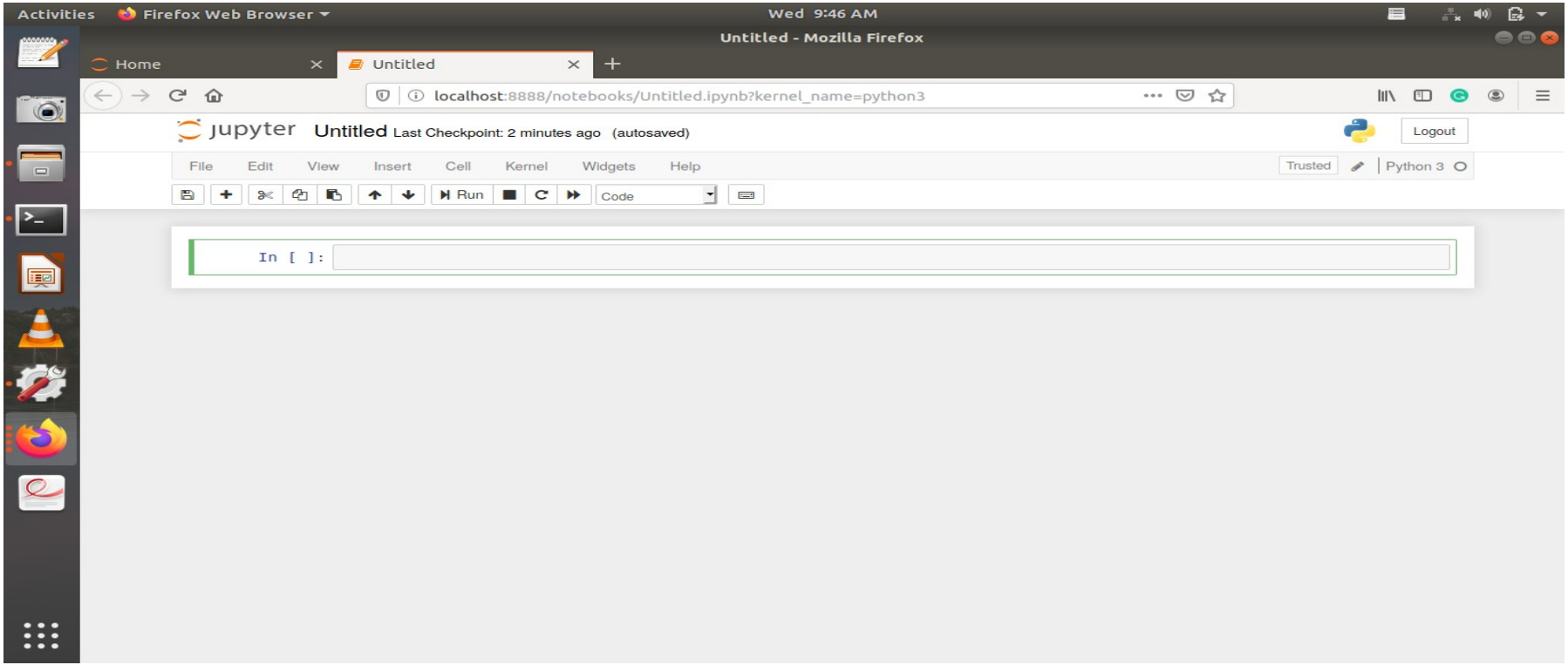
Jupyter


Activities
Firefox Web Browser
Wed 9:40 AM
Home - Mozilla Firefox
Home
localhost:8888/tree
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size
The notebook list is empty.

Creating a python file



Jupyter Notebook



Installing OpenCV

Install package python-opencv :

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

Test:

```
import cv2

print(cv2.__version__)
```

Reading and Displaying an image

Method 1:

```
import cv2

img = cv2.imread("trolley.png")

cv2.imshow('Original Image', img)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

Activities Wed 10:07 AM

Untitled - Mozilla Firefox

Inbox (8) - coe18d003@i | Home | Untitled

localhost:8889/notebooks/Untitled.ipynb

jupyter Untitled Last Checkpoint: 23 minutes ago (unsaved changes)

Logout

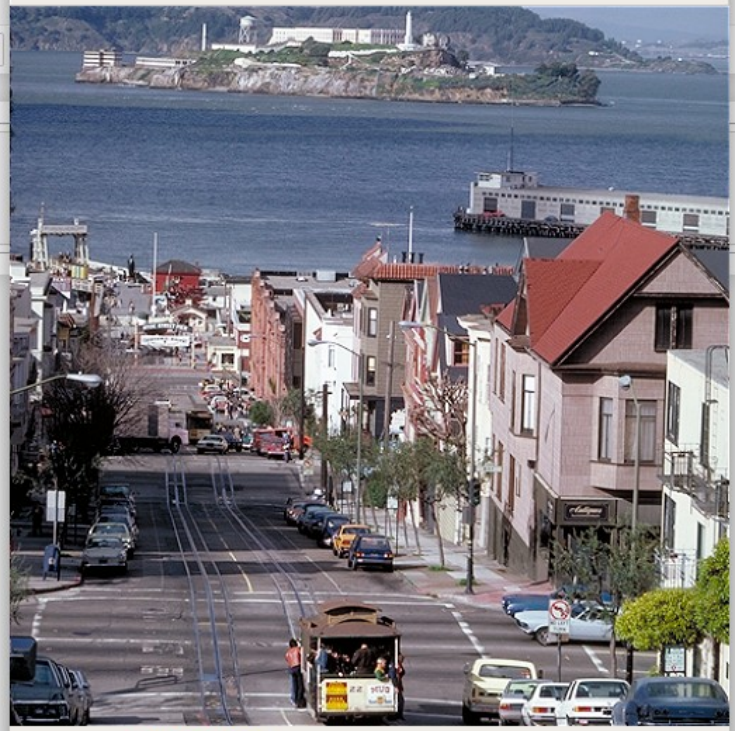
File Edit View Insert Cell Kernel Widgets Help

Python 3

Run Code

In [*]:
```
import cv2
img = cv2.imread("trolley.png")
cv2.imshow('Original Image', img)
cv2.waitKey(0)
```

Original Image



(x=8, y=103) ~ R:70 G:90 B:124

Displaying an image

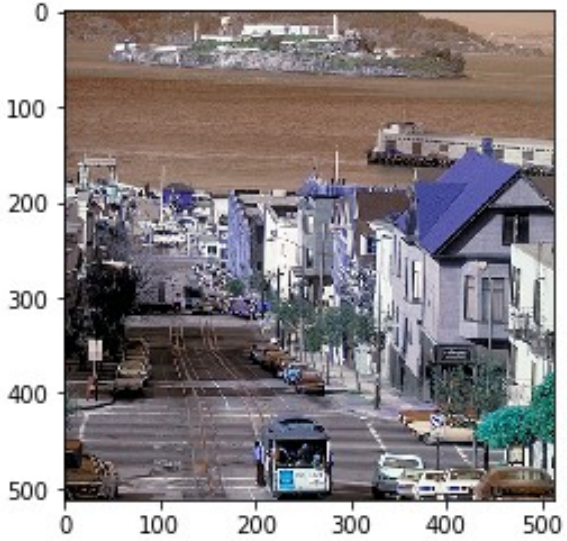
Method 2:

```
import cv2

from matplotlib import pyplot as plt

im = cv2.imread('trolley.png')

plt.imshow(im)

plt.show()
```

```
#print(img.shape[0:2])
```

In [10]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
im = cv2.imread('trolley.png')
plt.imshow(im)
plt.show()
```



Size and Shape of an image

print(img.size)

print(img.shape)

print(img[0,0])

print(img)

In [17]:

```
print(img.size)
print(img.shape)
print(img[0,0])
print(img)
```

```
786432
(512, 512, 3)
[67 53 45]
[[[ 67  53  45]
  [131 101  94]
  [138 103  92]
  ...
  [239 202 183]
  [240 205 187]
  [255 242 217]]

 [[ 89  74  61]
  [ 86  62  49]
  [100  69  53]
  ...
  [190 165 143]
  [193 160 145]
  [238 197 179]]

 [[118  89  76]
  [102  72  58]
  [105  71  59]
  ...
  [194 164 145]
  [192 162 141]
```

RGB to Gray Color Conversion - Method 1

```
import cv2

# Reading color image as grayscale

gray = cv2.imread("trolley.png",1)

# Showing grayscale image

cv2.imshow("Grayscale Image", gray)

cv2.waitKey(0)

cv2.destroyAllWindows()

i
```

Home | Untitled

localhost:8888/notebooks/Untitled.ipynb

jupyter Untitled Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Python 3

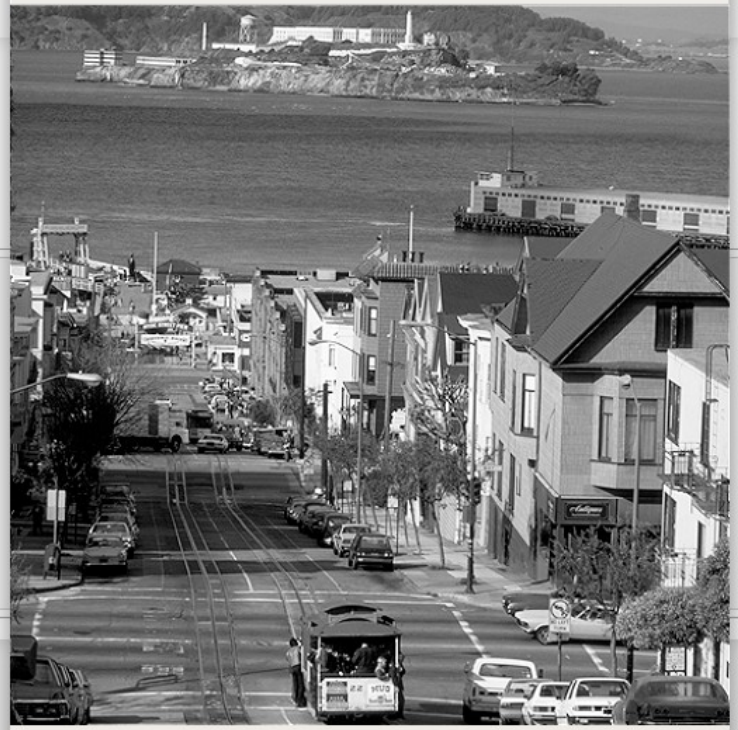
Run Code

```
[[  0   2   2]
 [  0   1   1]
 [  0   1   1]
 ...
 [ 90  67  61]
 [ 66  55  47]
 [ 79  63  55]]]
```

In [*]:

```
import cv2

# Reading color image as grayscale
gray = cv2.imread("trolley.png",0)

# Showing grayscale image
cv2.imshow("Grayscale Image", gray)

# waiting for key event
cv2.waitKey(0)

# destroying all windows
cv2.destroyAllWindows()
```

Grayscale Image



(x=362, y=5) ~ L:94

RGB to Gray Color Conversion - Method 2

```
import cv2

img = cv2.imread("trolley.png")

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imshow("Converted Image",gray)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

Activities Wed 11:07 AM

Untitled - Mozilla Firefox

Home | Untitled

localhost:8888/notebooks/Untitled.ipynb

jupyter Untitled Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

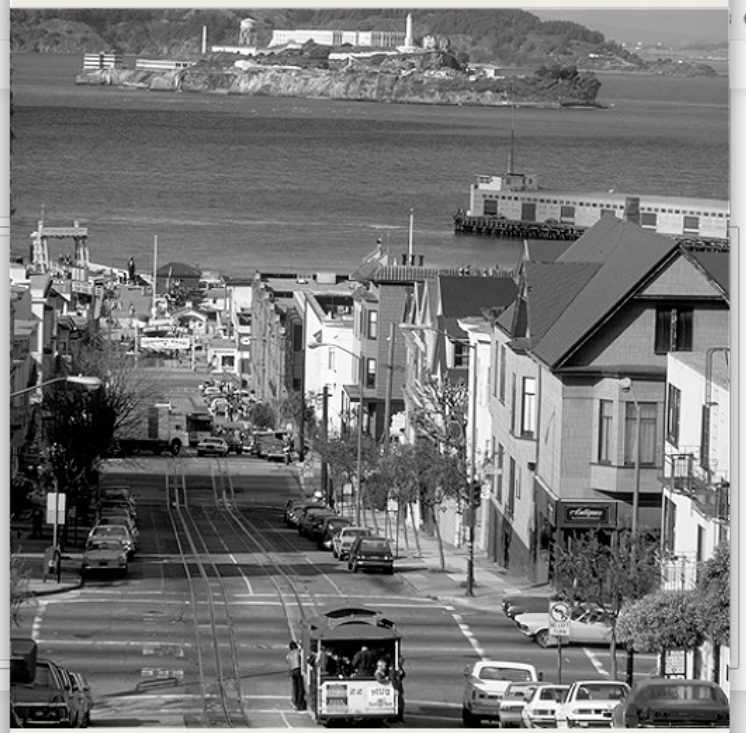
Run Code

```
cv2.waitKey(0)

# destroying all windows
cv2.destroyAllWindows()
```

In [*]:

```
import cv2

# Reading color image
img = cv2.imread("trolley.png")

# Converting color image to grayscale image
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Showing the converted image
cv2.imshow("Converted Image",gray)

# waiting for key event
cv2.waitKey(0)

# destroying all windows
cv2.destroyAllWindows()
```

Converted Image



(x=222, y=202) ~ L:184

Color Conversion

```
import cv2

from matplotlib import pyplot as plt

img = cv2.imread('trolley.png',0)

ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)

ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)

ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)

ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)

ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)
```

Cont..

```
titles = ['Original
Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']

images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in range(6):

    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')

    plt.title(titles[i])

    plt.xticks([]),plt.yticks([])

plt.show()
```

- **THRESH_BINARY**

$$\texttt{dst}(x,y) = \begin{cases} \texttt{maxval} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ 0 & \text{otherwise} \end{cases}$$

- **THRESH_BINARY_INV**

$$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{maxval} & \text{otherwise} \end{cases}$$

- **THRESH_TRUNC**

$$\texttt{dst}(x,y) = \begin{cases} \texttt{threshold} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{src}(x,y) & \text{otherwise} \end{cases}$$

- **THRESH_TOZERO**

$$\texttt{dst}(x,y) = \begin{cases} \texttt{src}(x,y) & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ 0 & \text{otherwise} \end{cases}$$

- **THRESH_TOZERO_INV**

$$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{src}(x,y) & \text{otherwise} \end{cases}$$

Activities Firefox Web Browser Wed 11:20 AM

Untitled - Mozilla Firefox

Home | Untitled

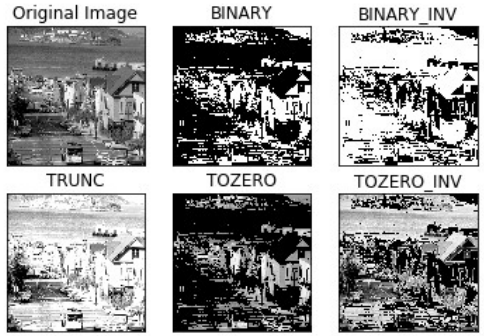
localhost:8888/notebooks/Untitled.ipynb

jupyter Untitled Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [24]:
```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('trolley.png',0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)
titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])
plt.show()
```



Writing an image

```
img = cv2.imread("trolley.png")

# Converting color image to grayscale image

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imwrite('gray_trolley.png',gray)
```

Activities Files Wed 11:37 AM

Untitled - Mozilla Firefox

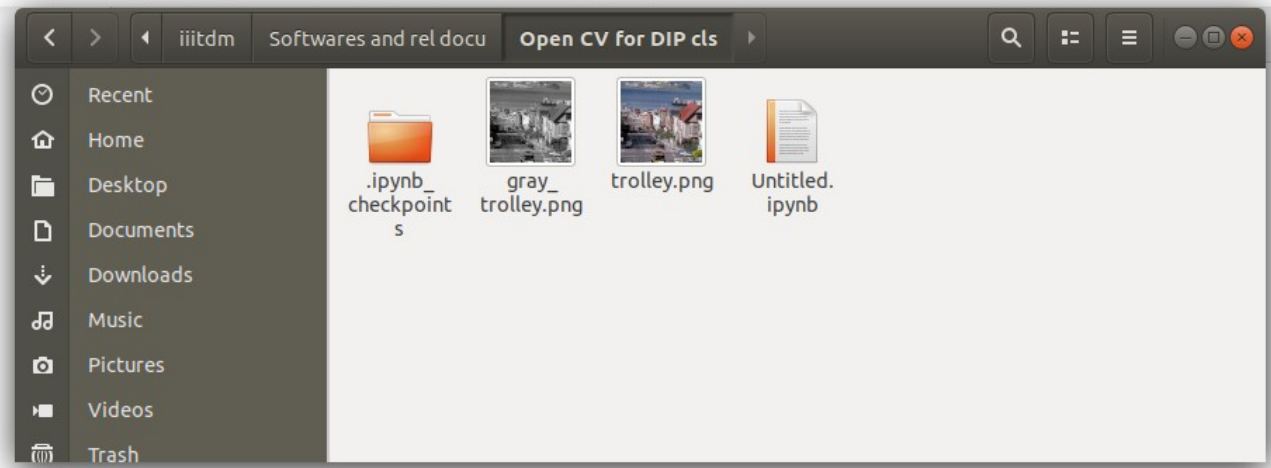
Home | Untitled

localhost:8888/notebooks/Untitled.ipynb

jupyter Untitled Last Checkpoint: 2 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code



In [30]:
```
img = cv2.imread("trolley.png")

# Converting color image to grayscale image
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imwrite('gray_trolley.png',gray)
```

Out[30]: True

Geometric Transformation

Resize : Method 1:

```
import cv2

img = cv2.imread('binary_trolley.png')

height, width = img.shape[:2]

res = cv2.resize(img,(2*width, 2*height), interpolation = cv2.INTER_CUBIC)

cv2.imshow("Resized Image",res)

cv2.waitKey(0);  cv2.destroyAllWindows()
```

Geometric Transformation

Resize : Method 2:

```
import cv2

img = cv2.imread('trolley.png')

res = cv2.resize(img,None,fx=2, fy=2, interpolation = cv2.INTER_CUBIC)

cv2.imshow("Resized Image",res)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

Example

Geometric Transformation

Rotation : Method 1:

```
import cv2

img = cv2.imread('trolley.png',0)

rows,cols = img.shape

M = cv2.getRotationMatrix2D((cols/2,rows/2),60,1)

dst = cv2.warpAffine(img,M,(cols,rows))

cv2.imshow("Rotated Image",dst);  cv2.waitKey(0); cv2.destroyAllWindows()
```

Geometric Transformation

Rotation : Method 2:

```
import cv2

img = cv2.imread('trolley.png',0)

R = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)

cv2.imshow("Rotated Image",R)

cv2.waitKey(0); cv2.destroyAllWindows()
```

Example

Geometric Transformation

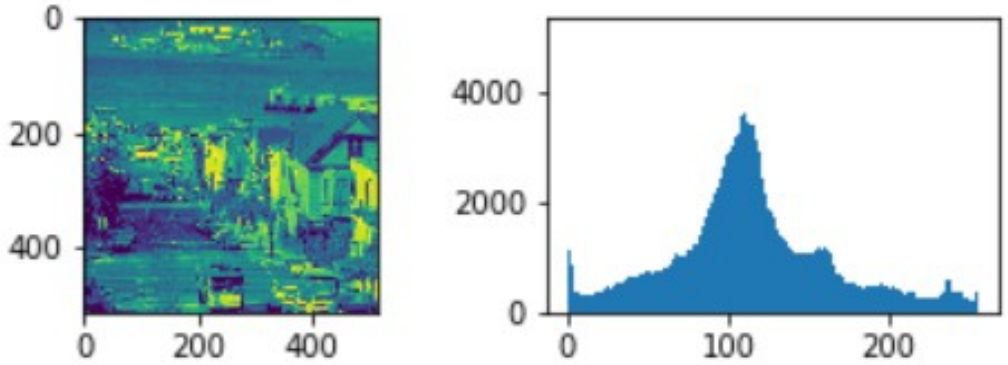
Translation:

```
import cv2

import numpy as np

img = cv2.imread('trolley.png',0)

rows,cols = img.shape

M = np.float32([[1,0,100],[0,1,50]])

dst = cv2.warpAffine(img,M,(cols,rows))

cv2.imshow('img',dst); cv2.waitKey(0); cv2.destroyAllWindows()

print(img)
```

Histogram

```
import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('trolley.png',0)

plt.subplot(221),plt.imshow(img)

plt.subplot(222),plt.hist(img.ravel(),256,[0,256]); plt.show()
```

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('trolley.png',0)
#plt.imshow(img)
plt.subplot(221),plt.imshow(img)
plt.subplot(222),plt.hist(img.ravel(),256,[0,256]); plt.show()
#plt.hist(img.ravel(),256,[0,256]); plt.show()
```



Adding Gaussian Noise

```
import cv2
import numpy as np
from skimage.util import random_noise
 img = cv2.imread("42049.jpg")
noise_img =
random_noise(img,mode='gaussian',mean=0,var=0.01)
noise_img = np.array(255*noise_img, dtype = 'uint8')
cv2.imwrite('blur_42049.png',noise_img)
cv2.imshow('blur',noise_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Adding Salt and Pepper Noise

```
import cv2
import numpy as np
from skimage.util import random_noise
img = cv2.imread("42049.jpg")
noise_img = random_noise(img, mode='s&p',amount=0.1)
noise_img = np.array(255*noise_img, dtype = 'uint8')
cv2.imwrite('sp_42049.png',noise_img)
cv2.imshow('blur',noise_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Median Filter

```
import cv2

from matplotlib import pyplot as plt

img = cv2.imread('42049.jpg')

median = cv2.medianBlur(img,5)
```

Cont..

```
plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(median),plt.title('Smooth Image’)
plt.xticks([]), plt.yticks([])
plt.show()
```

Gaussian Filter

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('blur_42049.png')

blur = cv2.GaussianBlur(img,(11,11),0)

plt.subplot(121),plt.imshow(img),plt.title('Noisy Image')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(blur),plt.title('Smooth Image')
plt.xticks([]), plt.yticks([])
plt.show()
```

Example

Noisy Image

Smooth Image

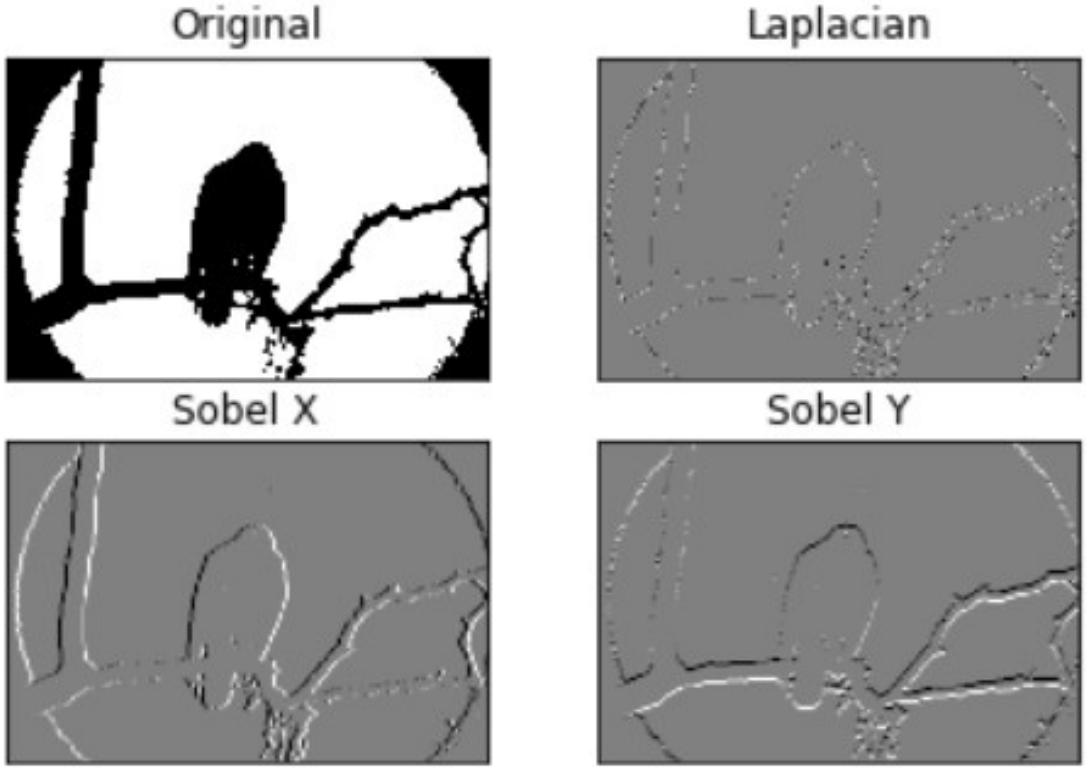
Edge Detection

```
import cv2

from matplotlib import pyplot as plt

img = cv2.imread('42049.jpg',0)

ret,thresh1 = cv2.threshold(img,172,255,cv2.THRESH_BINARY)

laplacian = cv2.Laplacian(thresh1,cv2.CV_64F)

sobelx = cv2.Sobel(thresh1,cv2.CV_64F,1,0,ksize=5)

sobely = cv2.Sobel(thresh1,cv2.CV_64F,0,1,ksize=5)
```

Cont..

```
plt.subplot(2,2,1),plt.imshow(thresh1,cmap='gray'); plt.title('Original'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,2),plt.imshow(laplacian,cmap='gray'); plt.title('Laplacian'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,3),plt.imshow(sobelx,cmap='gray'); plt.title('Sobel X'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,4),plt.imshow(sobely,cmap='gray'); plt.title('Sobel Y'), plt.xticks([]), plt.yticks([]); plt.show()

cv2.imshow('laplacian',laplacian); cv2.waitKey(0); cv2.destroyAllWindows()
```

Output



Edge Detection

```
import cv2

from matplotlib import pyplot as plt

img = cv2.imread('42049.jpg',0); edges = cv2.Canny(img,100,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')

plt.title('Original Image'), plt.xticks([]), plt.yticks([])
```

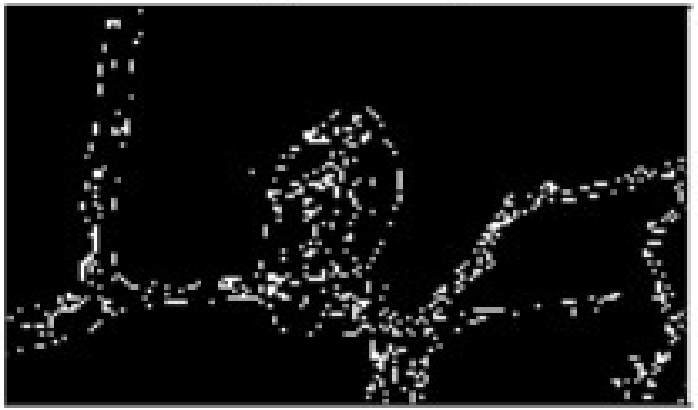
Cont..

```
plt.subplot(122),plt.imshow(edges,cmap = 'gray')

plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()

cv2.imshow('canny',edges)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

Output

Original Image

Edge Image



Morphological Operation

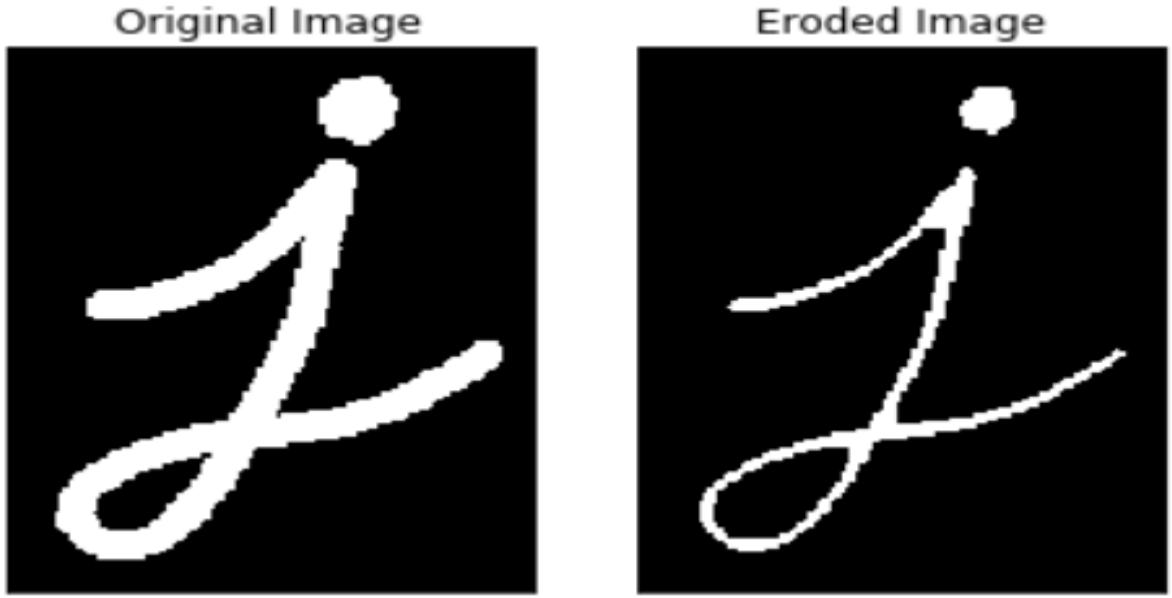
Erosion :

```
import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('j.png',0)

kernel = np.ones((5,5),np.uint8)

erosion = cv2.erode(img,kernel,iterations = 1)
```

Cont..

```
plt.subplot(121),plt.imshow(img,cmap = 'gray')

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(erosion,cmap = 'gray')

plt.title('Eroded Image'), plt.xticks([]), plt.yticks([])

plt.show()
```

Output



Morphological Operation

Dilation :

```
import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('j.png',0)

kernel = np.ones((5,5),np.uint8)

dilation = cv2.dilate(img,kernel,iterations = 1)
```

Cont..

```
plt.subplot(121),plt.imshow(img,cmap = 'gray')

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(dilation,cmap = 'gray')

plt.title('Dilated Image'), plt.xticks([]), plt.yticks([])

plt.show()
```

Original Image

Dilated Image

Morphological Operation

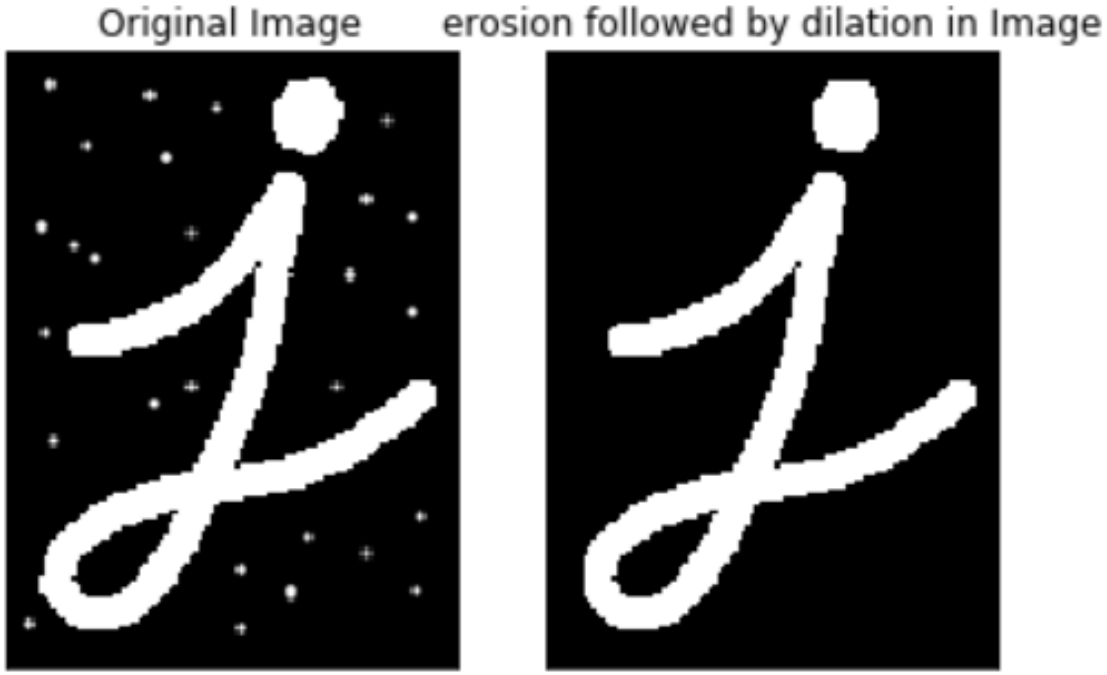
Opening :

```
import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('jforopening.png',0)

kernel = np.ones((5,5),np.uint8)

opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```

Cont..

```
plt.subplot(121),plt.imshow(img,cmap = 'gray')

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(opening,cmap = 'gray')

plt.title('erosion followed by dilation in Image'), plt.xticks([]), plt.yticks([])

plt.show()
```

Output



Morphological Operation

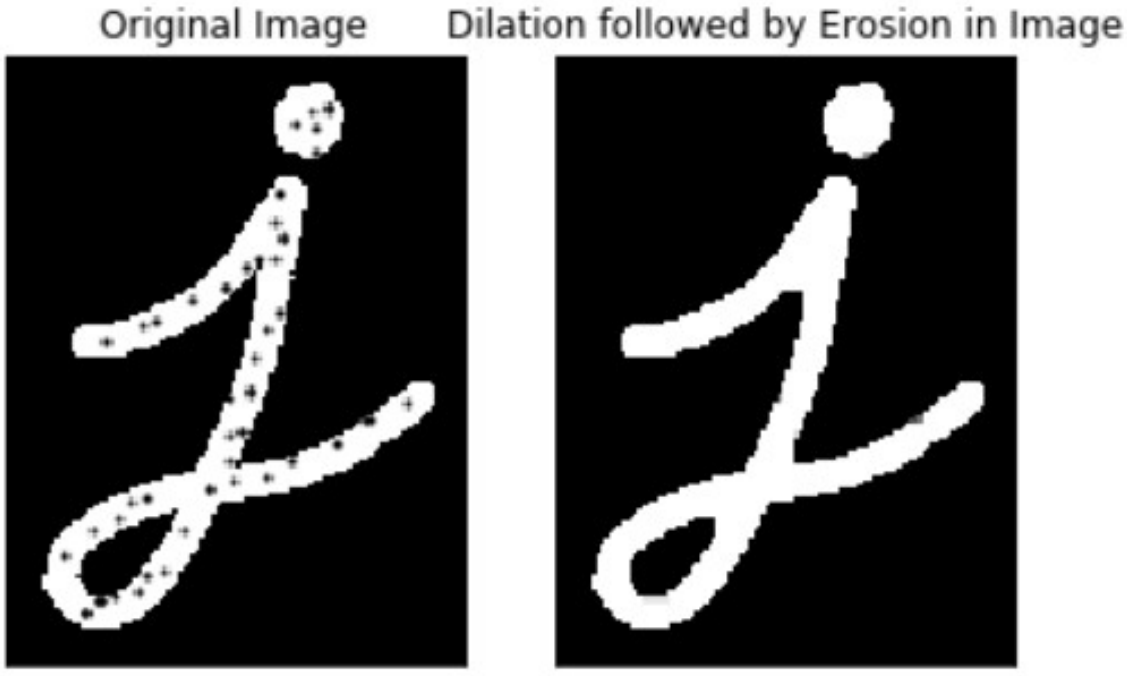
Closing :

```
import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('jforclosing.png',0)

kernel = np.ones((5,5),np.uint8)

closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

Cont

```
plt.subplot(121),plt.imshow(img,cmap = 'gray')

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(closing,cmap = 'gray')

plt.title('Dilation followed by Erosion in Image'), plt.xticks([]), plt.yticks([])

plt.show()
```

Output



References

1. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html
2. https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html?highlight=cvtcolor
3. https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html?highlight=warpaffine#warpaffine
4. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html
5. https://note.nkmk.me/en/python-opencv-numpy-rotate-flip/

Thank You