Histogram processing

import cv2

import numpy as np

from matplotlib import pyplot as plt

%matplotlib inline

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

cv2.imshow("original",img)

histogram=cv2.calcHist([img],[0],None,[256],[0,256])

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

color=('b','g','r')

for \_i,col in enumerate(color):

histogram1=cv2.calcHist([img],[i],None,[256],[0,256])

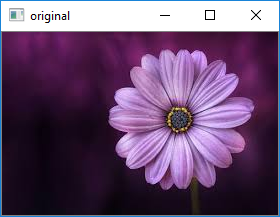
plt.plot(histogram1,color=col)

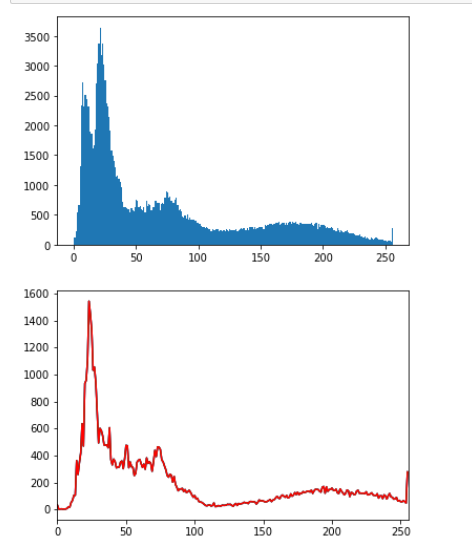
plt.xlim([0,256])

plt.show()

cv2.waitKey(0)

cv2.distoryAllWindows()





Spatial filters

import cv2

import numpy as np

image=cv2.imread('elephant.jpg')

cv2.imshow('original image',image)

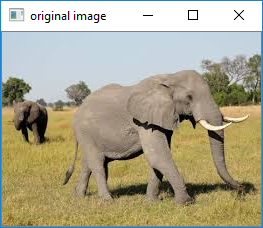
kernel\_3x3=np.ones((3,3),np.float32)/9

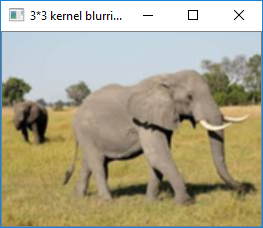
blurred=cv2.filter2D(image,-1,kernel\_3x3)

cv2.imshow('3\*3 kernel blurring',blurred)

cv2.waitKey(0)

cv2.destoryAllWindow()





Histogram processing by applied mask

import cv2

import numpy as np

from matplotlib import pyplot as plt

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

%matplotlib inline

mask= np.zeros(img.shape[:2],np.uint8)

mask[50:200,50:400]=255

masked\_img=cv2.bitwise\_and(img,img,mask=mask)

hist\_full=cv2.calcHist([img],[0],None,[256],[0,256])

hist\_mask=cv2.calcHist([img],[0],mask,[256],[0,256])

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

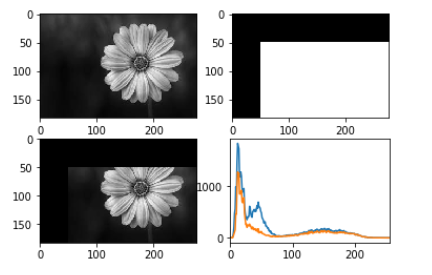
plt.subplot(222),plt.imshow(mask,'gray')

plt.subplot(223),plt.imshow(masked\_img,'gray')

plt.subplot(224),plt.plot(hist\_full),plt.plot(hist\_mask)

plt.xlim([0,256])

plt.show()



Average filter

import cv2

import numpy as np

image=cv2.imread('elephant.jpg')

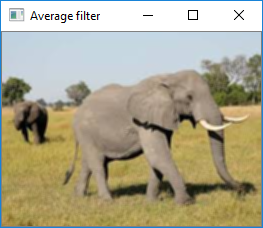
cv2.imshow('original image',image)

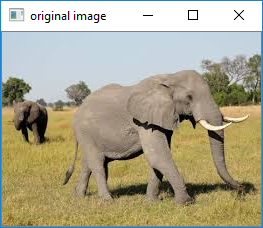
avg=cv2.blur(image,(3,3))

cv2.imshow('Average filter',avg)

cv2.waitKey(0)

cv2.destroyAllWindows()





median filter

import cv2

import numpy as np

image=cv2.imread('elephant.jpg')

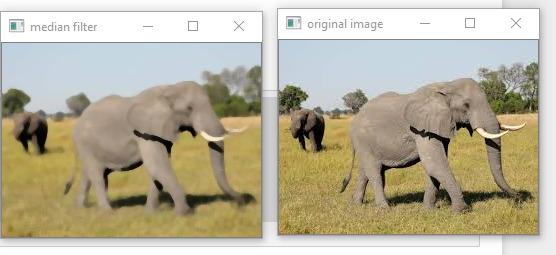
cv2.imshow('original image',image)

medi=cv2.medianBlur(image,5)

cv2.imshow('median filter',medi)

cv2.waitKey(0)

cv2.destroyAllWindows()



Gaussian filter

import cv2

import numpy as np

image=cv2.imread('elephant.jpg')

cv2.imshow('original image',image)

gauss=cv2.GaussianBlur(image,(7,7),0)

cv2.imshow('Gaussian filter',gauss)

cv2.waitKey(0)

cv2.destroyAllWindows()

