**Lab 09 DIP (Morphological Operators)**

AliAmirKhawaja 344200

MueedHassan 347391

10-B

**Task 1**

Code:

import cv2

from PIL import Image, ImageChops

import numpy as np

from google.colab.patches import cv2\_imshow

im=cv2.imread("/content/lara.jpg")

im=cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

cv2\_imshow(im)

#finding out resolution

resolution=im.shape

print("Resolution:",resolution)

#converting image to array

im=np.array(im)

kernal = np.ones((3,3))

#inverting the image, as a function of erosion and dilation happens on white pixels, not black

for x in range(0,resolution[0]):

for y in range(0,resolution[1]):

if im[x][y]<100:

im[x][y]=255

else:

im[x][y]=0

#Closing

im=cv2.dilate(im,kernal)

im=cv2.erode(im,kernal)

#Erosion

im=cv2.erode(im,kernal)

im=cv2.dilate(im,kernal)

#inverting the image

for x in range(0,resolution[0]):

for y in range(0,resolution[1]):

if im[x][y]>100:

im[x][y]=0

else:

im[x][y]=255

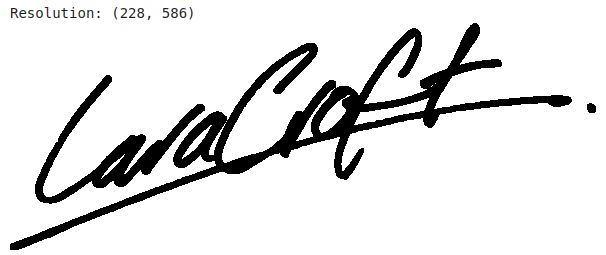
im=Image.fromarray(im)

im

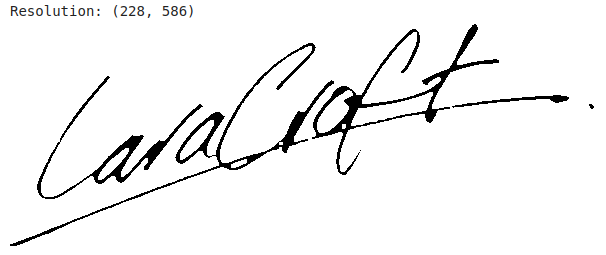
**Original**



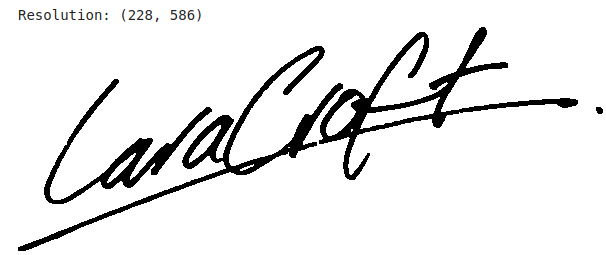
**Dilate:**

****

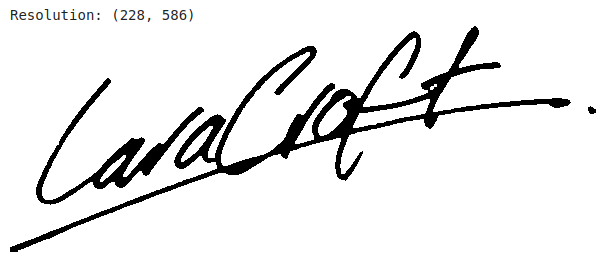
**Erosion:**

****

**Opening (Erosion, then Dilation)**

****

**Closing (Dilation, then Erosion)**



**Task 2**

Code:

#Libraries Used

import cv2

from PIL import Image, ImageChops

import numpy as np

from google.colab.patches import cv2\_imshow

#Reading Image File

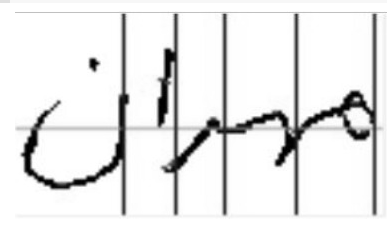
im=cv2.imread("/content/mehran.jpg")

#Converting the image to greyscale

im=cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

#showing the image

cv2\_imshow(im)



resolution=im.shape

print("Resolution:",resolution)

im=np.array(im)

kernal = np.ones((5,5)) #using a kernal of 5x5 to erode the image

kernal1 = np.ones((11,11)) #using a kernal of 11x11 to dilate the image

#First converting the image into a binary image

#intensities less than 125 are mapped to zero

#intensities of more than 125 are mapped to 255

for x in range(0,resolution[0]):

for y in range(0,resolution[1]):

if im[x][y]<125:

im[x][y]=255

else:

im[x][y]=0

im=cv2.erode(im,kernal)

im=cv2.dilate(im,kernal1)

#inverting the pixel to get final results

for x in range(0,resolution[0]):

for y in range(0,resolution[1]):

if im[x][y]>125:

im[x][y]=0

else:

im[x][y]=255

im=Image.fromarray(im)

im

**Output Image**

