college-assignment (/github/sachi1406/college-assignment/tree/main)
/
IP_Expt_7_Morphology_Sachi_Shah.ipynb (/github/sachi1406/college-assignment/tree/main/IP_Expt_7_Morphology_Sachi_Shah.ipynb)

Open in Colab
(https://colab.research.google.com/github/sachi1406/college-assignment/blob/main/IP_Expt_7_Morphology_Sachi_Shah.ipynb)

Experiment No. 7 : Morphological Operations

- Name: Sachi Shah
- Roll No.: C094
- Batch: EC1
- Sap Id: 70321018081

Aim :

1. Perform erosion on the given test image using a square structuring element of size 11x11,15x15, 45x45.
2. Perform dilation on the given test image with broken text characters using a suitable structuring element to show that the broken segments were joined.
3. Oserve the effect of opening and closing on the given test images. Analyze and compare the effect of structuring elements of different sizes on the test images.

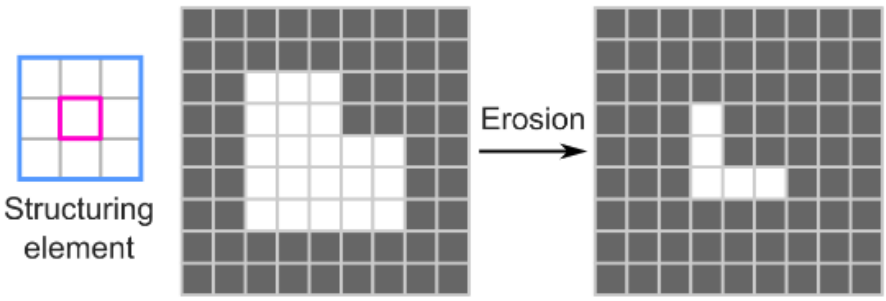
Import Libraries

In [1]:

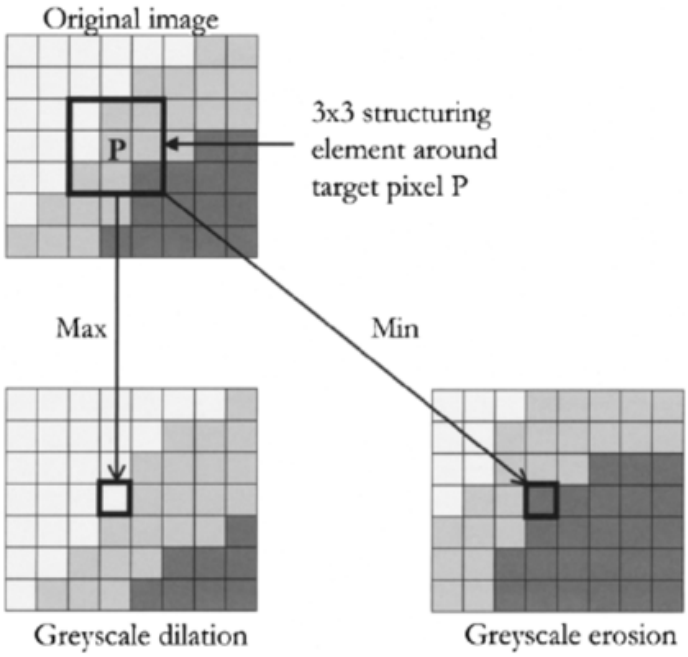
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Erosion :

In the erosion operation, if for EVERY pixel that is true (1 / shaded) in the structuring element, the corresponding pixel in the image is also true (1/shaded), then the output pixel is set to true (1/shaded). Otherwise, the output pixel is set to false (0/unshaded).



Erosion and Dilation for of Grey scale image



Python code for performing erosion

In [19]:

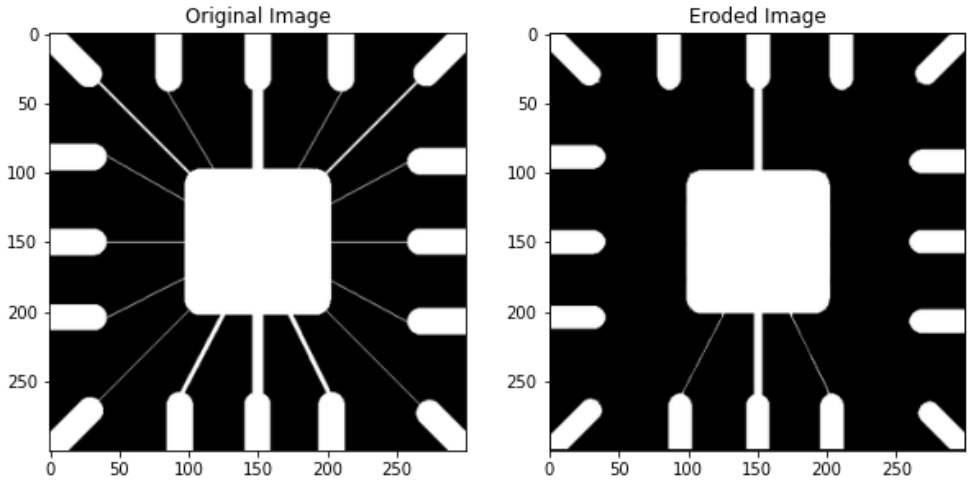
```
#Read the image for erosion squares.tif
img1 = cv2.imread('Image_for_erosion.png',0)
```

In [20]:

```
#Define the square structuring element (SE)
#k=11, 15, 45 - different sizes of the structuring element
r,c = img1.shape
erode1 = np.zeros((r,c) , dtype = int)
for i in range(1,r-1):
    for j in range(1,c-1):
        temp = [img1[i-1,j-1] , img1[i-1,j] , img1[i-1,j+1] ,
                img1[i,j-1] , img1[i,j] , img1[i,j+1] ,
                 img1[i+1,j-1] , img1[i+1,j] , img1[i+1,j+1]]
        erode1[i,j] = min(temp)

plt.figure(figsize = (10,10))

plt.subplot(121)
plt.imshow(img1,cmap='gray')
plt.title('Original Image')

plt.subplot(122)
plt.imshow(erode1,cmap='gray')
plt.title('Eroded Image')
```

Out[20]:

```
Text(0.5, 1.0, 'Eroded Image')
```

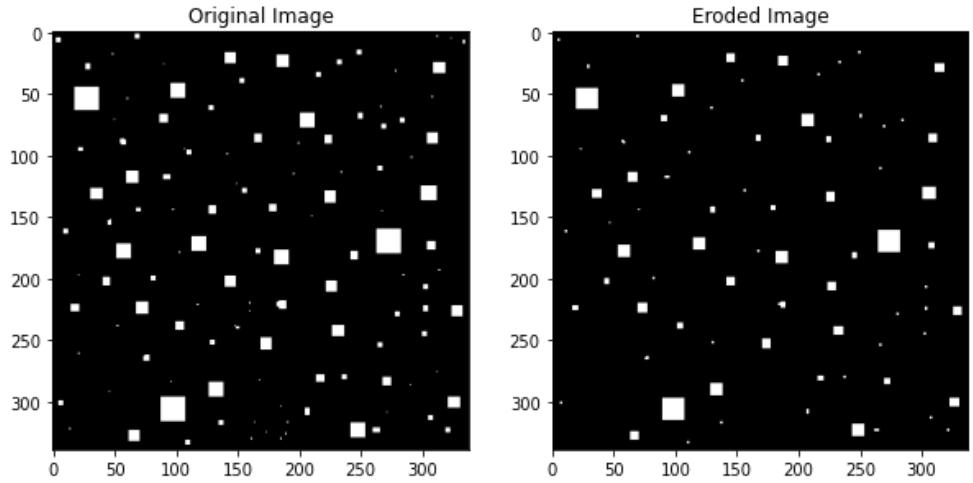


In [11]:

```
img2 = cv2.imread('squares.tif',0)
r,c = img2.shape
erode2 = np.zeros((r,c) , dtype = int)
for i in range(1,r-1):
    for j in range(1,c-1):
        temp = [img2[i-1,j-1] , img2[i-1,j] , img2[i-1,j+1] ,
                img2[i,j-1] , img2[i,j] , img2[i,j+1] ,
                 img2[i+1,j-1] , img2[i+1,j] , img2[i+1,j+1]]
        erode2[i,j] = min(temp)

plt.figure(figsize = (10,10))

plt.subplot(121)
plt.imshow(img2,cmap='gray')
plt.title('Original Image')

plt.subplot(122)
plt.imshow(erode2,cmap='gray')
plt.title('Eroded Image')
```

Out[11]:

```
Text(0.5, 1.0, 'Eroded Image')
```

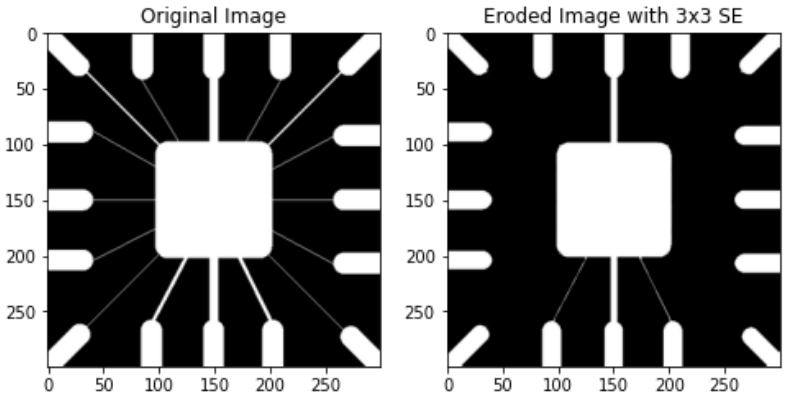


In [22]:

```
#Erosion using the built-in function for morphology
SE = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))

eroded_img = cv2.erode(img1,SE)

plt.figure(figsize = (8,8))

plt.subplot(121)
plt.imshow(img1,cmap='gray')
plt.title('Original Image')

plt.subplot(122)
plt.imshow(eroded_img,cmap='gray')
plt.title('Eroded Image with 3x3 SE')
```

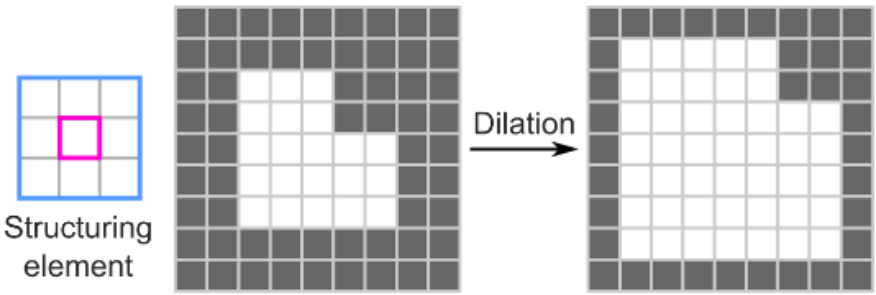
Out[22]:

```
Text(0.5, 1.0, 'Eroded Image with 3x3 SE')
```



Dilation

In the dilation operation, if for ANY pixel that is true (1/shaded) in the structuring element, the corresponding pixel in the image underneath is also true (1/shaded) , then the output pixel is set to true. Otherwise, the output pixel is set to false.



In [12]:

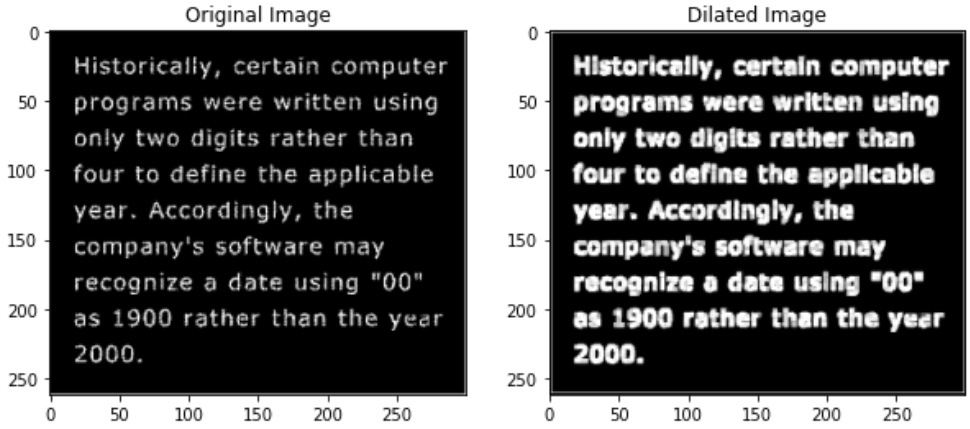
```
#Read the image and find the size of the image
img1_dilation = cv2.imread('Image_for_dilation.png',0)
r,c = img1_dilation.shape
```

In [14]:

```
#Implement the code for dilating the image
dilate1 = np.zeros((r,c) , dtype = int)
for i in range(1,r-1):
    for j in range(1,c-1):
        temp = [img1_dilation[i-1,j-1] , img1_dilation[i-1,j] , img1_dilation[i-1,j+1] ,
                img1_dilation[i,j-1] , img1_dilation[i,j] , img1_dilation[i,j+1] ,
                 img1_dilation[i+1,j-1] , img1_dilation[i+1,j] , img1_dilation[i+1,j+1]]
        dilate1[i,j] = max(temp)

plt.figure(figsize = (10,10))

plt.subplot(121)
plt.imshow(img1_dilation,cmap='gray')
plt.title('Original Image')

plt.subplot(122)
plt.imshow(dilate1,cmap='gray')
plt.title('Dilated Image')
```

Out[14]:

```
Text(0.5, 1.0, 'Dilated Image')
```



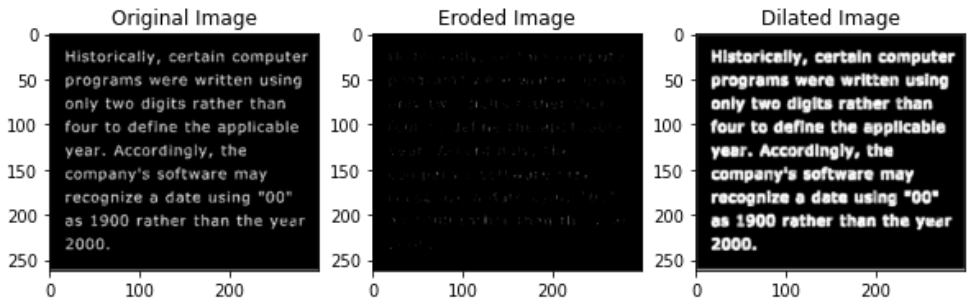
Erosion & Dilation using built in function

In [15]:

```
#Erosion using CV2 built in function to obtain structuring element and perform erosion
img5 = cv2.imread('Image_for_dilation.png',0)
SE = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))

E = cv2.erode(img5,SE)
D = cv2.dilate(img5,SE)

#Plot the original image, image obtained by hard coding and image obtained by built in function
plt.figure(figsize = (10,10))

plt.subplot(131)
plt.imshow(img5,cmap='gray')
plt.title('Original Image')

plt.subplot(132)
plt.imshow(E,cmap='gray')
plt.title('Eroded Image')

plt.subplot(133)
plt.imshow(D,cmap='gray')
plt.title('Dilated Image')
```

Out[15]:

```
Text(0.5, 1.0, 'Dilated Image')
```



Effect of different sizes of structuring element

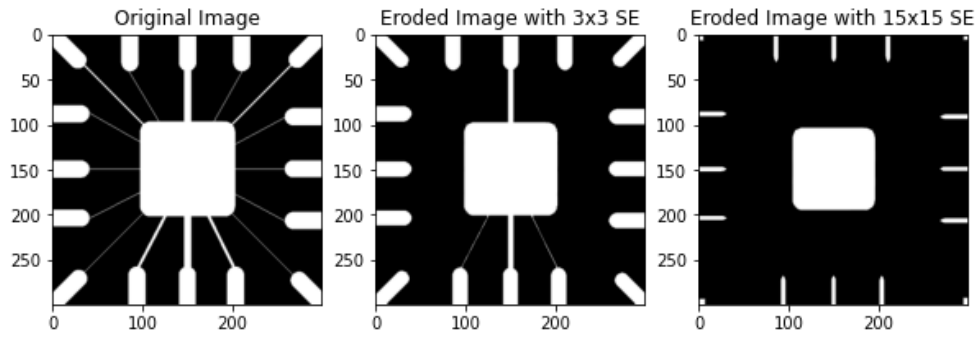
For Erosion

In [16]:

```
#Erosion using CV2 built in function to obtain structuring element and perform erosion for different size o
img6 = cv2.imread('Image_for_erosion.png',0)

SE1 = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))
E1 = cv2.erode(img6,SE1)

SE2 = cv2.getStructuringElement(cv2.MORPH_RECT,(15,15))
E2 = cv2.erode(img6,SE2)

#Plot the images
plt.figure(figsize = (10,10))

plt.subplot(131)
plt.imshow(img6,cmap='gray')
plt.title('Original Image')

plt.subplot(132)
plt.imshow(E1,cmap='gray')
plt.title('Eroded Image with 3x3 SE')

plt.subplot(133)
plt.imshow(E2,cmap='gray')
plt.title('Eroded Image with 15x15 SE')
```

Out[16]:

```
Text(0.5, 1.0, 'Eroded Image with 15x15 SE')
```



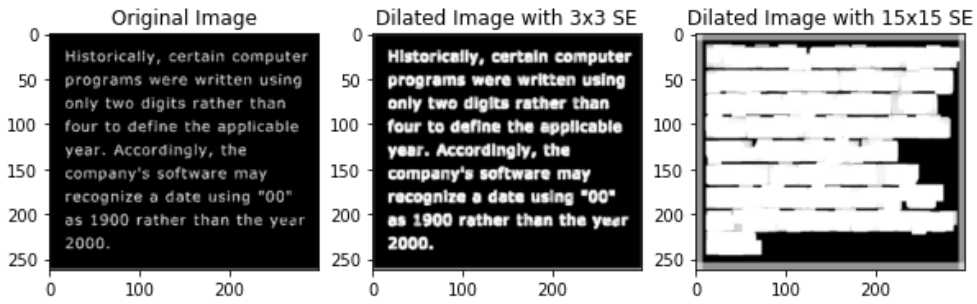
For Dilation

In [25]:

```
#Dilation using CV2 built in function to obtain structuring element and perform erosion for different size
img7 = cv2.imread('Image_for_dilation.png',0)

SE1 = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))
D1 = cv2.dilate(img7,SE1)

SE2 = cv2.getStructuringElement(cv2.MORPH_RECT,(15,15))
D2 = cv2.dilate(img7,SE2)

#Plot the images
plt.figure(figsize = (10,10))

plt.subplot(131)
plt.imshow(img7,cmap='gray')
plt.title('Original Image')

plt.subplot(132)
plt.imshow(D1,cmap='gray')
plt.title('Dilated Image with 3x3 SE')

plt.subplot(133)
plt.imshow(D2,cmap='gray')
plt.title('Dilated Image with 15x15 SE')
```

Out[25]:

```
Text(0.5, 1.0, 'Dilated Image with 15x15 SE')
```



Students should observe the effect of different size of structuring element and comment on the same

1. When Dilation and Erosion is applied on given text image, Dilation works better as it thickens the white pixel and improves readability.
2. When erosion is applied using different SE, erosion using a large SE gives more thining effect then small SE.
3. Larger the size of SE, more is the erosion effect.(also dilation effect)

Morphological opening

Morphological opening of an image is basically erosion followed by dilation: A∘B=OPEN(A,B)=D(E(A))

In the example below, application of opening and closing is shown to filter the noise in the given fingerprint image. First the image is opened using erosion followed by dilation operation using the given structuring element. Then on the opened image, closing operation of dilation followed by erosion operation is performed.

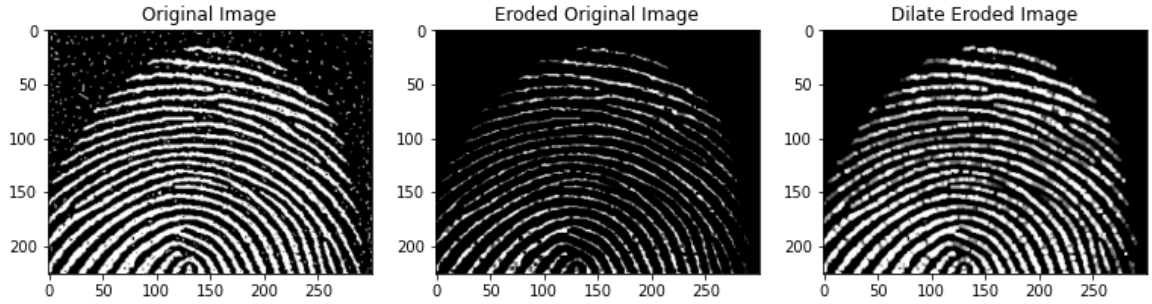
It is seen that after applying the opening followed by closing operation, the noise on the finger print image is filtered.

In [29]:

```
#Read the image for opening
img4_opening = cv2.imread('noisy_fingerprint.png',0)

#Define the structuring element using inbuilt CV2 function
SE = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))

#Erode the image
O1  = cv2.erode(img4_opening,SE)

#Dilate the eroded image.
O2  = cv2.dilate(O1,SE)

#Plot all the images
plt.figure(figsize = (13,13))

plt.subplot(131)
plt.imshow(img4_opening,cmap='gray')
plt.title('Original Image')

plt.subplot(132)
plt.imshow(O1,cmap='gray')
plt.title('Eroded Original Image')

plt.subplot(133)
plt.imshow(O2,cmap='gray')
plt.title('Dilate Eroded Image')
```

Out[29]:

```
Text(0.5, 1.0, 'Dilate Eroded Image')
```



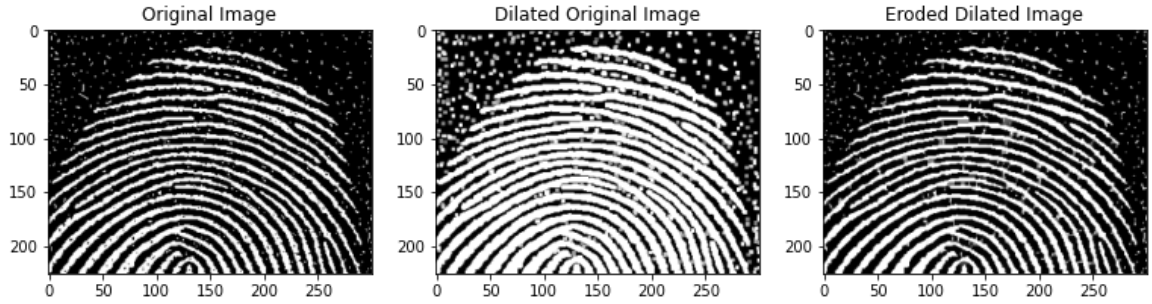
Morphological closing

In [28]:

```
#Read the image for closing
img4_closing = cv2.imread('noisy_fingerprint.png',0)

#Define the structuring element using inbuilt CV2 function
SE = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))

#Dilate the image
C1  = cv2.dilate(img4_closing,SE)

#Erode the dilated image.
C2  = cv2.erode(C1,SE)

#Plot all the images
plt.figure(figsize = (13,13))

plt.subplot(131)
plt.imshow(img4_closing,cmap='gray')
plt.title('Original Image')

plt.subplot(132)
plt.imshow(C1,cmap='gray')
plt.title('Dilated Original Image')

plt.subplot(133)
plt.imshow(C2,cmap='gray')
plt.title('Eroded Dilated Image')
```

Out[28]:

```
Text(0.5, 1.0, 'Eroded Dilated Image')
```

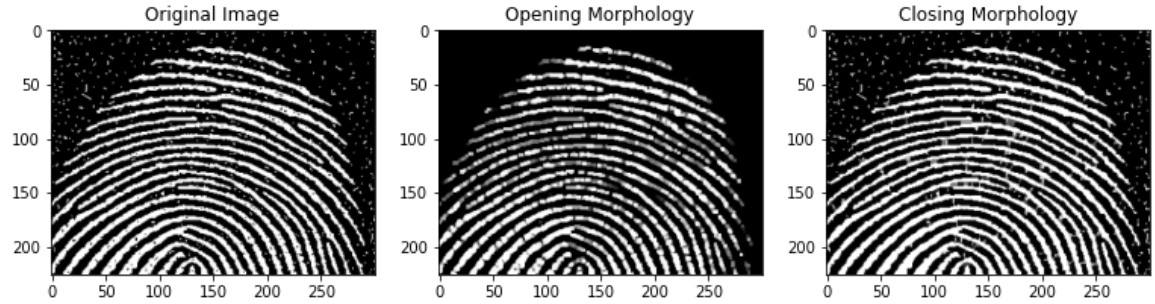


In [31]:

```
#Comparing the results for Opening and Closing
plt.figure(figsize = (13,13))

plt.subplot(131)
plt.imshow(img4_opening,cmap='gray')
plt.title('Original Image')

plt.subplot(132)
plt.imshow(O2,cmap='gray')
plt.title('Opening Morphology')

plt.subplot(133)
plt.imshow(C2,cmap='gray')
plt.title('Closing Morphology')
```

Out[31]:

```
Text(0.5, 1.0, 'Closing Morphology')
```



Comment: For the given fingerprint image, opening morphological operation is the appropriate choice.

Conclusion :

1. We implemented code for Erosion, Dilation, Closing and Opening.
2. It was observed that:
   a. Erosion deletes a layer of pixels from the boundary of the object. Erosion provides one level of thinning to the object in the image.
   b. Dilation adds a layer of pixel to the boundary of the object. Dilation provides one level of thickening to the object in the image. Moreover, larger the size of SE, more is the Erosion/Dilation.
3. Opening = Erosion followed by dilation
4. Closing = Dilation followed by erosion
5. Opening helps in filling slots/holes while retaining Bridges as seen from results of fingerprint image.
6. Closing retains all holes/white slots & breaks bridges.
   Hence, opening is suitable choice for fingerprint detection.