Implementation-of-Filters

Aim:

To implement filters for smoothing and sharpening the images in the spatial domain.

Software Required:

Anaconda - Python 3.7

ALGORITHM:

Step 1:

Import the necessary modules.

Step 2:

Perform smoothing operation on a image.

- Average filter(Box Filter)
- Weighted average filter
- Gaussian Blur
- Median filter

Step 3:

Perform sharpening on a image.

- Laplacian Kernel
- Laplacian Operator

Step 4:

Display all the images with their respective filters.

Program:

Developed By: **Shafeeq Ahamed. S**

Register Number: **212221230092**

1. Smoothing Filters

i) Using Averaging Filter

```
kernal = np.ones((11,11),np.float32)/121
img_box_filter = cv2.filter2D(img_rgb,-1,kernal)

cv2.imshow("Box Filter",img_box_filter)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.title("Box")
plt.imshow(img_box_filter)
plt.show()
```

ii) Using Weighted Averaging Filter

```
kernal_weighted_avg = np.array([[2,2,2],[4,8,4],[2,4,2]])/15
img_w_avg_filter = cv2.filter2D(img_rgb,-1,kernal_weighted_avg)

cv2.imshow("Weighted Avg Filter",img_w_avg_filter)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.title("W-Avg")
plt.imshow(img_w_avg_filter)
plt.show()
```

iii) Using Gaussian Filter

```
img_gaussian = cv2.GaussianBlur(src = img_rgb, ksize = (11,11), sigmaX=0,sigmaY=0)

cv2.imshow("Gaussian Filter",img_gaussian)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.title("Gaussian")
plt.imshow(img_gaussian)
plt.show()
```

iv) Using Median Filter

```
img_median = cv2.medianBlur(src = img_rgb, ksize = 7)

cv2.imshow("Median Filter",img_median)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.title("Median")
plt.imshow(img_median)
plt.show()
```

2. Sharpening Filters

i) Using Laplacian Kernal

```
kernal_Laplacian = np.array([[1,2,1],[1,-5,1],[2,1,0]])
img_laplacian_kernal = cv2.filter2D(img_rgb,-1,kernal_Laplacian)

cv2.imshow("Laplacian Filter",img_laplacian_kernal)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.title("Laplacian Kernal")
plt.imshow(img_laplacian_kernal)
plt.show()
```

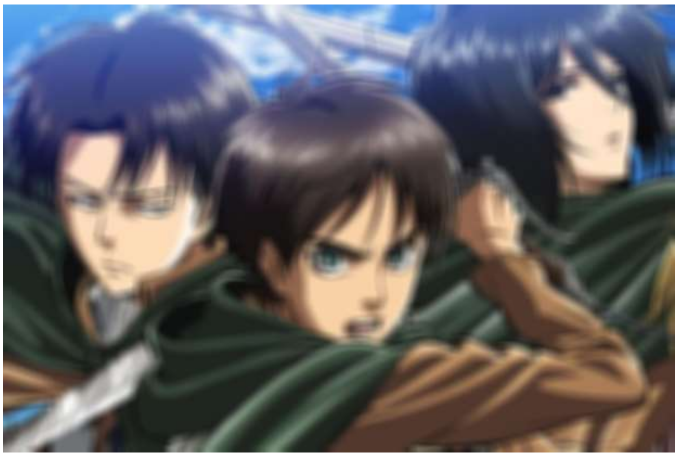
ii) Using Laplacian Operator

```
img_laplacian = cv2.Laplacian(img_rgb,cv2.CV_64F)

cv2.imshow("Laplacian",img_laplacian)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.title("Laplacian Operator")
plt.imshow(img_laplacian)
plt.show()
```

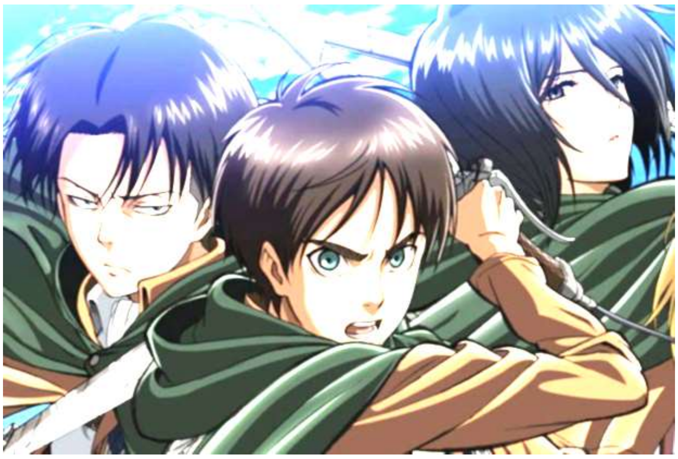
OUTPUT:

1. Smoothing Filters

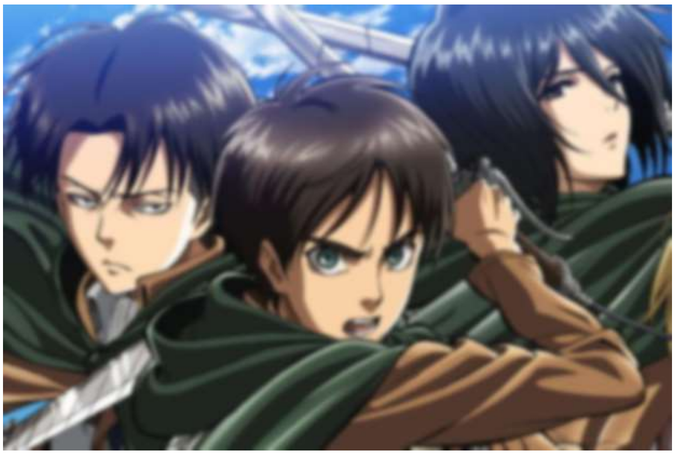
i) Using Averaging Filter

Original	Averaging Filter
	

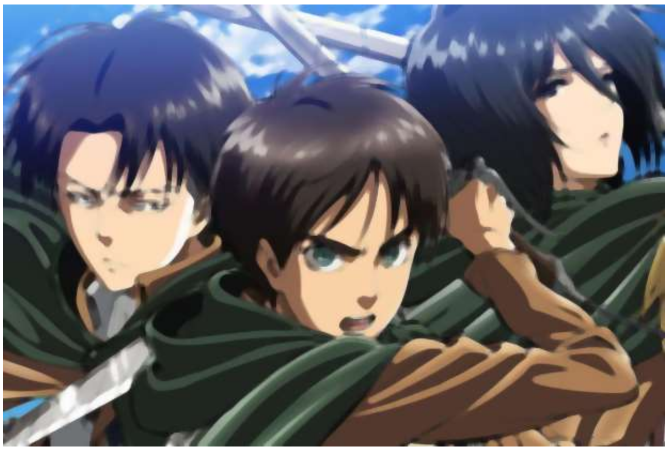
ii) Using Weighted Averaging Filter

Original	Weighted Averaging Filter
	

iii) Using Gaussian Filter

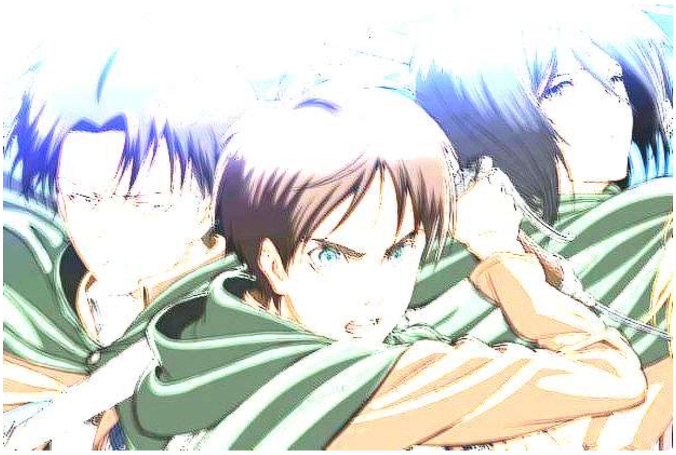
Original	Gaussian Filter
	

iv) Using Median Filter

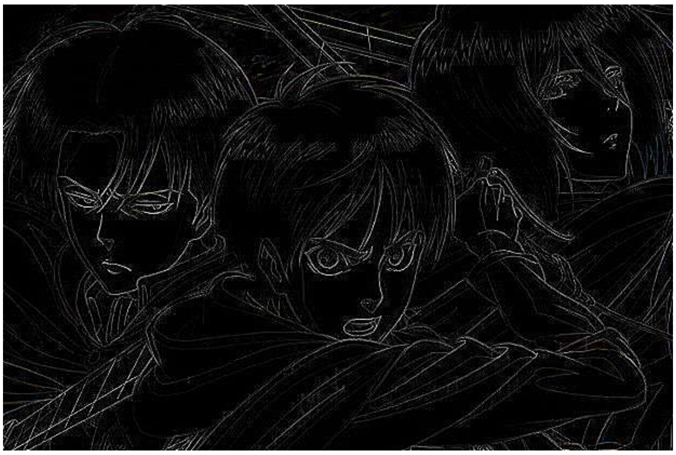
Original	Median Filter
	

2. Sharpening Filters

i) Using Laplacian Kernal

Original	Laplacian Kernal
	

ii) Using Laplacian Operator

Original	Laplacian Operator
	

Result:

Thus the filters are designed for smoothing and sharpening the images in the spatial domain.