

8. Perform transformation using the OpenCV library.

```
In [23]: import cv2
import numpy as np
import matplotlib.pyplot as plt
import math as mp
img=cv2.imread("edge.png",0)
img=cv2.resize(img,(400,400))
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
w,h=img.shape[:2]
```

```
In [22]: translation_mat=np.array([[1,0,100],[0,1,50]],dtype=np.float32)

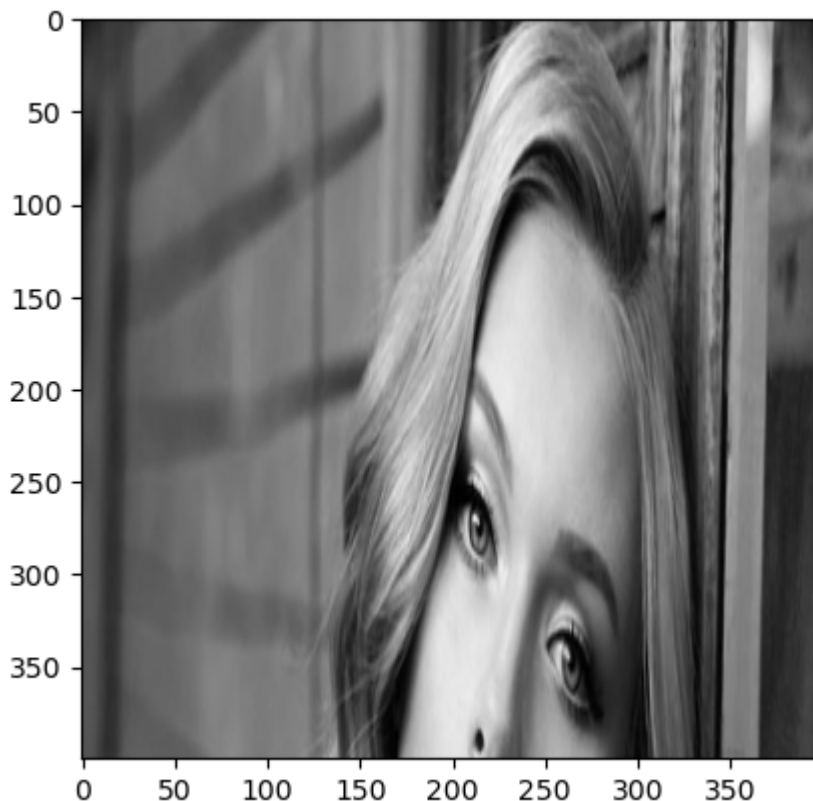
trans=cv2.warpAffine(img,translation_mat,(h,w))
```

```
In [21]: angle=30
scale=1
center=(h//2,w//2)
rotation_matrix = cv2.getRotationMatrix2D(center, angle, scale)
rotate = cv2.warpAffine(img, rotation_matrix, (h,w))
```

```
In [25]: scaling_mat=np.array([[1,0,1],[0,2,1]],dtype=np.float32)

scale=cv2.warpAffine(img,scaling_mat,(h,w))
```

Out[25]: <matplotlib.image.AxesImage at 0x1080fe85e10>



```
In [18]: shearing_mat=np.array([[1,0.2,0],[0.5,1,0]],dtype=np.float32)

shear=cv2.warpAffine(img,shearing_mat,(h,w))
```

```
In [33]: rf_x_mat=np.array([[1,0,0],[0,-1,w]],dtype=np.float32)

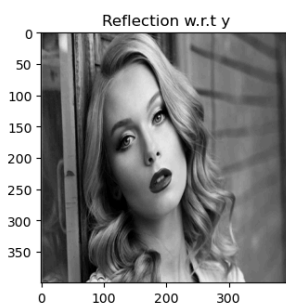
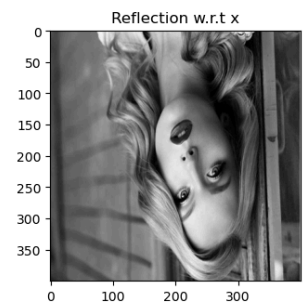
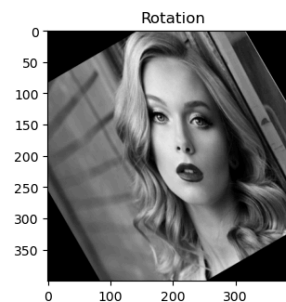
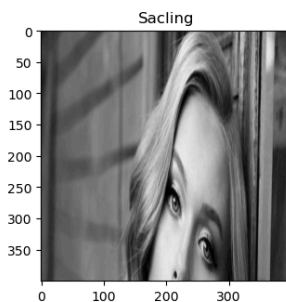
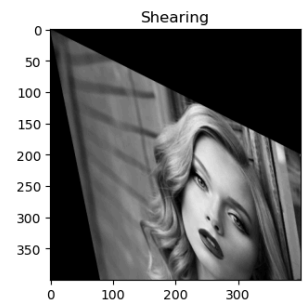
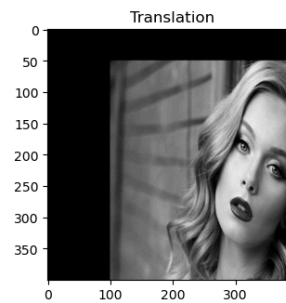
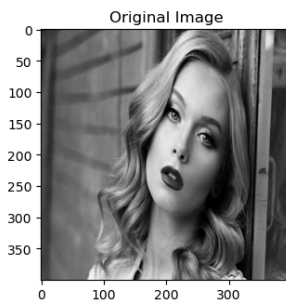
reflect_x=cv2.warpAffine(img,rf_x_mat,(h,w))
```

```
In [34]: rf_y_mat=np.array([[ -1,0,h],[0,1,0]],dtype=np.float32)

reflect_y=cv2.warpAffine(img,rf_y_mat,(h,w))
```

```
In [36]: plt.figure(figsize=(15,10))
plt.subplot(3, 3, 1), plt.imshow(img, cmap='gray'), plt.title('Original Image')
plt.subplot(3, 3, 2), plt.imshow(trans, cmap='gray'), plt.title('Translation')
plt.subplot(3, 3, 3), plt.imshow(shear, cmap='gray'), plt.title('Shearing')
plt.subplot(3, 3, 4), plt.imshow(scale, cmap='gray'), plt.title('Sacling')
plt.subplot(3, 3, 5), plt.imshow(rotate, cmap='gray'), plt.title('Rotation')
plt.subplot(3, 3, 6), plt.imshow(reflect_x, cmap='gray'), plt.title('Reflection w.r
plt.subplot(3, 3, 7), plt.imshow(reflect_y, cmap='gray'), plt.title('Reflection w.r

plt.tight_layout()
plt.show()
```



```
In [ ]:
```