Source Code :

```
# scaling
import cv2
import numpy as np

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

res = cv2.resize(img,None,fx=2, fy=2, interpolation = cv2.INTER_CUBIC)

#OR

height, width = img.shape[:2]
res = cv2.resize(img,(2*width, 2*height), interpolation = cv2.INTER_CUBIC)
cv2.imwrite('scaling_output.jpg', res)

#-------------------------------------------------------------------------------#
# Translation

import cv2
import numpy as np

img = cv2.imread('lord_hanuman_angry_image.jpg',0)
rows,cols = img.shape

M = np.float32([[1,0,100],[0,1,50]])
dst = cv2.warpAffine(img,M,(cols,rows))

cv2.imshow('img',dst)
cv2.waitKey(0)
cv2.destroyAllWindows()

#-----------------------------------------------------------------#
# Rotation
import cv2

img = cv2.imread('lord_hanuman_angry_image.jpg',0)
rows,cols = img.shape

M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
dst = cv2.warpAffine(img,M,(cols,rows))
cv2.imshow('Rotation Result', dst)
cv2.waitKey(0)
cv2.destroyAllWindows()

#-----------------------------------------------------------#
# Affine Transformation
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

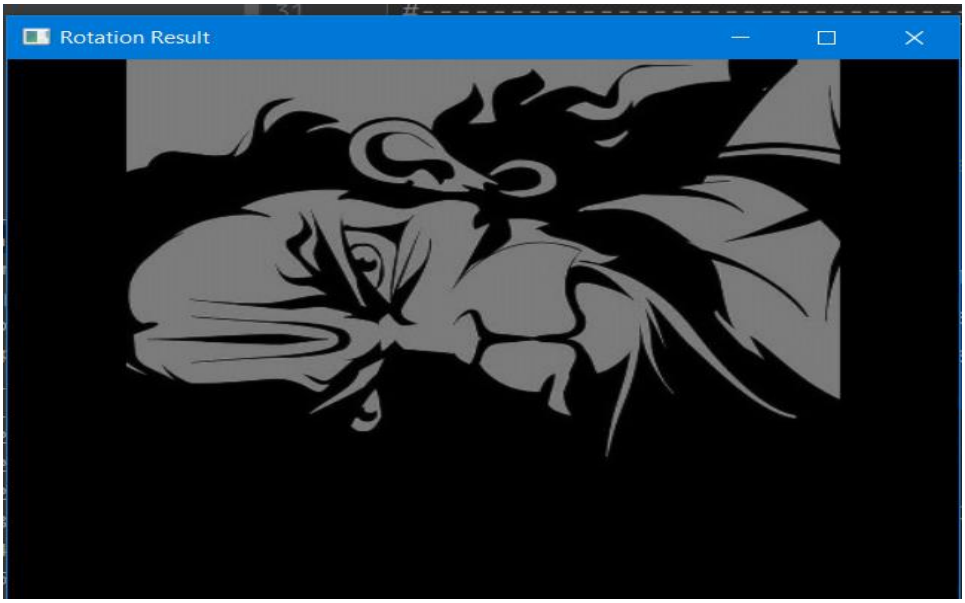
```
img = cv2.imread('lord_hanuman_angry_image.jpg')
rows,cols,ch = img.shape

pts1 = np.float32([[50,50],[200,50],[50,200]])
pts2 = np.float32([[10,100],[200,50],[100,250]])

M = cv2.getAffineTransform(pts1,pts2)

dst = cv2.warpAffine(img,M,(cols,rows))

plt.subplot(121),plt.imshow(img),plt.title('Input')
plt.subplot(122),plt.imshow(dst),plt.title('Output')
plt.show()


#------------------------------------------------------------------------------#
# perspective transformation
import cv2
import numpy as np
import matplotlib.pyplot as plt
img = cv2.imread('lord_hanuman_angry_image.jpg')
rows,cols,ch = img.shape

pts1 = np.float32([[56,65],[368,52],[28,387],[389,390]])
pts2 = np.float32([[0,0],[300,0],[0,300],[300,300]])

M = cv2.getPerspectiveTransform(pts1,pts2)

dst = cv2.warpPerspective(img,M,(300,300))

plt.subplot(121),plt.imshow(img),plt.title('Input')
plt.subplot(122),plt.imshow(dst),plt.title('Output')
plt.show()
```

Output :

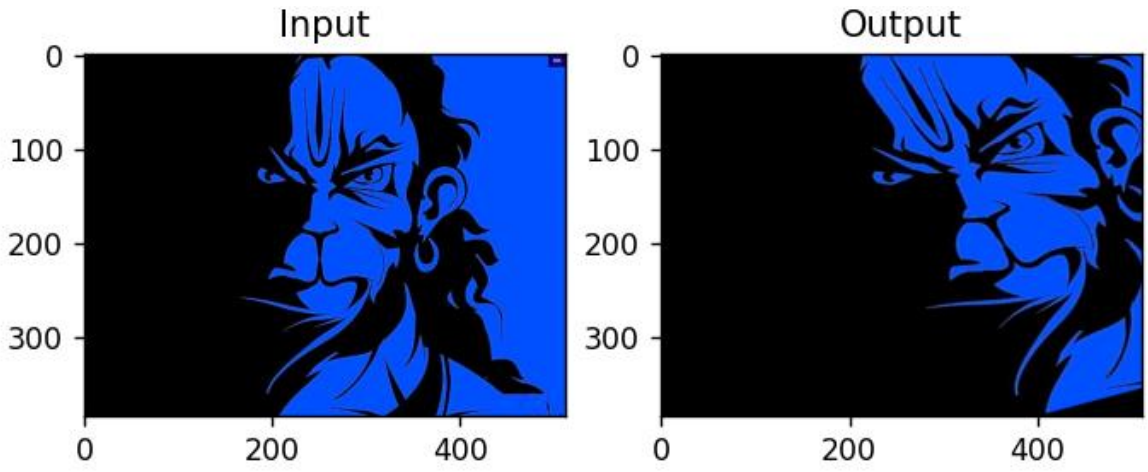
Scaling : input **Output**

Translation

Rotation



Affine Transformation :



Perspective transformation: