

Image-Transformation

AIM:

To perform image transformation such as Translation, Scaling, Shearing, Reflection, Rotation and Cropping using OpenCV and Python.

SOFTWARE REQUIRED:

Anaconda - Python 3.7

ALGORITHM:

Step 1:

Import the necessary libraries and read the original image and save it as a image variable.

Step 2:

Translate the image.

Step 3:

Scale the image.

Step 4:

Shear the image.

Step 5:

Reflect of image.

Step 6:

Rotate the image & Crop the image.

Step 7:

Display all the Transformed images.

Program:

Developed By: **Shafeeq Ahamed. S**

Register Number: **212221230092**

i)Image Translation

```
M=np.float32([[1,0,200],
              [0,1,250],
              [0,0,1]])

translated_img=cv2.warpPerspective(input_img,M,(cols,rows))

plt.axis('off')
plt.imshow(translated_img)
plt.show()
```

ii) Image Scaling

```
M=np.float32([[1.5,0,0],
              [0,1.5,0],
              [0,0,1]])

scaled_img=cv2.warpPerspective(input_img,M,(cols,rows))

plt.axis('off')
plt.imshow(scaled_img)
plt.show()
```

iii)Image shearing

```
M_x=np.float32([[1,0.2,0],
                [0.2,1,0],
                [0,0,1]])

sheared_img = cv2.warpPerspective(input_img,M_x,(cols,rows))

plt.axis('off')
plt.imshow(sheared_img)
plt.show()
```

iv)Image Reflection

```
M_x=np.float32([[1,0,0],
                [0,-1,rows],
                [0,0,1]])

reflected_img_xaxis=cv2.warpPerspective(input_img,M_x,(cols,rows))

plt.axis('off')
plt.imshow(reflected_img_xaxis)
plt.show()

M_x=np.float32([[-1,0,cols],
                [0,1,0],
                [0,0,1]])

reflected_img_yaxis=cv2.warpPerspective(input_img,M_x,(cols,rows))

plt.axis('off')
plt.imshow(reflected_img_xaxis)
plt.show()
```

v)Image Rotation

```
angle1=np.radians(5)
angle2=np.radians(35)
angle3=np.radians(5)
angle4=np.radians(45)
M=np.float32([[np.cos(angle1),-(np.sin(angle2)),0],
              [np.sin(angle3),np.cos(angle4),0],
              [0,0,1]])
rotated_img=cv2.warpPerspective(input_img,M,(cols,rows))

plt.axis('off')
plt.imshow(rotated_img)
plt.show()
```

vi)Image Cropping

```
cropped_img=input_img[150:1500,1000:3000]

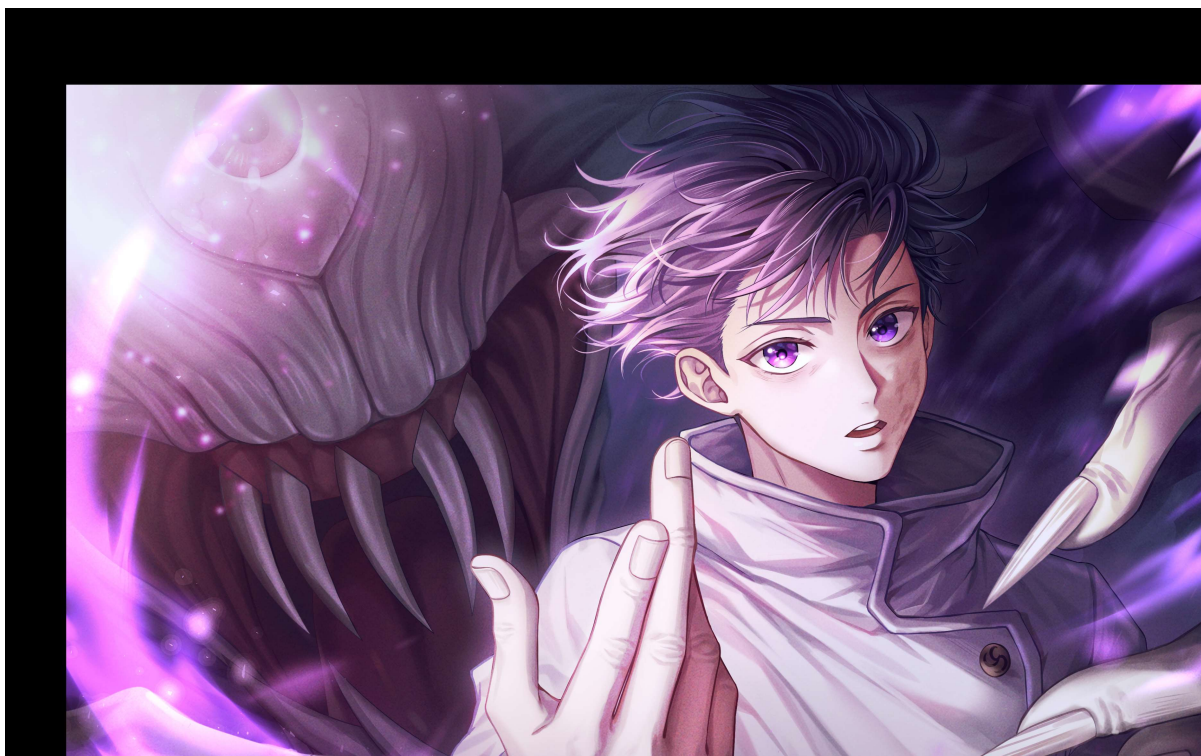
plt.axis('off')
plt.imshow(cropped_img)
plt.show()
```

Output:

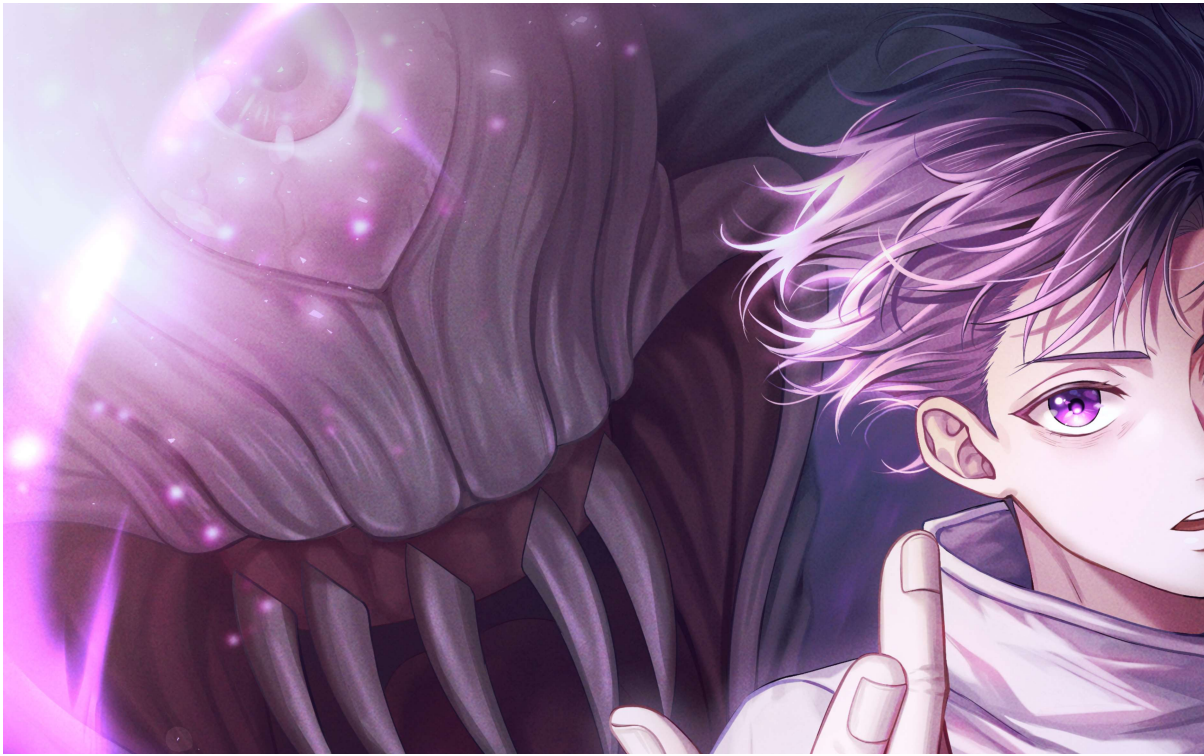
Original Image



i)Image Translation



ii) Image Scaling



iii) Image shearing



iv)Image Reflection



v)Image Rotation



vi)Image Cropping



Result:

Thus the different image transformations such as Translation, Scaling, Shearing, Reflection, Rotation and Cropping are done using OpenCV and python programming.