

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

image_path = '/content/cvlab1img.jpg'
image = cv2.imread(image_path)

if image is None:
    print("Image not found! Please check the path.")
    exit()

image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(12, 8))
plt.subplot(241)
plt.imshow(image_rgb)
plt.title("Original Image")
plt.axis('off')

# 1. Image Translation
rows, cols = image.shape[:2]
translation_matrix = np.float32([[1, 0, 50], [0, 1, 50]]) # Shift by 50 pixels in both x and y
translated_image = cv2.warpAffine(image, translation_matrix, (cols, rows))
translated_image_rgb = cv2.cvtColor(translated_image, cv2.COLOR_BGR2RGB)

plt.subplot(242)
plt.imshow(translated_image_rgb)
plt.title("Translated Image")
plt.axis('off')

# 2. Reflection
reflected_image = cv2.flip(image, 1) # Flip horizontally
reflected_image_rgb = cv2.cvtColor(reflected_image, cv2.COLOR_BGR2RGB)

plt.subplot(243)
plt.imshow(reflected_image_rgb)
plt.title("Reflected Image")
plt.axis('off')

# 3. Rotation
center = (cols // 2, rows // 2)
rotation_matrix = cv2.getRotationMatrix2D(center, 45, 1) # Rotate 45 degrees
rotated_image = cv2.warpAffine(image, rotation_matrix, (cols, rows))
rotated_image_rgb = cv2.cvtColor(rotated_image, cv2.COLOR_BGR2RGB)

plt.subplot(244)
plt.imshow(rotated_image_rgb)
plt.title("Rotated Image")
plt.axis('off')

# 4. Scaling
scaled_image = cv2.resize(image, None, fx=1.5, fy=1.5, interpolation=cv2.INTER_LINEAR)
scaled_image_rgb = cv2.cvtColor(scaled_image, cv2.COLOR_BGR2RGB)

plt.subplot(245)
plt.imshow(scaled_image_rgb)
plt.title("Scaled Image")
plt.axis('off')

# 5. Cropping
cropped_image = image[50:200, 50:200] # Crop a specific region
cropped_image_rgb = cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB)

plt.subplot(246)
plt.imshow(cropped_image_rgb)
plt.title("Cropped Image")
plt.axis('off')

# 6. Shearing in x-axis
shear_matrix_x = np.float32([[1, 0.5, 0], [0, 1, 0]])
sheared_image_x = cv2.warpAffine(image, shear_matrix_x, (int(cols * 1.5), rows))
sheared_image_x_rgb = cv2.cvtColor(sheared_image_x, cv2.COLOR_BGR2RGB)

plt.subplot(247)
plt.imshow(sheared_image_x_rgb)
plt.title("Sheared X")
```

```
plt.axis('off')

# 7. Shearing in y-axis
shear_matrix_y = np.float32([[1, 0, 0], [0.5, 1, 0]])
sheared_image_y = cv2.warpAffine(image, shear_matrix_y, (cols, int(rows * 1.5)))
sheared_image_y_rgb = cv2.cvtColor(sheared_image_y, cv2.COLOR_BGR2RGB)

plt.subplot(248)
plt.imshow(sheared_image_y_rgb)
plt.title("Sheared Y")
plt.axis('off')

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



Original Image



Translated Image



Reflected Image



Rotated Image



Scaled Image



Cropped Image



Sheared X



Sheared Y

