

```

# Install required libraries if not already installed
!pip install opencv-python pillow matplotlib

# Import necessary libraries
import cv2
import numpy as np
from google.colab import files
from PIL import Image
import matplotlib.pyplot as plt

# Step 1: Upload the image
uploaded = files.upload()

# Open the image using PIL and convert it to a format OpenCV understands
image_path = list(uploaded.keys())[0]
image = Image.open(image_path)
image_cv = np.array(image)

# Convert from RGB to BGR (since OpenCV uses BGR by default)
image_cv = cv2.cvtColor(image_cv, cv2.COLOR_RGB2BGR)

# Step 2: Select the 4 points in the original image (you can adjust these as needed)
# Example points (e.g., top-left, top-right, bottom-right, bottom-left)
pts1 = np.float32([[100, 100], [500, 100], [500, 500], [100, 500]])

# Define the destination points for the perspective transformation (the shape will be a rectangle)
pts2 = np.float32([[0, 0], [400, 0], [400, 400], [0, 400]])

# Step 3: Get the perspective transformation matrix
matrix = cv2.getPerspectiveTransform(pts1, pts2)

# Step 4: Apply the perspective transformation
transformed_image = cv2.warpPerspective(image_cv, matrix, (400, 400))

# Step 5: Convert the transformed image back to RGB for displaying with matplotlib
transformed_image_rgb = cv2.cvtColor(transformed_image, cv2.COLOR_BGR2RGB)

# Step 6: Display the original and transformed images
plt.figure(figsize=(10, 5))

# Display original image

```

```
plt.subplot(1, 2, 1)
plt.imshow(image)
plt.title("Original Image")
plt.axis('off')

# Display the transformed image
plt.subplot(1, 2, 2)
plt.imshow(transformed_image_rgb)
plt.title("Transformed Image")
plt.axis('off')

plt.show()

# Optional: Save and download the transformed image
cv2.imwrite("transformed_image.jpg", transformed_image)
files.download("transformed_image.jpg")
```

Original Image



Transformed Image

