# 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 an 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: Define the translation values (move the image by 100 pixels in the X direction and 50 pixels in the Y direction)

Tx = 100 # Shift along the X axis (horizontal move)

Ty = 50 # Shift along the Y axis (vertical move)

# Define the translation matrix

translation\_matrix = np.float32([[1, 0, Tx], [0, 1, Ty]])

# Step 3: Apply the translation using cv2.warpAffine

rows, cols, \_ = image\_cv.shape

translated\_image = cv2.warpAffine(image\_cv, translation\_matrix, (cols, rows))

# Step 4: Convert the translated image back to RGB for displaying with matplotlib

translated\_image\_rgb = cv2.cvtColor(translated\_image, cv2.COLOR\_BGR2RGB)

# Step 5: Display the original and translated images side by side

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 translated (moved) image

plt.subplot(1, 2, 2)

plt.imshow(translated\_image\_rgb)

plt.title("Translated Image")

plt.axis('off')

plt.show()

# Optional: Save and download the translated image

cv2.imwrite("translated\_image.jpg", translated\_image)

files.download("translated\_image.jpg")

