1. **Load and display an image using OpenCV and perform basic operations like resizing, cropping, and rotation using Matplotlib.**

**Title:**

Load and display an image using OpenCV and perform basic operations like resizing, cropping, and rotation using Matplotlib.

**Aim:**

To develop a Python application using OpenCV and Matplotlib to load an image and perform basic operations like displaying, resizing, cropping, and rotation.

**Software and Hardware Requirements:**

* Software: Python 3.x, OpenCV, Matplotlib
* Libraries: cv2, matplotlib, pyplot
* Hardware: PC with minimum 4GB RAM.

**Description:**

 **OpenCV** is an open-source computer vision and image processing library.

 **Matplotlib** is used to plot and visualize image data in Python.

 Image operations:

* **Reading**: cv2.imread()
* **Resizing**: cv2.resize()
* **Cropping**: Using NumPy array slicing
* **Rotation**: Using transformation matrix (cv2.getRotationMatrix2D and cv2.warpAffine)
* **Display**: Using matplotlib, pyplot, imshow().

**Source Code:**

**NOTE:**  
The source code provided below includes **clear and detailed explanations** for **each line**.  
**All comments are written beside or above the actual code lines** using # symbol to help you understand **what each line does**.

# Import the OpenCV library for image processing

import cv2

# Import pyplot from matplotlib for displaying images in notebook or GUI

from matplotlib import pyplot as plt

# ---------- Load the Image ----------

# Load the image file 'loki.jpg' into a variable called image

image = cv2.imread('loki.jpg') # Replace with the actual image file name

# Check if the image is loaded successfully

if image is None:

print ("Image not found. Make sure the image is in the same folder.")

exit () # Exit the program if image is not found

# Convert the image from BGR (OpenCV default) to RGB format (Matplotlib expects RGB)

image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

# ---------- Display the Original Image ----------

# Display the original image using matplotlib

plt.imshow(image) # Show the image

plt.title('Original Image') # Set title of the image window

plt.axis('off') # Hide axis for better visual

plt.show() # Show the image on screen

# ---------- Resize the Image ----------

# Resize the image to 300x300 pixels using cv2.resize()

resized = cv2.resize(image, (300, 300)) # Resize to width=300, height=300

# Display the resized image

plt.imshow(resized) # Show resized image

plt.title('Resized Image (300x300)') # Title for resized image

plt.axis('off') # Hide axis

plt.show() # Display the image

# ---------- Crop the Image ----------

# Crop the image using array slicing [y1:y2, x1:x2]

cropped = image[100:400, 100:400] # Crops the region starting from (100,100) to (400,400)

# Display the cropped portion

plt.imshow(cropped) # Show cropped image

plt.title('Cropped Image') # Title for cropped image

plt.axis('off') # Hide axis

plt.show() # Display the cropped image

# ---------- Rotate the Image ----------

# Get the dimensions of the original image

(h, w) = image.shape[:2] # h = height, w = width of the image

# Calculate the center point of the image for rotation

center = (w // 2, h // 2) # Find the center coordinates

# Create a rotation matrix to rotate the image by 45 degrees around its center

matrix = cv2.getRotationMatrix2D(center, 45, 1.0) # 45 degree angle, 1.0 is scaling factor

# Apply the rotation to the image using warpAffine

rotated = cv2.warpAffine(image, matrix, (w, h)) # Rotate image based on matrix

# Display the rotated image

plt.imshow(rotated) # Show rotated image

plt.title('Rotated Image (45 Degrees)') # Title for rotated image

plt.axis('off') # Hide axis

plt.show() # Show final image

**Output:**

* **Original Image** – Displays the input image.
* **Resized Image** – 300x300 pixels version.
* **Cropped Image** – Portion from center of the image.
* **Rotated Image** – Rotated by 45 degrees.

Each image is shown using **Matplotlib** with titles.

**Result:**

The image was successfully loaded, displayed, and basic operations like resizing, cropping, and rotation were performed using OpenCV and Matplotlib in Python.