**Lab 1- Loading Image Formats Tutorial**

**Aim:** Loading Different Image Formats for Computer Vision Tasks.

**Requirements:**

* Windows PC
* VS code/ Jupyter Notebook

**Procedure:**

**Step1 :-** Installation of Necessary Libraries

To follow along, ensure you have the following libraries installed:

* OpenCV (cv2)
* Pillow (PIL)
* imageio

You can install these libraries using pip:

**pip install opencv-python pillow imageio**

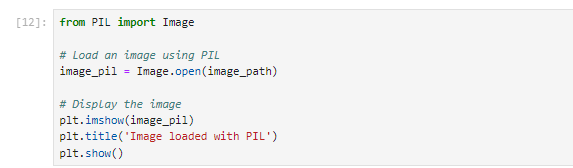
**Step2 :-** Loading Images with OpenCV

OpenCV is a powerful library for computer vision tasks. It reads images in BGR format by default.



**Step3 :-** Loading Images with PIL (Pillow)

PIL (Pillow) is a widely-used library for image processing in Python. It reads images in RGB format by default.





**Step4 :-** Loading Images with imageio

imageio is another versatile library for reading and writing images in various formats.





**Step5 :-** Handling Different Image Formats

These libraries can handle various image formats such as JPEG, PNG, BMP, and more. Let's load images of different formats using each library.













**Results:** Loading images of various formats using different Python libraries, including OpenCV, PIL (Pillow), and imageio has run successfully.

**Lab 2- Image Resizing, Cropping, and Rotation**

**Aim:** Image Resizing, Cropping, and Rotation: Adjusts the size, shape, and orientation of images.

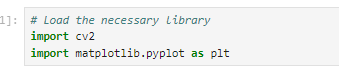
**Requirements:**

* Windows PC
* VS code/ Jupyter Notebook

**Procedure:**

**Step1 :-** Image resized

* Load necessary libraries



* Load an image



* Convert the image from BGR (OpenCV format) to RGB (Matplotlib format)



* Resize image to 256x256 pixels



* Display the original and resized images





**Step2 :-** Crop image

* Crop image to a region (x, y, width, height)



* Display the original and resized images





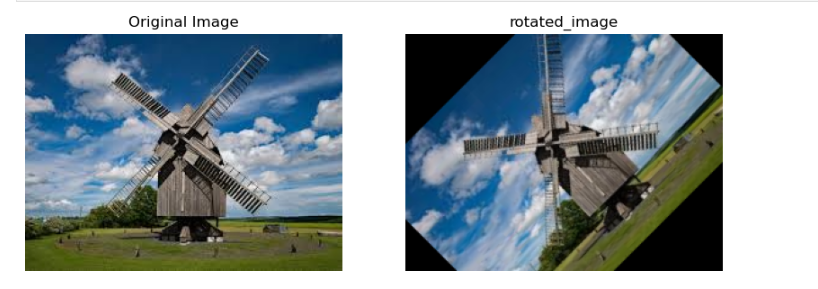
**Step3 :-** Rotate image

* Rotate image by 45 degrees



* Display the original and resized images





**Results :** Image Preprocessing Techniques was successfully executed.

**Lab 3 – Image Denoising**

**Aim:** ToReduces noise and smoothens images to improve quality through ***Image Denoising and Smoothing.***

To Enhances the contrast and brightness of images for better visibility using ***Histogram Equalization and Contrast Enhancement*.**

**Requirements:**

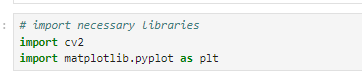
* Windows PC
* VS code/ Jupyter Notebook

**Procedure:**

**Step1 :- Denoising**

Denoising removes unwanted noise from images.

* import necessary libraries



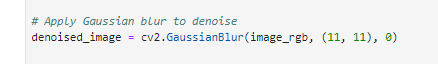
* Load an image



* Convert the image from BGR (OpenCV format) to RGB (Matplotlib format)



* Apply Gaussian blur to denoise



* Display the original and resized images





**Step2 :-** Histogram Equalization

Histogram equalization enhances the contrast of an image by spreading out the most frequent intensity values.

* Convert to grayscale

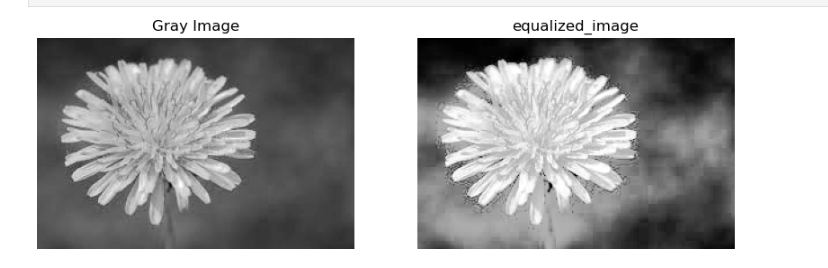


* Apply histogram equalization



* Display the original and resized images





**Results:** Toreduces noise and smoothens images to improve quality through **Image Denoising and Smoothing and**to enhances the contrast and brightness of images for better visibility using **Histogram Equalization and Contrast Enhancement** were successfully executed.