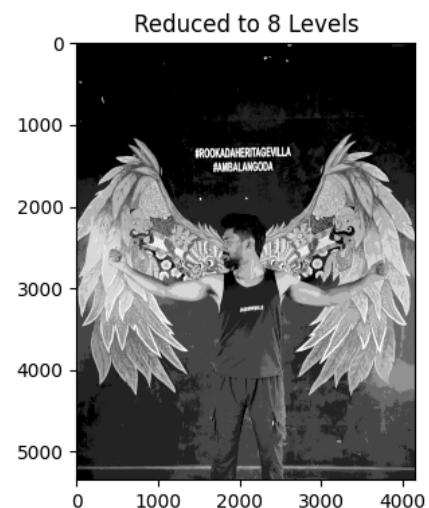


This project mainly focus on the basic image processing operations using Python and OpenCV. The goal is to perform four different tasks on a grayscale image.

### 1. Reduce Intensity Levels

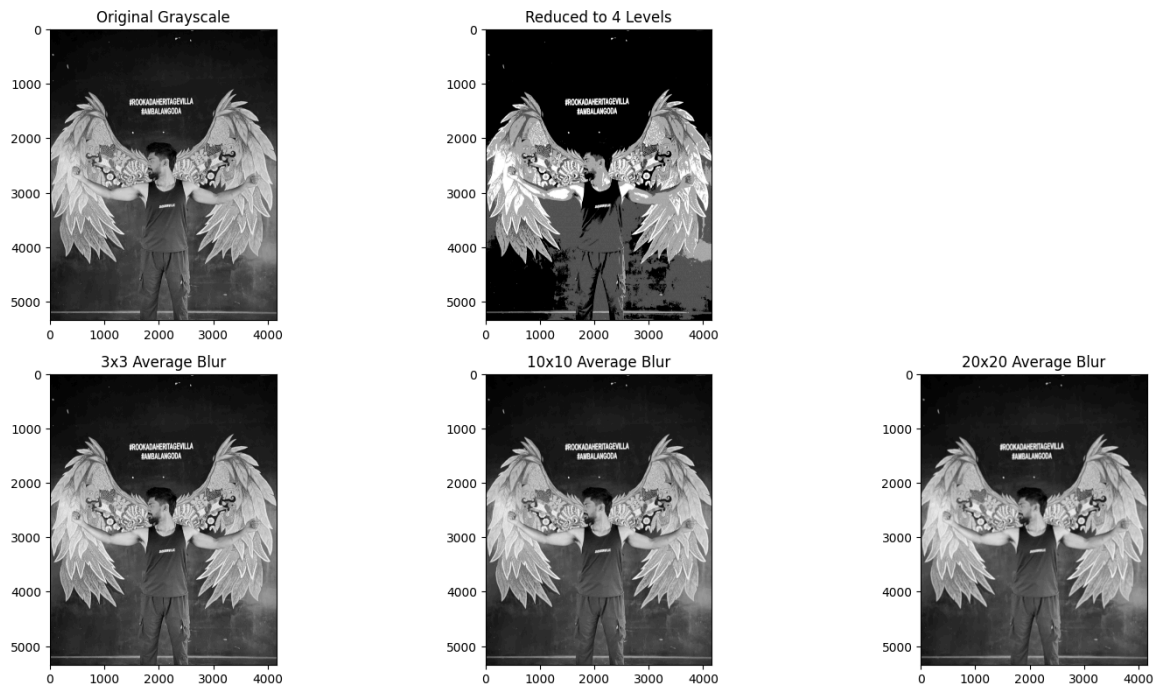
Images normally have 256 levels of brightness (from 0 to 255). For the assignment, I wrote a program to reduce the number of brightness levels to a smaller number, like 2, 4, 8, etc. This helps simulate how images look with fewer shades of gray. The user should enter how many levels they want, and the program adjusts the image accordingly.



### 2. Spatial Averaging (Smoothing)

In this part, I applied a blurring (smoothing) technique to the image using three different square regions:

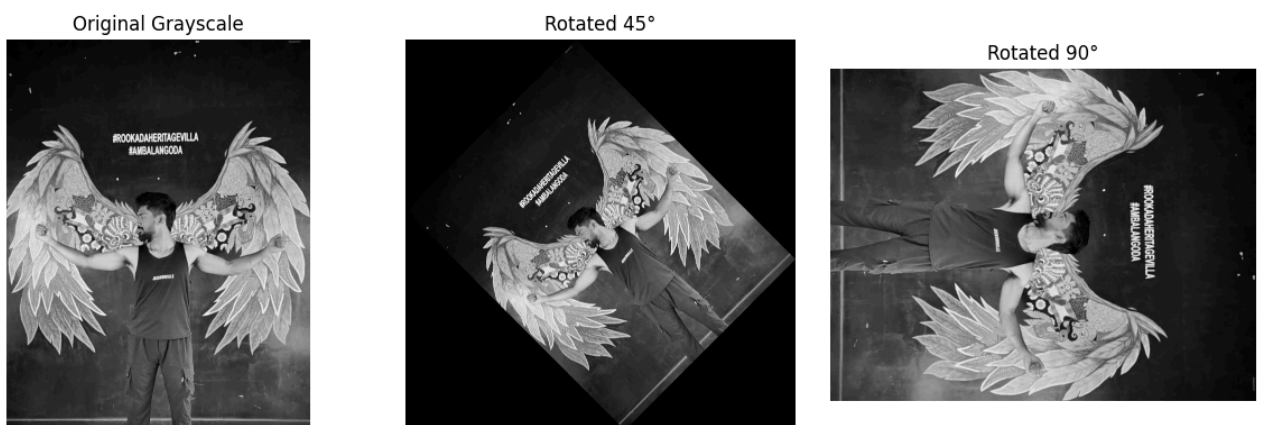
- A small 3x3 region
- A medium 10x10 region
- A large 20x20 region



This means each pixel is replaced by the average value of its surrounding pixels. Larger regions result in a more blurred image. This technique is useful to reduce noise.

### 3. Rotate Image

I rotated the input image by:  
45 degrees, and 90 degrees.



The 90-degree rotation uses a built-in OpenCV function. The 45-degree rotation is done using a mathematical transformation, which rotates the image around its center and resizes it to fit the new shape.

#### 4. Block Averaging (Reduce Spatial Resolution)

In this part, I divided the image into small square blocks (3x3, 5x5, and 7x7) without overlapping, and replaced all pixels in each block with the average value of that block. This makes the image look more pixelated or low-resolution, which simulates reducing the image quality. It's like seeing a zoomed-out or simplified version of the image.

3x3 Block Averaged



5x5 Block Averaged



7x7 Block Averaged

