**EC7212 – Computer Vision and Image Processing**

**Take Home Assignment 1**

Github Repo: <https://github.com/NisalDeZoysa/EC7212-CVIP-Assignment-01.git>

1. To reduce the number of intensity levels in an image from 256 to 2, in integer powers of 2. The desired number of intensity levels needs to be a variable input to your program.

def reduce\_intensity\_levels(image, levels):

    factor = 256 // levels

    reduced\_image = (image // factor) \* factor

    return reduced\_image

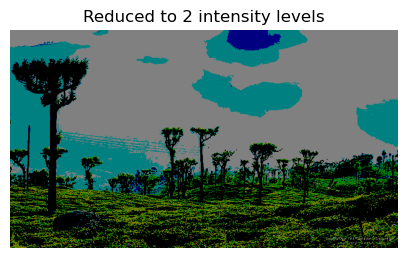
# Try different levels: 2, 4, 8, 16, 32, etc.

for level in [2, 4, 8, 16, 32, 64, 128]:

    reduced = reduce\_intensity\_levels(image, level)

    show\_image(f"Reduced to {level} intensity levels", reduced)

**Results:**



A group of trees in a field

AI-generated content may be incorrect.

A green field with trees and mountains in the background

AI-generated content may be incorrect.A green field with trees and mountains in the background

AI-generated content may be incorrect.A green field with trees and mountains in the background

AI-generated content may be incorrect.

A green field with trees and mountains in the background

AI-generated content may be incorrect.

1. Load an image and then perform a simple spatial 3x3 average of image pixels. Repeat the process for a 10x10 neighborhood and again for a 20x20 neighborhood.
2. def spatial\_average(image, kernel\_size):
3. return cv2.blur(image, (kernel\_size, kernel\_size))
4. for size in [3, 10, 20]:
5. blurred = spatial\_average(image, size)
6. show\_image(f"Spatial Average {size}x{size}", blurred)

**Results:**

A green field with trees and mountains in the background

AI-generated content may be incorrect.

A blurry image of a green field

AI-generated content may be incorrect.

A blurry image of a green field

AI-generated content may be incorrect.

3. Rotate an image by 45 and 90 degrees.

# Rotation

def rotate\_image(image, angle):

    (h, w) = image.shape[:2]

    center = (w // 2, h // 2)

    matrix = cv2.getRotationMatrix2D(center, angle, 1.0)

    return cv2.warpAffine(image, matrix, (w, h))

rotated\_45 = rotate\_image(image, 45)

rotated\_90 = rotate\_image(image, 90)

show\_image("Rotated 45°", rotated\_45)

show\_image("Rotated 90°", rotated\_90)

**Results:**

A green field with trees

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

4. For every 3×3 block of the image (without overlapping), replace all the corresponding 9 pixels by their average. This operation simulates reducing the image spatial resolution. Repeat this for 5×5 blocks and 7×7 blocks.

# Block Averaging

def block\_average(image, block\_size):

    h, w = image.shape[:2]

    result = image.copy()

    for y in range(0, h, block\_size):

        for x in range(0, w, block\_size):

            block = image[y:y+block\_size, x:x+block\_size]

            if block.size == 0:

                continue

            avg\_color = block.mean(axis=(0, 1), dtype=int)

            result[y:y+block\_size, x:x+block\_size] = avg\_color

    return result

for block in [3, 5, 7]:

    reduced = block\_average(image, block)

    show\_image(f"Block Average {block}x{block}", reduced)

**Results:**

A green field with trees and mountains in the background

AI-generated content may be incorrect.

