E9 241 Digital Image Processing
Assignment 02

Name: Bitupan Arandhara Sr. No.: 25910

Q1. Spatial Filtering and Binarization:

Results:

Box blurring was applied on the input image where a spatial box filter of size m x m was used. The box filter was constructed as a matrix of ones and then normalized by dividing each element by m^2.

The input image was blurred using the generated box filter of sizes $\{5,\ 29,\ 129\}$. The blurred image was then binarized using Otsu's Binarization algorithm. For each filter size, the blurred image, its corresponding histogram, and the final binarized image were generated.

The optimal within-class variances and their corresponding Otsu thresholds are tabulated below:

Kernel Size (m)	Optimal Otsu Threshold	Optimal Within-Class Variance
5	125	165.36
29	127	327.94
129	76	405.76

The **minimum optimal within-class variance** was observed for **kernel size 5.**

The blurred image, binarized image and the histogram of the blurred image are presented in the figure below:

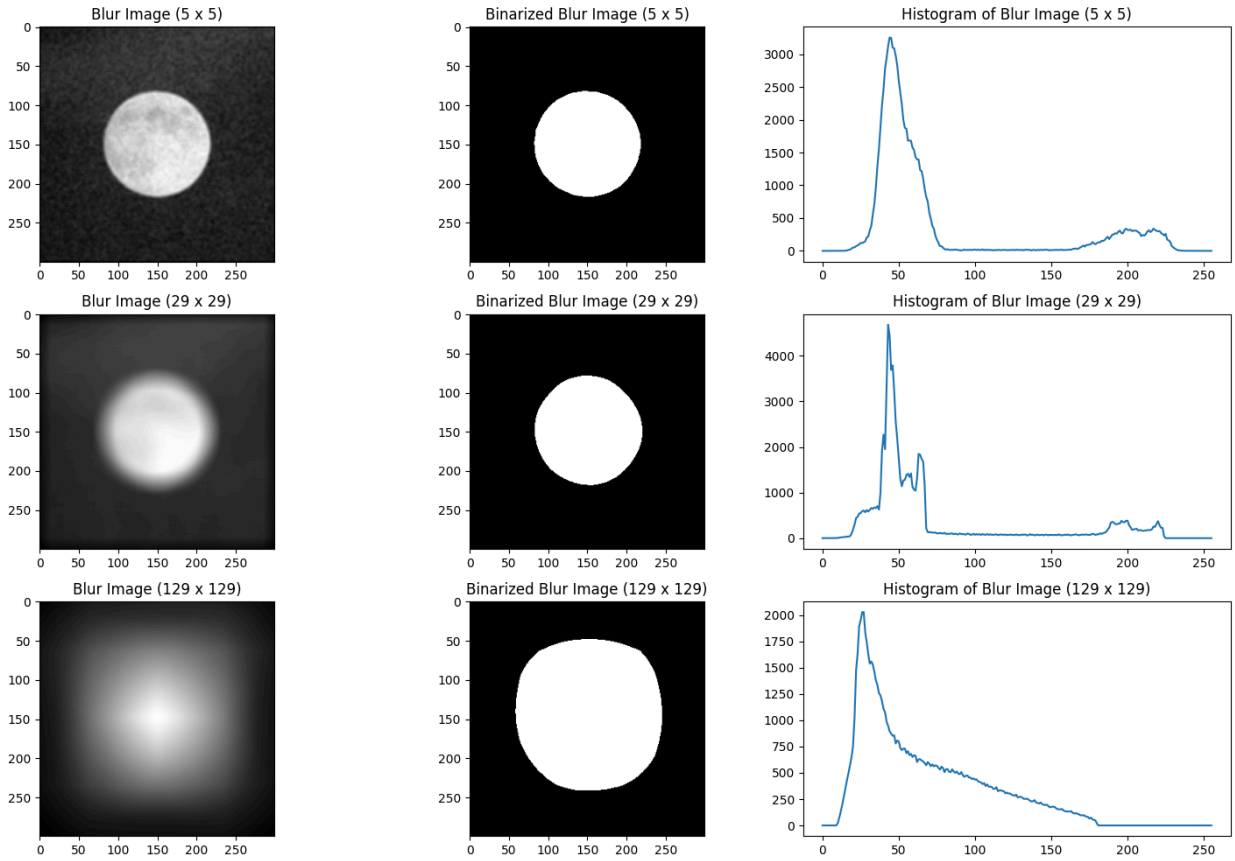

Figure1. The output images for kernel sizes 5x5, 29x29, and 129x129. Each row displays the blurred image, the binarized result after applying Otsu's method, and the histogram of the blurred image.

Inferences:

1. **Effect of Blurring on Noise:**

   The primary effect of the box filter is noise reduction. With the smallest kernel (5x5), the moon's texture is still visible, but much of the high-frequency salt-and-pepper noise is reduced. As the kernel size increases to 29x29 and 129x129, the blurring becomes much more pronounced, effectively averaging out all the noise and finer details.

2. **Effect on Histogram:**

   The histogram changes significantly with the degree of blurring

- For **m=5**, the histogram shows two relatively distinct peaks, corresponding to the background (noise) and the foreground (moon). This separation is what allows Otsu's method to find a good threshold.
- As **m increases**, the two peaks in the histogram merge and become a single, narrower peak. This is because excessive blurring averages the pixel values of the moon and the background. The overall intensity range of the image is also reduced.

Q2. Scaling and Rotation with Interpolation:

Results:

The two transformation sequences were applied to the original image. The resulting images, along with the original and the difference image, are shown below.

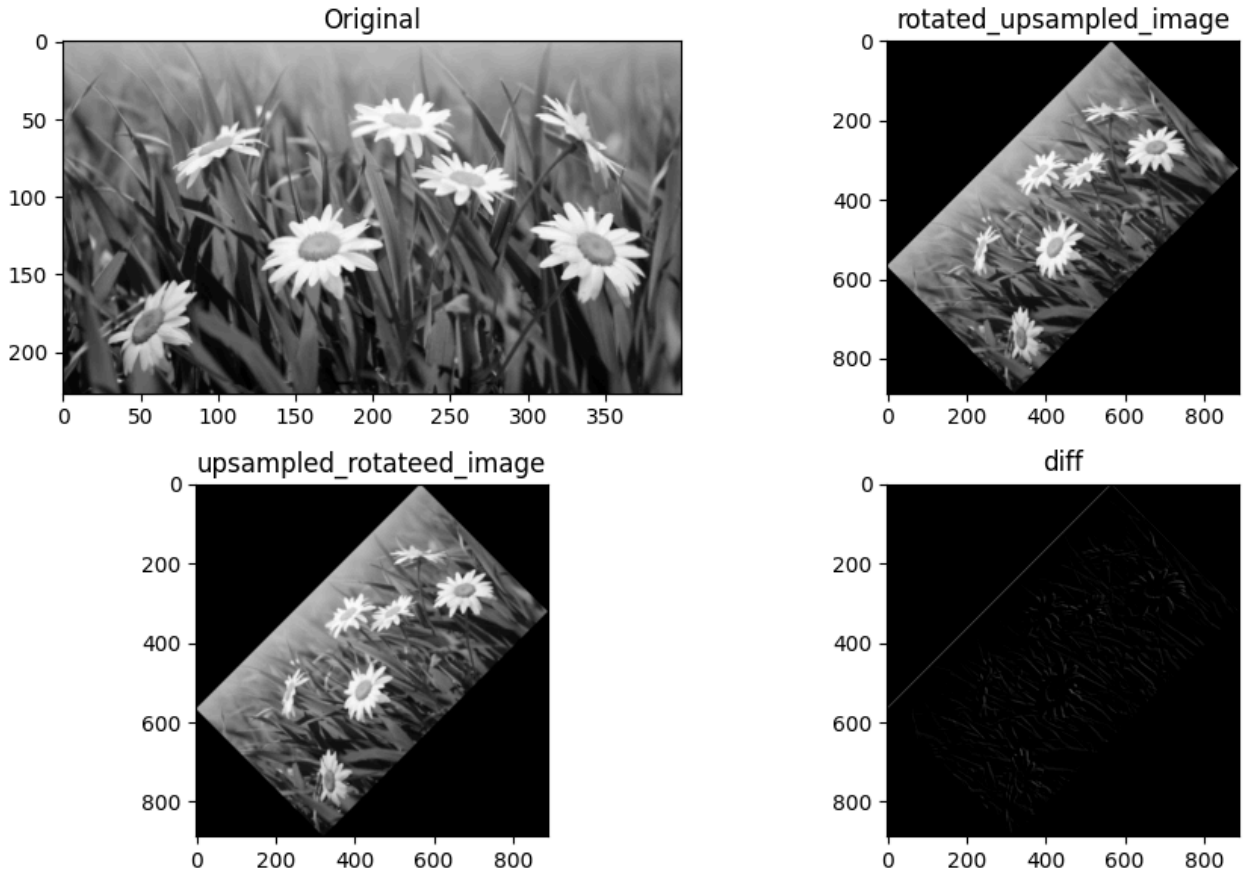

Figure 2: Comparison of the two transformation sequences. (Top-left) Original Image. (Top-right) Result 1: Upsample then Rotate. (Bottom-left) Result 2: Rotate then Upsample. (Bottom-right) The difference between Result 1 and Result 2.

The difference image ($diff$) was calculated as $Result\ 1\ -\ Result\ 2$. The range of pixel intensity values in this difference image was analyzed:

- Range of $diff$ before clipping: **Minimum = -159.0, Maximum = 150.0**
- Range of $diff$ after clipping to [0, 255]: **Minimum = 0, Maximum = 150**

The non-zero range of the difference image immediately indicates that the two resulting images are **not identical.**

Inferences:

1. **Commutativity of Operations**:

   The experiment clearly demonstrates that upsampling and rotation are **not commutative operations.** The result will be different when we apply rotation first and then upsample it and when we apply upsample first and then rotate the upsampled image.

2. **Explanation for the Difference:**

   The discrepancy arises from how bilinear interpolation interacts with the data at each stage.

   - **Scenario 1 (Upsample then Rotate):** We first create a new image by interpolating the original high-resolution image. This new, larger image is then rotated. The rotation step performs a *second* interpolation on this already-interpolated data.
   - **Scenario 2 (Rotate then Upsample):** We first rotate the original image. This involves interpolating values from the original grid to find the pixel values for the rotated grid. Then, we upsample this result, which performs *another* interpolation on the rotated, interpolated data.
   - Since interpolation is an approximation (a weighted average of neighbors), applying it at different stages with different neighboring pixels will lead to different final values. Each interpolation step introduces a small amount of error, and compounding these approximations in a different order leads to a different final output.

3. **Analysis of the Difference Image**:

   The $diff$ image is not random noise; it highlights exactly where the two methods differ most. When plotting the clipped difference image, we are viewing the absolute difference wherever $Result\ 1$ was greater than $Result\ 1$. The brightest areas in the difference image correspond to the **edges and high-texture regions** of the flowers and leaves. Interpolation errors are most pronounced in areas of high frequency (sharp changes in

intensity). In smoother, more uniform regions of the image, the difference is minimal (darker in the $diff$ image).

Q3. Image Sharpening:

Methodology:

The sharpening effect was achieved using the **unsharp masking** algorithm. The process consists of the following steps:

1. **Create a Blurred Image:** A blurred (or "unsharp") version of the original image is created using a 5x5 box filter.

2. **Create a Mask:** The blurred image is subtracted from the original image. The result is a "mask" that contains the high-frequency components of the image, i.e., the edges and details.

$$mask = original\_image - blurred\_image$$

3. **Add Mask to Original:** This detail mask is then scaled by a factor derived from the input parameter p and added back to the original image. The formula used is:

$$sharpened\_image = original\_image + (p * 10) * mask$$

By controlling the weight of the mask with p, we can adjust the strength sharping effect. Finally, the resulting pixel values are clipped to the [0, 255] range.

Results:

The function was tested on the $study.png$ image with p values of $0,\ 0.1,\ 0.2,\ 0.5,\ 0.75,\ and\ 1$.

The function was also tested on two additional images, an owl and a fox.

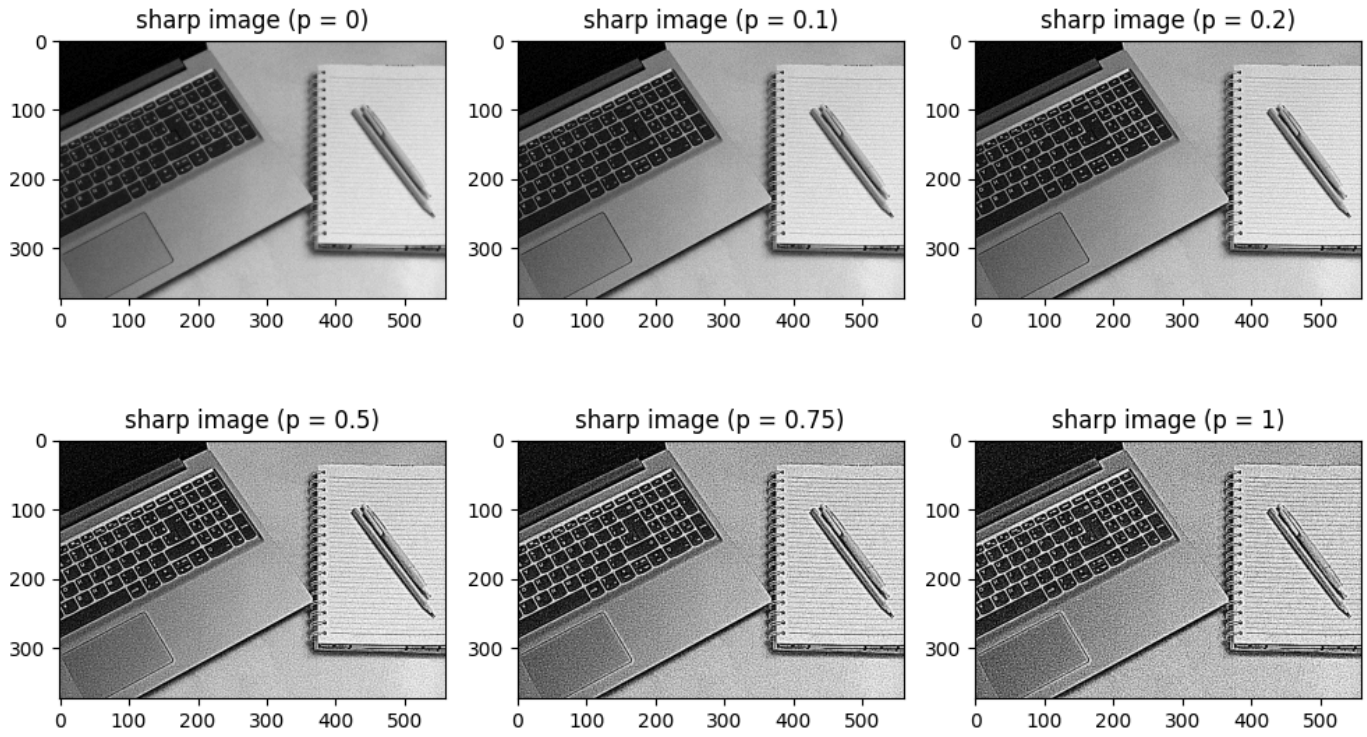

Figure 3: The 'study.png' image sharpened with different values of the parameter p. The sharpening effect visibly increases as p approaches 1.

Figure 4: The owl image with no sharpening (p=0, left) and strong sharpening (p=0.5,right). The texture of the feathers and the tree bark are significantly enhanced.

Figure 5: The fox image with mild sharpening (p=0.2, left) and strong sharpening (p=1, right).

Inferences:

1. **Effectiveness of Parameter**:

   The parameter p provides an intuitive and effective control over the sharpening process.

   - At $p = 0$, the output is identical to the input, as required.
   - As p **increases**, details become progressively clearer. In the image $study.png$, the lines on the notepad and the texture of the laptop surface become more defined.
   - At $p = 1$, the sharpening is very strong. This leads to the creation of **halos**—light and dark artifacts along strong edges. This is a characteristic effect of strong unsharp masking and is visible around the pen and notebook edges in Figure 3.

2. **Noise Amplification**:

   A key side effect of sharpening is the amplification of any existing noise in the image. In Figure 1, the subtle grain in the background of the original image becomes much more pronounced as p increases.

3. **Application to Textured Images:**

   The tests on the owl and fox images show that the technique works very well on images with rich natural textures. The sharpness added to the owl's feathers and the fox's fur makes the images appear more vivid and detailed. However, even in these cases, excessive sharpening can make the image look harsh.