Lab report

Object counting

Vũ Trung Thành - 20220066

1 Introduction

The goal of this experiment is to develop and evaluate a image processing pipeline capable of accurately counting objects in a given set of images. These images feature multiple subjects captured against relatively uniform, yet potentially noisy, backgrounds.

This report outlines the development of the processing pipeline, including the choice of preprocessing techniques, segmentation methods, and object counting strategies. The pipeline was tested on a set of images with varying characteristics, and the results were evaluated based on the accuracy of object detection and the total object count. The effectiveness and limitations of the pipeline are discussed, and suggestions for improvement are provided.

2 Data

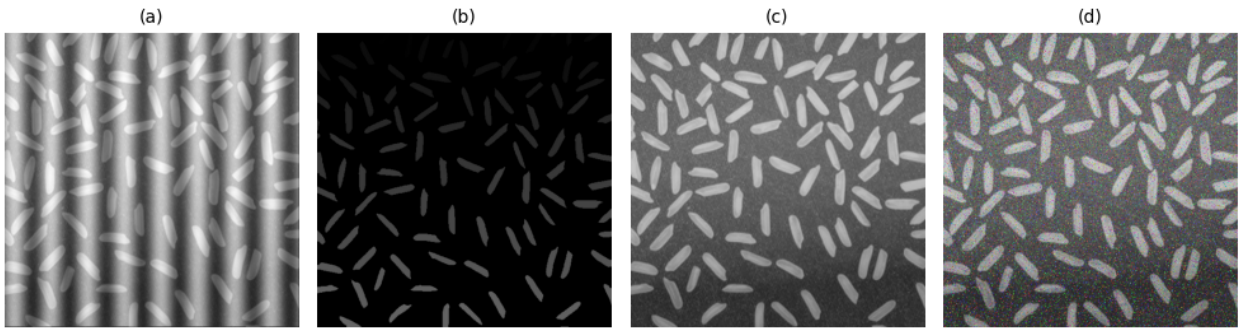


Figure 1: Original images.

The dataset consists of four images, each containing multiple uniform objects (likely rice grains) with varying visual features and distortions. Image (a) has a high-intensity sinus noise on top of it. In image (b), the objects are distinct and appear well-defined against a pure black background; however, their brightness varies, becoming darker from the bottom to the top. Images (c) and (d)

are quite similar, with backgrounds that transition from dark at the bottom to lighter at the top. However, image (d) differs by having salt-and-pepper noise, which is absent in image (c).

3 Method

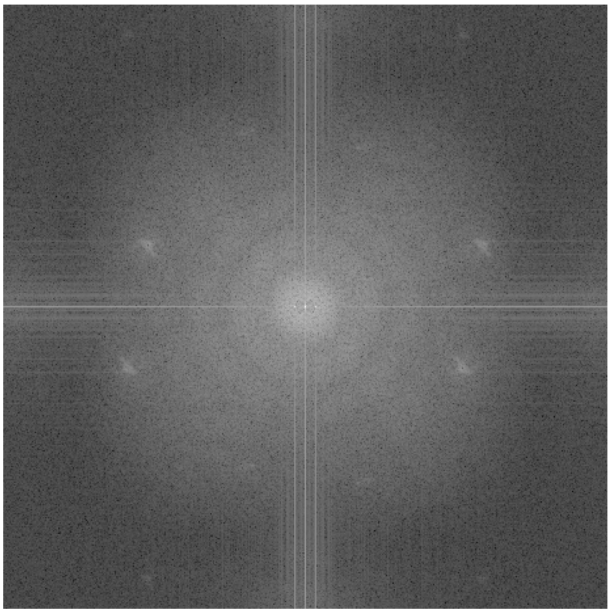
Before processing, the images were first converted to grayscale, as non-gray colors only appear in image (d) due to the presence of salt-and-pepper noise. Following the conversion, the processing pipeline proceeded as outlined below:

3.1 Remove sinusoidal noise

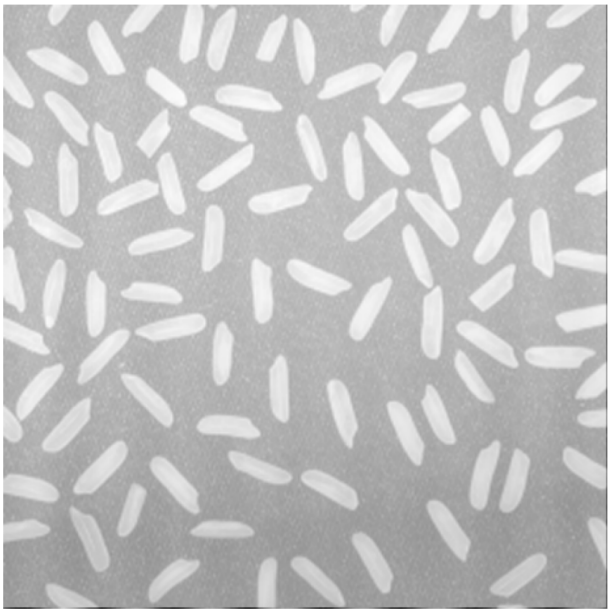
The removal of sinusoidal noise was prioritized, as performing other processing steps beforehand could interfere with the sinusoidal signal layered on top of image (a).

The process began with applying a Fourier Transform (FT) with scaling and shifting to convert the original image into the frequency domain. In this domain, the bright spots corresponding to the sinusoidal noise were selectively filtered out, **excluding** the central bright spot, which represents the primary image features.

Afterward, the image was transformed back to the spatial domain, resulting in images with the sinusoidal noise effectively removed.



(a) Image (a), which has two bright spots apart from the center.



(b) Image (a) after filtering.

Figure 2: Removal of sinusoidal noise using FT.

3.2 Median filter

Next, a median filter with a kernel size of 3 was applied to remove the salt-and-pepper noise in image (d). Subsequently, the images were sharpened using unsharp masking, as the filtering step caused a slight blurring effect. Sharpening improved the separation of objects in later processing steps.

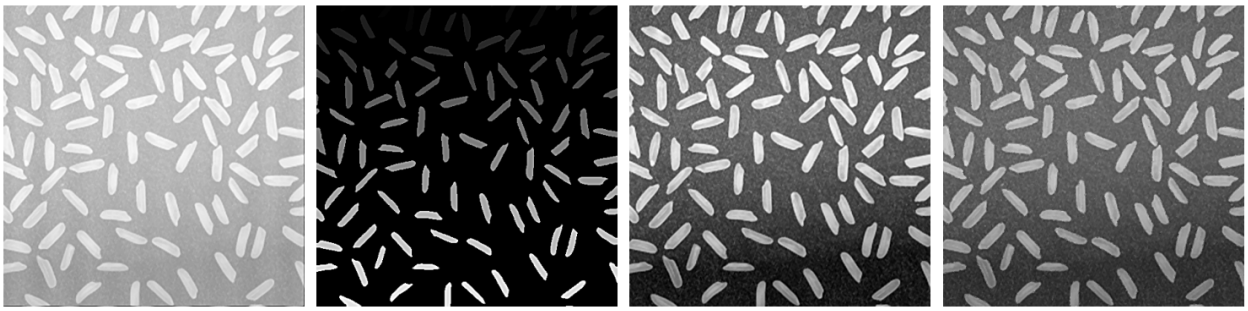


Figure 3: Images after median filter and sharpening.

3.3 Correct background illumination

When observing the images, it was noted that images (a), (c), and (d) did not have uniform backgrounds—the background was brighter at the top of the image compared to the bottom. To address this, background illumination correction was performed.

First, the backgrounds were approximated by applying morphological opening using a disk-shaped structuring element with a diameter of 50 pixels. This operation removed the objects, leaving only the background behind.

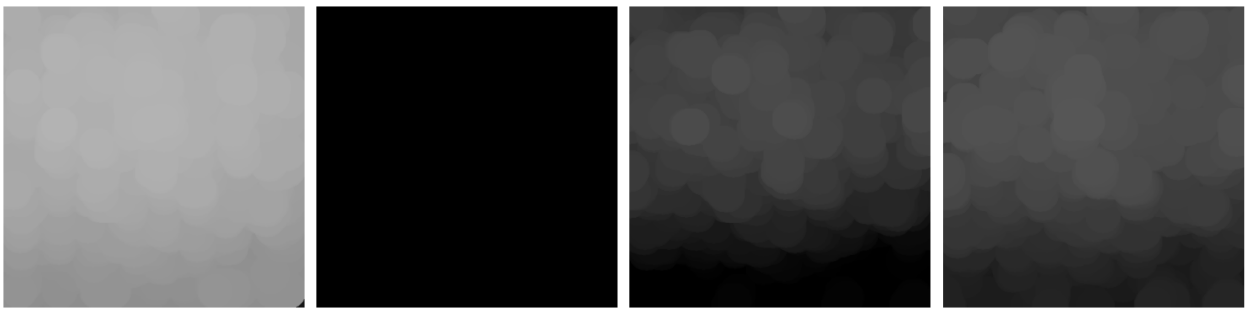


Figure 4: Background approximation.

Next, the background approximation was subtracted from the original images to produce results with uniform backgrounds. However, this process introduced some bright white noise near the edges of objects, resembling salt-and-pepper noise. Since the frequency of this noise was relatively low, a median filter was not applied to preserve the separation between objects as much as possible.

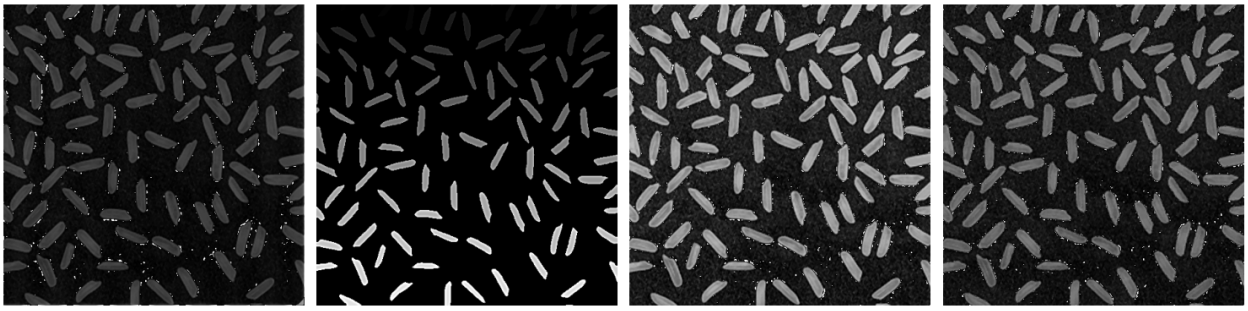


Figure 5: Images after background correction.

3.4 Gamma correction

At this stage, there was a clear separation between the background and foreground, and the object colors were relatively uniform — except for image (b), where the objects remained very dark at the top and bright at the bottom.

To address this, gamma correction was applied to reduce the contrast between the brighter and darker regions. Values of $\gamma < 1$ were tested to enhance the visibility of the objects in image (b).

After experimentation, gamma value of $\gamma = 0.15$ was selected. This value had a good balance between making the dark objects in image (b) more visible without over-brightening the other images.

3.5 Binarization

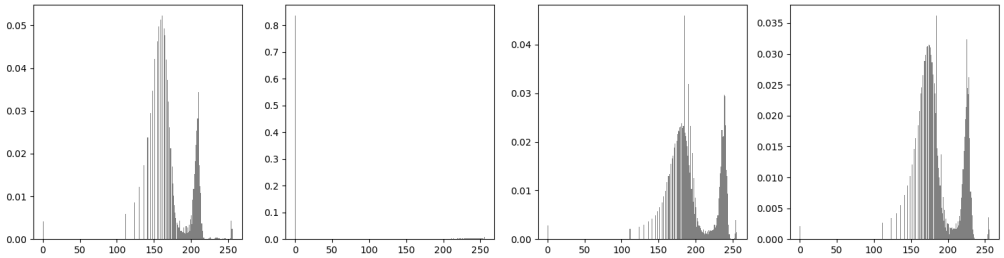


Figure 7: Histogram of the images after gamma correction.

By analyzing the histograms of the images, it is clear that each image requires a distinct threshold value for binarization. For instance, image (a) requires a threshold of approximately 180, while image (b) needs a small non-zero threshold. The last two images, (c) and (d), require a threshold value of around 225.

Due to the different threshold value, Otsu's method for automatic image thresholding was used to determine an optimal threshold dynamically. Following thresholding, morphological opening with a disk-shaped kernel of diameter 6 was applied to eliminate thresholding artifacts.

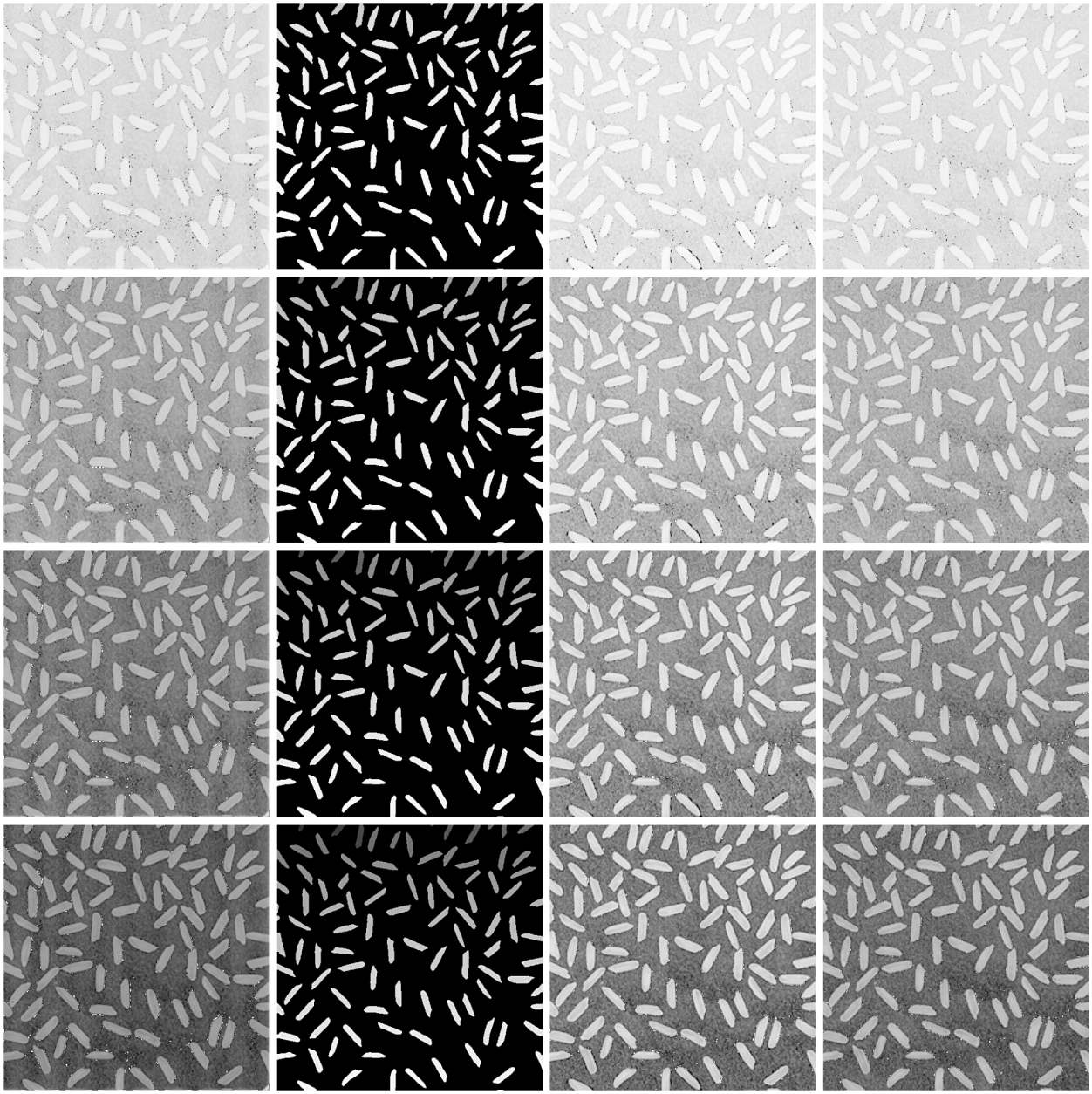


Figure 6: Images with gamma correction. From top to bottom: $\gamma = 0.05, 0.15, 0.25, 0.35$.

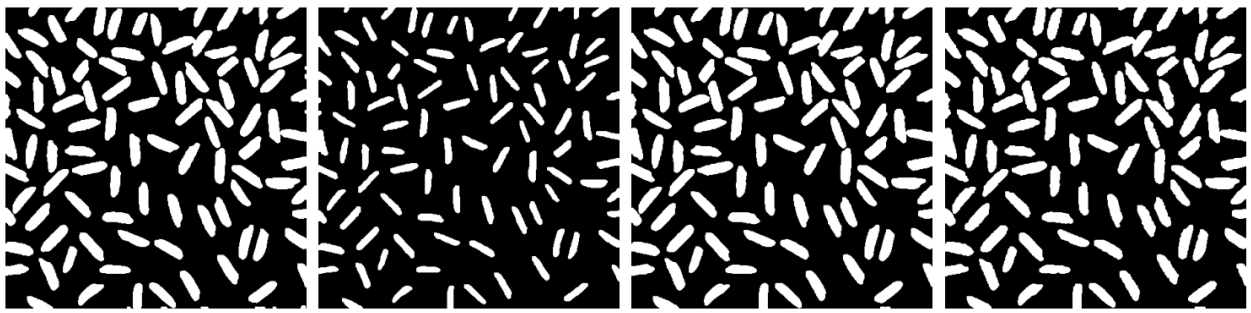


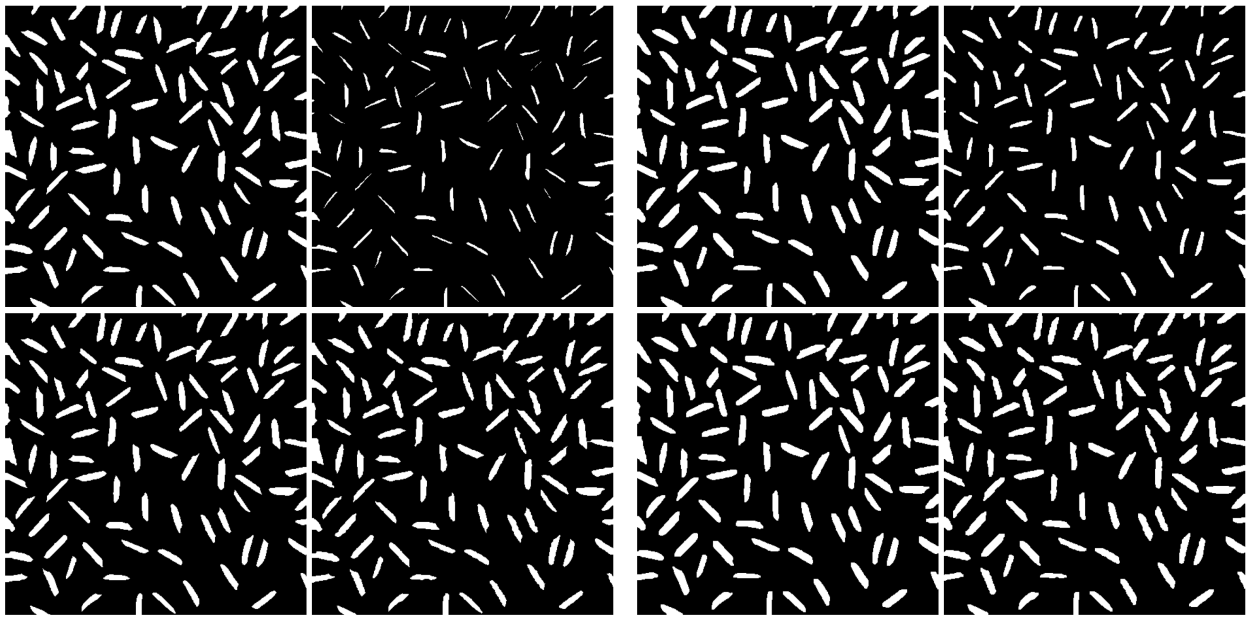
Figure 8: Images after binarization.

3.6 Object separation

Some objects remained connected together after binarization. While erosion is commonly used to address this issue, applying it caused small objects in image (b) to disappear.

As an alternative, distance transformation with thresholding was used. The distance transformation acts as a soft erosion, where foreground pixels near the border are not turned into background pixels but are instead assigned values proportional to their distance from the background.

Subsequently, a threshold of 0.3 times the maximum distance was applied to the transformed image. This approach produced significantly better results compared to simple erosion, preserving small objects while effectively separating overlapping ones.



(a) Erosion with a 2x2 kernel was applied iteratively for 5 steps. In image (b), some smaller objects started to disappear, while in the other images, the objects were just beginning to separate.

(b) Distance transformation with thresholding was applied. The objects were well-separated, and the smaller objects in image (b) remained intact.

Figure 9: Comparison of two methods.

3.7 Object counting

After separating the objects, the final step was straightforward: the number of objects was determined by counting the number of connected components in the images.

4 Result and discussion

After counting, all images were found to contain the same number of objects, totaling 99 each. The watershed segmentation results, using the separated images as seeds for the algorithm, are also

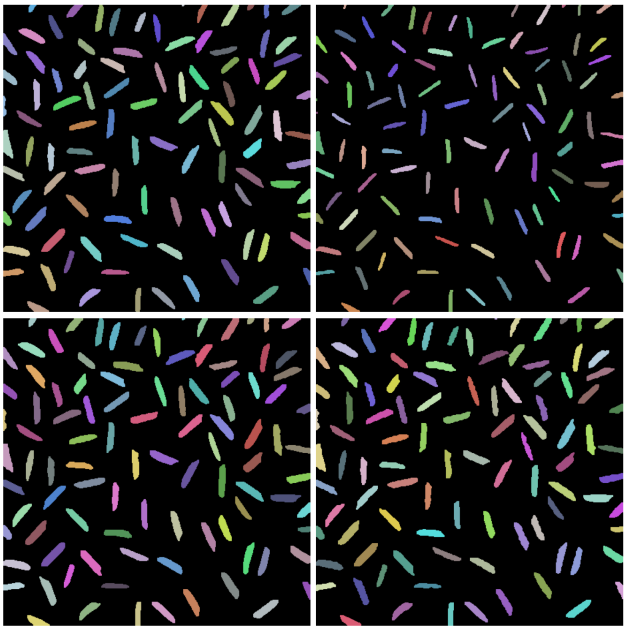


Figure 10: Counting results.

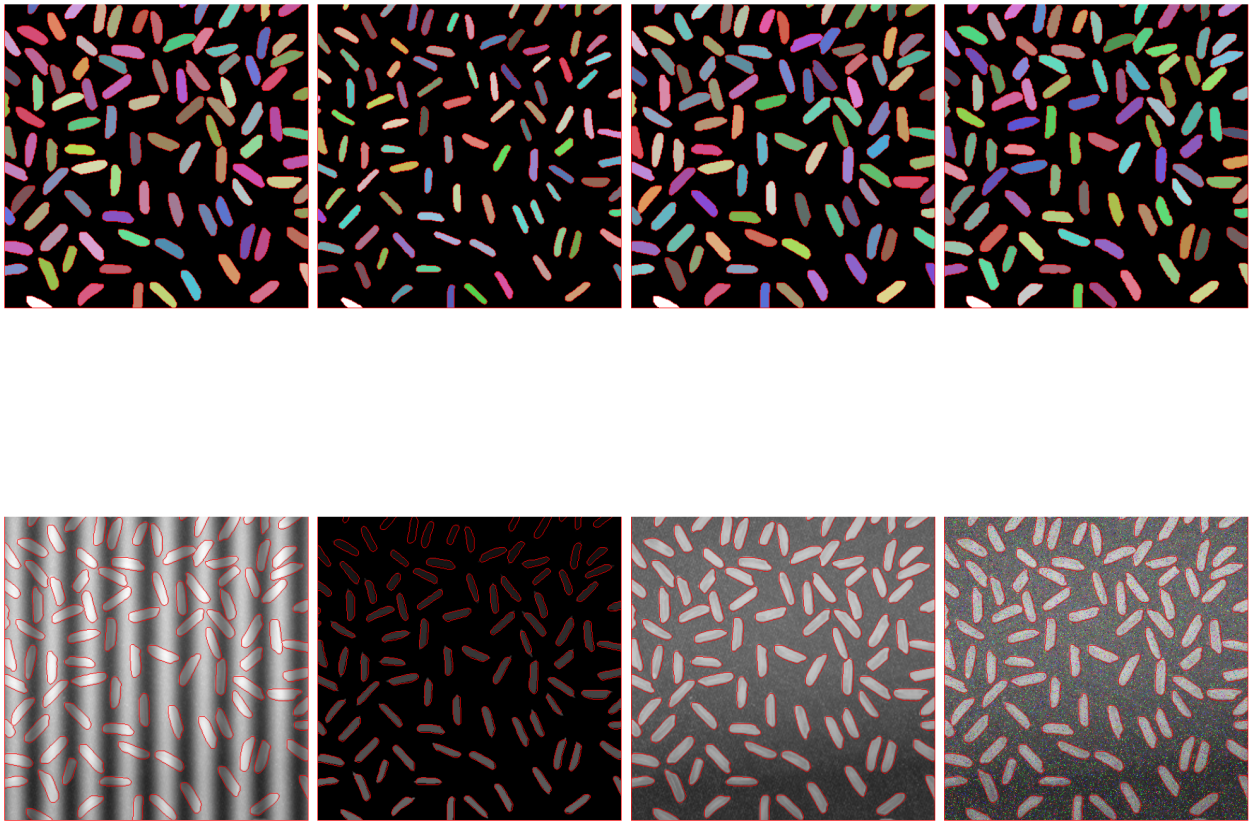


Figure 11: Watershed result on images from Figure 8 and original images.

provided.

Although the results appear to be promising, some issues still need to be addressed:

1. Some small objects located at the edges of the images were not detected, resulting in their loss during the process.

2. While image (b) showed the same total number of objects as the other images, closer inspection revealed that one object was incorrectly split into two parts, and another object was completely missed. This likely occurred because the objects in image (b) are smaller, and their intensity levels in the original image were much closer to the background, making them harder to distinguish.

5 Conclusion

This experiment successfully developed a processing pipeline to count objects in images with varying noise and background conditions. Techniques such as Fourier Transform, morphological operations, gamma correction, and distance transformation were used to address challenges like noise, non-uniform illumination, and object separation.

The pipeline determined the number of objects in all images to be 99, demonstrating its effectiveness. However, issues such as missed objects and inaccuracies with smaller or low-contrast objects highlight some rooms for improvement.

Future work can focus on improving edge object detection, implementing adaptive parameter selection, enhancing small and low-contrast object detection, and experimenting with more robust denoising methods.

Appendix

The link to the implementation can be found at https://github.com/thanh309/BKAI-CV-Lab-Practice.