Student: Nare Hovhannisyan
Class: Image Processing

Final Project

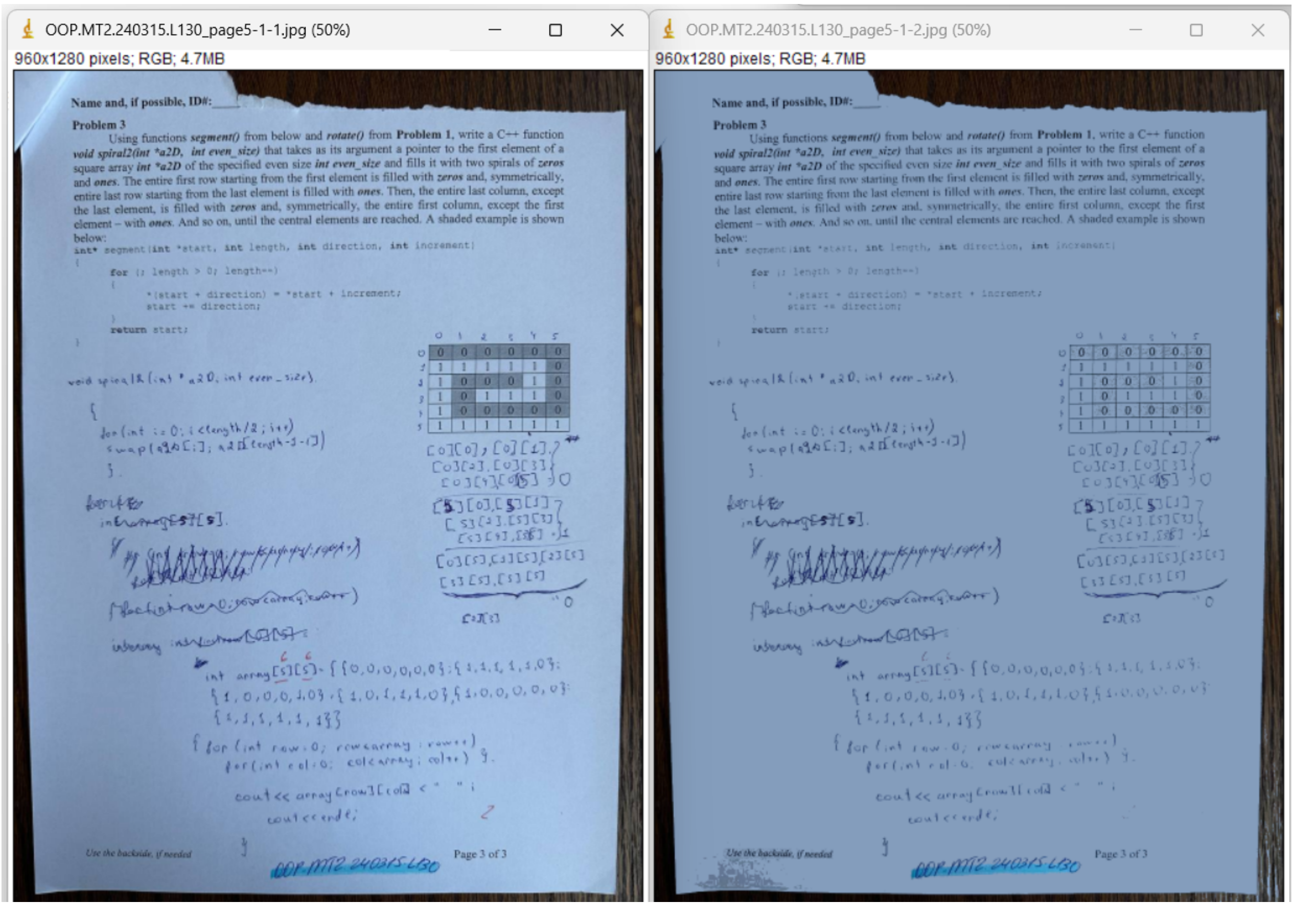
- **Introduction**
- **Adjusting background to have similar color**
- **Histogram Comparison to identify handwritten vs printed region**
- **Comparison of count of pixels with higher intensity**
- **Removing Background by Thresholding and Background Subtraction**
- **Analyzing the printed and handwritten images by analyze particles**

Introduction

This project aims to do image processing on images collected beforehand, preprocess them to be able to run machine learning or deep learning algorithms on them and get some insights from a person's handwriting. The images are taken by phone. The extension is .png. The pictures are A4 images which have printed text + handwritten text on them. In the scope of this project I tried to come up with ideas of how we can preprocess those images.

Adjusting background to have similar color

The first idea that came to my mind was to try to make the background color even in all parts of the image. Since the pictures are taken by phone and they don't have ideal lighting, we can observe that different pixels of the background have different color intensities.
For this purpose I have created a **pluginAdjustBackground** which sets the background color to one particular shade. I have tried several different shades and this shade is the most optimal in a sense that it does the least damage to the text on the paper. The plugin can be found in the repository.
Following is the comparison before and after applying the filter to the image.



We can observe that the blue color still remains blue while all parts of the background have the same color intensity. There are also drawbacks with this approach. Although the blue color still remained blue the shades are being modified a bit which can affect further analysis of the image. Also, another drawback is that if some text is not very visible even before applying this filter the filter will make those parts even worse and not readable.

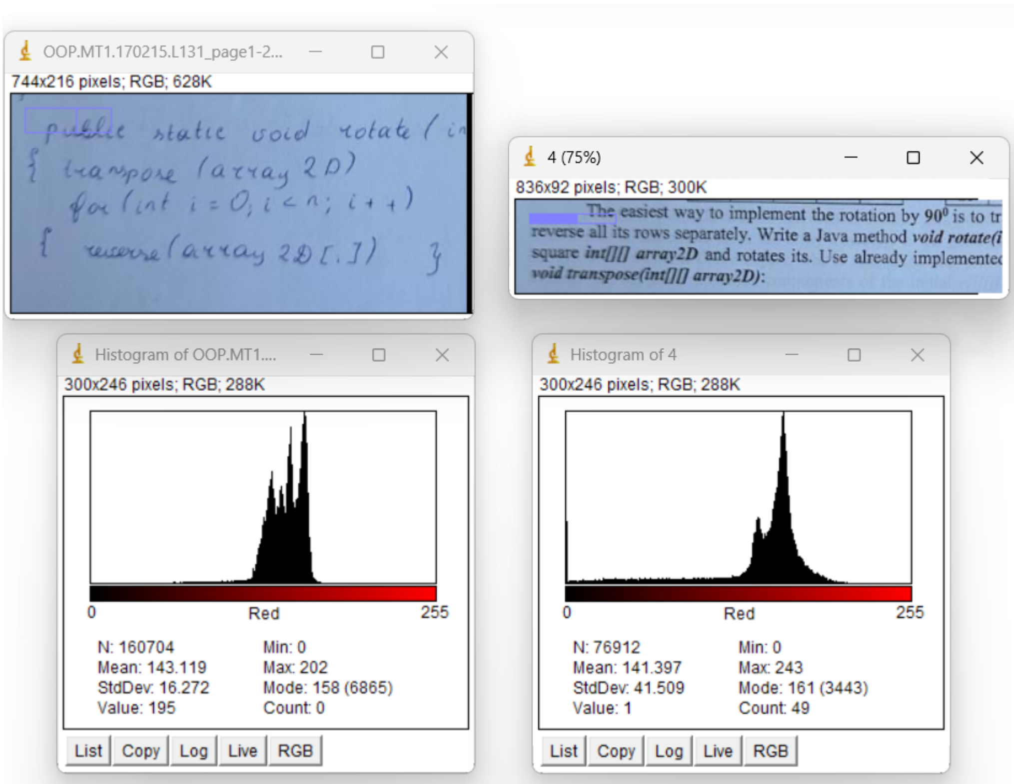
Overall to conclude, this approach can be helpful in some scenarios but it also has some drawbacks.

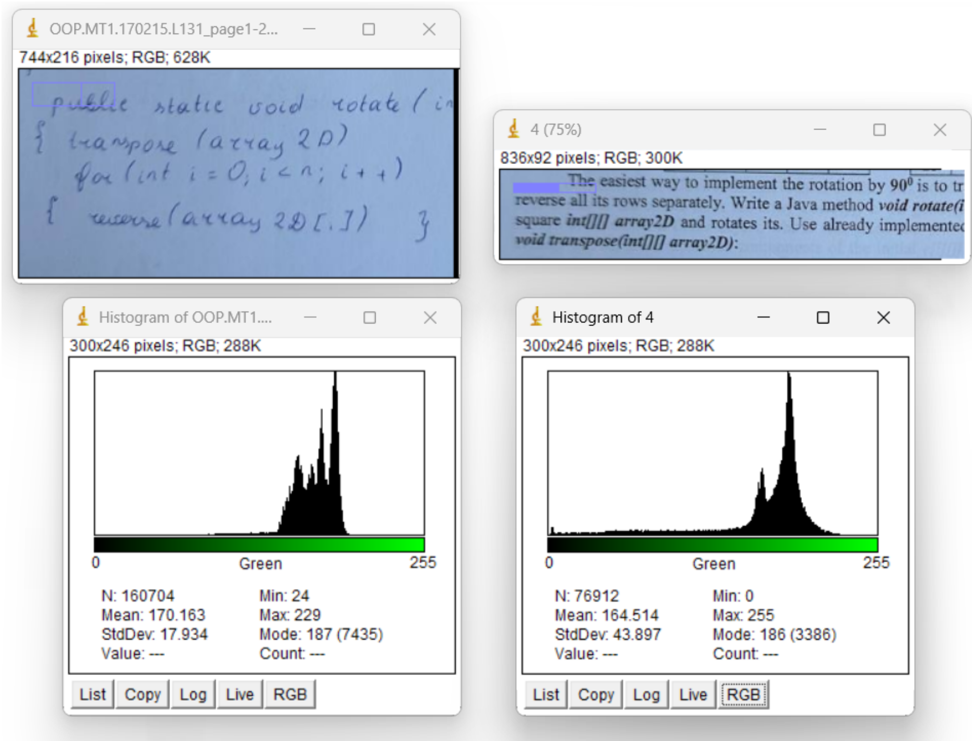
Histogram Analysis

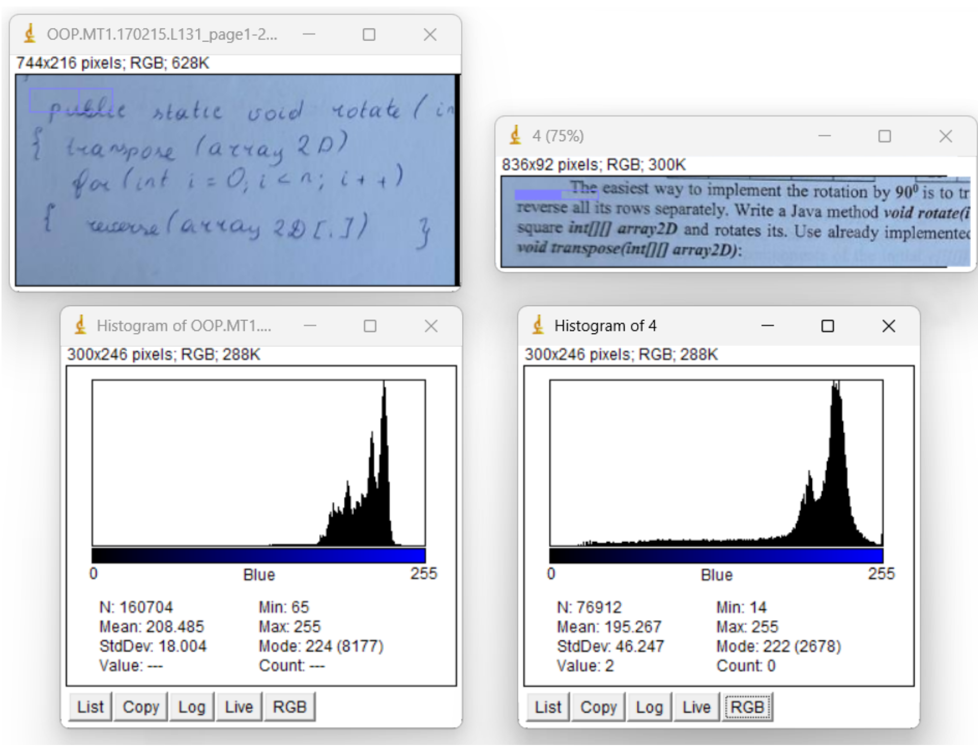
Another feature that I have noticed which can be used for preprocessing is that handwritten text, though not always, usually is written with blue color. This can be used to detect and differentiate them from printed text. For this we can use histogram analysis.

I have taken one example image from the project and cropped the handwritten part, constructed the histogram separately, and cropped the printed text part and constructed the histogram separately. Let's compare them to understand the differences.

On the following images, the left part is the written text and its histogram, on the right is the printed text and its histogram.







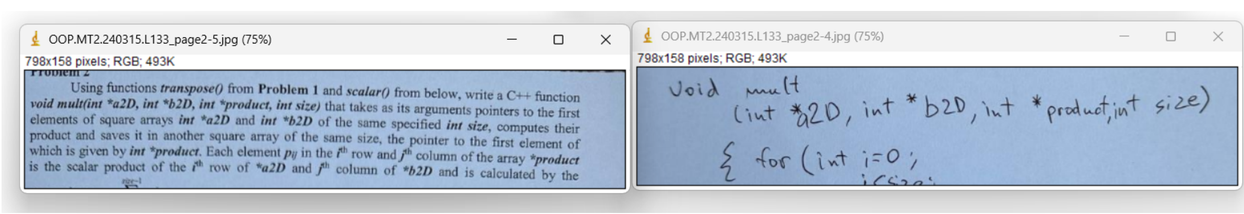
Comparing the histograms of two images reveals distinct characteristics in the histograms for the Red (R), Green (G), and Blue (B) channels, as well as in the combined RGB histogram.

Image with Handwriting:
Red Channel (R): We observe less peaks towards the higher intensity values.
Green Channel (G): Similar to the Red Channel, minimal peaks towards higher intensity values.
Blue Channel (B): Higher peaks towards the higher intensity values, indicating the presence of blue color.

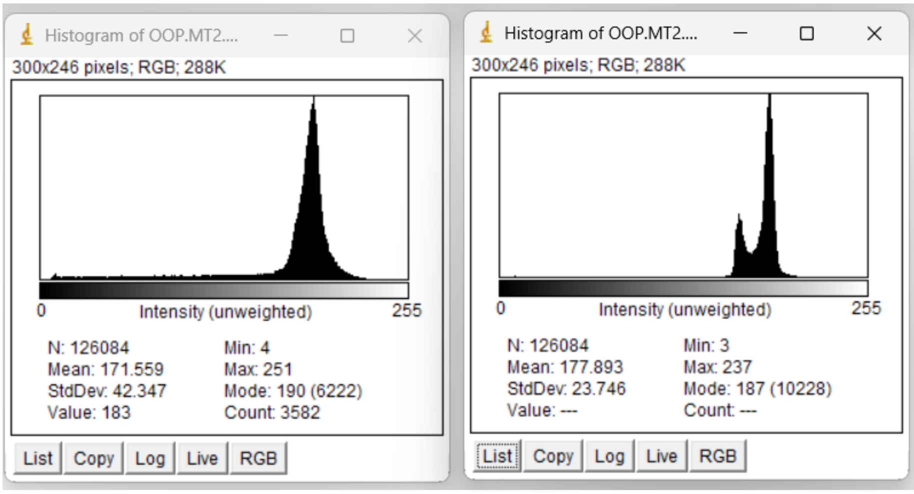
Image with Printed Black Text:
Red Channel (R): We also observe values towards the lower intensity values, representing the presence of dark text (black).
Green Channel (G): Similar to the Red Channel, we still observe values also towards lower intensity values.
Blue Channel (B): Lower peaks towards all intensity values, minimal contribution from the blue color.

Comparison of count of pixels with higher intensity

Another idea that can help to differentiate the printed text region from handwritten region is the fact that printed text is usually written with less spacing than the handwriting. This means that if we take an equal sized selection from printed text and handwritten region the printed text selection will have more counts of pixels that have less intensity overall meaning more pixels with dark colors. For this analysis I have taken equal selection and tried to compare them.



When we analyze their histograms we observe that for the left(printed text) image we have more points appearing with less intensity than for the right image.



To make this distinction more clear I have taken out the values and their counts to an excel file and counted for image1 and image2 the number of points that have intensity less than 160. Here are the results:

	printed	handwritten
Total count of points with intensity < 160	21617	5325

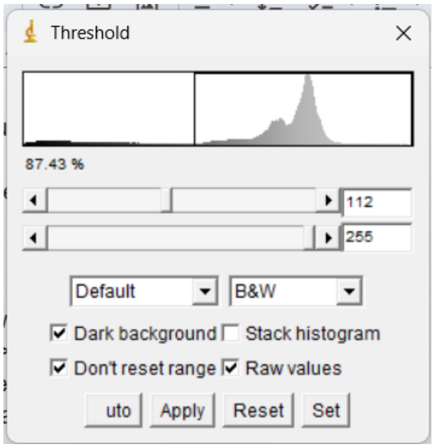
We can see that the count of pixels that have intensity less than 160 for printed text is clearly bigger than for handwritten text. This can be used to differentiate the printed text from handwriting.

Removing Background by Thresholding and Background Subtraction

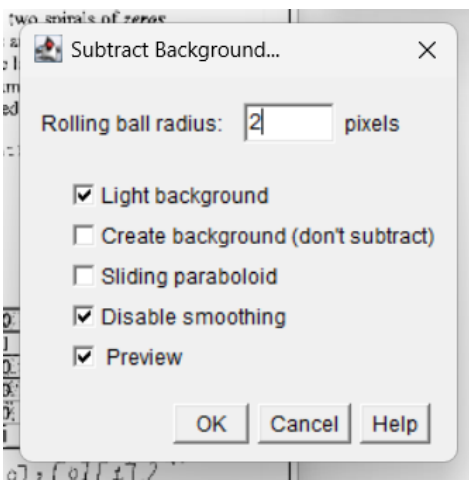
We can use thresholding + subtract background for background and foreground separation of the image.

Thresholding is used in image processing for tasks like image segmentation, object detection, or feature extraction. It simplifies the image by separating objects or regions of interest from the background.

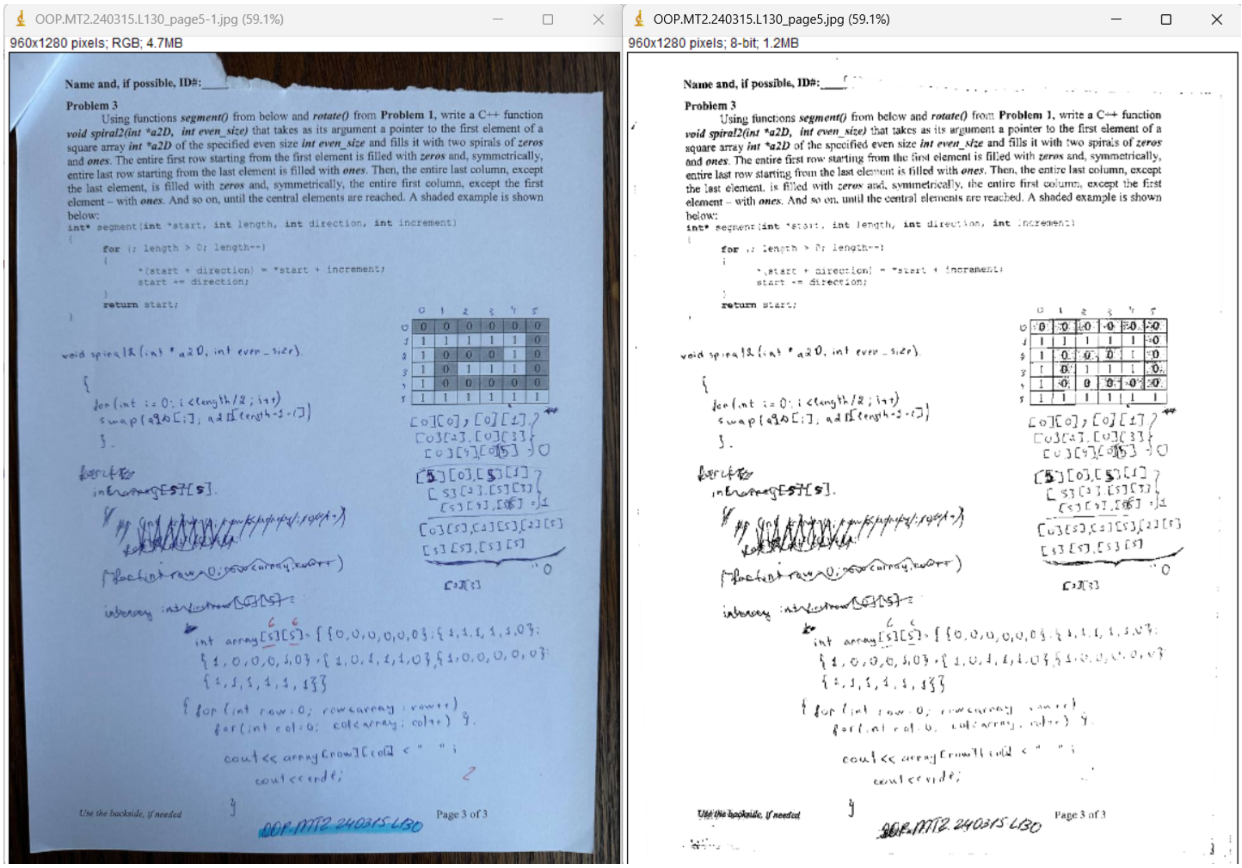
To use thresholding and background subtraction first we need to convert the image into grayscale image. For that we can do Image -> Type -> 8 bit. After this we need to select Image -> Adjust -> Threshold. We need to set the threshold to be 112, I have found that this is the optimal value which clearly separates the background with black but also doesn't distort the text on the image.



After applying thresholding to the image we can subtract the background with Process -> Subtract Background with the following configurations. It will subtract the background from the image. I have found that the optimal value for radius is 2.\



Lets see the results before and after applying Thresholding + Subtract Background.



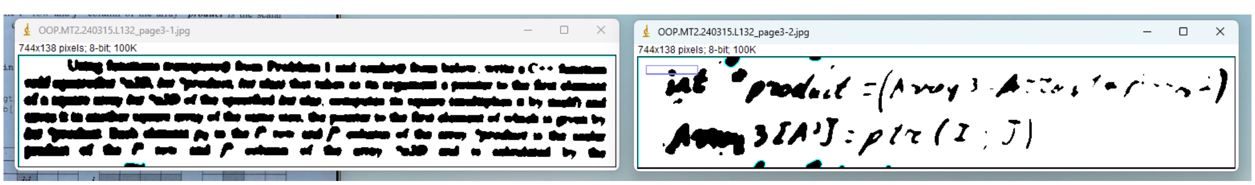
Conclusion: We can use this technique to remove the background from the images. This will help us to concentrate on the text of the image.

Analyzing the printed and handwritten images by analyze particles

To analyze particles first we need to convert the images to appropriate states. For that I have applied blurring to printed and handwritten texts

- Process -> Filters -> Gaussian Blur
- Convert Image to 8bit image by Image -> Type -> 8bit
- Adjusting threshold

The results are the following:



Then I run set measurements and analyze particles to observe the differences.
By comparing the results for printed and handwritten we observe several differences:

- **Mean**: 165.708 for printed and 231.223 for handwritten. A lower mean value, such as 165.708, suggests that the average pixel intensity within the particles detected in the printed text image is relatively lower or darker. This could indicate that the printed text regions are less bright on average compared to the handwritten text image. A higher mean value 231.223, suggests that the average pixel intensity within the particles detected in the handwritten text image is relatively higher or brighter. This could indicate that the handwritten text regions are, on average, brighter compared to the printed text image.

- **Angle**: 179.979 for printed and 0.0005083 for handwritten. 179.979: For printed text an angle very close to 180 degrees suggests that the major axis of the best-fit ellipse for the particles detected in the printed text image is almost horizontal or oriented parallel or antiparallel to the reference axis (x-axis). Visually, this would appear as a nearly horizontal orientation of the particles. For handwritten text a very small angle, close to zero. This indicates that the major axis of the best-fit ellipse for the particles detected in the handwritten text image is nearly vertical or close to aligning with either the x-axis or y-axis, resulting in a nearly vertical orientation of the particles.

- **Kurt**: -1.605 for printed and 5.828 for handwritten. A negative kurtosis value typically indicates a distribution that is flatter or less peaked than a normal distribution. It suggests that the intensity distribution within the particles in the printed text image is relatively more spread out than a normal distribution, with fewer extreme values (outliers) and less pronounced tails compared to a normal distribution. A positive kurtosis value generally indicates a distribution that is more peaked or has heavier tails than a normal distribution. This suggests that the intensity distribution within the particles in the handwritten text image has more extreme values or outliers compared to a normal distribution. It indicates a greater concentration of pixel intensities around the mean.

- **Ferret**: 756.69 for printed and 381.951 for handwritten. A Feret diameter of 756.69 indicates that the largest distance between any two points along the perimeter of the particles detected in the printed text image is 756.69 pixels. This suggests that the printed text regions contain particles or objects with larger linear extents or widths compared to the handwritten text or other detected features in the image. A Feret diameter of 381.951 indicates that the largest distance between any two points along the perimeter of the particles detected in the handwritten text image is 381.951 pixels. This suggests that the handwritten text regions contain particles or objects with smaller linear extents or widths compared to the printed text or other detected features in the image.

- **Perim**: 1769.799 for printed and 924.518 for handwritten. A larger perimeter value like 1769.799 suggests that the boundary of the particles in the printed text image is more extensive or more irregular compared to the handwritten text or other detected features in the image. A perimeter value of 924.518 suggests that the boundary of the particles in the handwritten text image is less extensive or less irregular compared to the printed text or other detected features in the image.