

Semester Thesis

Image Processing and Nanoparticle Quantification for Effective Biomarker Detection in Sepsis Diagnostics

Autumn Term 2023

Contents

Abstract	iii
1 Introduction	1
1.1 Background & Motivation	1
1.1.1 Neonatal Sepsis	1
1.1.2 NEOSENS, a 'start-up in creation'	1
1.1.3 The Technology	2
1.2 Objectives	2
1.2.1 Challenges	2
1.2.2 My work	3
2 Literature Review	5
2.1 Metrics	5
2.2 Endpoint Analysis	6
2.2.1 Image processing	6
2.2.2 Nanoparticle identification and counting	7
2.3 Time-resolved Analysis	9
3 Code Structuration	11
3.1 Modularization	11
3.2 Configuration and Dependency Management	11
3.3 Documentation	12
4 Endpoint Images Database	13
4.1 Database	13
4.2 Region of interest and Masking	14
4.3 Thresholding Algorithms	15
4.3.1 Global Fixed Thresholding Algorithm	15
4.3.2 Local Adaptive Thresholding Algorithm	18
4.3.3 Global Median Thresholding	20
4.3.4 Otsu Thresholding Algorithm	22
4.4 Processing our Dataset: Noise Reduction	23
4.4.1 Metrics Implementation	24
4.4.2 Gaussian Filtering	25
4.4.3 K-space Filtering	27
4.4.4 Background Subtraction	29
4.4.5 Optimal methods	32
4.5 Processing our Dataset: Contrast Enhancement	33
4.5.1 CLAHE	33
4.6 Final Pipeline and Potential Improvements	35
4.6.1 Results Discussion	36
4.6.2 Potential Improvements	37

5 Time-resolved Images Database	39
5.1 Creation of a New Database	39
5.1.1 Imaging Setup	39
5.1.2 From Video to Database	41
5.2 Endpoint Analysis	41
5.3 Time-resolved Analysis	43
5.3.1 Temporal Average	43
5.3.2 Differential Imaging	44
6 Conclusion	47
Bibliography	50
A Appendix	51
A.1 Local Adaptive Thresholding Algorithm	51
A.1.1 Constant Value Tuning	51
A.1.2 Window Size Tuning	52

Abstract

Neonatal Sepsis is a life-threatening medical emergency that requires immediate treatment upon suspicion, necessitating an early-stage diagnosis for effective treatment. The emergence of new optical biosensing technologies is showing promising results for fast and accurate biomarker concentration detection, which in turn requires performant image analysis software for the detection of single nanoparticles.

In this thesis, image processing methods for nanoparticle processing and detection have been tested and compared to find the optimal pipeline. This comparison was made using various analytical metrics and visual inspection. The behaviors and efficacy of these techniques have been reported.

The results indicate that processing using Gaussian filters and K-space filters offers a significant improvement in image quality. When combined with the right segmentation tool, such as a global fixed threshold, it can provide a low limit of detection ($< 0.05\text{ng/mL}$) with a short computation time, making it suitable for testing in clinical conditions.

Chapter 1

Introduction

1.1 Background & Motivation

1.1.1 Neonatal Sepsis

Neonatal Sepsis (NS) is a life-threatening medical emergency where immediate antibiotic treatment is crucial. NS is considered a global health concern, accounting for 13% of neonatal mortality. Among those, 99% occur in Low- and Middle-Income Countries (LMICs).

At an early stage, NS is hard to diagnose, as it has very aspecific clinical signs that can overlap with other conditions. Treating newborns with NS is considered "the nightmare of all neonatologists" as described by Egyptian neonatologists from Ain Shams Hospital. NS is accompanied by uncertainty and emotional burden. Neonatologists in LMICs mostly rely on their "clinical sense" for the early diagnosis of NS, as current laboratory tests are expensive, invasive and often not reliable.

This situation leads to the overdiagnosis and the overtreatment of approximately 12% of all neonates with antibiotics for weeks. This long and unnecessary treatment adds huge financial costs for the whole healthcare system, threatens neonates' health and contributes to the global increase of antibiotic resistance.

1.1.2 NEOSENS, a 'start-up in creation'

NEOSENS is a 'start-up in creation' that I co-founded with Marco Fumagalli and Karim Zahra.

We are developing a diagnostic device that combines optical biosensing technology with an advanced predictive model. This integration aims to generate a risk score for sepsis, to assist both trained and untrained medical professionals in making informed decisions about the admission of newborns to the Neonatal Intensive Care Unit (NICU).

We plan to launch a pilot project with our prototype at Ain Shams Hospital in April 2023. Our goal is to test, and validate a rapid, accurate, and robust diagnostic solution that can adapt to the varying clinical parameters of a new environment, such as temperature fluctuations, humidity levels, quality of consumables, and transportation conditions.

1.1.3 The Technology

The optical biosensing technology[1] we are using has been developed at the Bio-nanophotonic System Laboratory led by Prof. Dr. Hatice Altug at EPFL. This plasmonic biosensor puts together the phenomenon of Extraordinary Optical Transmission, with a common ELISA sandwich immunoassay to allow the detection of proteins on a gold chip surface.

When a gold nanoparticle (AuNP) binds to the surface, it induces a local change in the refractive index of the surrounding medium, which in turn reduces the intensity of light detected in the far-field.

A bright-field imaging setup is employed to identify these distortions. Binding events manifest as high-contrast spots on plasmonic intensity heat maps, allowing us to infer the number of bound AuNPs from the image and accurately deduce the concentration of proteins.

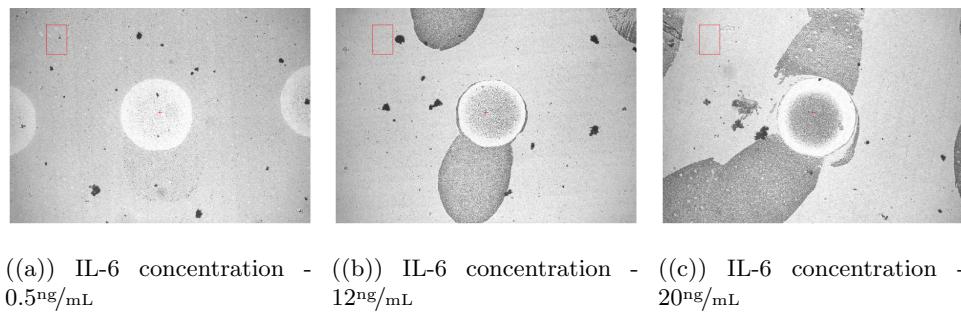


Figure 1.1: Raw images obtained with a bright-field imaging setup

1.2 Objectives

Our focus for this semester thesis is to develop a robust image processing system that can efficiently and accurately differentiate surface-bound gold AuNPs from those in solution, count individual AuNPs, and achieve a Limit of Detection (LOD) of $< 0.01\text{ng/mL}$ - all within a timeframe of 5 minutes.

1.2.1 Challenges

Adapting a reader device for industrial use, particularly in LMICs, presents unique challenges due to the differing properties from those of a laboratory microscope. Factors such as cost constraints, transportation, and other operational parameters introduce several potential challenges in our pilot project, including:

- **Environmental Interference:** Non-specific binding, material noise, and ambient noise can interfere with the accurate detection of bound AuNPs.
- **Differentiation:** It's essential to differentiate between surface-bound AuNPs and those freely moving in solution to ensure accuracy in biomarker detection.
- **Sample Inconsistencies:** Variations in sample backgrounds can lead to inaccurate values and nonspecific adsorption issues.
- **Detection Sensitivity:** The detection limit is defined by the minimum number of biomarker molecules per unit area on the sensor surface, which translates to identifying a single AuNP linked via one biomarker molecule within the full field of view.

1.2.2 My work

The thesis aims to implement state-of-the-art methods from the literature on similar problems, identify relevant metrics to quantify the results, and select the most suitable methods that best address our specific needs.

To achieve our objective, we have separated the problem into **two primary tasks**.

1. Determine the optimal image processing techniques that minimize environmental noise, such as non-specific binding, material noise, and background variations. We want to enhance the contrast between the dark spots, indicative of AuNPs, and the background, thus facilitating their detection and subsequent quantification.
2. Accurately identify and measure the concentration of AuNPs, represented as dark spots, using a method that ensures a LOD of less than $0.01^{\text{ng/mL}}$.

The entire computational process must be designed for efficiency, delivering results within a five-minute timeframe.

Chapter 2

Literature Review

This chapter provides an in-depth overview of existing literature on digital imaging and nanoparticle counting. The focus is on understanding the metrics used to differentiate techniques, their challenges and advantages, and the criteria for discussing results. Additionally, this chapter discusses traditional methods for endpoint analysis, categorized and explained in various subsections. A contrasting approach using time-resolved images for dynamic quantification is also explored.

2.1 Metrics

In evaluating the effectiveness of digital imaging techniques, several key metrics are considered:

- **Limit of Detection (LOD):** The LOD is the lowest quantity of a substance that can be distinguished from the absence of that substance (a blank value) with a stated confidence level.
- **Signal-to-Noise Ratio (SNR):** This metric is crucial for assessing the clarity and quality of the captured image. The SNR compares the Region of Interest (ROI) count against the background count.

$$\text{Formula: } SNR = \frac{\mu_{signal}}{\sigma_{noise}} \quad (2.1)$$

where μ_{signal} is the mean intensity of the signal (e.g., the ROI) and σ_{noise} is the standard deviation of the background noise.

- **Contrast-to-Noise Ratio (CNR):** The CNR is used to measure the contrast of a signal (e.g. AuNPs) to the background noise.

$$\text{Formula: } CNR = \frac{|\mu_{signal} - \mu_{background}|}{\sigma_{noise}} \quad (2.2)$$

where μ_{signal} is the mean intensity within the ROI, $\mu_{background}$ is the mean intensity within the background region, and σ_{noise} is the standard deviation of the background noise.

- **Weber Contrast:** Weber contrast is used to measure the difference in intensity between an object and its background relative to the background intensity.

$$\text{Formula: } \text{WeberContrast} = \frac{|\mu_{signal} - \mu_{background}|}{\mu_{background}} \quad (2.3)$$

where μ_{signal} is the mean intensity of the object and $\mu_{background}$ is the mean intensity of the background.

- **Structural Similarity Index (SSIM):** SSIM is a method for measuring the similarity between two images.

$$\text{Formula: } \text{SSIM} = \frac{(2\mu_{original}\mu_{proc} + c_1)(2\sigma_{original,proc} + c_2)}{(\mu_{original}^2 + \mu_{proc}^2 + c_1)(\sigma_{original}^2 + \sigma_{proc}^2 + c_2)} \quad (2.4)$$

where $\mu_{original}$ and μ_{proc} are the average pixel values, and $\sigma_{original}$, σ_{proc} , and $\sigma_{original,proc}$ are the variances and covariance of the original and processed images respectively. Constants c_1 and c_2 are used to stabilize the division.

2.2 Endpoint Analysis

As shown in Figure 1.1, the initial database we worked on had images taken at a specific time point - two hours after deposition (stable state) of the AuNPs. Literature relevant to this method was explored, categorizing the process into, first, image processing and then, AuNPs counting.

2.2.1 Image processing

Effective image processing is crucial for endpoint analysis. The primary goal is noise reduction to enhance the SNR. Techniques include:

Filtering

Various filtering methods are employed to address different types of noise, such as shot noise, pattern noise, or artifacts from moving objects in the far field. The right filter selection significantly improves image clarity.

K-space filtering has been used in [2] and has shown great results for highly precise detection of single binding events with a wide dynamic range ($LOD \approx 2.8\text{pg/mL}$). They performed a noise reduction in the image sequence by performing a spatial fast Fourier transform, itself known for its capability of selective attenuation of high-frequency components.

Other research has revealed that among various filtering techniques, **local averaging methods** (such as mean and Gaussian filters) produce the most favorable outcomes [3].

These methods of averaging are realized through the convolution of the image $I(x, y)$ with a mask $m(x, y)$, as described by the equation:

$$I(x, y) * m(x, y) = \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} I(x+i, y+j)m(i, j) \quad (2.5)$$

For the mean filter, the mask $m(x, y)$ is defined as $\frac{1}{w^2}$, which ensures an equal contribution from each pixel in the neighborhood. In contrast, the mask for the Gaussian filter $m(x, y) = G(x, y)$ is given by $\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$.

Background subtraction

This technique involves removing the background noise from the image, which is essential for isolating and accurately counting the AuNPs.

As detailed in the referenced research paper[4], background noise can be effectively reduced by subtracting an out-of-focus image from the original image, followed by the subtraction of a median-filtered image. This process can achieve a LOD of approximately 1.8pg/mL

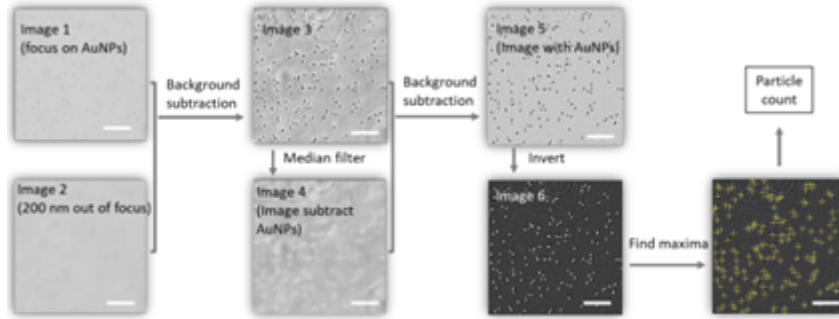


Figure 2.1: Schematic of image processing methods - Supplement Figure S3. from *Gradient-Based Rapid Digital Immunoassay for High-Sensitivity Cardiac Troponin T (hs-cTnT) Detection in 1 μ L Plasma*[4]

After enhancing the SNR, the next step is to increase the contrast between AuNPs and the background. To facilitate the differentiation of pixels with or without surface-bound AuNPs, we apply contrast enhancement techniques.

Contrast Enhancement

Among various methods, Contrast Limited Adaptive Histogram Equalization (CLAHE)[5, 6] is particularly suited for our application.

Unlike ordinary Adaptive Histogram Equalization or high-pass filtering, CLAHE does not excessively amplify noise, which could otherwise lead to inaccuracies and potential false positives in AuNP identification.

Its key parameters include the clip limit, α , limiting the enhancement to reduce noise amplification and grid size, $w \times h$, defining the regions for histogram equalization. The process involves clipping the histogram at α and redistributing excess pixels across all intensities, thus ensuring a more uniform enhancement.

2.2.2 Nanoparticle identification and counting

While numerous methods exist for processing endpoint images, for AuNP identification one type is almost universally used: Thresholding algorithms.

Thresholding algorithms

Though simple to implement, thresholding algorithms remain the state-of-the-art method for identifying AuNPs in bright-field imaging. There are numerous variations of this algorithm, each with its own advantages and drawbacks.

The most basic form is the **Global Fixed thresholding**, which has been noted in multiple papers for its effectiveness, especially when the optimal threshold is determined. For instance, in this paper[1], the optimal value was chosen to detect the maximum number of AuNPs while maintaining a low rate of false positives. Another approach, as seen in this research[6], involves using a threshold set at half the intensity of the maximum spike.

However, images are subject to intensity variations depending on the characteristics of the optical setup. Therefore, dynamically determining the global threshold value is crucial for robust results that can adapt to these variations.

In their paper[7], Zhang et al. employed **Median Global Fixed thresholding** to dynamically set the global threshold for binarization. They based the threshold on the median value of the simulated signal image obtained prior to the addition of the detection antibody, effectively representing the background.

Furthermore, instead of applying a global threshold to the entire image, a local threshold for each pixel can be calculated, making the process more robust to variations in illumination within the image. In their research[5], a unique pipeline was implemented involving an **Adaptive threshold** followed by binarization across multiple equally-sized sub-Fields of View (FOV). These sub-FOVs are then averaged to obtain a mean count.

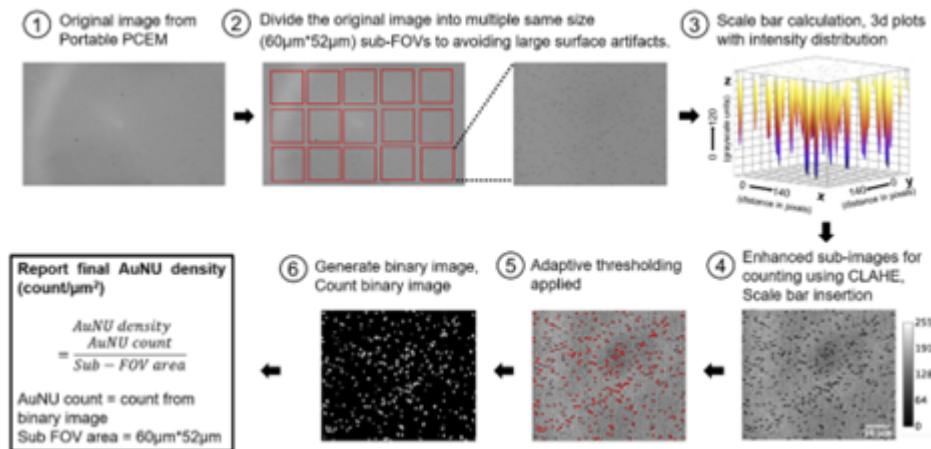


Figure 2.2: Schematic of image processing steps - Figure 3. from *A compact photonic resonator absorption microscope for point of care digital resolution nucleic acid molecular diagnostics[5]*

Adaptive threshold computes the threshold $T(x, y)$ for a pixel at (x, y) based on the average intensity in a surrounding window of size w , adjusted by a constant C . The formula is given by:

$$T(x, y) = \frac{1}{w^2} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} I(x + i, y + j) - C \quad (2.6)$$

where $I(x, y)$ is the pixel intensity.

AuNPs quantification

Once the AuNPs are labeled through binarization using a thresholding algorithm, the next step is to count the number of AuNPs. Two methods are available for this purpose.

1. Estimating the number of AuNPs, which can lead to some inaccuracies due to aggregation or noise.
2. Quantifying the number of pixels identified as AuNPs. By comparing this pixel count with reference values, we can deduce the concentration of biomarker molecules in the blood.

2.3 Time-resolved Analysis

Time-resolved imaging is extensively utilized in nanoplasmonic digital imaging to capture a series of images or frames at distinct time points, enabling the analysis of temporal changes. This method is particularly suited to our problem, where dynamic changes are observed due to the binding of AuNPs, the movement of unbound AuNPs in the far-field, and the presence of background noise.

After extensive research into the existing literature related to my project, many studies were found to incorporate time-resolved image processing techniques to enhance performance compared to the techniques presented in section 2.2.

It is important to note that the techniques outlined in section 2.2 can also be applied to a database of time-resolved images, although this may sometimes be demanding in terms of computational efficiency.

Temporal Average:

Temporal averaging (as used in [2, 3, 4, 8, 9] ...) is one of the most efficient ways to enhance image quality and reduce noise for the detection of bound AuNPs on plasmonic nanoarray surfaces. By averaging the intensity values of a pixel over a set window size, we can eliminate random noise and moving objects from the images.

The process is described mathematically as follows: if I_1, I_2, \dots, I_n represent the intensity values of a pixel in successive frames, the temporally averaged intensity \bar{I} is calculated using the formula

$$\bar{I} = \frac{1}{n} \sum_{i=1}^n I_i \quad (2.7)$$

where n is the number of frames used in the averaging process, also called window size.

Differential Analysis:

Differential bright-field imaging analysis is a technique used to capture and analyze changes between two or more images, thereby highlighting and tracking single AuNPs binding events in real time. This method involves subtracting an initial image frame from subsequent ones, enabling the detection of changes between the frames, such as newly bound AuNPs.

The efficiency of this technique lies in its ability to significantly reduce background and noise through subtraction, with high local spikes indicating the presence of newly bound AuNPs (as demonstrated in these papers[2, 8, 9]). This dynamic approach allows for rapid results but also as it can quantify AuNPs binding over a specific time frame. By comparing the evolution of a standard curve with a calibration curve, it determines the correct fit, offering a clear and quantifiable measure of biomarker concentration.

Chapter 3

Code Structuration

Before going into the details of my research, I would like to make a side note and clarify that the code written will be utilized by future engineers working at NEOSENS. Having a codebase that is clear, understandable, well-structured, and adhering to good conventions is essential for efficient work.

This chapter outlines the practices adopted in the research to achieve these objectives.

3.1 Modularization

For clear organization, the first step involves dividing the code into various modules, each responsible for a specific functionality.

Currently, this includes:

- *image-processing* module for accurate counting.
- *nanoparticles_counting* module for AuNPs identification and quantification.
- *image-metrics* module for calculating metrics and image parameters.
- *video-to-database* module for converting videos into a usable database format for further quantification.
- ...

3.2 Configuration and Dependency Management

Efficient understanding and management of global variables, such as the threshold value for our Global Fixed thresholding algorithm or the window size for our Temporal average, are achieved through a *config.json* file.

Similarly, our code requires multiple third-party libraries, such as *OpenCV*, *NumPy*, *Pandas*, and *Scikit-Image*. It is, therefore, necessary to maintain a *requirements.txt* file that lists these needed libraries and facilitates their efficient installation for any new user.

3.3 Documentation

The final step of structuring was to write comprehensive documentation, which includes comments in the code, README files, and other documentation resources.

A typical example is the *load_grayscale_images* function. This function includes a docstring, serving as documentation to outline the function's main purpose and its parameters. Type annotations indicate the expected types of values and return, complemented by error handling to prevent unnecessary debugging issues.

```
# Helper Functions
def load_grayscale_images(directory_path: str) -> Tuple[List[np.ndarray], List[str]]:
    """
    Load images from the specified directory and convert them to grayscale.

    Args:
        directory_path (str): The file path of the directory containing the images.

    Returns:
        tuple: A tuple containing two elements:
            - A list of grayscale images.
            - A list of filenames corresponding to the loaded images.

    Raises:
        FileNotFoundError: If the specified directory does not exist.
        ValueError: If no images are found in the directory.
    """
    if not os.path.exists(directory_path):
        raise FileNotFoundError(f"The specified directory does not exist: {directory_path}")

    image_filenames = [filename for filename in os.listdir(directory_path) if filename.endswith(ALLOWED_IMAGE_EXTENSIONS)]
    if not image_filenames:
        raise ValueError("No images found in the specified directory.")

    print(f"Attempting to load {len(image_filenames)} images...")
    grayscale_images = [cv2.imread(os.path.join(directory_path, filename), cv2.IMREAD_GRAYSCALE) for filename in image_filenames]

    if not grayscale_images:
        raise ValueError("One or more images could not be loaded. Please check the file paths and formats.")

    return grayscale_images, image_filenames
```

Figure 3.1: Main module - *load_grayscale_images* function - convention style

Of course, everything was organized using a version control system, namely GitHub. Adopting the right practices from the beginning is crucial to prevent time loss in the future.

Even though it is not directly research-related, I felt it was important to explain these practices that are often overlooked in personal projects and during our studies but are essential when working as a team, as we will in the future.

Chapter 4

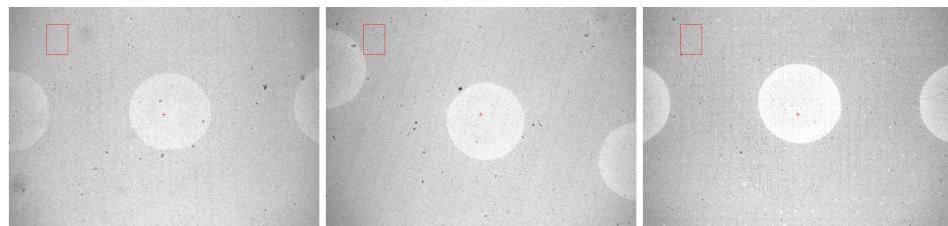
Endpoint Images Database

This chapter goes into the details of the implementation and the results of methods identified during the literature research. The implementation is clearly explained, including our chosen approach, as well as the challenges encountered during the coding process. The results will be discussed, compared, and justified with a straightforward objective: to find an optimal pipeline for our specific case and to fulfill the requirements outlined in Section 1.2.

4.1 Database

In this section, I would like to provide clarity regarding the specific images used and outline the reasoning behind their selection.

The initial dataset comprised 46 raw images, each representing various concentrations of biomarker IL-6 in serum. These concentrations included control samples (blank, 0^{ng/mL}) and samples with biomarker concentrations of 0.05^{ng/mL}, 0.1^{ng/mL}, 0.3^{ng/mL}, 0.5^{ng/mL}, 1^{ng/mL}, 5^{ng/mL}, 12^{ng/mL}, and 20^{ng/mL}. For each concentration level, between 4 to 6 images were used to ensure a comprehensive representation and analysis.



((a)) Blank sample (0^{ng/mL}) ((b)) 0.05^{ng/mL} concentration ((c)) 0.1^{ng/mL} concentration

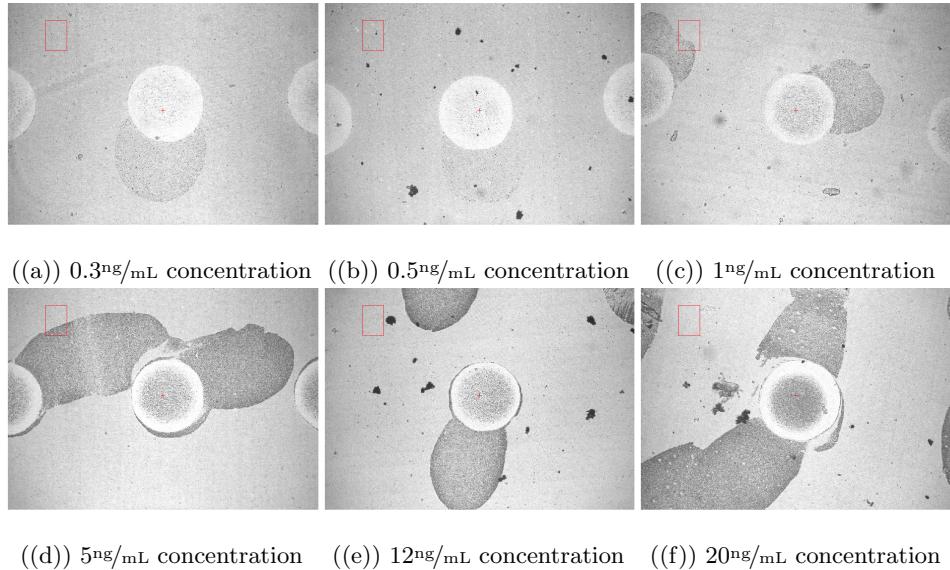


Figure 4.2: Endpoint Images Dataset - Representative images of IL-6 concentrations in serum.

4.2 Region of interest and Masking

Implementation

In the provided images, Aggregations of AuNPs and significant noise in certain areas that could compromise the accuracy of our results are observed. To mitigate this, we designate a ROI.

As shown in Figure 4.2, our ROI is typically a circular area centrally located within the image. The initial step involved determining the ROI and applying a binary mask to eliminate irrelevant information for subsequent quantification.

The ROI's radius was established visually and empirically after numerous trials to be 85 pixels. This dimension was selected to keep the broadest area possible, thereby extending our counting range and enhancing accuracy.

Using the *OpenCV* library, we defined the center of the ROI using the command *cv2.EVENT & LBUTTONDBLCLK* and subsequently drew the circle with *cv2.bitwise & and(image, image, mask=masks[i])*.

Results & Discussion

A binary mask specific to this ROI was created, which is applied to each image using the formula *cv2.bitwise & and(image, image, mask=masks[i])*.

The outcomes were as follows:

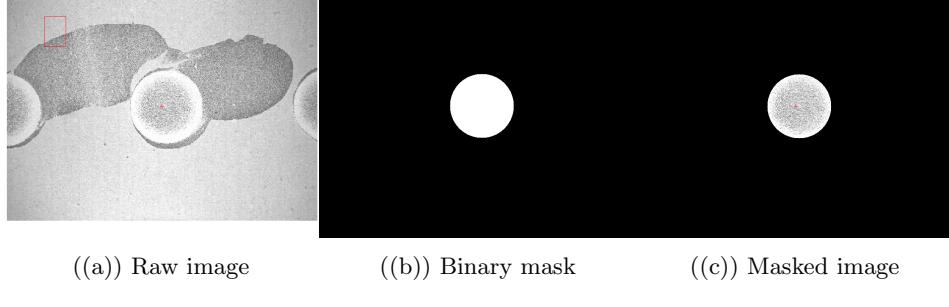


Figure 4.3: Masking process and result

With the ROI now defined and isolated, we can proceed with the identification of AuNPs.

4.3 Thresholding Algorithms

To optimize the quantification of AuNPs, the first step was to implement and compare various identification and counting techniques referenced in Section 2.2.2. These included the global fixed thresholding algorithm, global median thresholding algorithm, local adaptive thresholding, and Otsu's thresholding algorithm.

The simplest solution is often the best starting point, as we might not need to process any images to achieve our objective, which includes a low LOD, computationally efficient software, and a low rate of false positives.

For the implementation, we used the *OpenCV2* library, which provides pre-implemented functions for efficient thresholding. Our goal was to determine the optimal parameters for each algorithm to achieve the best results.

4.3.1 Global Fixed Thresholding Algorithm

Implementation

This algorithm has a single parameter to optimize: the threshold value.

If a pixel's intensity is below this threshold value, the pixel is labeled as 1; if above, it is labeled as 0. For my dataset, iteration over more than 40 potential threshold values ranging from 0 to 255 has been computed.

At each iteration:

1. Apply the segmentation technique to identify and label pixels belonging to AuNPs.
2. Retrieve the ROI and applied the masks.
3. Count the white pixels (intensity value of 255).

From the mean and standard deviation calculated over each concentration, we determined the optimal threshold value.

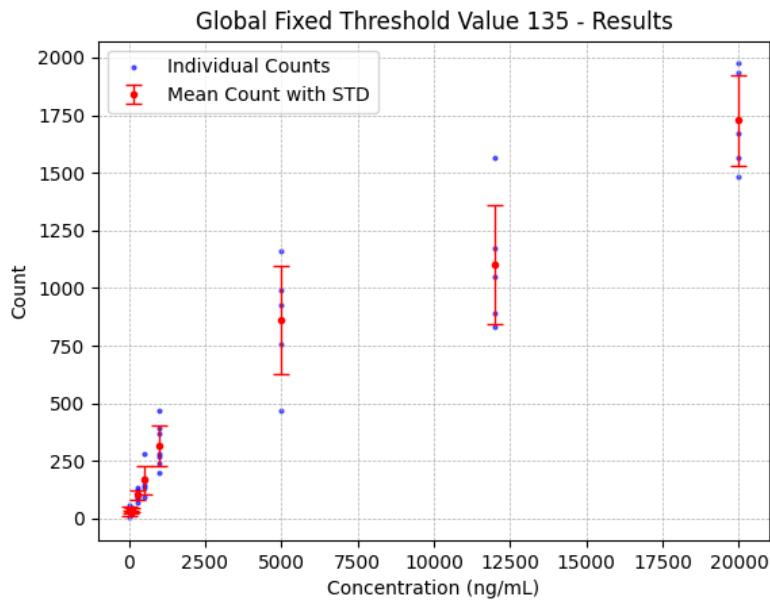


Figure 4.4: White pixel counts results for optimal fixed thresholding algorithm

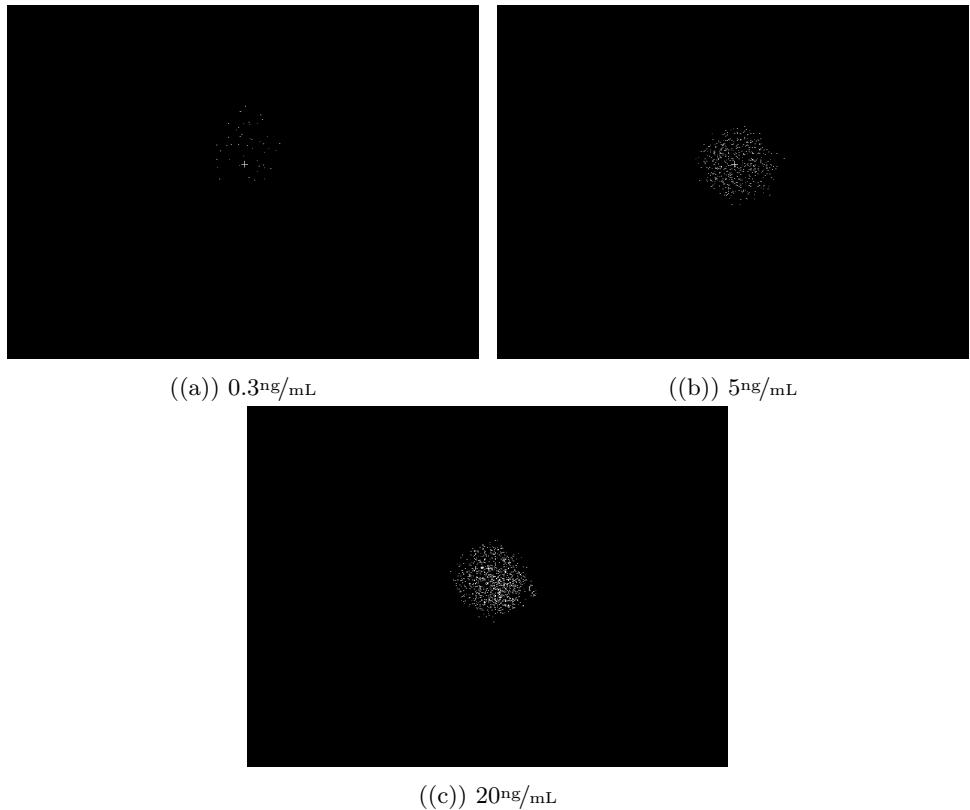


Figure 4.5: Results after binarization over multiple concentrations - Global Fixed thresholding algorithm

4.3.2 Local Adaptive Thresholding Algorithm

Implementation

As explained in Section 2.2.2, Local Adaptive thresholding allows the threshold to vary throughout the image, depending on the local neighborhood of the pixel being analyzed.

This method is particularly useful for images with varying lighting conditions. In our case, we observed a phenomenon known as 'vignetting', which leads to darker illumination toward the edges of the pictures due to the physical properties of the camera lens.

Furthermore, this technique preserves details, potentially enhancing the accuracy of our AuNP identification.

This algorithm requires fine-tuning of two parameters to find the optimal combination for our specific use case: the window size (w) and the constant (C). For each pixel, if its intensity is below the locally calculated threshold (refer to eq. 2.6), the pixel is labeled as 1; if it is above, the pixel is labeled as 0.

For my dataset, I applied adaptive thresholding to 120 different combinations, with neighborhood window sizes ranging from 3 to 21 and constant values ranging from 0 to 50.

Each iteration followed a similar pipeline as the global fixed thresholding.

Results & Discussion

As observed from Tables A.1 and A.2 (See Appendix), the impact of varying the constant value on the results is significant.

There is a need to find a trade-off between achieving a high dynamic range (which involves having a low blank count and high counts for higher concentrations as we increase C) and minimizing the number of false negatives. We observed that, above a certain value ($C = 20$ in our case), the number of false positives increased significantly, resulting in the count for a lower concentration (5^{ng/mL}) becoming higher than that for 20(^{ng/mL}).

This behavior was consistent across all the window size values tested.

Surprisingly, variations in the window size did not significantly affect our results (See Tables A.3 & A.4 in Appendix). The LOD, standard deviation, and dynamic range remained similar. This could be due to the relative uniformity of the images across different regions.

To optimize the parameters, we chose a constant value $c = 15$ and the smallest window size possible ($w = 3$) to reduce computation time, as window size did not significantly impact the results.

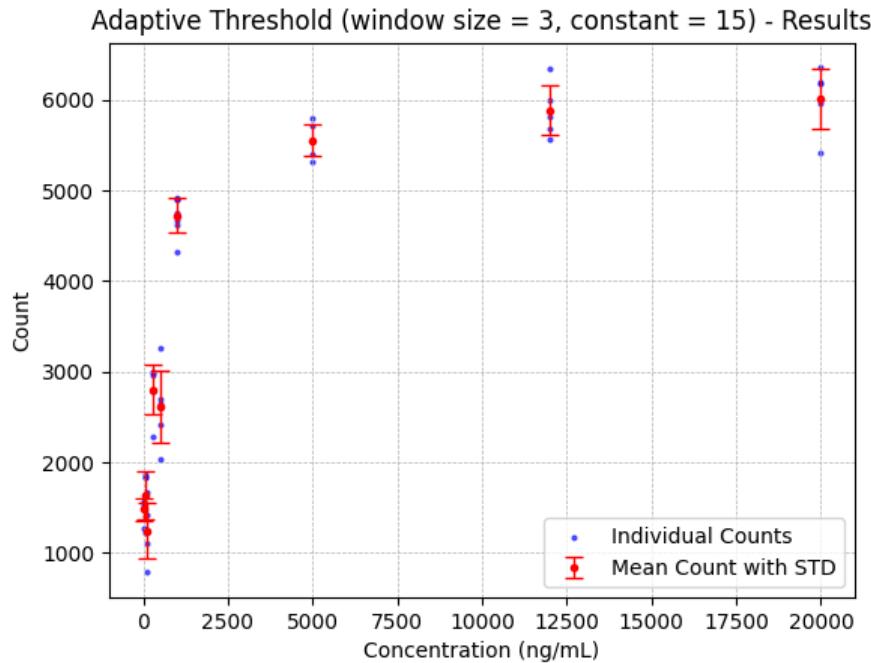


Figure 4.6: White pixel counts results for optimal Adaptive thresholding algorithm

While using Adaptive thresholding might not be the optimal solution in our current scenario, it remains a viable option should we encounter local variations in image intensity in the future.

At present, this method results in a LOD of $0.1(\text{ng/mL})$, which is higher than what we achieve using the Global Fixed threshold algorithm. The larger standard deviation observed with Adaptive thresholding limits the robustness of our approach.

Adaptive thresholding is a valuable tool to consider, however the Global Fixed thresholding algorithm currently offers more precise and reliable results for our specific application.

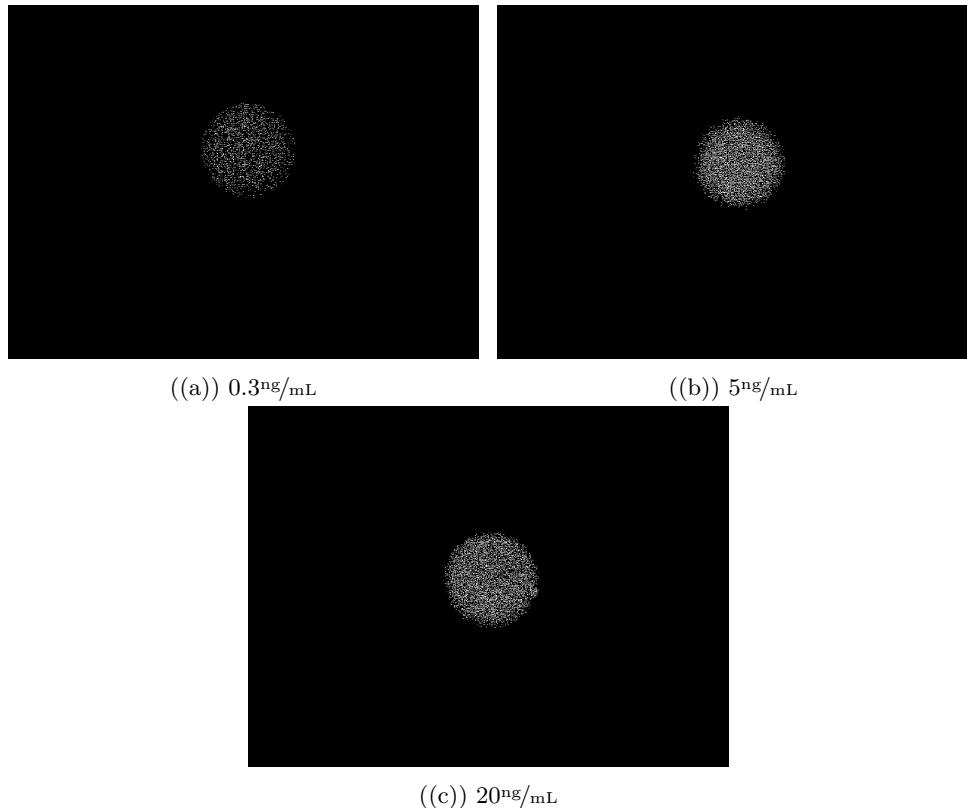


Figure 4.7: Results after Binarization over Multiple Concentrations - Adaptive Thresholding Algorithm

4.3.3 Global Median Thresholding

Implementation

Having observed that Global methods might be a suitable solution for our requirements, we still face challenges with large standard deviations and sometimes overlapping counts, as shown in Figure 4.6. Therefore, it is necessary to find a way to improve robustness and reproducibility.

To implement this method, we followed the approach used by Zhang et al. in their paper[7], where the threshold value is set to the median value of the blank sample. The threshold value was calculated by averaging the median values across our four blank samples.

Results & Discussion

The median threshold value derived from our database is 203.

However, this method did not yield satisfactory results; we observed overlapping averages, poor LOD, and larger standard deviations. The Median thresholding algorithm does not appear to be a viable solution for endpoint analysis in our case.

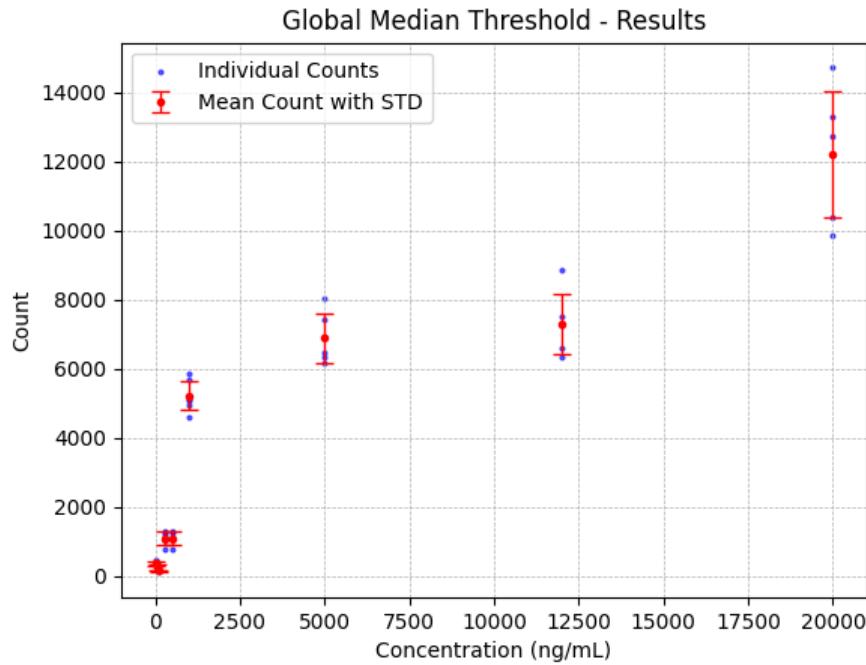


Figure 4.8: White pixel counts results for Median thresholding algorithm

The Median thresholding algorithm behaves similarly to a Global Fixed thresholding algorithm once the threshold value is determined. Since we already know that the optimal threshold value for fixed thresholding in our case is 135, we are unlikely to achieve better results using the Median thresholding algorithm.

Nevertheless, It is important to explore this method further with time-resolved images. In such scenarios, the median value of AuNP deposition might be more representative and could potentially lead to robust and accurate results.

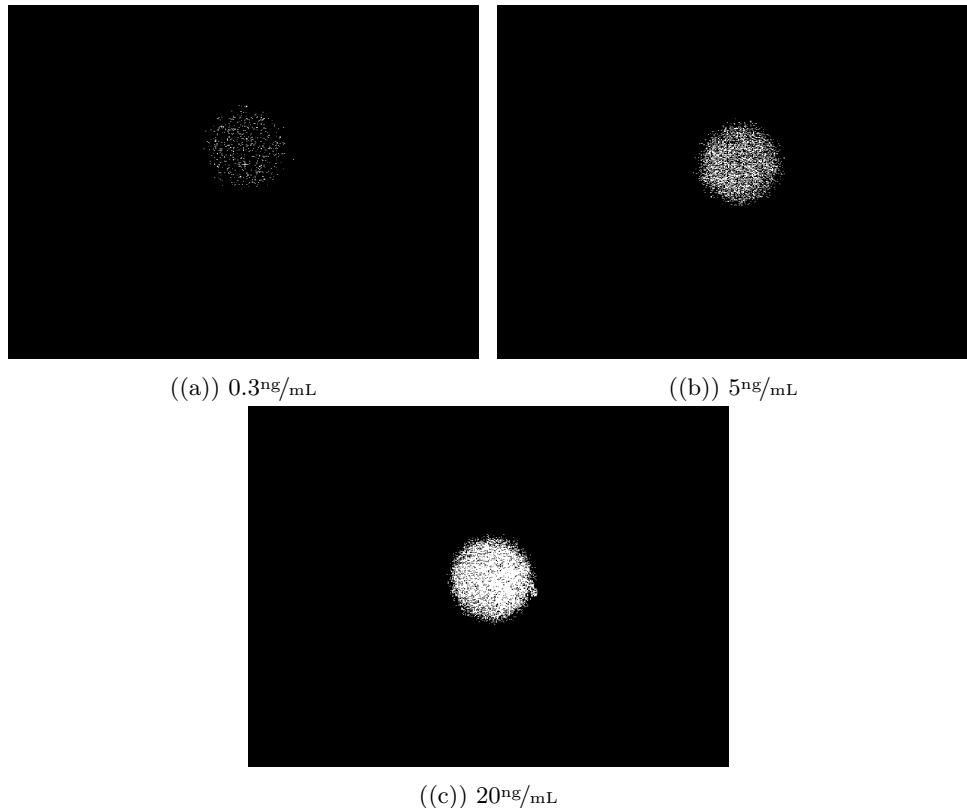


Figure 4.9: Results after Binarization over Multiple Concentrations

4.3.4 Otsu Thresholding Algorithm

Implementation

To conclude our exploration of thresholding algorithms, I was intrigued to implement the Otsu Algorithm, out of curiosity. The Otsu thresholding algorithm is an automatic thresholding technique that segments an image into foreground and background by maximizing inter-class variance and minimizing intra-class variance.

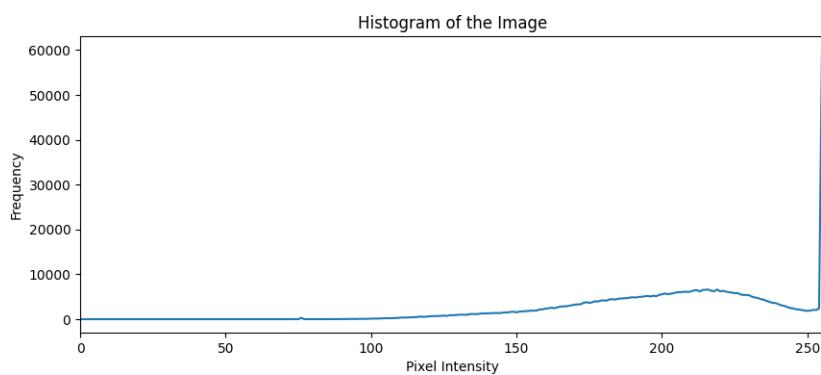


Figure 4.10: Histogram of pixel intensity of an image of concentration 5^{ng}/mL

The Otsu algorithm provides excellent results for bimodal images, where the foreground and background are distinctly separate. As represented in Figure 4.10, the

histogram of an image with a concentration of 5ng/mL exhibits two peaks. However, the peak close to the intensity of 255 is significantly larger than the one around 220.

Results & Discussion

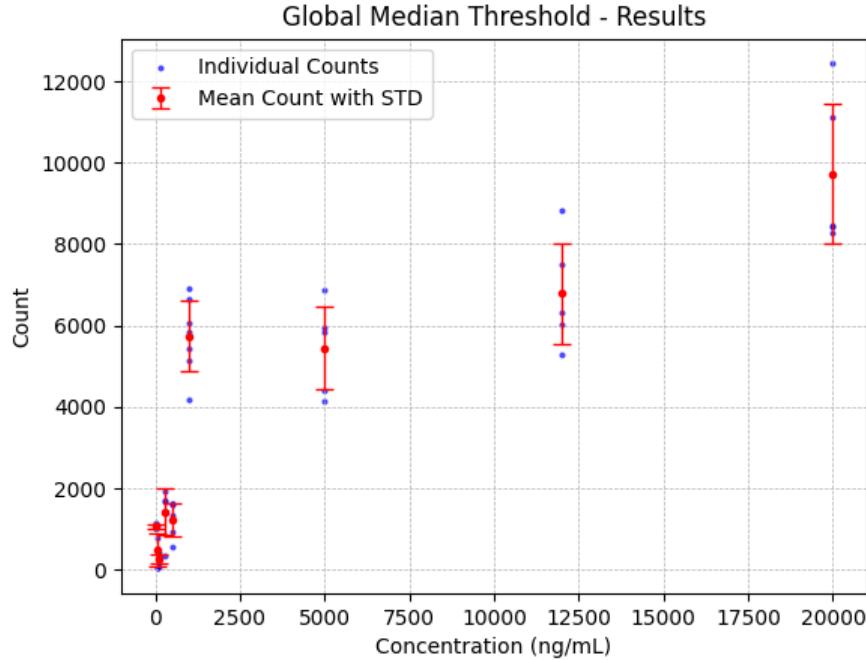


Figure 4.11: White pixel counts results for Otsu threshold algorithm

As illustrated in Figure 4.11, the Otsu algorithm did not yield favorable results, with a LOD close to 1ng/mL .

The issue arises because the peak around 220 is due to background noise rather than the binding of AuNPs. This was confirmed after encountering the issue and comparing histograms for different concentrations, which revealed a similar profile, indicating that the peaks originated from background noise.

4.4 Processing our Dataset: Noise Reduction

In the previous section (4.3), we explored various quantification techniques applicable to similar problems, their implementation, and the results they yield. Reflecting on our initial objectives (Section 1.2), it is apparent that not all criteria have been met, particularly the targeted LOD ($< 0.01\text{ng/mL}$).

Our next approach is thus to process images with two primary objectives:

- To reduce background noise and thereby increase the SNR.
- To enhance the contrast between the foreground (bound AuNPs) and the background.

4.4.1 Metrics Implementation

Multiple metrics were employed to quantify our results, including SNR (which we aim to maximize), CNR (also to maximize), SSIM (which measures the degree of similarity and should ideally be high), Weber Contrast (desired to be high), and, of course, visual inspection.

As discussed in Section 2.1 on Metrics, the calculation of SNR, CNR, and Weber Contrast necessitates a reference background signal. Hence, we used the same technique as in Section 4.2 (ROI retrieval using a binary mask) to extract the background noise, facilitating the computation of metrics for each image.

A significant challenge in computing the metrics correctly to quantify the results of my processing arose from the functionalization of surfaces with antibodies for AuNP binding. This process manifested as brighter spots on the chip's surface. Consequently, the background for AuNP binding differed from the initial background used to compute SNR, CNR, and Weber Contrast.

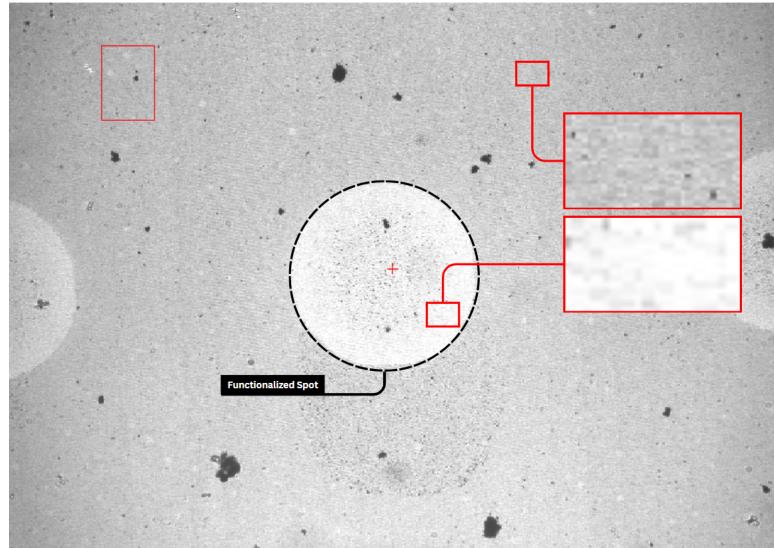


Figure 4.12: Schematic background problem

To address this, the bright spot from the blank sample was considered as the background. This solution is based on the assumption that the background noise is independent of the IL-6 concentration in the solution.

Another issue encountered was related to bright field imaging, where my foreground features dark spots. For instance, if we consider the same noise standard deviation for two scenarios—one with a signal (dark spot) and one without (only the functionalized bright spot)—the SNR calculation for the scenario without a signal will be higher because the mean signal (numerator) is larger.

The solution is to compute the SNR of the inverted image, resulting in the signal manifesting as high-intensity spots.

For a comparative perspective against other methods (described below), Table 4.3 presents the metrics for each raw image used as a benchmark.

Table 4.3: Raw Images Quality Metrics

Concentration (ng/mL)	SNR	CNR	Weber Contrast	SSIM
0	1.9	0.00	0.00	1.0
0.05	1.4	0.48	0.25	1.0
0.1	0.8	1.14	0.60	1.0
0.3	1.3	0.62	0.33	1.0
0.5	1.4	0.48	0.25	1.0
1	2.9	0.95	0.50	1.0
5	3.0	1.06	0.56	1.0
12	3.7	1.76	0.92	1.0
20	5.0	3.12	1.63	1.0

4.4.2 Gaussian Filtering

Implementation

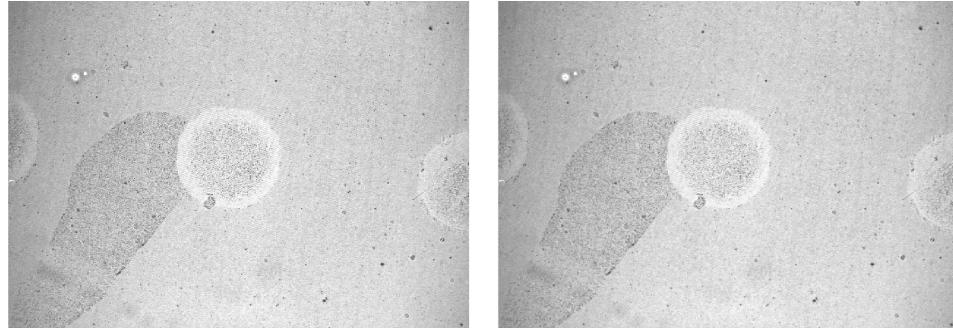
Initially, Gaussian filtering was implemented, which, according to the literature review, yields the most favorable outcomes for background noise reduction. However, I had some doubts about its effectiveness for my specific problem, as the AuNPs bindings are discerned at a small resolution (1-3 pixels), and smoothing the image might result in a loss of detail.

To implement Gaussian filtering, we used the OpenCV2 library, specifically the function `cv2.GaussianBlur(image, (kernel_size, kernel_size), sigma)`. Gaussian filtering has two parameters to optimize: its kernel size and sigma. As explained in Equation 2.5, increasing the kernel size accounts for more neighboring pixels and results in more extensive blurring. Increasing sigma leads to more pronounced blurring.

We tested 25 different combinations with kernel sizes ranging from 5 to 29 and sigma values from 0.5 to 2.5 on a dataset of 9 images (one per concentration level) and analyzed the results.

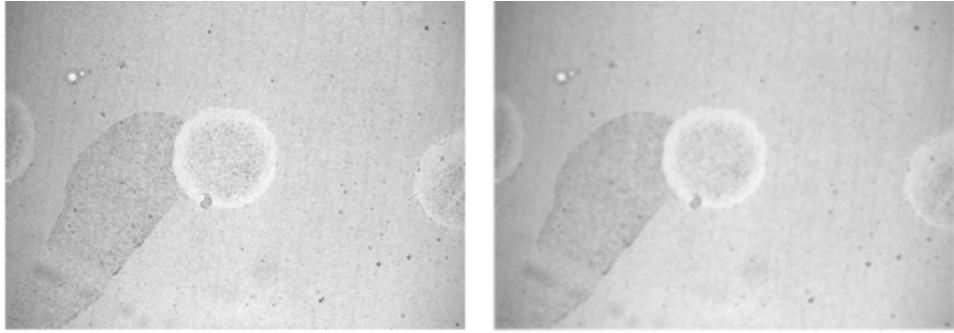
Results & Discussion

For the sake of conciseness, the results are displayed as an average of the metrics of the images, with one line representing one set of parameters.



(a)) Raw image

(b)) Gaussian filter kernel 5 sigma 0.5



((a)) Gaussian filter kernel 17 sigma 1.5 ((b)) Gaussian filter kernel 29 sigma 2.5

Figure 4.14: Comparison of raw and Gaussian filtered images

Several insights can be deduced from our results. First, visual inspection revealed that we need to find a balance between noise reduction and detail preservation. Excessive blurring can diminish the intensity of dark spots, which may even vanish, even so it is essential to maintain high contrast between our signal and the background.

Table 4.4: Gaussian filtered images quality metrics

Filter		SNR	CNR	Weber Contrast	SSIM
Raw		2.38	1.07	0.558	1.00
Kernel	Sigma				
5	0.5	2.77	0.70	0.346	0.35
5	1.0	2.74	0.64	0.294	0.34
5	1.5	2.82	0.53	0.274	0.33
5	2.0	2.58	0.41	0.212	0.32
5	2.5	2.69	0.45	0.204	0.32
11	0.5	2.61	0.39	0.200	0.35
11	1.0	2.51	0.24	0.104	0.34
11	1.5	2.27	0.12	0.054	0.33
11	2.0	2.18	0.12	0.067	0.32
11	2.5	2.43	0.11	0.063	0.32
17	0.5	2.30	0.13	0.063	0.35
17	1.0	2.31	0.10	0.048	0.34
17	1.5	2.34	0.10	0.054	0.33
17	2.0	2.34	0.14	0.046	0.32
17	2.5	2.52	0.19	0.088	0.32
23	0.5	2.39	0.09	0.045	0.35
23	1.0	2.38	0.09	0.043	0.34
23	1.5	2.36	0.13	0.067	0.33
23	2.0	2.32	0.11	0.063	0.32
23	2.5	2.49	0.12	0.065	0.32
29	0.5	2.40	0.09	0.049	0.35
29	1.0	2.43	0.11	0.054	0.34
29	1.5	2.33	0.11	0.047	0.33
29	2.0	2.37	0.16	0.088	0.32
29	2.5	2.49	0.09	0.046	0.32

This behavior is also shown in Table 4.4. It shows that as we increased the kernel size or sigma, the Weber contrast decreased.

Similarly, there is a decline in SNR with intensified filtering, attributable to the signal's disappearance and reduced intensity.

Optimal results were achieved with minimal filtering using the smallest set of parameters ($k = 5$, $\sigma = 0.5$), which led to a 16% increase in SNR (from 2.38 to 2.77) at the expense of reduced CNR and Weber contrast.

4.4.3 K-space Filtering

Implementation

With the same objective as with a Gaussian filter, K-space filtering was implemented. K-space filtering involves suppressing some frequency components of an image in its Fourier transform domain. Considering that the goal is to reduce background noise, K-space filtering has been used as a low-pass filter (LPF). It eliminates high-frequency components, often associated with noise, to retain only the low-frequency components related to our main image features.

K-space filtering is known to be efficient in reducing noise and patterns, but it sometimes comes with a loss of image fine textures, leading to overly smoothed images and the removal of some main features.

K-space Filtering Implementation is as follows:

1. Conversion to Frequency Domain: The image is transformed from the spatial domain to the frequency domain using the Fast Fourier Transform (FFT) with the numpy library, especially `np.fft.fft2`.
2. Applying the LPF: A LPF is applied to each image using a squared binary mask with a length equivalent to the given cut-off frequency.
3. Conversion Back to Spatial Domain: The filtered frequency domain representation is then shifted back using `np.fft.ifft2`, resulting in a noise-reduced version of the original image.

Challenges emerged during implementation, notably the appearance of random black pixels in the spatial representation of filtered images (Figure4.15). Initial hypotheses suggested that it was due to Fourier Transform Edge Effects, which could be solved through edge smoothing, significantly diminishing the prevalence of black pixels.

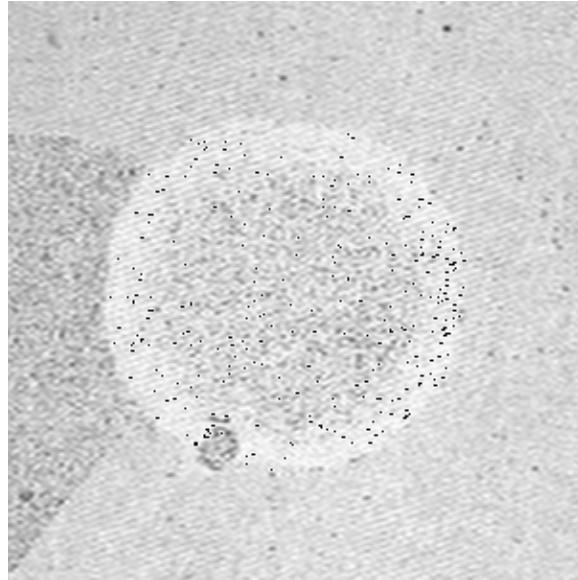


Figure 4.15: Wrap-around effect example

However, the black pixels could not be completely removed, so the second hypothesis was the correct one: We were dealing with the **wrap-around effect** due to using an incorrect representation. Storing the result of an inverse Fourier transform as an unsigned 8-bit integer (*uint8*) can cause negative values and those exceeding the maximum of 255 to be incorrectly mapped, leading to visual artifacts like random black pixels.

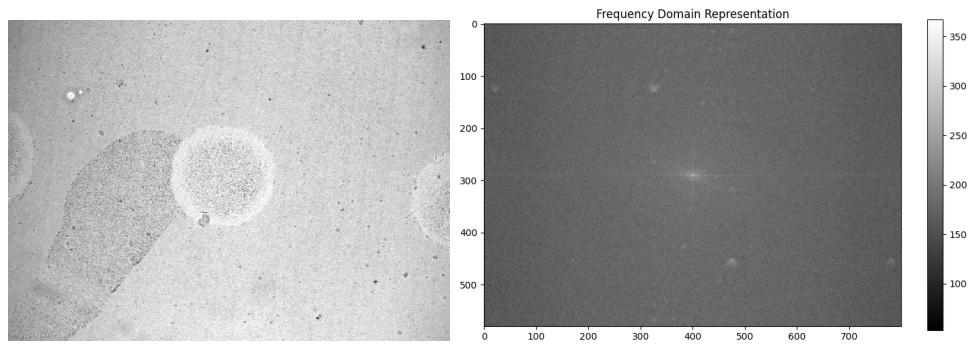
The solution was to use a more accurate representation: *float64*.

Finding the optimal cut-off frequency is not straightforward. The next step was to understand which components are represented in the Frequency domain and should be discarded.

58 cut-off frequencies were tested, ranging from 10 to 300 in increments of 5, on a dataset of 9 images (one per concentration level).

Results & Discussion

To understand the effects of the filtering, the frequency domain representations are visualized.



((a)) Spatial domain representation

((b)) Frequency domain representation

Figure 4.16: Spatial & frequency domain representation of a raw image

In the frequency domain representation (Figure 4.16(b)), 2 main problems can be identified:

1. **Instrumentation artifacts:** The two bright spots represent instrumentation artifacts caused by the equipment during image acquisition. There is a diagonal repeating pattern bringing noise to the image. It is necessary to choose the adapted cut-off frequency to block them.
2. **Uniform brightness across the spectrum:** It indicates an evenly distributed frequency content, meaning there is high noise in the spatial image. Removing all the noise is not possible; however, our goal is to cut high-frequency components while keeping the details, which will be one of our focuses while choosing the cut-off frequency.

A cut-off frequency below 160 was used to eliminate the pattern, as shown in Figure 4.17(b).

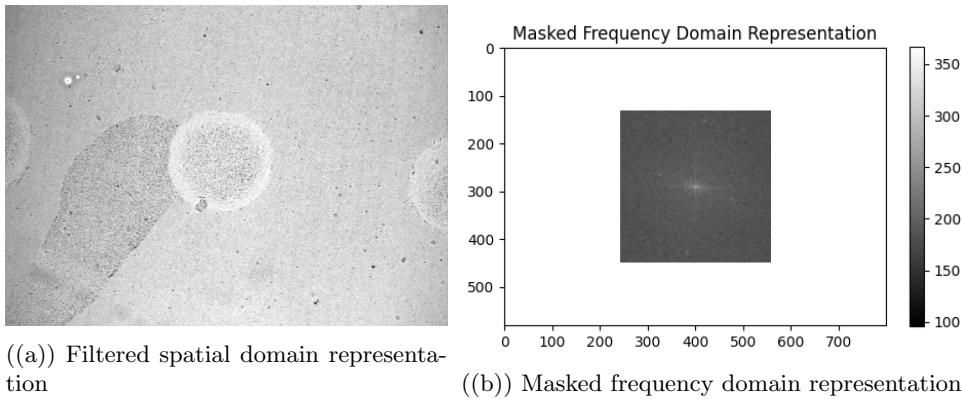


Figure 4.17: Spatial & frequency domain representation of a filtered image

Quality metrics were not taken into account to find the optimal cut-off frequency. From visual inspection, the cut-off frequency of 160 was selected to retain maximal image detail while concurrently filtering out instrumentation artifacts.

4.4.4 Background Subtraction

Implementation

In our study, we implemented a processing technique based on the research presented by Jing et al. in their paper[4]. Background subtraction is widely recognized for its effectiveness in removing background noise and, more importantly, in enhancing contrast, especially in microscopic imagery.

This technique was tailored to suit our specific requirements. Therefore, instead of using an out-of-focus image as the background, we used an averaged image derived from four blank sample images.

Our pipeline comprises three steps, as outlined in Figure 2.1:

1. Subtract the averaged image from all the raw images.
2. Compute a median-filtered version of these resulting images.
3. Subtract the median-filtered version from the original images.

A median filter operates by replacing the value of a pixel with the median value of its neighboring pixels. The kernel size, a key parameter of the filter, determines the number of neighboring pixels considered in the calculation. While this method is effective for removing salt-and-pepper noise, it can lead to the loss of fine details and is computationally intensive.

At first, only the results of the background subtraction will be analyzed. Subsequently, the entire process will be studied, using a set of 13 kernel size for the median filter, which varies from 3 to 30, across a dataset of 9 images (representing one image per concentration level).

Results & Discussion

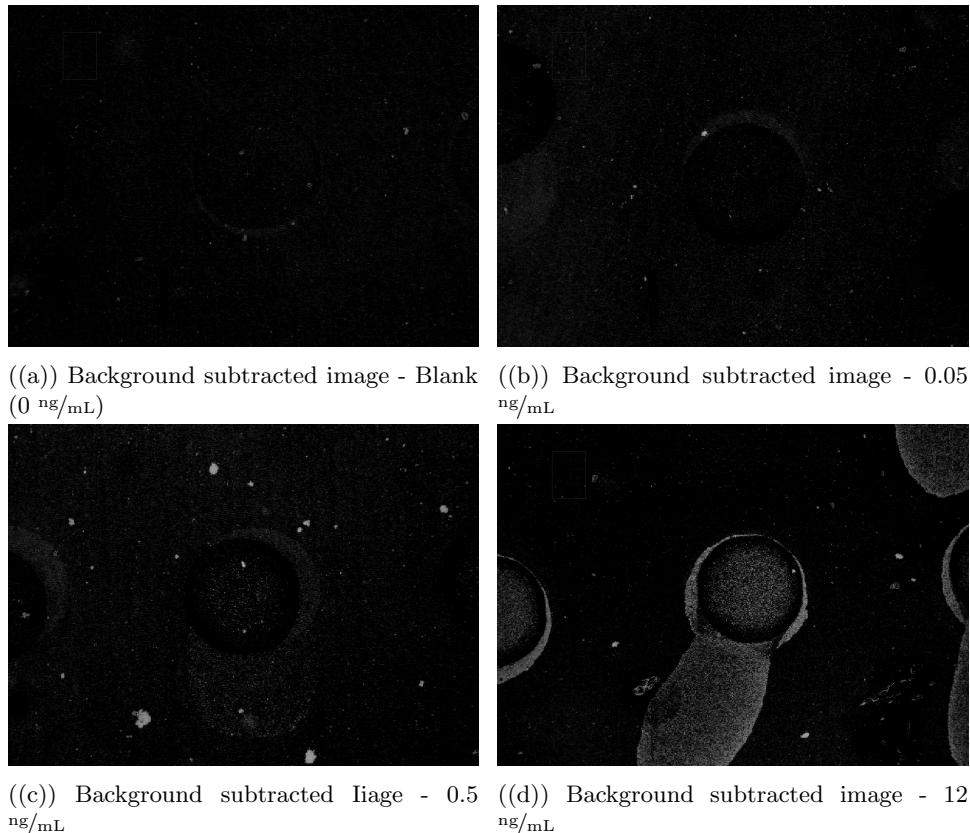


Figure 4.18: Results after background subtraction for different concentrations

From visually inspecting Figure 4.18, a transition from a bright to a dark background is observed. The remaining visible features include the bound AuNPs, aggregated AuNPs, and residual noise, which now appear clearer, likely indicating higher contrast compared to the raw images.

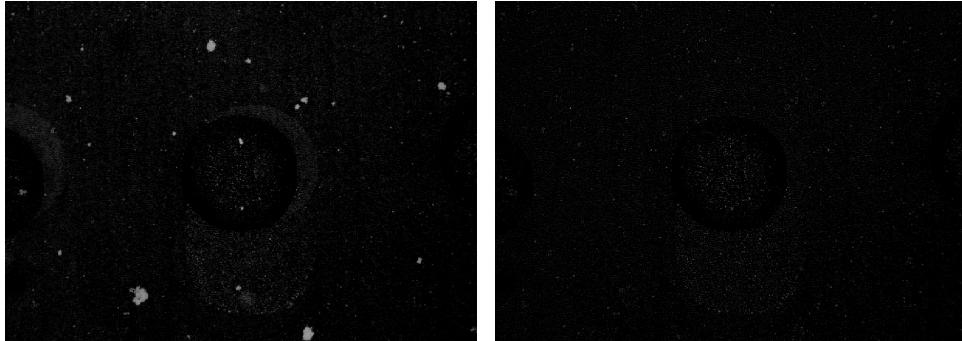
Table 4.5: Background subtracted images quality metrics

Concentration (ng/mL)	SNR	CNR	Weber Contrast	SSIM
0	0.6	0	0	0.01
0.05	0.5	0.13	0.21	0.01
0.1	0.2	0.44	0.696	0.01
0.3	0.7	0.07	0.114	0.01
0.5	0.8	0.19	0.301	0.01
1	2.7	2.06	3.268	0.01
5	3.5	2.82	4.484	0.01
12	4.3	3.64	5.79	0.01
20	6.7	6.06	9.624	-0.12
Average (Raw)	2.38	1.07	0.56	1.00
Average (Bg Subtracted)	2.22	1.71	2.72	0.01

As shown in Table 4.5, the CNR has increased on average by 60%, as anticipated. However, the SNR has slightly decreased, which was unexpected. This could be attributed to poor background estimation from the averaged blank image or possibly due to over-subtraction of the signal, where the background subtraction was too aggressive, inadvertently removing parts of the actual signal.

Examining the behavior for each concentration independently may show on the reason for these variations. For lower concentrations, the CNR has decreased, but it has increased for higher concentrations. During CNR calculation, the observed variations can be attributed to remaining noise and aggregated AuNPs, as seen in Figure 4.18(a), potentially skewing the results.

One approach to suppress the aggregated AuNPs involves subtracting a median filter with a small kernel (in our case, kernel = 3), as demonstrated in Figure 4.19. With a small kernel, the filtered images lose single AuNPs that are about 1 pixel in size, but aggregates, which are several pixels in size, remain visible in the filtered version and are discarded during the subtraction.

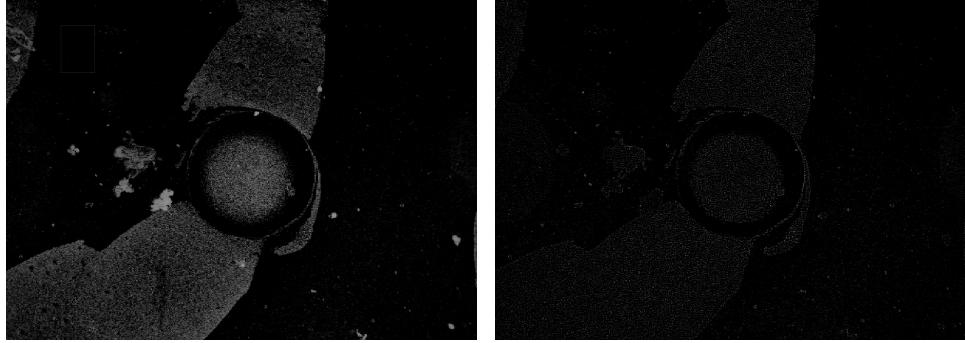


((a)) Background subtracted image with AuNPs aggregates - 0.5 ng/mL ((b)) Median filter subtracted image without AuNPs aggregates - 0.5 ng/mL

Figure 4.19: Comparison of background subtracted and median filter subtracted images highlighting suppression of AuNPs aggregates

However, this method has limitations, particularly in cases where the concentration

of bound AuNPs is very high. In such scenarios, single AuNP might be discarded, similar to aggregates, as seen in Figure 4.20, thus limiting the accuracy of results at high concentrations.



((a)) Background subtracted image - 20 ng/mL ((b)) Median filter subtracted image at high concentration - 20 ng/mL

Figure 4.20: Technique limitation at high concentration (20 ng/mL)

While it is a promising method to implement, this technique is limited by the wide range of concentrations required for our work. As previously demonstrated, the intensity of the signal is significantly diminished by the subtraction of the median filter. This not only reduces the signal component but also leads to a high number of false negatives.

4.4.5 Optimal methods

Having implemented multiple methods separately to reduce noise and analyzed them, we compared the following combinations:

- Gaussian Filtering
- Kspace Filtering
- Background Subtraction
- Gaussian Filtering then K-space filtering
- K-space filtering then Gaussian filtering
- Gaussian Filtering then Background Subtraction
- Kspace Filtering then Background Subtraction
- Gaussian Filtering then Kspace filtering then Background subtraction
- ...

From the quality metrics results as presented in Table 4.6 and through visual inspection, the most effective method for noise reduction was found to be K-space filtering followed by Gaussian filtering. Initially, K-space filtering with a cut-off frequency of 160 effectively removes noisy patterns caused by instrumentation artifacts. Subsequently, Gaussian filtering gently smooths the image ($k = 5$, $\sigma = 0.5$), reducing noise while preserving the details and signal.

While background subtraction is an effective method for obtaining accurate images at high concentrations, it performs less effectively at low concentrations. High

performance at low concentrations is typically our focus, as it closely affects our LOD.

Table 4.6: Optimal Set Quality Metrics

Concentration (ng/mL)	SNR	CNR	Weber Contrast	SSIM
0	2.9	0.0	0.00	0.23
0.05	2.5	1.4	0.25	0.23
0.1	1.3	1.9	0.59	0.22
0.3	2.0	1.0	0.33	0.21
0.5	2.0	0.7	0.25	0.22
1	5.1	1.5	0.50	0.21
5	4.8	3.0	0.75	0.19
12	5.9	2.9	0.92	0.20
20	5.9	3.8	1.63	0.62

4.5 Processing our Dataset: Contrast Enhancement

Now that we have significantly reduced noise across our dataset, the signal in our images has become clearer. From an external perspective, further processing to enhance contrast is not necessary, as we already have a satisfactory CNR and Weber contrast.

4.5.1 CLAHE

Implementation

This section focuses on our second primary objective, as mentioned earlier: enhancing the contrast between the foreground and background. CLAHE is a technique used to enhance local contrast while preventing noise amplification, a common issue with contrast enhancement techniques.

However, efficient techniques often come with drawbacks. CLAHE is computationally intensive and sensitive to its two parameters: grid size and clip limit.

We tested a set of 20 combinations with clip limits ranging from 0.5 to 8.0 and grid sizes ranging from (8,8) to (64,64), using a dataset of 9 images (one per concentration level).

Results & Discussion

Table 4.7: Comparison of Quality Metrics

Set of Parameters		SNR	CNR	Weber Contrast	SSIM
Reference Image		3.60	1.80	0.58	0.26
Clip limit	Grid size				
0.5	(8,8)	3.58	1.74	0.58	0.39
0.5	(16,16)	3.58	1.73	0.57	0.40
0.5	(32,32)	3.54	1.76	0.57	0.41
0.5	(64,64)	3.07	1.60	0.60	0.27
1.0	(8,8)	3.45	1.62	0.61	0.34
1.0	(16,16)	3.37	1.80	0.55	0.34
1.0	(32,32)	3.29	1.76	0.58	0.41
1.0	(64,64)	3.07	1.60	0.60	0.27
2.0	(8,8)	3.46	1.75	0.61	0.25
2.0	(16,16)	3.32	1.78	0.55	0.25
2.0	(32,32)	3.31	1.75	0.59	0.29
2.0	(64,64)	3.07	1.60	0.61	0.27
4.0	(8,8)	3.51	1.82	0.61	0.15
4.0	(16,16)	3.25	1.94	0.59	0.15
4.0	(32,32)	3.07	1.87	0.63	0.16
4.0	(64,64)	3.02	1.76	0.68	0.15
8.0	(8,8)	3.21	1.85	0.64	0.09
8.0	(16,16)	3.04	2.07	0.67	0.08
8.0	(32,32)	2.97	1.80	0.64	0.08
8.0	(64,64)	2.85	1.66	0.61	0.08

As observed in Table 4.7, the metrics quantifying contrast have increased with the use of CLAHE, particularly as the clip limit and grid size are increased.

However, an increase in the clip limit also amplifies the noise contrast, as illustrated in Figure 4.21(d)). This amplification leads to a reduction in the SNR, which is undesirable in our context.

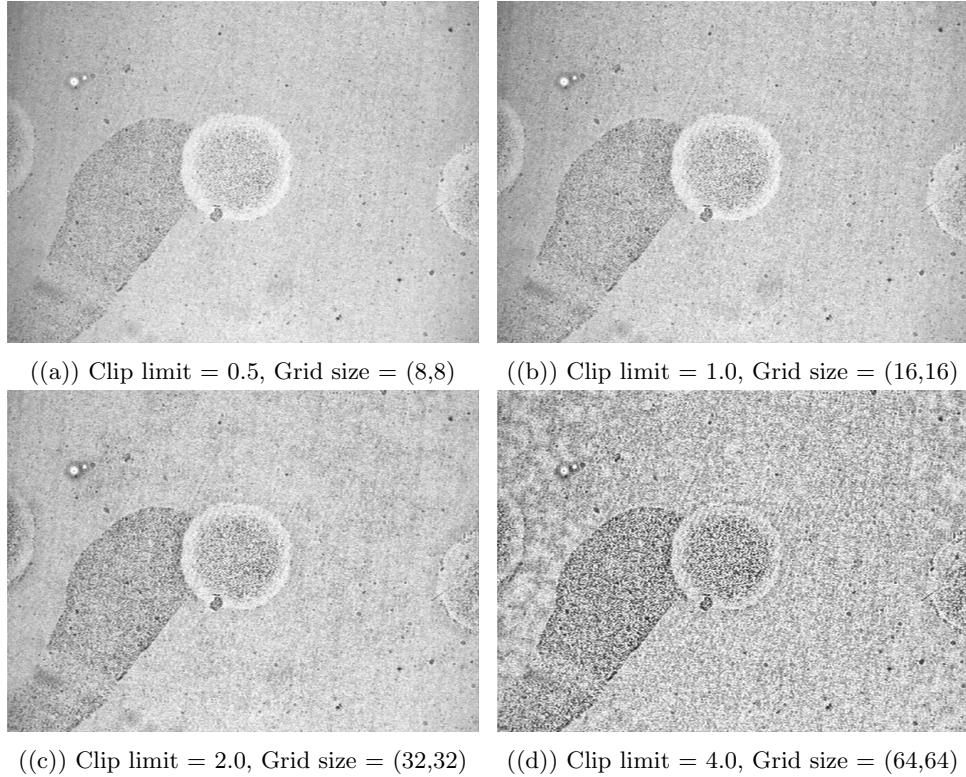


Figure 4.21: CLAHE processed images (1 ng/mL) for a set of parameters

From visual inspection and evaluation of the metrics, it is evident that random noise persists even after prior processing and is further amplified by this technique.

The Weber contrast shows a maximum increase of 17% compared to the reference image, accompanied by an 18% decrease in SNR. Based on these findings, it was decided not to include CLAHE in our final pipeline. However, its potential utility warrants consideration for future testing with a new image acquisition setup for endpoint image analysis.

4.6 Final Pipeline and Potential Improvements

This Section concludes our analysis of endpoint image analysis techniques. Multiple methods were studied, examining their behaviors, advantages, and inconveniences, to develop a final pipeline that represents the best and simplest solution for our case according to our research.

The pipeline includes:

1. **K-space filtering** with a cut-off frequency of 160.
2. **Gaussian filtering** with a kernel size of 5 and sigma of 0.5.
3. **ROI selection** and masking.
4. **Global Fixed thresholding algorithm** with a threshold value of 135.

4.6.1 Results Discussion

Table 4.8: Results of our final pipeline

Concentration (ng/mL)	Mean	Standard Deviation
0	23	14
0.05	30	4
0.1	35	9
0.3	105	22
0.5	165	67
1	316	80
5	854	232
12	1099	254
20	1748	205

As shown in Table 4.8, an improvement using our pipeline can be observed. Now, our LOD has decreased to below 0.05^{ng}/mL with a slight increase in the dynamic range. Processing the images, especially the blank sample, allowed us to reduce the blank count and, therefore, increase accuracy.

However, it is not currently possible to accurately determine new images and would require more images with concentrations between 0 and 0.05^{ng}/mL.

Please refer to the resulting binary images (Figure 4.23) and the concentration vs. count plot (Figure 4.22) for comparison with previous data if desired.

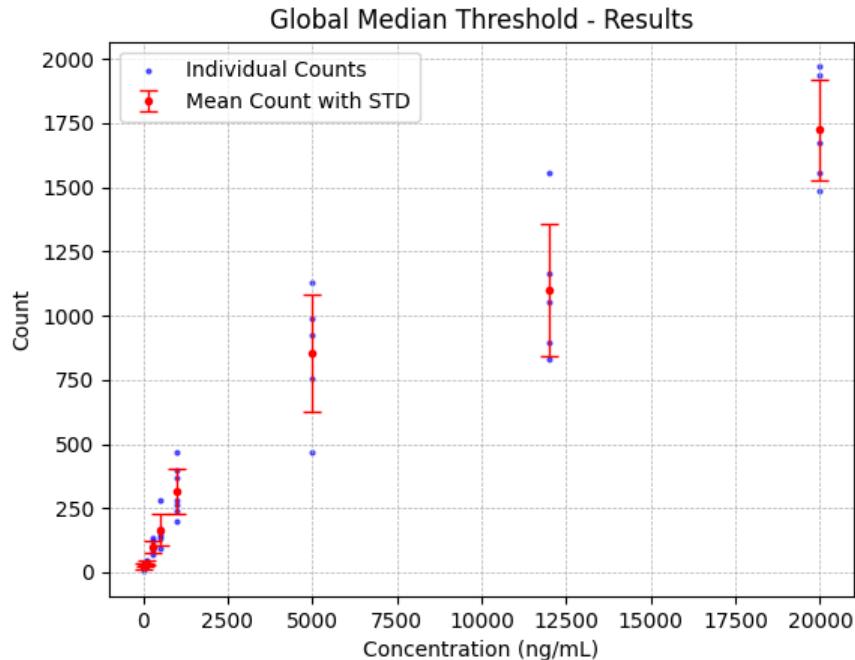


Figure 4.22: White pixel counts results for optimal pipeline

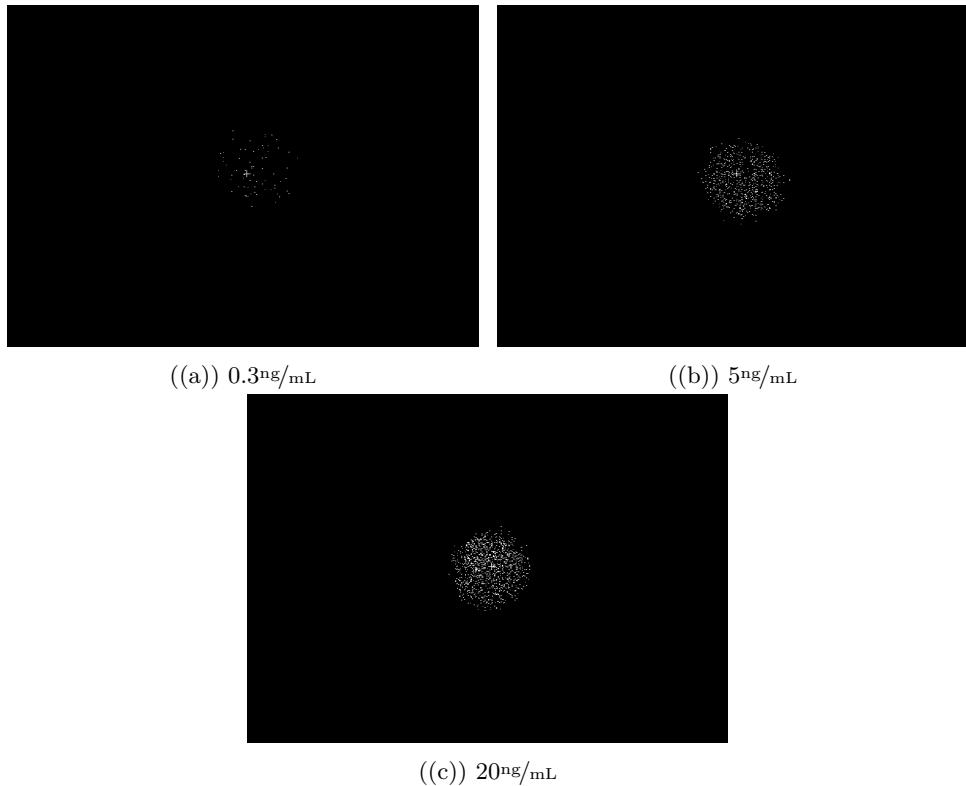


Figure 4.23: Results after binarization over multiple concentrations

4.6.2 Potential Improvements

During this section, we explored interesting techniques that could be powerful if used correctly. Some metrics, such as reproducibility, have not been taken into account, which could potentially make some methods (e.g. the global median threshold) more appealing.

With a more extensive dataset, we could have also made a better approximation of the background, making the background subtraction method more efficient as well.

Please be aware that this pipeline, chosen based on the research conducted with available resources, can undoubtedly be further improved.

Our next path of improvement involves exploring time-resolved methods in the following section.

Chapter 5

Time-resolved Images Database

As concluded from the endpoint images analysis, we are limited in our performance and cannot attain our requirements set in Section 1.2.

As clarified in the literature review, promising techniques can be employed in a time-resolved setup. Therefore, creating a new dataset of time-resolved images is my next step during this thesis.

5.1 Creation of a New Database

Our current advancement in the bioassay part of our device was a limiting factor for me. Being unable to retrieve real-time images on functionalized chips with our imaging setup, the decision was made to change the setup and work on unfunctionalized chips.

5.1.1 Imaging Setup

Equipment:

- **Microscope: Nikon Eclipse Ti[10]**

This advanced inverted research microscope system offers high-end features and adaptability for diverse imaging methods, providing stability and flexibility for complex imaging applications.

- **Camera: Iris 15 Scientific CMOS[11]**

A high-sensitivity, high-speed camera that provides a large field of view and high resolution, ideal for capturing dynamic processes and small details.

Acquisition Parameters:

- **Resolution:** 2528 x 1480 pixels

The high-resolution setting allows for detailed imaging, enabling the capture of small-intensity distortions due to AuNPs binding.

- **Exposure Time:** 20 ms

A short exposure time is utilized to minimize motion blur and to capture rapidly occurring events, ensuring sharp images.

- **Frame Rate:** 5 frames per second (fps)

The selected frame rate strikes a balance between temporal resolution and data volume, suitable for observing moderate-speed processes over time.

- **Duration:** 2 hours (trimmed to 36 minutes and 30 seconds)

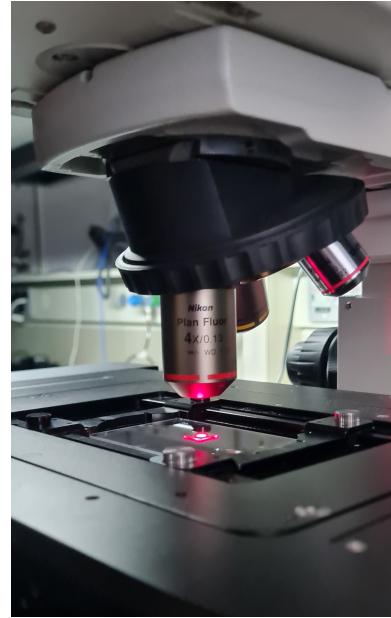
The extended duration of the acquisition was intended to monitor until evaporation of all the solution, with the final dataset focusing on 36.5 minutes of activity (before complete evaporation).

- **Magnification:** 10x Iris Objective and 4x Nikon CFI Plan Fluor Objective

The combination of a 10x magnifying Iris objective with a 4x Nikon CFI Plan Fluor objective offers a compounded magnification, ideal for observing AuNPs with high precision. Total magnification: x40



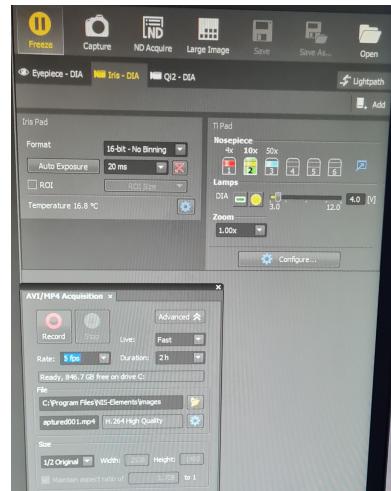
((a)) Nikon Eclipse Ti Microscope



((b)) Objectives Used in Imaging



((c)) Iris 15 Scientific CMOS Camera



((d)) Imaging Acquisition Parameters

Figure 5.1: Components and settings of the imaging setup

The imaging setup described is optimized for high-quality, plasmonic imaging and appears to be designed to minimize shot noise as much as possible.

5.1.2 From Video to Database

A Python script was coded and used to convert a video into an image database that can be further used in the image processing algorithm. The code functions as follows:

- Video editing to keep only the interesting part.
- Retrieval of frames (5 frames per second in our case, resulting in 1095 images) and storage as raw images.
- Reduction of shot noise by computing a temporal average over 50 raw images, done in batches of 200 images (results in 219 pre-processed images).

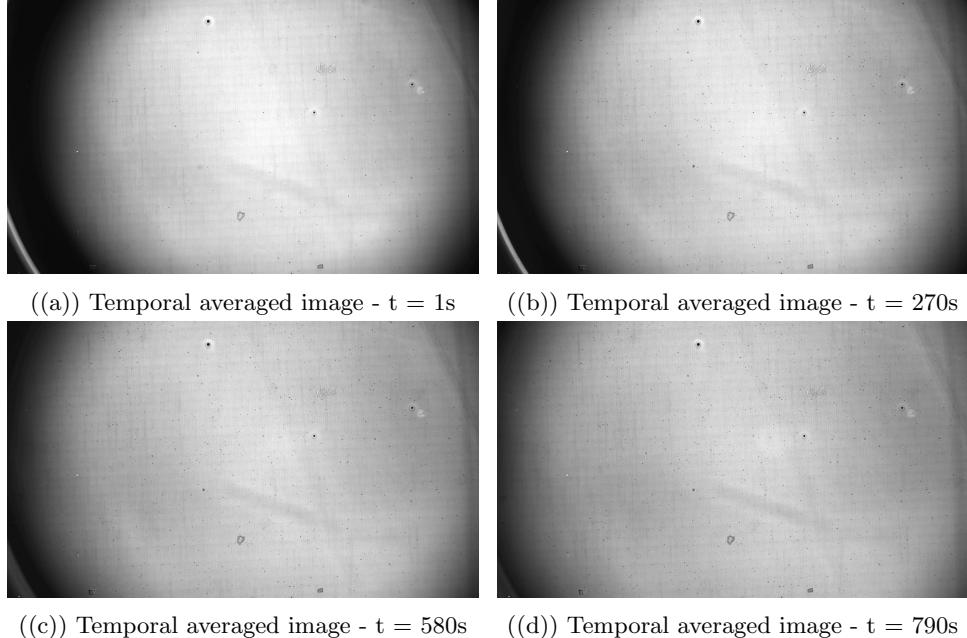


Figure 5.2: Images from Time-resolved Database

5.2 Endpoint Analysis

Before implementing Time-resolved techniques, one aspect needs clarification: the imaging setup used does not represent the quality of images that will be acquired on your prototype. The creation of this database aims to enable the implementation and to study the behavior of time-resolved techniques.

An endpoint analysis has been conducted for reference, but processing techniques did not need to be implemented due to the low background noise in the images. Using cutting-edge technologies, the images are not representative of the problem but allow the testing of these methods.

The pipeline implemented was the optimal one without processing and includes:

1. ROI Selection and Masking (ROI radius has been increased to 500)
2. Global fixed thresholding algorithm with a threshold value of 135

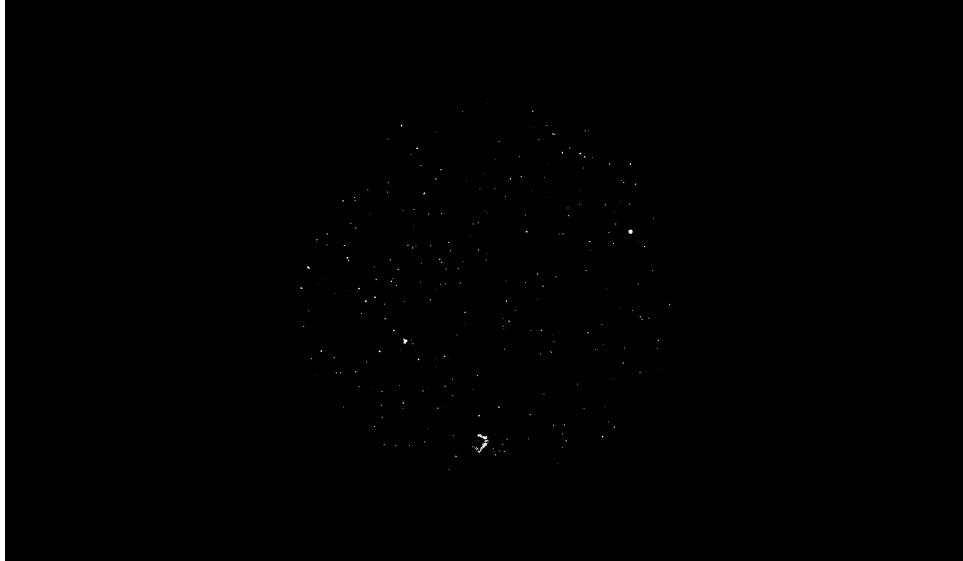


Figure 5.3: Segmented Image at time $t = 730\text{s}$

In the resulting image (Figure 5.3), we can clearly identify and segment the AuNPs. This is due to the lack of background noise and a high magnification. Indeed, high magnification allows us to reduce the number of AuNPs per pixel, thus avoiding saturation of dark spots in the image and the problems associated with it.

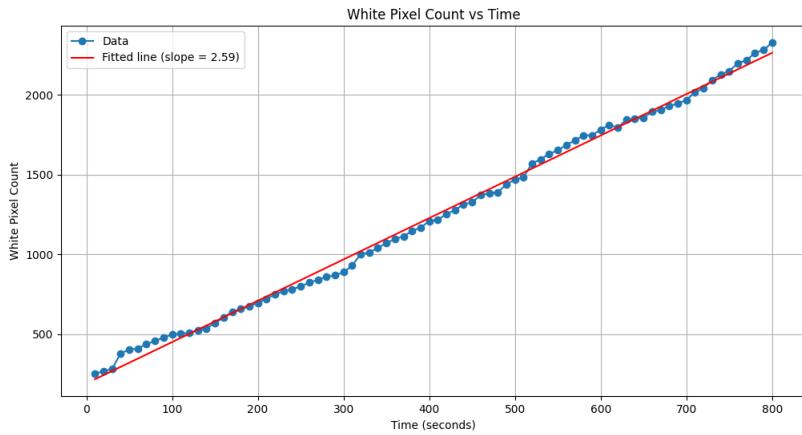


Figure 5.4: plot of evolution of Binding of AuNPs across time

Figure 5.4 clearly shows the evolution of AuNPs binding over time. We can approximate a linear fit and thus a binding rate (slope).

The endpoint techniques work correctly with these images, but how do they compare to time-resolved techniques?

5.3 Time-resolved Analysis

5.3.1 Temporal Average

Implementation

Temporal averaging can efficiently reduce background noise as it tends to cancel out random fluctuations and enhance signal clarity.

The effectiveness of temporal averaging depends on one parameter: window size. This refers to the number of images averaged together. The results are highly sensitive to the choice of window size. For a window size that is too high, this technique can lead to a loss of temporal resolution, but in our case of a slow dynamic process, we should be safe from negative effects.

Another issue is that temporal averaging is ineffective against non-random noise. In our case, we usually have instrumentation effects leading to patterns in the background, which cannot be reduced using temporal averaging.

For implementation, temporal averaging was performed during the conversion from video to the database, as explained in Section 5.1.2.

Results & Discussion

As expected, no significant improvement was shown (Table 5.1) in the SNR for these images. The great capacity of temporal averaging to remove random noise is not necessary, considering the use of cutting-edge image acquisition equipment.

Table 5.1: Results of Temporal Averaging process

Frame	SNR	SSIM
frame 07051 - time: 23:30.00	6.7	1
frame 07052 - time: 23:30.20	6.7	1
frame 07053 - time: 23:30.40	6.7	1
frame 07054 - time: 23:30.60	6.7	1
frame 07055 - time: 23:30.80	6.7	1
frame 07056 - time: 23:31.00	6.7	1
frame 07057 - time: 23:31.20	6.7	1
frame 07058 - time: 23:31.40	6.7	1
...
frame 07093 - time: 23:38.40	6.8	1
frame 07094 - time: 23:38.60	6.8	1
frame 07095 - time: 23:38.80	6.8	1
frame 07096 - time: 23:39.00	6.8	1
frame 07097 - time: 23:39.20	6.8	1
frame 07098 - time: 23:39.40	6.8	1
frame 07099 - time: 23:39.60	6.8	1
frame 07100 - time: 23:39.80	6.8	1
averaged frame	6.8	0.95

5.3.2 Differential Imaging

Implementation

Differential imaging offers strong background noise removal and high contrast for AuNPs identification. Its approach to AuNPs quantification is different. Instead of identifying all AuNPs bound to the chip's surface, it counts the AuNPs that bind during a certain lapse of time.

Therefore, the resulting counts allow us to find an averaged slope. This slope correlates with the binding rate, which is itself related to the concentration of biomarkers in the solution.

This solution can allow us to perform real-time measurements, thus estimating the slopes accurately in a very short amount of time.

To implement differential imaging, one parameter to set is the time interval. Indeed, depending on the time between the two differentiated images, we will have different results. For a time interval that is too low, we won't see sharp, highly contrasted AuNPs on the differential image. For a time interval that is too high, we will lose accuracy and time-to-results.

Results & Discussion

Multiple time intervals were tested, and a trade-off was found at 30 seconds between images.

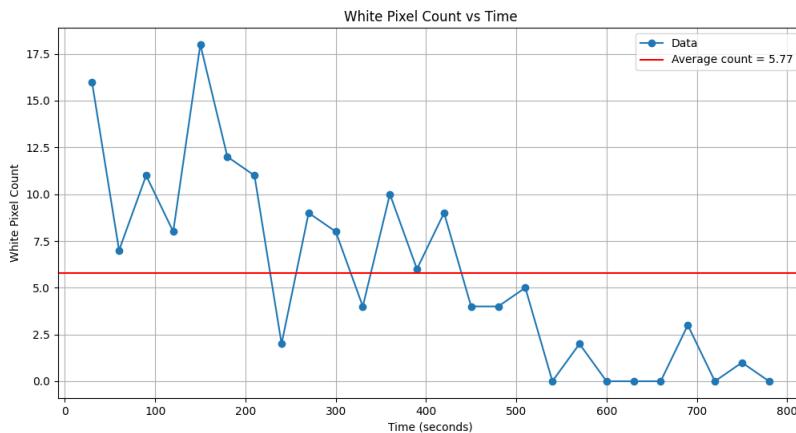


Figure 5.5: Results of Differential counting over time

As seen in Figure 5.5, the results are not as expected. A time interval of 30 seconds was used so the binding rate obtained from this technique and with endpoint analysis are the same (2.59 for 10s and 5.77 for 30s).

However, the legitimacy of the results can be doubted. The plot shows a decrease in binding rate to 0 when we arrive at a later stage, and in the endpoint analysis, the slope is approximately constant over time. The problem might arise from our database having illumination variation over time.

To conclude, differential analysis and temporal averaging have great potential. Unlike other endpoint analysis techniques, they require a deeper understanding to be effectively used in our case.

Chapter 6

Conclusion

This thesis embarked on a journey to develop an efficient image processing system for NEOSENS, a startup dedicated to revolutionizing the diagnosis of NS. The research was driven by a profound motivation to address the diagnostic challenges in LMICs, where NS is a major health concern.

Our efforts were focused on the design and implementation of a robust computational method that can differentiate and count surface-bound AuNPs with high precision. The goal was to achieve a LOD of less than 0.01ng/mL within a five-minute timeframe. This objective was particularly challenging due to the environmental and operational constraints prevalent in LMICs.

Throughout this research, several advanced image processing techniques were explored and tested. We successfully reduced background noise and enhanced the contrast between the AuNPs and the background, which was essential in accurately identifying and quantifying the AuNPs.

The results obtained have shown promise in reaching the LOD goal, marking a significant step towards a practical diagnostic solution for NS.

The research, however, does present some limitations. The datasets used are not exactly the same as those that will be employed in practice. This might impact the direct applicability and effectiveness of the developed algorithms in real clinical settings. Additionally, not all state-of-the-art techniques have been implemented. Research based on Machine Learning[12][13], as referenced in [12][13], appears promising and should be studied in the future.

A high-performing pipeline was developed, upon which NEOSENS can iterate to eventually have a software solution ready for use during the pilot project in Cairo, scheduled for April 2024.

Let's work towards ensuring that all newborns receive **the right care at the right time!**

Bibliography

- [1] A. Belushkin, F. Yesilkoy, and H. Altug, “Nanoparticle-enhanced plasmonic biosensor for digital biomarker detection in a microarray,” *ACS Nano*, vol. 12, no. 5, pp. 4453–4461, 2018, pMID: 29715005.
- [2] W. Jing, Y. Wang, Y. Yang, Y. Wang, G. Ma, S. Wang, and N. Tao, “Time-resolved digital immunoassay for rapid and sensitive quantitation of procalcitonin with plasmonic imaging,” *ACS Nano*, vol. 13, no. 8, pp. 8609–8617, 2019, pMID: 31276361.
- [3] I. Sidorenko, S. Nizamov, R. Hergenröder, A. Zybin, A. Kuzmichev, B. Kiwull, R. Niessner, and V. Mirsky, “Computer assisted detection and quantification of single adsorbing nanoparticles by differential surface plasmon microscopy,” *Microchimica Acta*, 08 2015.
- [4] W. Jing, Y. Wang, C. Chen, F. Zhang, Y. Yang, G. Ma, E. H. Yang, C. L. N. Snozek, N. Tao, and S. Wang, “Gradient-based rapid digital immunoassay for high-sensitivity cardiac troponin t (hs-ctnt) detection in 1 l plasma,” *ACS Sensors*, vol. 6, no. 2, pp. 399–407, 2021, pMID: 32985183.
- [5] S. Ghosh, N. Li, Y. Xiong, Y.-G. Ju, M. P. Rathslag, E. G. Onal, E. Falkiewicz, M. Kohli, and B. T. Cunningham, “A compact photonic resonator absorption microscope for point of care digital resolution nucleic acid molecular diagnostics,” *Biomed. Opt. Express*, vol. 12, no. 8, pp. 4637–4650, Aug 2021.
- [6] N. li, “Photonic resonator absorption microscopy for biosensing with digital resolution,” 2019.
- [7] W. Zhang, T. Dang, Y. Li, J. Liang, H. Xu, G. L. Liu, and W. Hu, “Digital plasmonic immunosorbent assay for dynamic imaging detection of protein binding,” *Sensors and Actuators B: Chemical*, vol. 348, p. 130711, 2021.
- [8] Y. Wang, Y. Yang, C. Chen, S. Wang, H. Wang, W. Jing, and N. Tao, “One-step digital immunoassay for rapid and sensitive detection of cardiac troponin i,” *ACS Sensors*, vol. 5, no. 4, pp. 1126–1131, 2020, pMID: 32180397.
- [9] Y. Wang, W. Jing, N. Tao, and H. Wang, “Probing single-molecule binding event by the dynamic counting and mapping of individual nanoparticles,” *ACS Sensors*, vol. 6, no. 2, pp. 523–529, 2021, pMID: 33284583.
- [10] N. Healthcare, “Eclipse ti series inverted microscopes,” [Accessed: 22/11/2023].
- [11] Photometrics, “Iris 15 scientific cmos camera,” [Accessed: 22/11/2023].

- [12] Z. Gao, Y. Song, T. Y. Hsiao, J. He, C. Wang, J. Shen, A. MacLachlan, S. Dai, B. H. Singer, K. Kurabayashi, and P. Chen, “Machine-learning-assisted microfluidic nanoplasmonic digital immunoassay for cytokine storm profiling in covid-19 patients,” *ACS Nano*, vol. 15, no. 11, pp. 18 023–18 036, 2021, pMID: 34714639.
- [13] X. Z. H. Z. Q. Z. C. Y. Y. L. T. X. Zhongwen Long, Xinyu Shen, “Plasmon-enhanced ultrasensitive digital imaging immunoassay for the quantification of micornas assisted by convolutional neural network analysis (adv. funct. mater. 6/2023),” *Advanced Functional Materials*, 2023.

Appendix A

Appendix

A.1 Local Adaptive Thresholding Algorithm

A.1.1 Constant Value Tuning

Table A.1: Examples of average white pixel counts for different adaptive constant values (window size = 3)

Concentration (ng/mL)	Constant Value				
	2	10	15	25	40
0	9727	3531	1482	204	44
0.05	9015	3502	1635	315	90
0.1	6796	2557	1246	283	103
0.3	8146	4325	2804	1266	565
0.5	8613	4234	2611	1137	494
1	9874	6401	4727	2578	1099
5	9649	6914	5557	3706	2004
12	9469	7037	5883	4175	2424
20	10385	7550	6017	3590	1440

Table A.2: Examples of standard deviation of white pixel counts for different adaptive constant values (window size = 3)

Concentration (ng/mL)	Constant Value				
	2	10	15	25	40
0	49	207	140	44	19
0.05	674	572	301	57	11
0.1	997	609	335	81	18
0.3	778	498	301	168	83
0.5	909	625	447	183	74
1	144	194	213	220	137
5	311	244	199	156	97
12	414	361	304	209	100
20	456	174	368	731	770

A.1.2 Window Size Tuning

Table A.3: Examples of average white pixel counts for different adaptive window sizes (Constant = 15)

Concentration (ng/mL)	Window Size				
	3	5	11	15	19
0	1482	2093	2148	2166	2209
0.05	1635	2147	2241	2282	2306
0.1	1246	1639	1720	1728	1744
0.3	2804	3216	3238	3247	3288
0.5	2611	3014	3074	3080	3075
1	4727	5134	5150	5172	5159
5	5557	5873	5911	5904	5893
12	5883	6150	6221	6241	6284
20	6017	6454	6656	6728	6766

Table A.4: Examples of standard deviation of white pixel counts for different adaptive window sizes (Constant = 15)

Concentration (ng/mL)	Window Size				
	3	5	11	15	19
0	140	308	229	223	244
0.05	301	383	353	361	358
0.1	335	432	413	408	408
0.3	301	403	339	347	355
0.5	447	444	451	452	466
1	213	176	154	141	143
5	199	208	204	215	215
12	304	300	309	321	322
20	368	285	266	209	193