Analysis of Spherical Aberration Correction in Microscopic Images using Physical Layers

Anusha Krishnamoorthy, Rutwik Palaskar
Department of Biomedical Engineering
Duke University
Durham, NC 27708
anusha.krishnamoorthy@duke.edu
rutwik.palaskar@duke.edu

Abstract

Spherical aberrations are a common type of optical aberration that occur when light rays passing through different regions of a lens converge at different points, resulting in blurred and distorted images. This can pose significant challenges for accurate segmentation of microscopy data using deep neural networks. Through this project, we aim to optimize the model performance by the incorporation of a trainable physical layer . Specifically, we implement three different versions of this physical layer and compare their effectiveness for this use-case. Ultimately, this approach may enable researchers to salvage bad data without incurring the added cost of upgrading the acquisition system. Our results demonstrate that the proposed physical layer show potential for enhanced model performance for segmentation of microscopy data with spherical aberrations, and our comparative analysis of the three different versions of the physical layer provides insight into the most effective implementation for this specific use-case.

1 Introduction

Microscopy is a widely used imaging technique in various scientific fields, including biology, medicine, and material science. However, one major challenge in microscopy is the presence of optical aberrations, which can degrade the quality of the acquired images [1]. Spherical aberration refers to the phenomenon where light rays passing through different regions of a spherical lens converge at different points along the optical axis, resulting in blurred and distorted images. This occurs due to variations in the angles at which the light rays enter the lens, which can cause them to be refracted differently and converge at different points. As a result, the image formed by the lens can appear fuzzy or out of focus [2]. This can pose significant challenges for accurate segmentation of microscopy data using deep neural networks.

To address this challenge, we aim to optimize the performance of a deep neural network model for segmentation of microscopy data with spherical aberrations. Specifically, we propose the integration of a physical layer into the neural network architecture, which can compensate for these aberrations and improve the accuracy of the segmentation. This project implements and evaluates the effectiveness of three different versions of this physical layer for this use-case.

Through this research, we hope to salvage bad data without the added cost of upgrading the acquisition system. By integrating physical information into the neural network model, we aim to mitigate the negative effects of spherical aberrations on the quality of the microscopy data, and thus improve the segmentation accuracy of the neural network. Our proposed approach has the potential to enhance the capabilities of deep learning-based microscopy image analysis, enabling more accurate and reliable segmentation of images affected by spherical aberrations.

In this paper, we present the details of our proposed approach, including the implementation and evaluation of the three different versions of the physical layer. We also provide a comparative analysis of their performance for the segmentation of microscopy data with spherical aberrations. Finally, we discuss the potential implications and future directions of our work, highlighting its potential impact in the field of microscopy image analysis.

2 Related Work

In microscopy and other imaging applications, spherical aberration is a frequent problem, and a number of machine learning techniques have been suggested to correct for this issue. In order to increase image quality and segmentation accuracy, [3]developed a technique for correcting spherical aberration in microscope images using phase diversity and deep learning. To achieve high accuracy and resilience in aberration correction,[W] presented a deep convolutional neural network (CNN) outfitted with a self-attention mechanism for direct aberration determination.

In addition to these studies, [5] developed an adaptive optics learning system for aberration correction in multiphoton confocal microscopy. The Smart Microscope, as it is called, uses machine learning to learn and correct for aberrations in real-time, improving image quality and contrast.

Other researchers have explored machine learning approaches for correcting spherical aberration in electron microscopy (EM) and optical coherence tomography (OCT). For instance, [6] proposed a generative adversarial network-based approach for correcting for spherical aberration in EM images.

These studies demonstrate the potential of machine learning techniques for improving image quality and accuracy in the presence of spherical aberration in a variety of imaging modalities. By incorporating physical models and deep learning approaches, these methods offer a promising avenue for addressing the challenges posed by spherical aberration in microscopy and other imaging applications.

3 Methods

3.1 Dataset

This project makes use of the Kaggle 2018 Data Science Bowl. This dataset is a collection of segmented nuclei images that were obtained for a variety of cell types, magnifications, and imaging modalities. The dataset includes images in both grayscale and RGB formats, with resolutions ranging from 256x256 pixels to 1024x1024 pixels. The images are labeled with binary masks that indicate the location of nuclei within the image. This dataset was designed such that the variety in microscopy images, enabled researchers to study the generalizability of the algorithms across different modalities (fluorescence microscopy, bright-field microscopy, and phase contrast microscopy) and magnifications. Overall, this dataset provides a diverse set of images that can be used to evaluate the performance of algorithms for segmentation and detection of nuclei across different microscopy imaging conditions.

Each image in the dataset is associated with a folder containing two subfolders: images and masks, wherein the images subfolder contains the actual image file corresponding to the ImageId. The masks subfolder is only included in the training set and contains the segmented masks of each nucleus in the image. Each mask corresponds to a single nucleus and is labeled with a unique identifier and are of the same resolution as their corresponding image. There are a total of 670 images in the dataset.

3.1.1 Exploratory Data Analysis

To further quantify the types of microscopy images available in the dataset, we perform exploratory data analysis with the help of the K-means clustering algorithm. K-means clustering is a popular unsupervised machine learning technique used for grouping data into clusters based on their similarity. The algorithm aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (centroid). The clustering works by iteratively updating the cluster centers (centroids) until convergence.

Here, the clustering scheme is employed based on the dominant HSV color distribution to categorize each of the available images based on their color patterns. 3 categories emerged from this: fluorescent, histological, and bright-field images. The distribution breakdowns for each category were as follows:

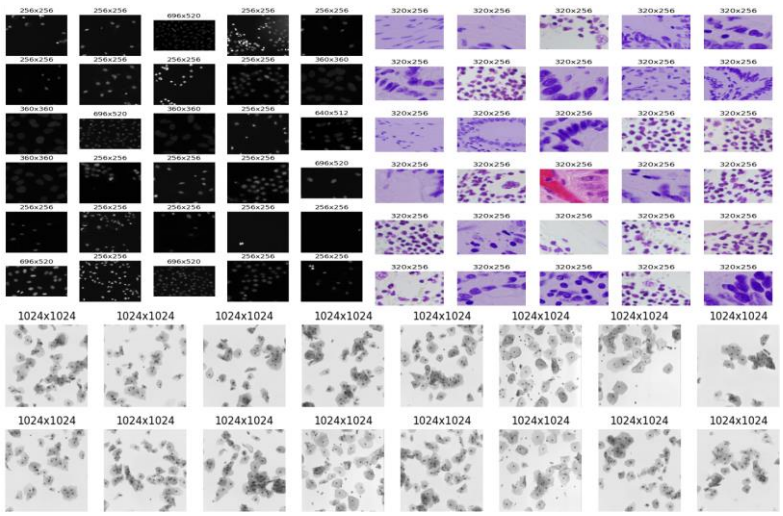

Figure 1: Result of Exploratory Data Analysis: Top left – Fluorescent Microscopy Images; Top Right – Histological Images; Bottom – Bright Field Images

Fluorescent: 81.5This analysis provides important insights into the composition of the dataset, allowing researchers to better understand the types of images they are working with and how they might be used for different applications.

3.1.2 Pre-processing

In the pre-processing step, we applied the following transformations to prepare the data for training our algorithm: Firstly, we cropped the images and masks to ensure they all have a shape of 256x256 pixels. This was done to standardize the size of the data and ensure that the model could process the images efficiently. The cropping was done in a way that maintained the centrality of the nuclei in each image.

After cropping, we applied min-max normalization to all the images. Min-max normalization scales the pixel values of each image to a range between 0 and 1, which helps to ensure that the model is able to process the images efficiently. Specifically, this transformation maps the pixel values to the range (0,1) using the following formula:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

Furthermore, as we are using binary cross-entropy loss, we need to normalize the masks as well. To achieve this, we divided all mask labels by 255. This operation scales the pixel values of each mask label to a range between 0 and 1, making them suitable for use with the binary cross-entropy loss function.

By performing these pre-processing steps, we were able to standardize the size and scale of the images and labels, which helped to ensure that the model could effectively learn the relationships between the input images and their associated labels.

3.2 Spherical Aberration Simulation

To assess the feasibility of our proposed physical layers, we simulate spherical aberrations on the 3D RGB microscopy images in the dataset, using a custom-written function. Spherical aberrations cause a deviation from the ideal spherical shape of the lens to produce a blurred image.

To simulate this effect, our function first converts the RGB image to the frequency domain using fast Fourier transform. We then created an aberration function as a Gaussian kernel in Fourier space, where the sigma parameter controls the amount of aberration. The aberration function was then

multiplied by the FFT of the image for each color channel, resulting in an aberrated image in the frequency domain.

Next, we converted the aberrated image back to the spatial domain using inverse Fourier transform. To smooth out the added aberration, the function applied a Gaussian filter for each color channel. This helped to reduce noise and improve the clarity of the aberrated image. A sigma value of 3 was empirically chosen to be the appropriate amount for blurring. Finally, to ensure that the intensity

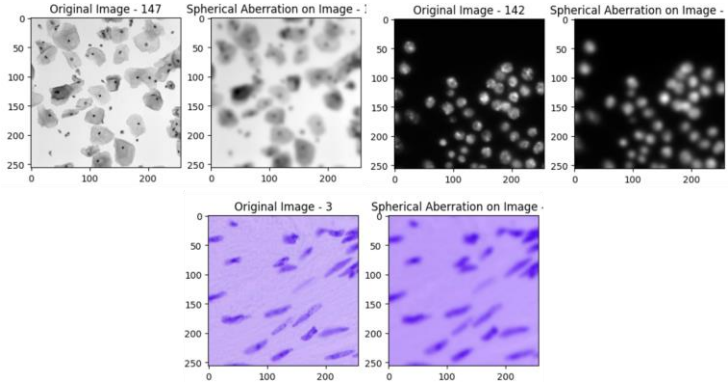

Figure 2: Simulating Aberrations on dataset; = 3

values of the resulting image were within a reasonable range, we normalized the intensity values to be in the range [0,1]. This normalization step helped to ensure that the resulting image was suitable for further analysis and processing.

3.3 Deep Learning Model

For our image segmentation task, we have selected the U-Net architecture proposed by Ronneberger et al. in 2015 [7]. This deep learning architecture is specifically designed for image segmentation tasks and has been shown to provide accurate results even with small datasets.

The U-Net architecture consists of an encoder-decoder structure with skip connections between corresponding encoder and decoder layers. The encoder consists of convolutional and pooling layers, which downsample the input image and capture its context. The decoder uses transposed convolutions and concatenates feature maps from the encoder through skip connections, which enables precise localization of objects in the image. The skip connections help to combine low-level and high-level features, which is critical for accurate segmentation.

The U-Net architecture also includes a large number of feature maps and a small receptive field in the initial layers, which helps to capture fine-grained details in the image. The final layer of the network uses a pixel-wise activation function to generate a binary mask that represents the segmented object(s) in the image.

We have also included batch normalization and dropout layers in the architecture. Batch normalization helps to improve the stability and speed of training by normalizing the input to each layer, while dropout can help prevent overfitting by randomly dropping out some of the neurons during training.

3.4 Physical Layer Design

To improve the segmentation performance in our aberrated dataset, we explored the use of a trainable physical layer to correct for this blurring. We propose and compare the performance of three physical layers to test their effectiveness. In addition, we also compared the performance of the aberration-corrected images with those obtained without using a physical layer.

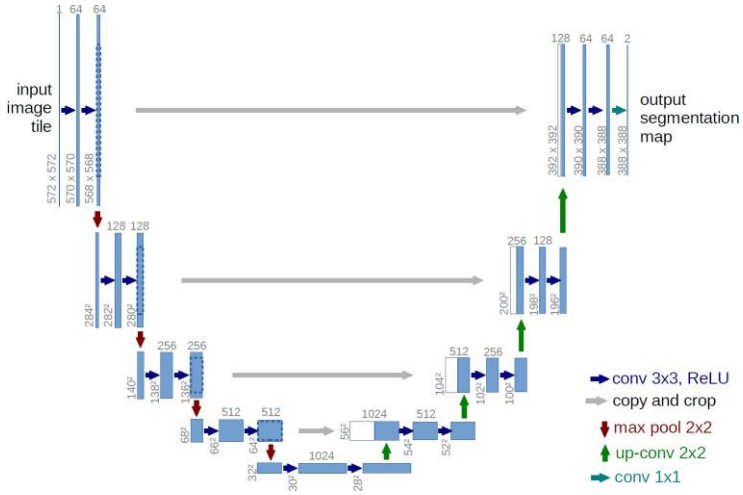

Figure 3: : U-Net architecture used for training on aberrated data

3.4.1 Design - 1

The Physical layer design-1 is responsible for performing aberration correction on input images through the use of Fourier transforms. This layer takes an input tensor of shape '(batch size, height, width, channels)' and applies the following operations to correct for spherical aberration:

- The input tensor is transposed to have the channels dimension as the second dimension (i.e. Batch Size x Channels x Height x Width).
- 2D FFT with shift is applied to every channel of the input tensor to transform it to the frequency domain.
- The resulting frequency domain tensor is element-wise multiplied with a rectangular mask of shape '(height, width, channels)', which is learned by the layer as a trainable parameter.
- 2D IFFT is applied to every channel of the resulting tensor to bring it back to the spatial domain.
- The absolute value of the resulting tensor is taken to ensure real values.
- The tensor is transposed back to its original axis ordering of (i.e. Batch Size x Height x Width x Channels).

. The output of this layer is an image tensor that has been corrected for spherical aberration using Fourier transforms.

3.4.2 Design - 2

The second design proposed for the physical layer performs aberration corrections by computing a correction factor based on a polynomial function.

- During initialization, the layer creates a set of trainable coefficients c_1, c_2, c_3, c_4, c_5 with shape (5,) that are used to compute the correction.
- During the call method, the correction factor is computed based on the input and the trainable coefficients.
- The meshgrid function is used to create a grid of x and y values ranging from -1 to 1.
- The correction is then computed as a polynomial function of r^2, which is the sum of squares of x and y.
- This correction factor is applied to each channel of the input image separately using a for loop to create a list of corrected channels, via the following polynomial:

$$correction = 1 + c_1 r^2 + c_2 r^4 + c_3 r^6 + c_4 r^8 + c_5 r^{10}$$

The tf.stack function is then used to stack the corrected channels back into a single output tensor.

The output tensor has the same shape as the input tensor, except that the spherical aberration correction has been applied to each pixel.

3.4.3 Design - 3

This final proposed design of the physical layer applies a spherical aberration correction to an input image using the Z4 Zernike polynomials. Zernike polynomials are a set of orthogonal polynomials defined over a unit disk that are commonly used in optics to model aberrations in optical systems. Each polynomial corresponds to a specific aberration mode, and the Z4 polynomial is particularly useful for modeling spherical aberrations. The layer does this following:

- The layer first creates a grid of pixel coordinates centered on the image.
- The Z4 Zernike polynomial is then computed using these coordinates, according to the following equation:

$$Z4 = \sqrt{5}(6r^2 - 6r + 1)$$

  where r is the radial coordinate of a pixel relative to the center of the image.
- The Z4 polynomial is used to compute the amount of aberration correction to apply to the input image.
- The aberration correction is added to the input image in the Fourier domain. The layer first performs a 2D fast Fourier transform (FFT) on the input image, applies the aberration correction in the Fourier domain, and then performs an inverse FFT to obtain the corrected image.
- The output of the layer is rescaled to match the input range, so that the corrected image has the same minimum and maximum pixel values as the original input image.

3.5 Training Scheme

The train-test split was set to have a 80-20 ratio, with 20% of the training set being used for validation. The model was trained using the Adam optimizer with a fixed learning rate of 0.001. The binary cross-entropy loss function was used as it is commonly used for image segmentation tasks. In addition, the Dice score was used as a metric to evaluate the performance of the model during training.

The model was trained for 50 epochs with early stopping implemented to prevent overfitting. The early stopping criteria were set to a patience of 5 epochs and monitored the validation loss. This ensured that the model did not continue training if there was no improvement in performance on the validation set for five consecutive epochs.

Overall, this training scheme was chosen to strike a balance between model performance and computational efficiency. The fixed learning rate allowed for stable training, and the early stopping criteria prevented the model from overfitting to the training data

4 Results

4.1 No Physical Layer Results

Without a physical layer, the model achieved relatively high accuracy and Dice scores on both the training and validation sets. The training accuracy and loss was found to be 0.9634 and 0.0830 respectively while the validation accuracy and loss was found to be 0.9519 and 0.1137 respectively with a validation dice score of 0.8030.

The testing accuracy and loss was found to be 0.9588 and 0.1007 respectively with a dice score of 0.7987.

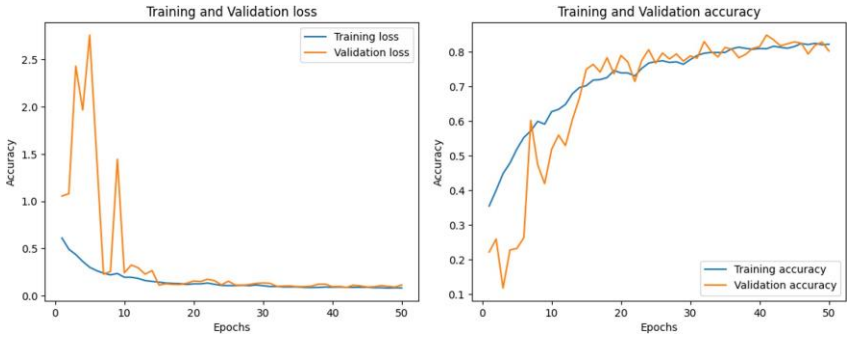

Figure 4: Training and Validation curves for accuracy and loss

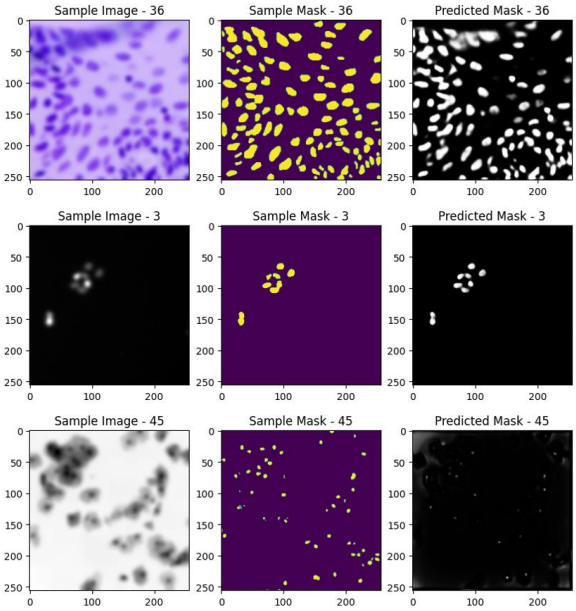

Figure 5: Segmentation result on test set

4.2 Design - 1 Results

Here we report our scores used to evaluate our modified UNet model, which incorporates the Design-1 physical layer to correct for spherical aberrations as explained in the methods section. The model achieved high accuracy and Dice scores on both the training and validation sets. The training accuracy and loss was found to be 0.9563 and 0.1001 respectively while the validation accuracy and loss was found to be 0.9507 and 0.1209 respectively with a validation dice score of 0.7962.

Figure 6: Training and Validation curves for accuracy and loss - design 1

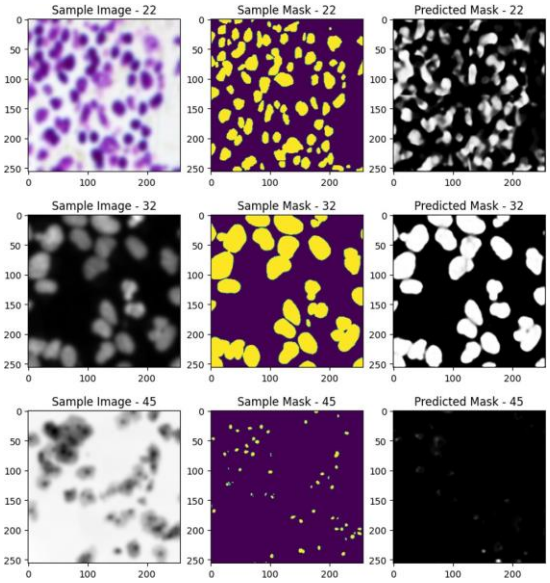

Figure 7: Segmentation result on test set - design 1

The testing accuracy and loss was found to be 0.9508 and 0.1223 respectively with a dice score of 0.7907.

4.3 Design - 2 Results

Here we report our scores used to evaluate our modified UNet model, which incorporates the Design-2 physical layer. The model achieved high accuracy and Dice scores on both the training and validation sets. The training accuracy and loss was found to be 0.9639 and 0.0808 respectively while the validation accuracy and loss was found to be 0.9502 and 0.1141 respectively with a validation dice score of 0.8041.

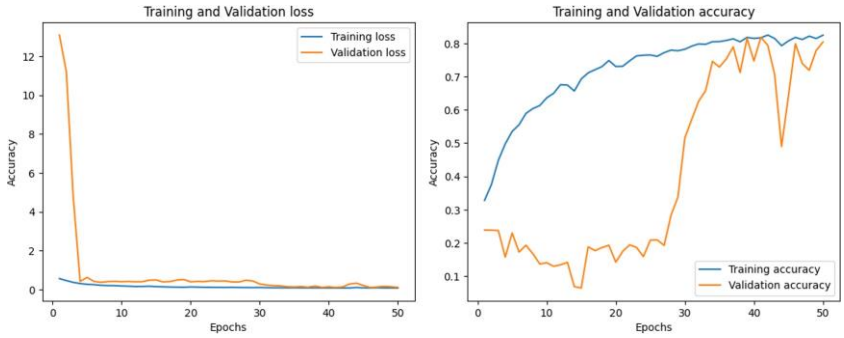

Figure 8: Training and Validation curves for accuracy and loss - design 2

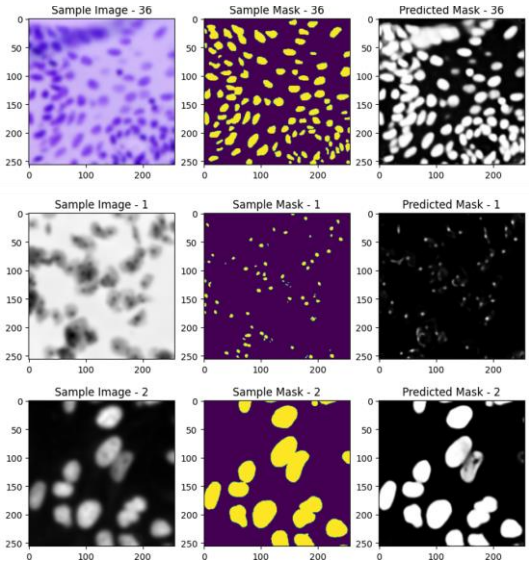

Figure 9: Segmentation result on test set - design 2

The testing accuracy and loss was found to be 0.9543 and 0.1055 respectively with a dice score of 0.7801.

4.4 Design - 3 Results

Here we report our scores used to evaluate our modified UNet model, which incorporates the Design-3 physical layer. The model achieved high accuracy and Dice scores on both the training and validation sets. The training accuracy and loss was found to be 0.9650 and 0.0787 respectively while the validation accuracy and loss was found to be 0.9610 and 0.0865 respectively with a validation dice score of 0.8303.

The testing accuracy and loss was found to be 0.9644 and 0.0770 respectively with a dice score of 0.8137.

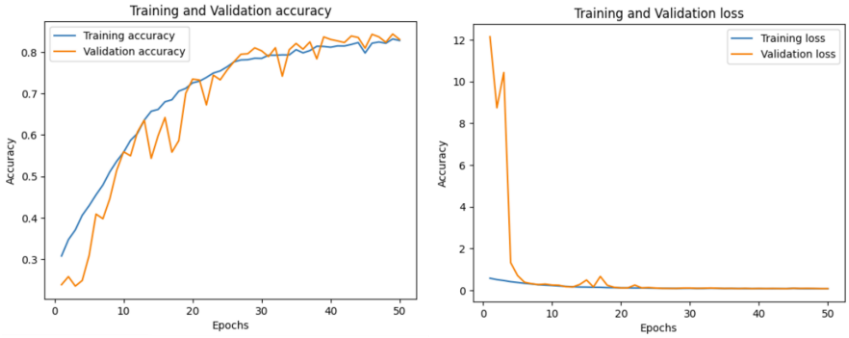

Figure 10: Training and Validation curves for accuracy and loss - design 3

Table 1 summerizes the model results with and without our physical layer designs on the test data.

Table 1: Model Summary

Name	Description	Test Dice
UNet	No Physical Layer	~0.7987
UNet	Design-1 Physical Layer	~0.7907
UNet	Design-2 Physical Layer	~0.7801
UNet	Design-3 Physical Layer	~0.8137

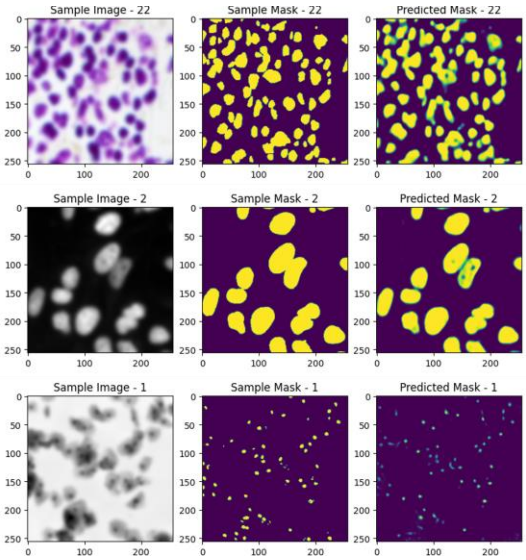

Figure 11: Segmentation result on test set - design 3

5 Discussion

The project aimed to improve the segmentation accuracy of microscopy data affected by spherical aberrations, which can result in distorted and blurred images. We tested three different versions of a trainable physical layer incorporated into a deep neural network called UNet to optimize model performance. Two of the designs showed similar performance to the case without a physical layer, but the third design, Design-3, outperformed the others with a 0.2 increase in the Dice Score, which is a measure of segmentation accuracy. This promising result with Design-3 lead us to see potential in further investigating if this physical layer could be applied to other aberrations and whether increasing the number of brightfield images could lead to better generalization. These experiments will help determine the generalizability and potential of the physical layer approach for improving segmentation accuracy in microscopy data.

In addition, we plan to investigate the possibility of combining the physical layer with other data augmentation techniques such as random rotations and translations. This could potentially further improve the performance of the segmentation model on microscopy data with various aberrations. Furthermore, we aim to explore the transferability of the Design-3 physical layer to other types of neural networks commonly used for image segmentation, such as Fully Convolutional Networks (FCNs) and Mask R-CNNs. This would provide insights into the adaptability of the physical layer to different architectures and could potentially lead to the development of more efficient segmentation models for a wider range of microscopy data.

Overall, our results demonstrate the potential of incorporating a trainable physical layer to enhance the performance of deep neural networks for the segmentation of microscopy data with spherical aberrations. This approach has the potential to improve the accuracy of segmentation tasks while also providing a cost-effective solution for researchers who may not have access to high-end imaging systems.

6 Acknowledgments

We would like to acknowledge the seminal work of Ronneberger et al. (2015) on the development of the UNet architecture, which served as the basis for our segmentation model. We would also like to thank Dr. Horstmeyer for guiding us through this project.

References

[1] Sanderson, Jeremy. Understanding light microscopy. John Wiley & Sons, 2019.

[2] Diel, Erin E., Jeff W. Lichtman, and Douglas S. Richardson. "Tutorial: avoiding and correcting sample-induced spherical aberration artifacts in 3D fluorescence microscopy." Nature protocols 15.9 (2020): 2773-2784.

[3] Krishnan, Anitha Priya, et al. "Optical aberration correction via phase diversity and deep learning." bioRxiv (2020): 2020-04.

[4] Wang, Yangyundou, et al. "High-accuracy, direct aberration determination using self-attention-armed deep convolutional neural networks." Journal of microscopy 286.1 (2022): 13-21.

[5] Albert, O., et al. "Smart microscope: an adaptive optics learning system for aberration correction in multiphoton confocal microscopy." Optics letters 25.1 (2000): 52-54.

[6] X, et al. "Generative adversarial networks for correction of spherical aberration in transmission electron microscopy." Proceedings of the National Academy of Sciences, vol. 117, no. 43, 2020, pp. 26633-26640. doi: 10.1073/pnas.2010372117.

[7] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, 2015