Phase Retrieval from Point Spread Function with Deep Learning Compared to Machine Learning

Wanting Xie

Abstract—Phase retrieval is a crucial challenge in various optics and imaging systems, enabling the recovery of phase information from intensity measurements. This process is vital in fields such as microscopy, holography, and astronomy. Traditional phase retrieval methods, including iterative algorithms like the Gerchberg-Saxton and Fienup algorithms, often depend heavily on prior system knowledge and can be computationally intensive and error-prone. Recent advancements in deep learning (DL) and machine learning (ML) have introduced innovative approaches that address phase retrieval problems with greater efficiency and accuracy. This project aims to evaluate the effectiveness of a deep learning-based phase retrieval method using Point Spread Function (PSF) and compare its performance to a machine learning-based approach.

Index Terms—Phase Retrieval, Point Spread Function, Zernike Polynomials, Machine Learning, Deep Learning

1 Introduction

PHASE retrieval is a fundamental problem in many areas of optics and imaging systems. It allows the recovery of phase information from intensity measurements, which is crucial in fields such as microscopy, holography, and astronomy. Traditional phase retrieval methods, such as iterative algorithms, rely heavily on prior knowledge of the system's characteristics and may be computationally expensive or prone to errors. With deep learning (DL) and machine learning (ML) techniques, there has been a surge in novel approaches to solve phase retrieval problems more efficiently and accurately. This project investigate the effectiveness of deep learning-based phase retrieval method using PSF and compare the performance to machine learning-based approach.

2 Related Work

Many algorithms, such as the Gerchberg-Saxton and Fienup algorithms, have been developed over the years to perform phase retrieval from intensity measurements. [1] These methods require multiple intensity measurements at different object-plane positions or various angles, and they are often iterative, requiring high computational power.

Recent advancements in deep learning have led to the development of neural networks for phase retrieval. Methods like convolutional neural networks (CNNs) and U-Net architectures have shown promising results in recovering phase from noisy and sparse data. [2], [3]

Machine learning techniques have also been explored for phase retrieval tasks. These methods, however, require careful feature extraction and may not perform as well as deep learning models when it comes to handling complex and large-scale data. [4]

3 Proposed Method

3.1 Basics

3.1.1 Point spread function

Point spread function (PSF) is a crucial concept in optics that describes how a point source of light is represented in an imaging system. Essentially, it measures the response of an imaging system to a point source or point object. It is a measure of intensity as a function of the x and y coordinates, with the optical path typically along the z-axis. In an ideal (aberration free) optical system, the PSF (Fig.1) would be a sharp peak at the focal point, indicating perfect focus. However, in real (with aberration) systems, various errors such as aberrations can cause the PSF (Fig.2) to spread, resulting in a less sharp image. PSF can be expressed as a function of apodization A and wavefront error ϕ.

$$PSF = |\mathcal{F}[A(u,v)e^{i\phi(u,v)}]|^2 \tag{1}$$

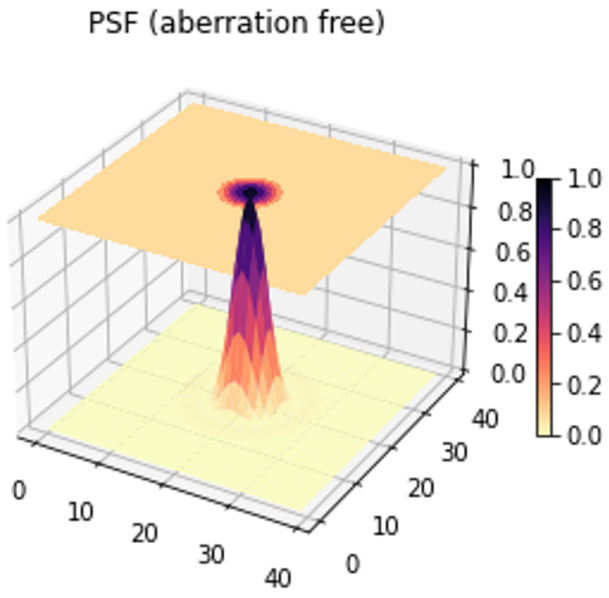


Fig. 1. PSF (Aberration Free)

The through-focus PSF, as illustrated in Fig.3, represents the PSF at various positions along the optical axis. The example provided shows the PSF ranging from -0.5 to +0.5

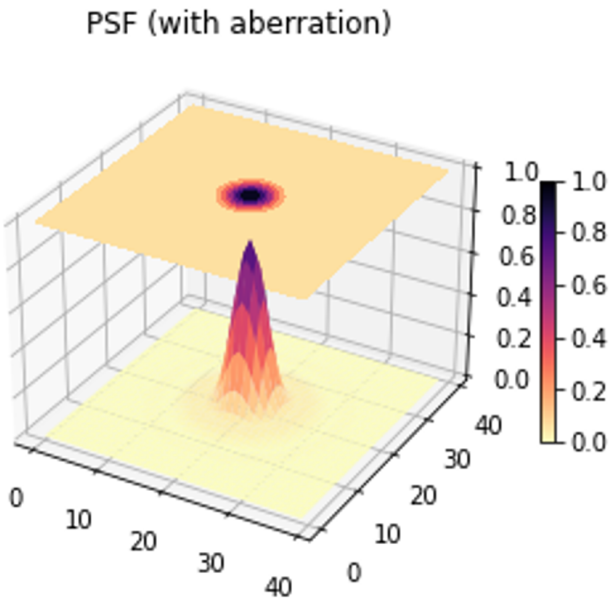


Fig. 2. PSF (with Aberration)

wavelengths (Waves) of defocus. It is evident that the PSF at defocused positions is more widely spread and contains a richer set of information, which is beneficial for model training. Previous studies have indicated that a focused PSF does not offer sufficient information to accurately estimate aberrations. In contrast, introducing defocus simplifies the process of determining aberration coefficients. These studies have also found that for identifying small aberrations, with an absolute value of less than 0.5 Waves, the optimal defocus range is between 0.1 and 0.2 Waves. This range provides the best balance, ensuring that the defocused PSF contains enough detail to facilitate precise aberration estimation. The broader spread of the defocused PSF captures more intricate details, making it a valuable asset in the training and refinement of models aimed at aberration correction. [5]

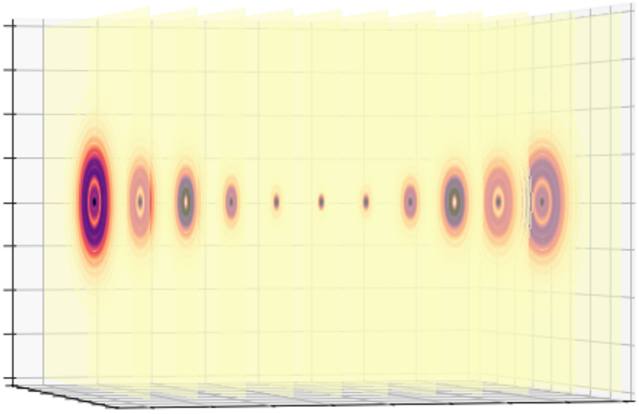


Fig. 3. Through-Focus PSF (-0.5 to +0.5 Wave)

3.1.2 *Wavefront Error*

Wavefront error (WFE) is a measure of the deviation of an optical wavefront from its ideal shape. In an ideal optical system, light waves would form a perfect spherical wavefront as they converge to a focal point. However, due to imperfections and aberrations in the optical system, the actual wavefront deviates from this ideal shape. (Fig.4)

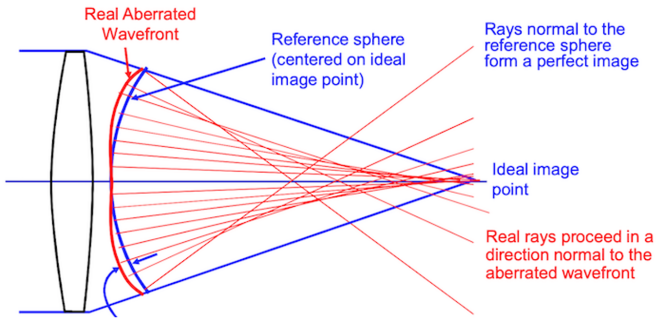
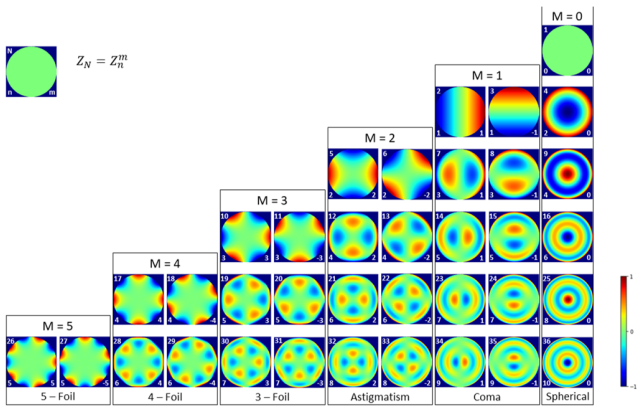


Fig. 4. Wavefront Error [6]

3.1.3 *Zernike polynomials*

Zernike polynomials are a sequence of polynomials that are orthogonal on the unit disk. Named after optical physicist Frits Zernike, laureate of the 1953 Nobel Prize in Physics and the inventor of phase-contrast microscopy. Zernike polynomials can be used to describe wavefront aberrations in an optical system. [7] Here we use up to n=36. (Fig. 5)

$$\phi(u,v) = \sum_{n=1}^{n} c_i Z_i(u,v) \tag{2}$$



$$\left.\begin{aligned} Z_{\text{even } j} &= [2(n+1)]^{1/2} R_n^m(r)\cos m\theta \\ Z_{\text{odd } j} &= [2(n+1)]^{1/2} R_n^m(r)\sin m\theta \end{aligned}\right\} \quad m \neq 0$$
$$Z_j = [(n+1)]^{1/2} R_n^m(r) \qquad m = 0 \tag{1}$$

where

$$R_n^m(r) = \sum_{s=0}^{(n-m)/2} \frac{(-1)^s (n-s)!}{s![(n+m)/2-s]![(n-m)/2-s]!} r^{n-2s}. \tag{2}$$

Fig. 5. Zernike Polynomials

3.2 Method

3.2.1 Optics Parameters

The following are the optical parameters used for the Point Spread Function (PSF) calculation. The wavelength of light utilized in this study is 550 nm. The numerical aperture (NA) of the optical system is set to 0.65, as illustrated in Fig.6. Based on these parameters, the pixel size is calculated using the formula 0.1×wavelength/NA, resulting in a pixel size of approximately 84.6 nm. Additionally, the dimensions of the image are specified, with both the width and height measuring 1025 pixels. These parameters are crucial for accurately modeling the PSF and ensuring precise image reconstruction.

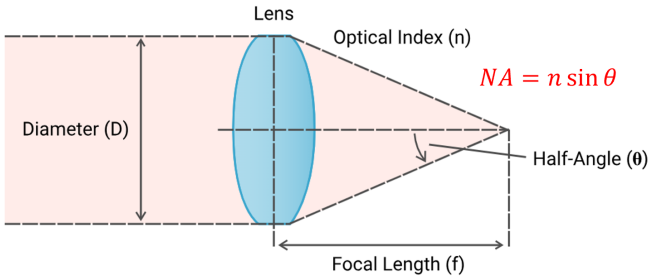


Fig. 6. Numerical Aperture (NA)

3.2.2 Generate Training Samples

In this study, a total of 100 and 500 samples of WFEs with RMS value within 70 miliwavelengths (mWaves) were randomly generated to simulate various optical aberrations. I chose 70 mWaves because he image quality is not sensibly degraded when the total wavefront error does not exceed 70 mWaves RMS. For each WFE sample, PSF images were generated across a specified focus range, spanning from -500 to +500 mWaves in increments of 50 mWaves. This comprehensive range ensures a detailed analysis of the PSF variations due to different WFEs. The subsequent step involved flattening each generated PSF image and normalizing it by the peak value of the aberration-free PSFs. This normalization process is crucial for accurately comparing the PSF images and assessing the impact of the WFEs. The detailed procedure and results are illustrated in Fig.7.

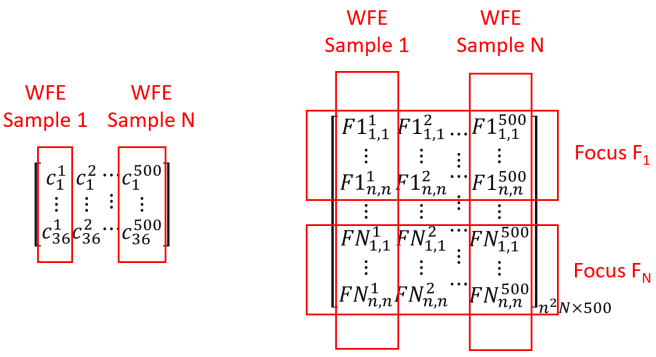


Fig. 7. Training Samples

3.2.3 Model Training

Two distinct methods were evaluated in this study. For the Deep Learning approach, various algorithms were explored to determine their effectiveness. Specifically, the Machine Learning method employed Linear Regression, while the Deep Learning method utilized a Fully Connected Neural Network.

1) Machine Learning (ML): Linear Regression (LR)
2) Deep Learning (DL): Fully Connected Neural Network (FCNN)

Deep Learning model was tuned to optimize the performance. Here are some key parameters. Each of these was assessed to identify the most suitable one for the given task.

- Hyperparameter Tuning
  - Learning Rate
  - Batch Size
  - Number of Epochs
- Model Architecture
  - Layer Configuration
  - Activation Functions
  - Dropout Rate
- Optimization Algorithms
  - Adam
  - AdamW
  - SGD
  - RMSprop
  - Adagrad
  - Adadelta

3.2.4 Testing

A total of 100 test WFEs, which serve as the ground truth, were generated along with their corresponding PSFs. Using these PSFs, the trained model was employed to retrieve the WFEs. The retrieved WFEs were then meticulously compared with the ground truth to evaluate the model's accuracy and performance. This comparison provided valuable insights into the model's ability to accurately reconstruct the wavefront errors from the given point spread functions. The modeling error is defined by

$$RSS\ Error = \sqrt{\sum_{n=1}^{n} |c_{i_{predict}} - c_{i_{truth}}|^2} \tag{3}$$

4 Experimental Results

4.1 Machine Learning

Initially, I tested the linear regression model to evaluate its performance. The model's error, when compared to the ground truth, was found to be 6.8 mWaves. The results, including both the predicted and true Zernike coefficients, are illustrated in Fig.8 and Fig.9. To ensure a straightforward comparison, I will apply the same test example to the subsequent models and algorithms.

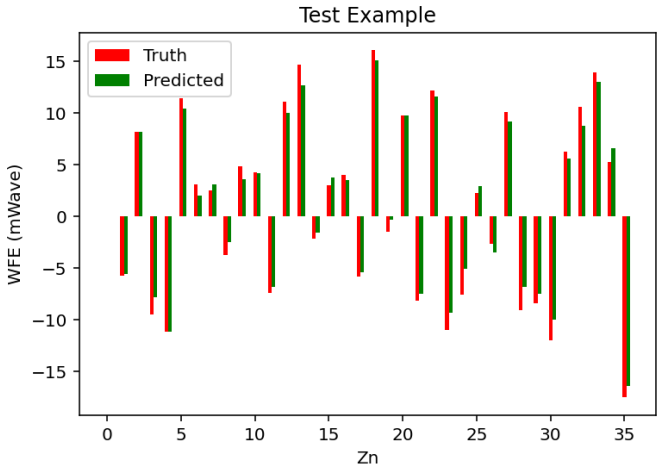


Fig. 8. One Test Example of Linear Regression - Zernike Coefficients

4.2 Deep Learning Model Tuning

Subsequently, I conducted tests on various deep learning algorithms, each configured with two layers. The models were trained using a dataset comprising 100 samples. Upon evaluating the results, it was evident that the Adam optimizer produced predictions that were the closest to the ground truth, outperforming the other algorithms tested.

In my experiments with the Adam optimization method, I explored the effects of increasing both the number of layers in the model and the number of training samples. The detailed results of these experiments are presented in Table 1. Interestingly, the data indicates that adding more layers to the model actually led to a decline in performance, yielding worse results compared to the original configuration. On the other hand, increasing the number of training samples resulted in a slight improvement in the model's accuracy, suggesting that while more data can enhance performance, the model's architecture also plays a crucial role in achieving optimal results.

TABLE 1
Compare Number of Layers and Training Samples for FCNN Adam

Number of Layers	Number of Training Samples	Error (mWaves)
Two	100	34.84
Four	100	46.80
Two	500	32.12
Four	500	46.29

Choosing the right activation function is crucial for the performance of a deep learning model, as it directly impacts the model's ability to learn and generalize from data. Several commonly used activation functions were tested with the Adam optimization method to evaluate their effectiveness. ReLU (Rectified Linear Unit) is widely used in hidden layers of neural networks due to its simplicity and effectiveness in mitigating the vanishing gradient problem, which can hinder the training of deep networks. The Sigmoid function, often employed in the output layer for binary classification problems, can suffer from vanishing gradients, making it less effective for deeper networks. Tanh (Hyperbolic Tangent) is similar to Sigmoid but outputs values between -1 and 1, making it zero-centered and potentially more effective in certain scenarios. Leaky ReLU addresses the "dying ReLU" problem by allowing a small, non-zero gradient when the unit is not active, thus ensuring that neurons do not become inactive during training. Softmax is used in the output layer for multi-class classification problems to produce a probability distribution over multiple classes, making it essential for tasks involving multiple categories. Swish, a newer activation function, has shown to outperform ReLU in some cases, particularly in deeper networks, due to its smooth and non-monotonic nature. Lastly, ELU (Exponential Linear Unit) helps to mitigate the vanishing gradient problem and can lead to faster learning by allowing the model to converge more quickly. Each of these activation functions has its strengths and weaknesses, and the choice of which to use often depends on the specific characteristics of the data and the architecture of the model. The experimental results indicate that the Tanh activation function yields the lowest error rate, as illustrated in Fig.11. This finding highlights the effectiveness of Tanh in minimizing errors compared to other activation functions tested. Furthermore, it was observed that increasing the number of layers in the neural network did not contribute to any significant improvement in performance. Despite the expectation that additional layers might enhance the model's accuracy, the results suggest that a deeper network does not necessarily lead to better outcomes in this particular case.

Based on the results of the tuning tests, it was determined that using 500 training samples, a neural network with two layers, the Tanh activation function, and the Adam optimization algorithm collectively provided the best overall performance. This combination of parameters was found to be the most effective in optimizing the model's accuracy and efficiency. The detailed outcomes of one specific test example, which illustrate the model's performance under these conditions, are presented in Fig.12 and Fig.13. The RSS error is 12.1 mWaves.

4.3 Machine Learning vs. Deep Learning Comparison

Figure 14 illustrates the RSS error distribution across 100 tests. The ML model achieves an average error of 4.4 mWaves with a standard deviation of 4.1 mWaves, whereas the DL model has a higher average error of 11.6 mWaves with a standard deviation of 4.6 mWaves. The best-performing DL result among the 100 tests still yields an error of 7 mWaves, which remains significantly worse than ML. In terms of computational efficiency, training the ML model on 500 samples takes only 6 seconds, compared to 781 seconds for the DL model. Overall, ML outperforms DL in both accuracy and computational efficiency.

Besides the possibility that the DL model is not tuned to its best, there are several reasons why linear regression might outperform a fully connected neural network in certain scenarios.

- Linear regression is a simpler model with fewer parameters, which makes it less prone to overfitting, especially when dealing with small datasets. In contrast, fully connected neural networks have many

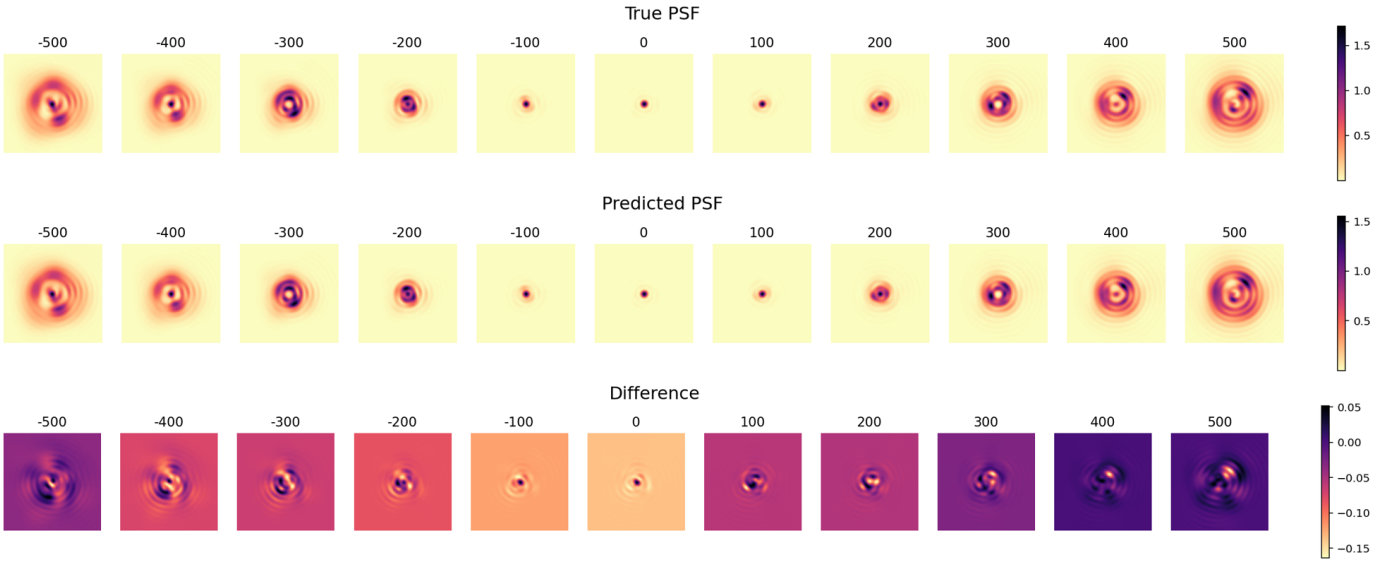


Fig. 9. One Test Example of Linear Regression - PSF

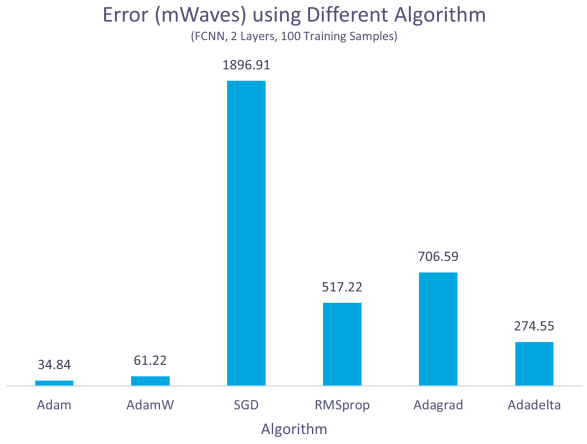


Fig. 10. Test Results of Different DL Algorithms

Fig. 12. One Test Example of Deep Learning - Zernike Coefficients

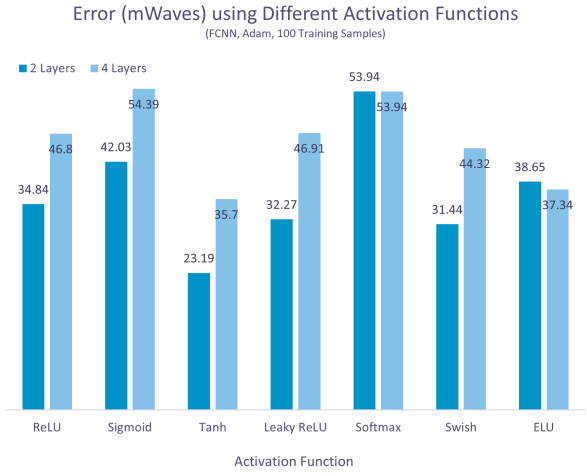


Fig. 11. Test Results of Different Activation Functions

parameters and can easily overfit the training data if not properly regularized.

- The loss surface for linear regression is convex, meaning it has a single global minimum. This makes it easier to find the optimal solution using methods like ordinary least squares. On the other hand, the loss surface for neural networks is non-convex, with many local minima, making optimization more challenging.
- Data Requirements: Neural networks generally require large amounts of data to perform well. If the dataset is small or not sufficiently diverse, a simpler model like linear regression might yield better results because it can generalize better from limited data.
- Linear regression is computationally less intensive compared to training a deep neural network. This can be an advantage when computational resources are limited or when quick results are needed.

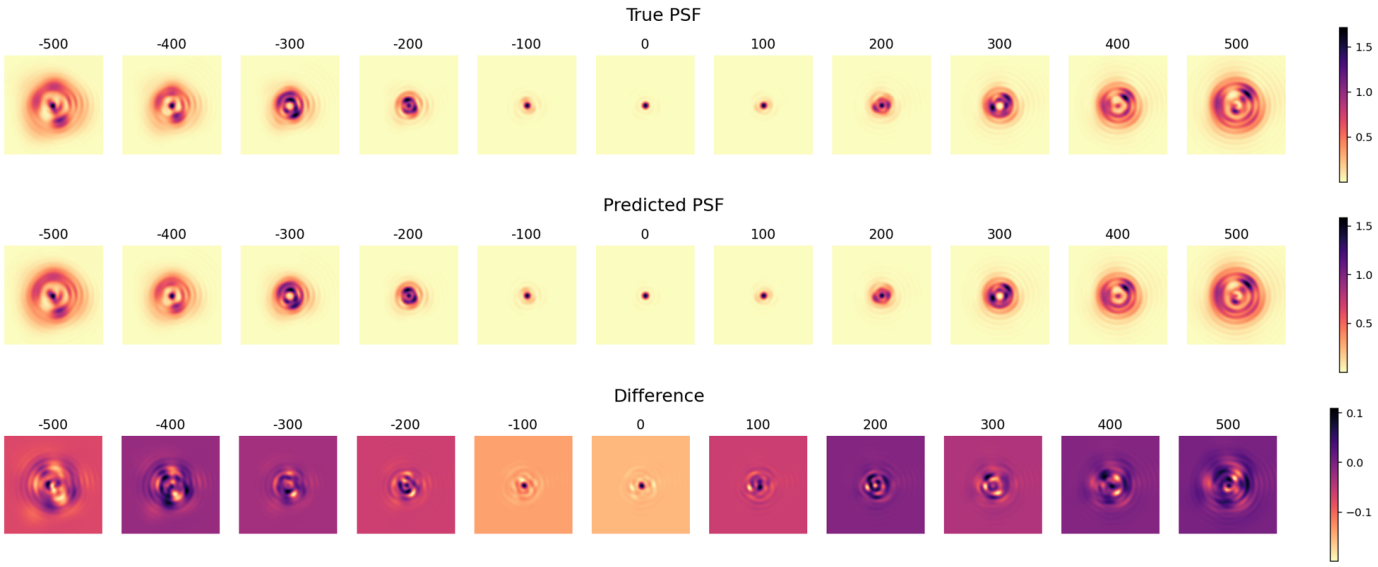


Fig. 13. One Test Example of Deep Learning - PSF

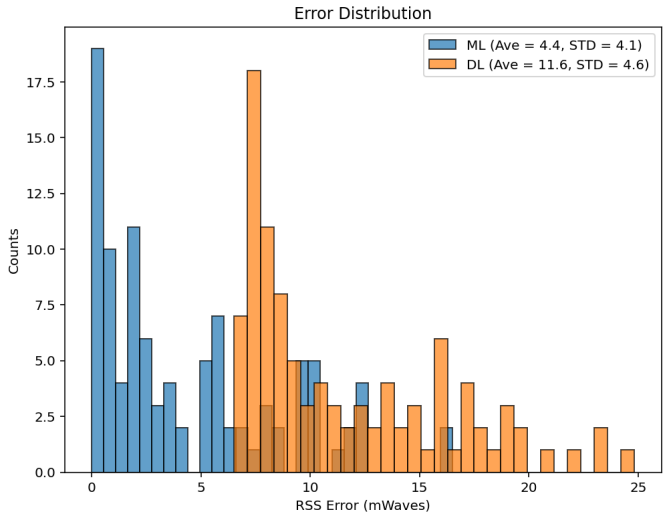


Fig. 14. RSS Error Distribution of 100 Tests: Machine Learning vs. Deep Learning

While neural networks are powerful tools for capturing complex patterns, linear regression can be more effective in scenarios where the data is limited, the relationship is linear, or computational efficiency is a priority. Deep learning remains highly promising, as models can be further optimized through tuning. Additionally, there are various other deep learning techniques, such as Convolutional Neural Networks (CNNs), that can be explored in the future to enhance performance.

5 Conclusion

In conclusion, this study reveals that while deep learning techniques have shown promise in various applications, they did not outperform traditional machine learning methods for phase retrieval in this particular case. The deep learning models, despite being optimized with the Point Spread Function (PSF) and various neural network architectures, yielded higher errors compared to the simpler linear regression model. This suggests that for the given dataset and problem complexity, the linear regression approach was more effective, likely due to its simplicity and lower risk of overfitting. The findings underscore the importance of selecting the appropriate model based on the specific characteristics of the data and the problem at hand. While deep learning holds potential for future improvements, traditional machine learning methods remain a robust and reliable choice for phase retrieval tasks in this study.

References

[1] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik (Stuttg.)*, vol. 35, p. 237–246, 1972.

[2] G. Ju, X. Qi, H. Ma, and C. Yan, "Feature-based phase retrieval wavefront sensing approach using machine learning," *Optics Express*, vol. 26, no. 24, pp. 31 767–31 783, 2018.

[3] A. P. Dzyuba, "Optical phase retrieval with the image of intensity in the focal plane based on the convolutional neural networks," *Journal of Physics: Conference Series*, vol. 1368, no. 2, p. 022055, 2019.

[4] N. Chimitt, A. Almuallem, and S. H. Chan, "Phase retrieval of a point spread function," *Unconventional Imaging, Sensing, and Adaptive Optics*, vol. 13149, pp. 220–224, 2024.

[5] O. Kalinkina, T. Ivanova, and J. Kushtyseva, "Wavefront parameters recovering by using point spread function," *CEUR Workshop Proceedings*, 2020.

[6] "Understanding rms wavefront error," https://www.opticsforhire.com/blog/rms-wavefront-error-explained/.

[7] J. Y. Wang and D. E. Silva, "Wave-front interpretation with zernike polynomials," *Applied Optics*, vol. 19, no. 9, pp. 1510–1518, 1980.