

# Compensation of Atmospheric Turbulence Phase Distortion Using Neural Network Adaptive Optics

Raaid Nawfee Hassan<sup>1a\*</sup>

<sup>1</sup>Department of Astronomy and Space, College of Science, University of Baghdad. Baghdad, Iraq

<sup>a\*</sup> Corresponding author: [raaid.hassan@sc.uobaghdad.edu.iq](mailto:raaid.hassan@sc.uobaghdad.edu.iq)

## Abstract

Adaptive optics revolutionizes telescopic resolution but faces cost, complexity, and calibration hurdles. Neural network adaptive optics (NNAO) offers promise by using neural networks to tailor corrections to telescopes and atmospheric conditions, by passing calibration and sensors. This MATLAB-based study examines NNAO's impact on astronomical image quality, revealing it as a cost-efficient solution that enhances adaptive optics in astronomy. The numerical simulation results were encouraging, with a compensation rate exceeding 50% due to favorable monitoring conditions. The results indicate that the dominant factor affecting image quality is the variance of wavefront aberrations. The Strehl ratio (SR) decreases from an average of 0.548 for a variance of 0.2 to 0.020 for a variance of 0.6, while the mean squared error (MSE) increases from an average of 0.613 to 5.414. However, the effect on peak signal-to-noise ratio (PSNR) is inconclusive. Furthermore, it was found that increasing the number of neurons and training ratio does not significantly impact the results obtained, but it noticeably affects the computational time required.

## Article Info.

### Keywords:

*Adaptive Optics, Optical Turbulence, Neural Network, MLP, Strehl Ratio.*

### Article history:

*Received: Sep.11, 2023*

*Revised: Dec. 9, 2023*

*Accepted: Dec.18, 2023*

*Published: Mar.01,2024*

## 1. Introduction

The Earth's atmosphere introduces distortions to light passing through it, resulting in blurred and distorted images of astronomical objects. To address this issue, adaptive optics (AO) technology has been developed to compensate for these distortions in real-time, thereby improving image quality and facilitating new discoveries in the field of astronomy [1-3]. Traditional AO systems typically incorporate a wavefront sensor to measure atmospheric distortions and a deformable mirror to correct them. However, these systems encounter limitations when correcting nonlinear aberrations and necessitate intricate models to estimate the wavefront distortions accurately [4-6]. Machine learning in AO was investigated as early as the 1990s [7-9]. At that time, artificial neural networks (ANN) were considered to be a good alternative for the wavefront sensing of single-aperture and array telescopes in the multiple mirror telescope (MMT) [10-12]. In recent years, a promising alternative called neural network adaptive optics (NNAO) has emerged to address these challenges. NNAO harnesses the power of neural networks to acquire the necessary correction parameters specific to a telescope and prevailing atmospheric conditions, thereby eliminating the need for calibration and wavefront sensors and overcoming the limitations in AO systems [13-15]. NN-based AO systems leverage machine learning algorithms to learn and model the nonlinear relationship between distorted and corrected wavefronts. This enables real-time correction of complex wavefront distortions without relying on detailed models, thereby enhancing the efficiency and accuracy of the AO process [16, 17]. Two kinds of machine learning-based methods are proposed to improve the Shack-Hartmann wavefront sensor (SHWFS) performance. One is to improve the gradient-based method, such as building the relationship between aberrations and gradients with nonlinear fitting tools such as ANN instead of simple matrix multiplication, or improving the spot centroid accuracy by doing the spot classification with ANN before centroid calculation.

The other is to extract the aberration from the SHWFS image directly with deep learning instead of calculating the gradients. In some applications, traditional special wavefront sensors (WFSs) are not allowed, and the imaging setup can be used as the PD WFS or single-image phase retrieval WFS. In these cases, deep learning can be used to solve the nonlinear phase retrieval problem without many iterations required by traditional Gauss-Seidel or Stochastic parallel gradient descent (SPGD) methods, etc. Besides improving the accuracy and speed of traditional narrow field-of-view WFS, deep learning can also be used to improve the performance of tomography WFS [18-20]. ANN possess the capability to understand the relationship between wavefront sensor data and necessary wavefront adjustments. This ability enables rapid and precise aberration compensation, outperforming conventional techniques in terms of both speed and accuracy.

This paper explores using neural networks to learn and apply the necessary corrections specific to a particular telescope and prevailing atmospheric conditions. A MATLAB implementation of NNAO is presented, and its effectiveness in compensating for atmospheric distortions and improving image quality in astronomical observations is demonstrated.

## 2. Von Karman Atmospheric Turbulence Model

The Earth's atmosphere is a complex and heterogeneous medium consisting of two types of media: discrete turbid particles and a continuous turbulent medium composed of moving molecules. These media give rise to various optical phenomena, including energy attenuation, intensity scintillation, angle-of-arrival fluctuations, beam wander, beam spread, and spot distortion [21-23]. These phenomena depend on differential optical path lengths rather than absolute values, and their random variations are described by spatial statistics and structure functions [24-26].

The structure function ( $D_n(r)$ ), which measures the mean square difference in the index of refraction,  $n(r)$ , between two spatial locations,  $r_1$  and  $r_2$ , is defined as [27-29]:

$$D_n(r) = \langle |n(r_1) - n(r_2)|^2 \rangle \quad (1)$$

where  $r$  is  $r_1 - r_2$ .

Now, over the inertial sub-range,  $l_0$  and  $L_0$ , Kolmogorov's theory further establishes the structure function and can be expressed in terms of structure constants [30-32]:

$$D_n(r) = \begin{cases} C_n^2 r^{2/3} & l_0 < r < L_0 \\ C_n^2 l_0^{-4/3} r^{2/3} & r < l_0 \end{cases} \quad (2)$$

The constant,  $C_n^2$  quantifies the strength of the optical turbulence. A  $C_n^2$  value, on the order of magnitude of  $10^{-17} \text{ m}^{-2/3}$ , is considered a weak turbulence, and strong turbulence values are on the order of  $10^{-13} \text{ m}^{-2/3}$ . The findings from atmospheric measurements indicate that the Earth's atmosphere contains a combination of Kolmogorov turbulence at lower altitudes and non-Kolmogorov turbulence at higher altitudes. Currently, two models exist to describe the structure of atmospheric turbulence in the Earth's atmosphere. The first model is a two-layer model, which proposes the turbulence in atmospheric measurements. Atmospheric turbulence in the Earth's atmosphere exhibits Kolmogorov turbulence at lower levels and non-Kolmogorov turbulence at higher levels. Two models describe the structure of atmospheric turbulence: a two-layer model and a three-layer model [33-35]. The two-

layer model posits Kolmogorov turbulence in the troposphere and non-Kolmogorov turbulence in the stratosphere. The revised two-layer model incorporates consistency between the structure of the refractive index and power spectral density. The three-layer model expands on the two-layer model by adding non-Kolmogorov turbulence in the free atmosphere.

$$\phi_{nB}(k, z) = 0.033 C_n^2(z) K^{-11/3} \quad (3)$$

$$\phi_{nF}(k, z) = 0.015 C_{nF}^2(z) K^{-10/3} \quad (5)$$

Eqs (3), (4), and (5) represent the mathematical expressions for the turbulence in each layer, with varying turbulence parameters and wave numbers [36-38]. where:  $z$  is a propagation distance that varies between  $z = 0$  and  $z = L$ ,  $C_n^2$  is Kolmogorov turbulent index of refraction structure parameter in the boundary layer (has unit of  $m^{-2/3}$ ),  $C_{nF}^2$  is the non-Kolmogorov turbulent index of refraction structure parameter in the free troposphere (has unit of  $m^{-1/3}$ ), and  $C_{nS}^2$  is the non-Kolmogorov turbulent index of refraction structure parameter in the stratosphere (of  $m^{-2}$  units). It is important to note that the structure of atmospheric turbulence varies with altitude and atmospheric conditions [39-41]. The coherence diameter of the atmosphere, denoted by  $r_0$ , is a commonly used parameter. It influences the phase power spectral density (PSD) for the Von Karman refractive index, as described by Eq. (6) [42-44]:

$$\Phi_{\phi}^{VK}(k) = 0.49 r_0^{-5/3} (k^2 + k_0^2)^{-11/3} \quad (6)$$

where:  $k$  is the transverse wavenumber,  $k < 1/L_0$ ,  $r_0$  is Fried coherent diameter, and  $k_0 = 2\pi/L_0$  where  $L_0$  is the outer scale.

### 3. The Main Steps of Neural Network based Adaptive Optics

The implementation of a Neural Network-based AO (NN-based AO) system can be divided into the main steps as follows:

1. Data preparation: The initial stage in constructing a neural network is to ready the data for the network's training. This process encompasses the gathering and arrangement of data, as well as cleaning and preprocessing. Specifically, it entails capturing images of a designated source, such as a star, utilizing a wavefront sensor and deformable mirror. The reference source should have a bright and stable point spread function (PSF) to ensure an accurate measurement of the wavefront distortion. The images are captured with different levels of wavefront distortion introduced by the deformable mirror [45].
2. Model definition: After the data has been prepared, the next step is to define the neural network model architecture. This involves selecting the appropriate type of network, specifying the number and types of layers, and setting the hyperparameters such as learning rate, batch size, and activation functions [46].
3. Training the network: Once the model is defined, the network is trained on the prepared data. During training, the network learns to recognize patterns and make predictions based on the input data. This involves using an optimization algorithm such as gradient descent to adjust the weights and biases of the network based on the errors or losses between the predicted and actual outputs [47].
4. Evaluation: After training, the performance of the network is evaluated on the validation set to ensure that it is not overfitting the training data. Various metrics such as accuracy, precision, and recall can be used to evaluate the performance [48].

5. Testing: Finally, the trained network is tested on the test set to measure its real-world performance. The testing results can be used to make improvements to the network or to fine-tune the hyperparameters for better performance [49].
6. Deployment: Once the model has been evaluated and tested, it can be deployed in a real-world application for which it was designed. In some cases, the model may need to be retrained periodically with new data to maintain its performance over time [50-52].

Following these steps, an NN-based AO system can be implemented to effectively correct wavefront distortions and improve image quality in astronomical observations.

#### 4. Multilayer Perceptron Neural Network Algorithm

Fig. 1 shows the various aspects of the multilayer perceptron neural network (MLP) algorithm, its progress, performance plots, training state, error histogram, and regression. Note that all the figures in the following paragraphs are the results of our numerical simulation for the multilayer perceptron (MLP) neural network algorithm.

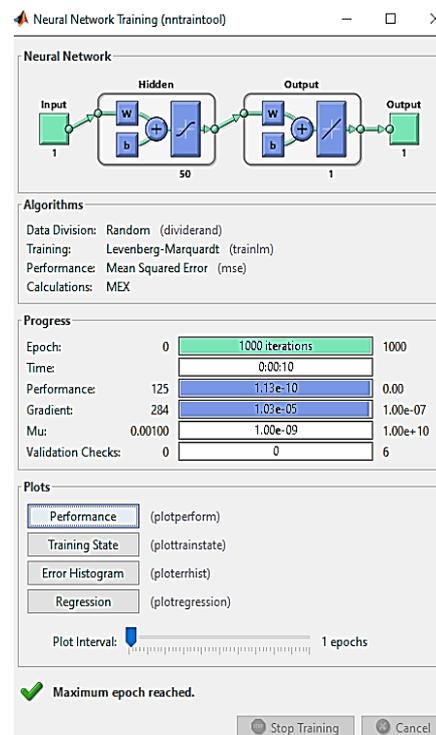


Figure 1: Graphical user interface (GUI) tool.

The neural network training (nntool) is a graphical user interface (GUI) MATLAB tool designed for training neural networks interactively as in Fig. 1. It offers insights into training progress and performance. An explanation of its sections and components is as follows:

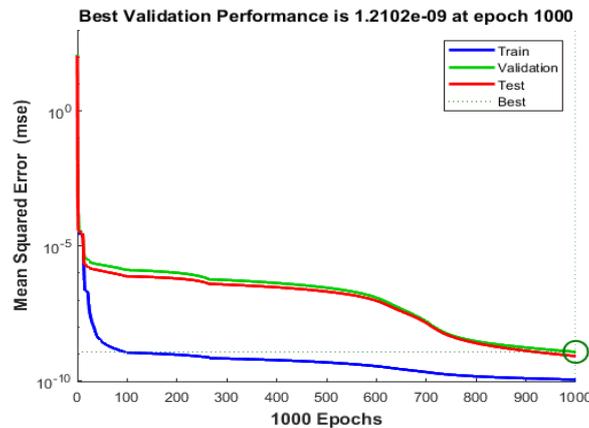
##### 1- Algorithms:

- The neural network used is a feedforward MLP neural network with a single hidden layer.
- The number of hidden units in the hidden layer is set to 50.
- The network is trained using the backpropagation algorithm, which involves iteratively adjusting the weights and biases to minimize the error between the predicted and desired wavefront corrections.

## 2- Progress and Plots:

The nntraintool GUI provides several plots to visualize the training progress and the neural network performance. These plots are updated dynamically during the training process. The following plots are typically included:

- 2.1. Performance Plot: Fig. 2 displays the training and testing performance of the neural network over epochs or iterations. It shows how the performance metric (e.g., mean squared error) changes as training progresses. The goal is to minimize the performance value.
- The program trains the neural network using the training data, consisting of wavefront sensor measurements and desired wavefront corrections
  - The network is trained for a specified number of epochs (1000 epochs in this case) using the training options.
  - After training, the network is used to predict wavefront corrections for a new wavefront sensor measurement.



**Figure 2: Performance Plot.**

2.2. Training State Plot: Fig. 3 represents gradient, adaptive learning rate parameter ( $\mu$ ) and validation failures (val. fail.) that provide insights into the training process and the performance of the neural network during different epochs.

- Gradient subplot: The gradient plot shows the gradients of the network's parameters (weights and biases) during training. It helps understand how the network's parameters are adjusted to minimize the error function. Monitoring the gradient helps to ensure that the network is converging towards a solution. Ideally, the gradient should decrease over epochs as the network learns to make more accurate predictions.
- The adaptive learning rate parameter ( $\mu$ ) subplot: The " $\mu$ " figure shows the adaptive learning rate parameter ( $\mu$ ) over epochs. The  $\mu$  parameter represents the current learning rate at each epoch. Monitoring the  $\mu$  values helps understand how the learning rate adapts over time and whether it is converging or fluctuating. Consistently decreasing or stabilizing  $\mu$  values indicates a good learning rate adjustment strategy.

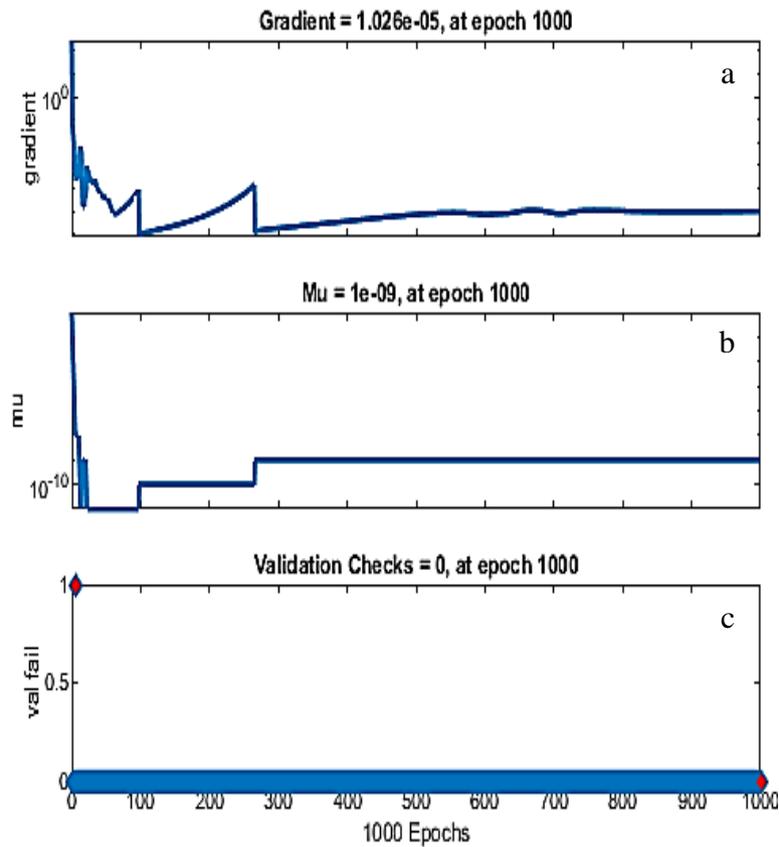


Figure 3: Training State Plots.

- Validation Failures (val. fail.) subplot: The val. fail. figure depicts the number of validation failures or misclassifications during training. It measures the performance of the neural network on a validation dataset, which is separate from the training dataset. By monitoring the val. fail., it is possible to assess the generalization capability of the network and identify epochs where the network's performance may have deteriorated. The goal is to minimize the number of validation failures over epochs, indicating improved accuracy and generalization.

2.3. Error Histogram: The error histogram (Fig. 4) represents the distribution of errors or residuals between the desired wavefront corrections and the predicted corrections. It provides insights into the error distribution and helps identify any bias or skewness.

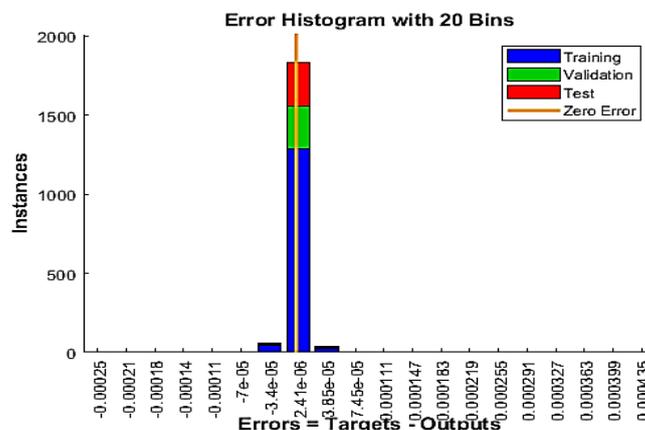
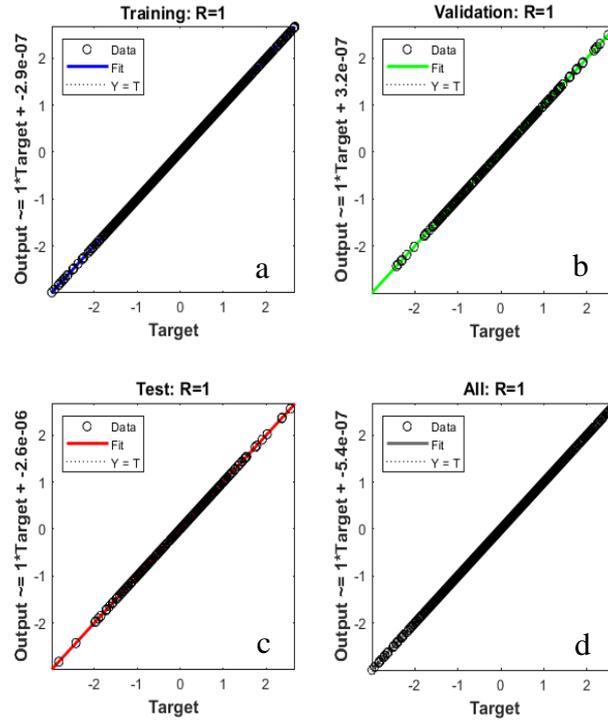


Figure 4: Error Histogram.

2.3 Regression Plot: The regression plot (Fig. 5) shows the relationship between the desired wavefront corrections and the predicted corrections. It helps assess the correlation and fit between the predicted and desired values. The predicted wavefront corrections are compared with the desired corrections to evaluate the accuracy of the regression model.

In Fig. (5 a,b,c,d), the y-axis represents the performance metric (accuracy) calculated on training, validation, test, and all test datasets, respectively.



**Figure 5: Regression Plots (a, b, c, d) for training, validation, test and all regressions progress respectively.**

## 5. Numerical Simulation Steps of NN Based Adaptive Optics System

The provided program is a numerical computer simulation of an NN-based AO system. A revised description of the simulation steps is as follows:

1. Set up the simulation parameters, including the wavelength ( $\lambda$ ), wave number ( $k$ ), Fried parameter ( $r_0$ ), outer scale ( $L_0$ ), aperture diameter ( $D$ ), number of grid points ( $n$ ), simulation domain size ( $L$ ), and grid spacing ( $dx$ ).
2. Generate a 2D wavefront sensor measurement by creating a random phase screen using the Von Karman model, and the resulting matrix is scaled by the wavefront variance.
3. Normalize the wavefront sensor measurements and desired wavefront corrections by subtracting the mean and dividing by the standard deviation.
4. Reshape the normalized wavefront sensor measurements and desired wavefront corrections into column vectors.
5. Split the data into training and testing sets using a specified training data ratio.
6. Extract the training and testing data based on the indices obtained from the previous step.
7. Create a feedforward neural network with a specified number of hidden units/neurons.
8. Train the neural network using the training data.

9. Generate a new wavefront sensor measurement by creating a random phase screen using the Von Karman model and scale it by the wavefront variance.
10. Normalize the new wavefront sensor measurement using the mean and standard deviation of the original wavefront sensor measurements.
11. Pass the normalized new measurement through the trained neural network to obtain the predicted wavefront corrections.
12. De-normalize the predicted wavefront corrections by multiplying by the standard deviation and adding the mean of the desired wavefront corrections.
13. Reshape the predicted wavefront corrections back into a 2D matrix with the same dimensions as the original wavefront.
14. Calculate performance metrics such as mean squared error (MSE), peak signal-to-noise ratio (PSNR), and Strehl ratio, which is the maximum intensity of the desired wavefront corrections divided by the square of the maximum absolute value of the predicted corrections.
15. Display the results using subplots. The first subplot shows the original wavefront sensor measurements; the second subplot shows the desired wavefront corrections; the third subplot shows the new wavefront sensor measurement; the fourth subplot shows the predicted wavefront corrections and an additional figure displays the difference between the new measurement and the predicted corrections.

## 6. Numerical Simulation Results and Discussion

The utilization of a neural network for adaptive optics compensation is demonstrated through a numerical simulation. In this simulation, the network is trained using wavefront sensor measurements and the desired wavefront corrections. Once trained, the network is applied to predict corrections for new measurements. The numerical simulation explores various conditions related to turbulence strength, training data ratio, and the number of neurons in the hidden layer. These conditions impact the performance of the neural network for adaptive optics compensation. Based on these different conditions, the simulation generates a visual representation of the results in the form of a figure with four subplots (Figs. 6, 8). The wavefront representation used in the simulation is derived from the Von Karman turbulence model.

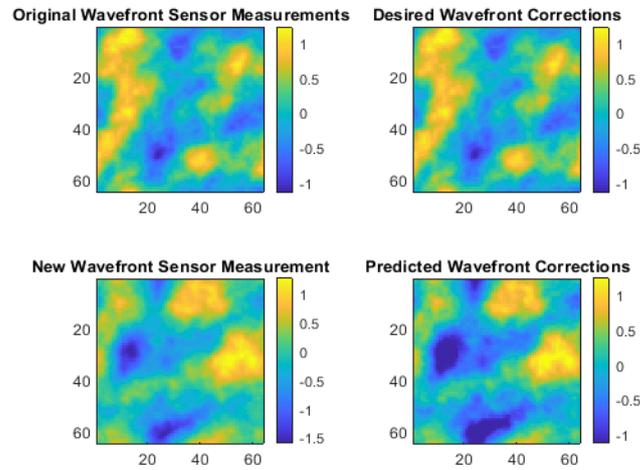
For different conditions related to the turbulence strength (variance amount of the simulated wavefront aberrations), training data ratio, and the number of neurons in the hidden layer, the simulation generates a visual representation of the results through a figure consisting of four subplots (Figs. 6, 8).

- Subplot 1 presents the original wavefront sensor measurements.
- Subplot 2 displays the desired wavefront corrections.
- Subplot 3 show cases of the new wavefront sensor measurement.
- Subplot 4 illustrates the predicted wavefront corrections.

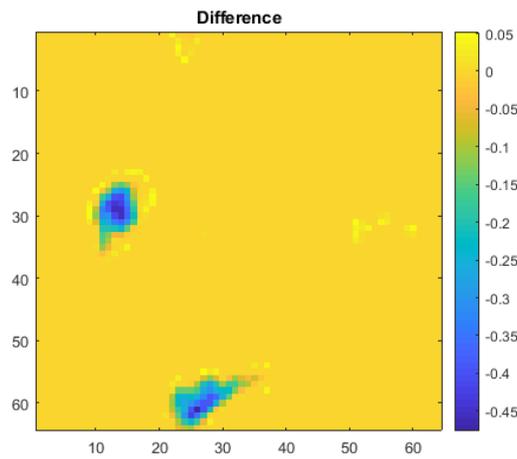
It is important to note that the wavefront representation in the simulation is derived from the Von Karman turbulence model, which accurately captures the characteristics of the simulated wavefront aberrations. Furthermore, separate Figs. 7, 9 are dedicated to show the difference or the residual error between the new measurement and the predicted corrections.

To assess the effectiveness of the neural network in compensating for wavefront aberrations under various conditions, Tables 1-9 present the performance metrics, including mean square error (MSE), peak signal-to-noise ratio (PSNR), and Strehl ratio (SR). These metrics provide valuable insights into the network's performance under different conditions of turbulence strength, training data ratio, and hidden layer neuron count. By examining these tables, one can gain insights into the performance of the neural network in terms of its ability to minimize MSE, maximize PSNR, and improve

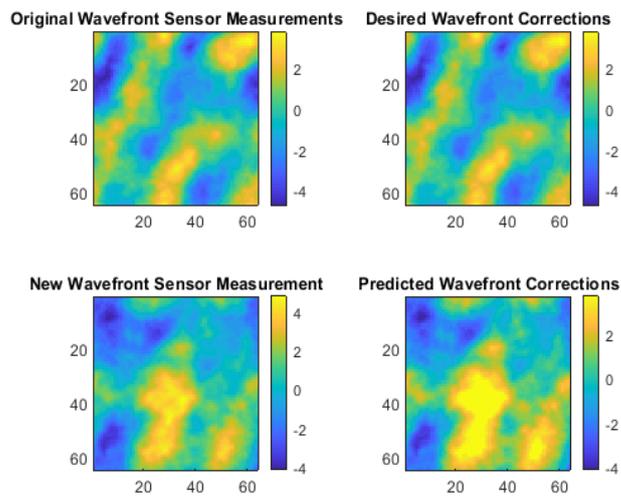
the SR. These metrics serve as quantitative indicators to evaluate the effectiveness of the neural network compensation.



**Figure 6: Von Karman phase screen (wavefront) with variance 0.2, training ratio 90% and neuron's no. 50.**



**Figure 7: Residual error.**



**Figure 8: Von Karman phase screen (wavefront) with variance 0.6, training ratio 90% and neuron's no. 50.**

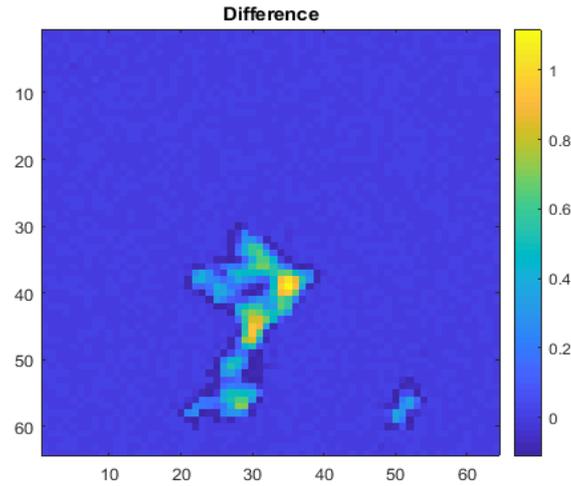


Figure 9: Residual error.

Table 1: Performance metrics MSE, PSNR and SR for the first case.

Training Ratio=90%, Variance=0.2, Neurons=50				
No.	iteration	MSE	PSNR	SR
1	14	0.570	3.808	0.560
2	20	0.628	4.225	0.533
3	14	0.852	2.330	0.426
4	815	0.371	5.477	0.689
5	41	0.934	5.748	0.392
6	11	0.796	2.023	0.450
7	119	0.543	3.496	0.580
8	11	0.459	4.208	0.631
9	1000	0.405	6.805	0.667
10	19	0.577	2.047	0.561

Table 2: Performance metrics MSE, PSNR and SR for the second case.

Training Ratio=60%, Variance=0.2, Neurons=50				
No.	Iteration	MSE	PSNR	SR
1	1000	0.762	5.178	0.466
2	17	0.351	6.703	0.703
3	18	0.832	5.050	0.434
4	194	0.358	7.418	0.698
5	18	0.667	6.328	0.513
6	12	0.650	3.984	0.522
7	39	0.349	7.263	0.704
8	10	0.899	5.420	0.406
9	17	0.646	7.066	0.523
10	17	0.453	8.427	0.635

**Table 3: Performance metrics MSE, PSNR and SR for the third case.**

Training Ratio=30%, Variance=0.2,  
Neurons=50

Iteration	MSE	PSNR	SR
11	1.095	2.352	0.334
1000	0.505	6.306	0.603
14	0.370	4.490	0.690
50	0.494	5.705	0.610
1000	0.345	6.979	0.707
18	0.581	6.902	0.559
12	0.481	7.612	0.618
7	0.469	5.074	0.625
9	0.438	5.691	0.645
12	0.556	4.862	0.573

**Table 4: Performance metrics MSE, PSNR and SR for the fourth case.**

Training Ratio=90%, Variance=0.6,  
Neurons=50

No.	Iteration	MSE	PSNR	SR
1	14	4.642	6.052	0.009
2	1000	11.341	2.785	0.118e-5
3	75	3.839	7.707	0.0215
4	25	2.904	5.830	0.054
5	24	7.260	3.826	7.030e-4
6	9	3.227	6.786	0.0397
7	9	6.706	3.984	0.0012
8	94	4.029	4.196	0.017
9	12	7.376	2.320	6.26e-4
10	254	2.823	8.780	0.0594

**Table 5: Performance metrics MSE, PSNR and SR for the fifth case.**

Training Ratio=60%, Variance=0.6,  
Neurons=50

Iteration	MSE	PSNR	SR
1000	7.483	4.012	5.624e-4
1000	5.590	3.231	0.0037
17	4.397	7.814	0.0123
12	3.098	7.765	0.045
15	6.398	5.996	0.001
8	5.651	5.506	0.003
27	3.71	7.675	0.024
20	5.978	5.275	0.002
42	3.331	5.449	0.0357
1000	6.402	4.115	0.0017

**Table 6: Performance metrics MSE, PSNR and SR for the sixth case.**

Training Ratio=30%, Variance=0.6,  
Neurons=50

No.	Iteration	MSE	PSNR	SR
1	9	4.353	4.727	0.0129
2	14	6.174	4.097	0.002
3	8	2.862	7.076	0.057
4	14	7.58	6.198	5.102e-4
5	27	7.614	3.138	4.931e-4
6	1000	4.062	6.877	0.017
7	10	3.222	7.747	0.039
8	13	6.251	2.867	0.0019
9	11	6.936	4.384	9.715e-4
10	8	6.412	6.175	0.0016

**Table 7: Performance metrics MSE, PSNR and SR for the seventh case.**

Training Ratio=90%, Variance=0.6,  
Neurons=100

Iteration	MSE	PSNR	SR
7	6.79	7.003	0.0011
1000	6.259	2.954	0.0019
10	4.636	6.547	0.0097
1000	3.379	4.774	0.034
11	5.577	3.863	0.0038
11	4.92	5.711	0.0073
19	4.133	5.95	0.16
17	6.217	5.429	0.002
15	4.582	3.331	0.0102
8	3.14	9.041	0.043

**Table 8: Performance metrics MSE, PSNR and SR for the eightieth case.**

Training Ratio=60%, Variance=0.6,  
Neurons=100

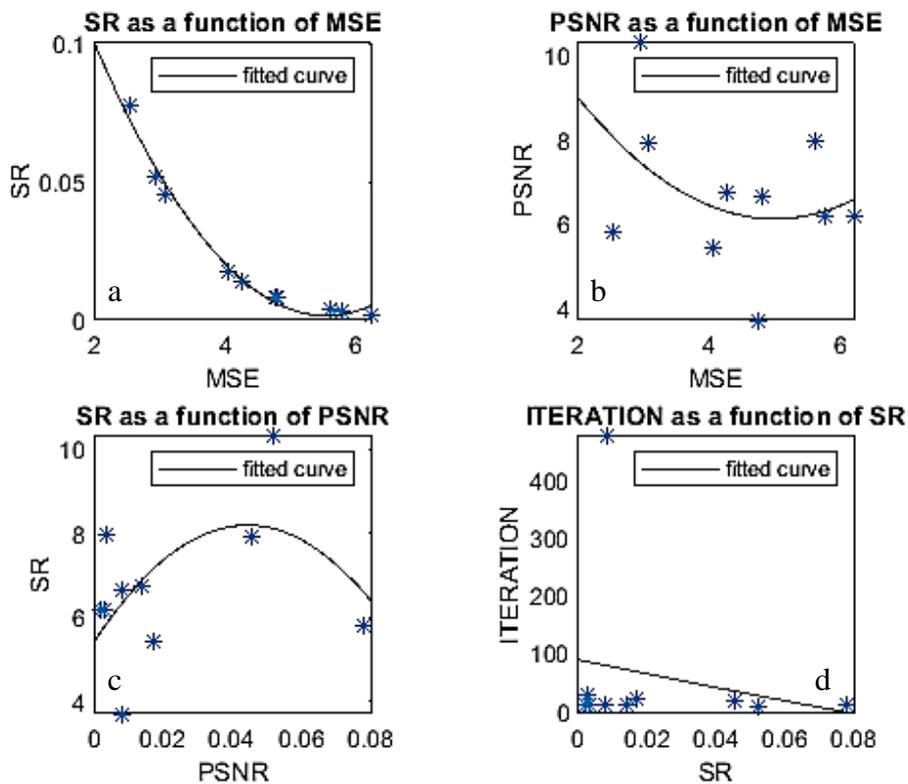
No.	Iteration	MSE	PSNR	SR
1	12	4.27	6.72	0.014
2	12	2.545	5.806	0.078
3	21	4.057	5.435	0.0173
4	477	4.764	3.706	0.0085
5	12	5.624	7.952	0.0036
6	16	3.090	7.903	0.0455
7	9	4.817	6.638	0.0081
8	8	2.957	10.332	0.052
9	12	6.231	6.184	0.002
10	28	5.793	6.167	0.003

**Table 9: Performance metrics MSE, PSNR and SR for the ninth case.**

Training Ratio=30%, Variance=0.6,  
Neurons=100

Iteration	MSE	PSNR	SR
9	3.404	8.353	0.0332
9	4.359	4.485	0.0123
10	4.834	8.568	0.008
8	4.03	5.001	0.0178
12	5.969	3.874	0.0026
10	4.728	3.424	0.0088
12	5.274	6.369	0.0051
7	3.672	6.392	0.0253
115	3.570	4.226	0.0281
16	5.759	5.848	0.0032

In each subplot of Figs. 10-18, representing the nine different cases, a scatter plot of the data points is presented. This scatter plot allows for the visualization of the relationship between the variables being studied. Along with the scatter plot, a curve fit is applied to the data points, which provides a smooth representation of the underlying trend or pattern observed in the data. This curve fit aids in understanding the overall behavior and relationship between the variables under consideration in the context of the specific case being examined.



**Figure 10: Performance metrics for first case.**

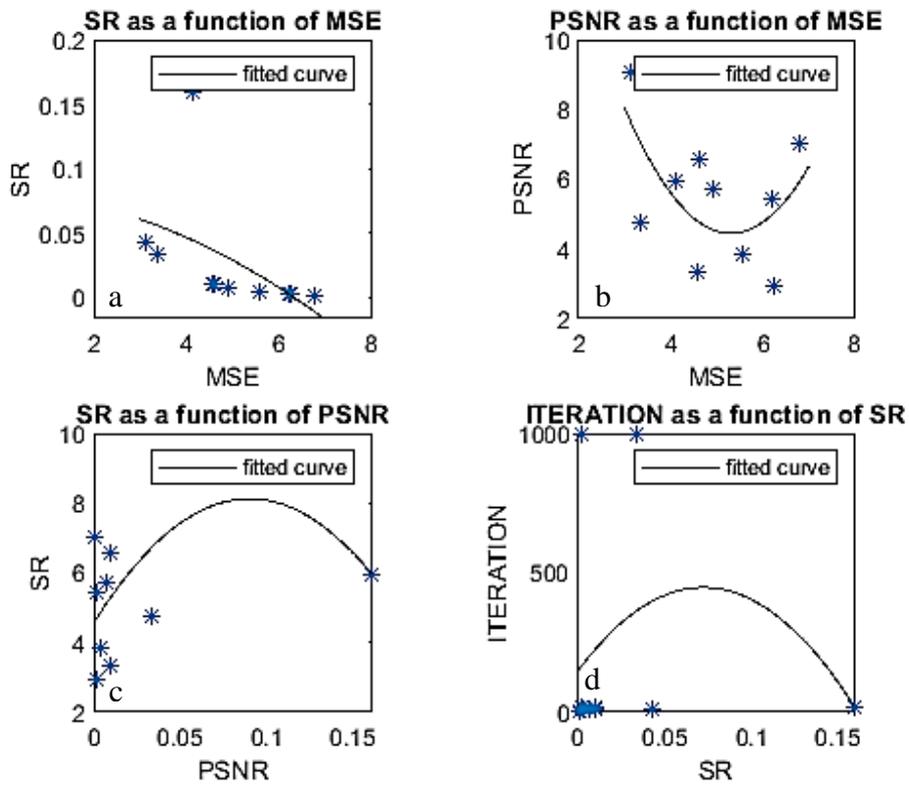


Figure 11: Performance metrics for second case.

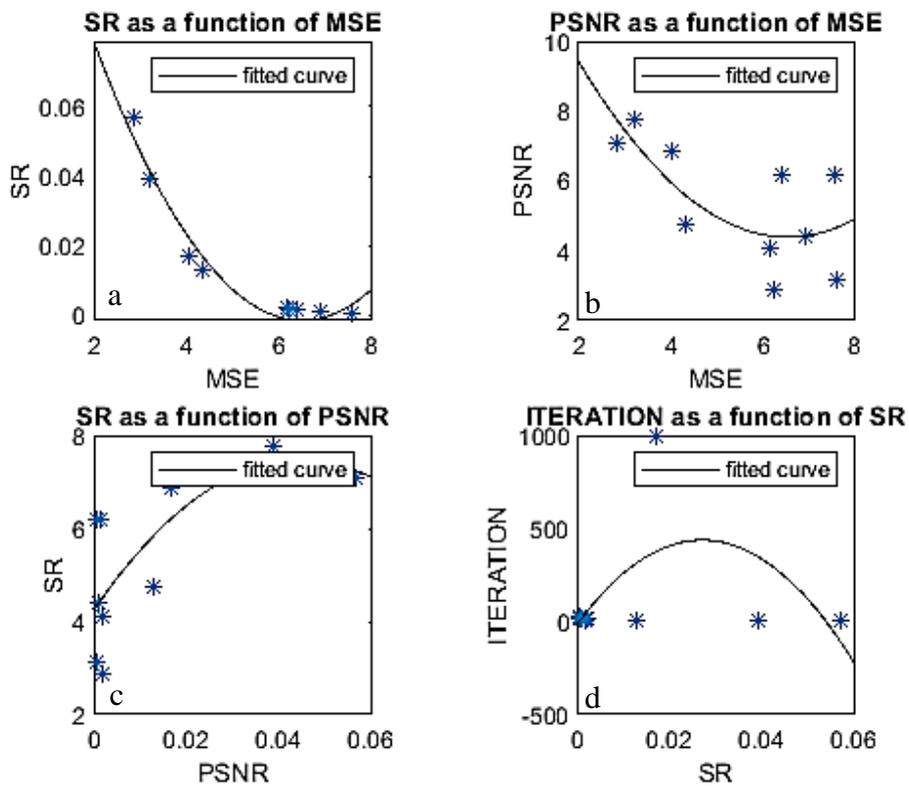


Figure 12: Performance metrics for third case.

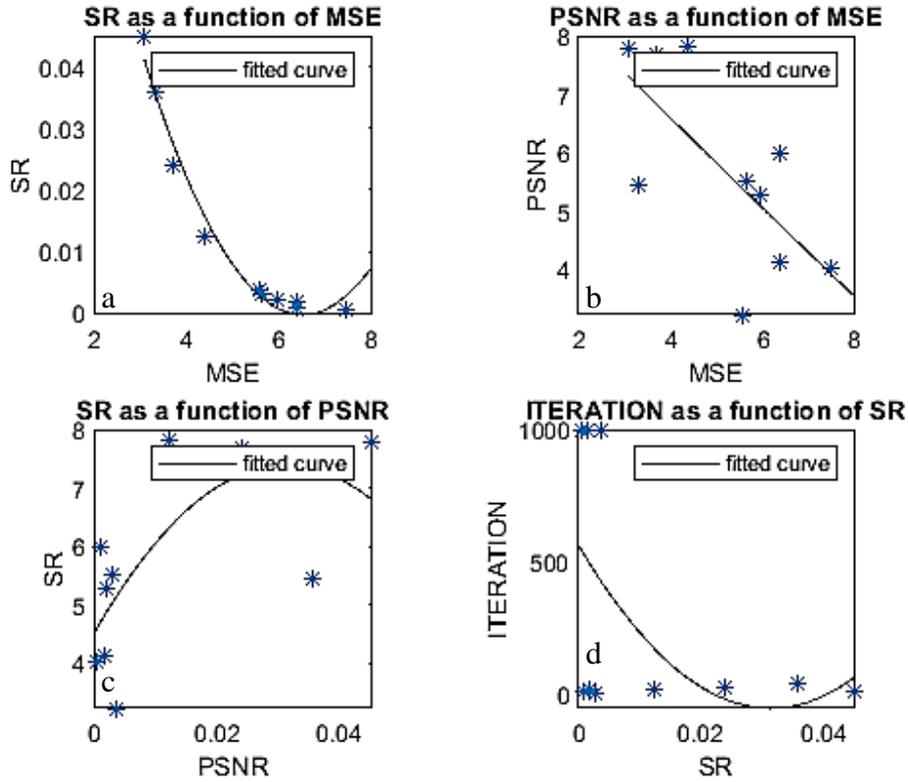


Figure 13: Performance metrics for fourth case.

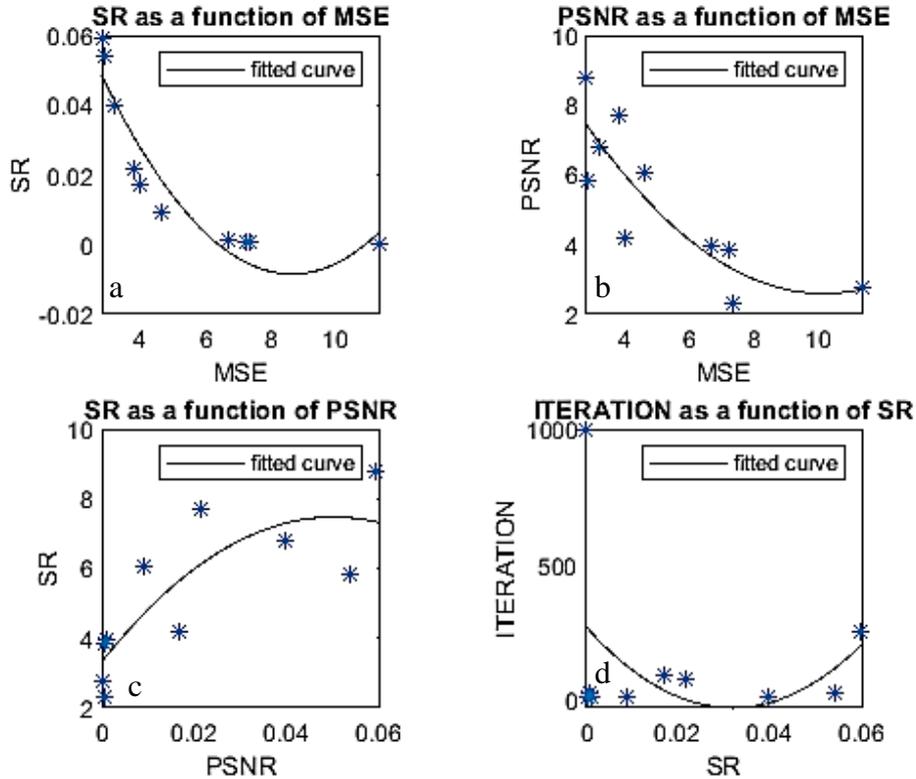


Figure 14: Performance metrics for fifth case.

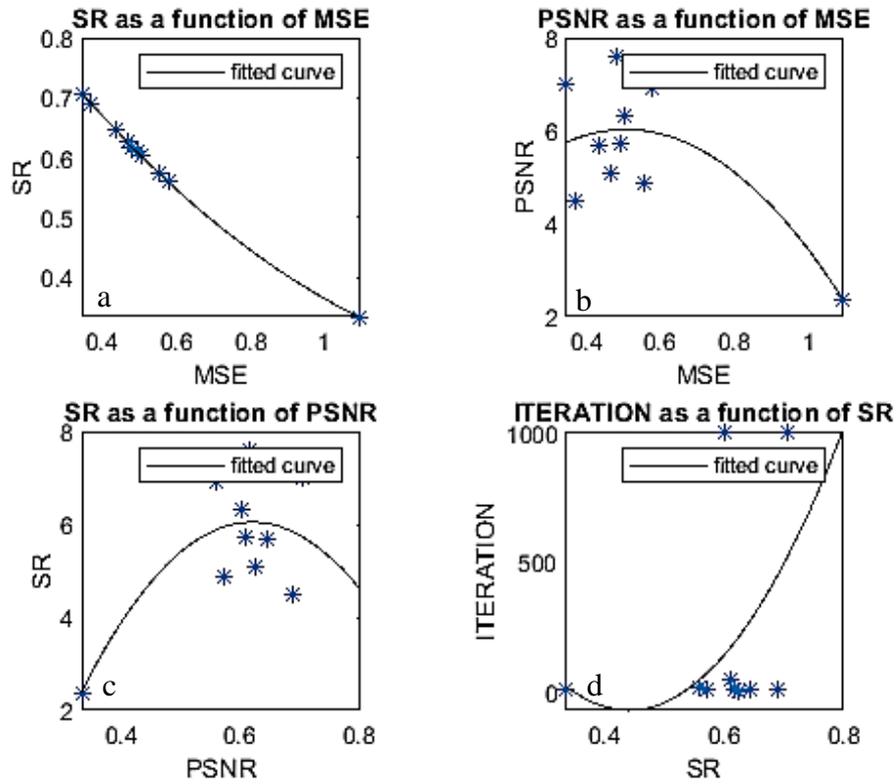


Figure 15: Performance metrics for sixth case.

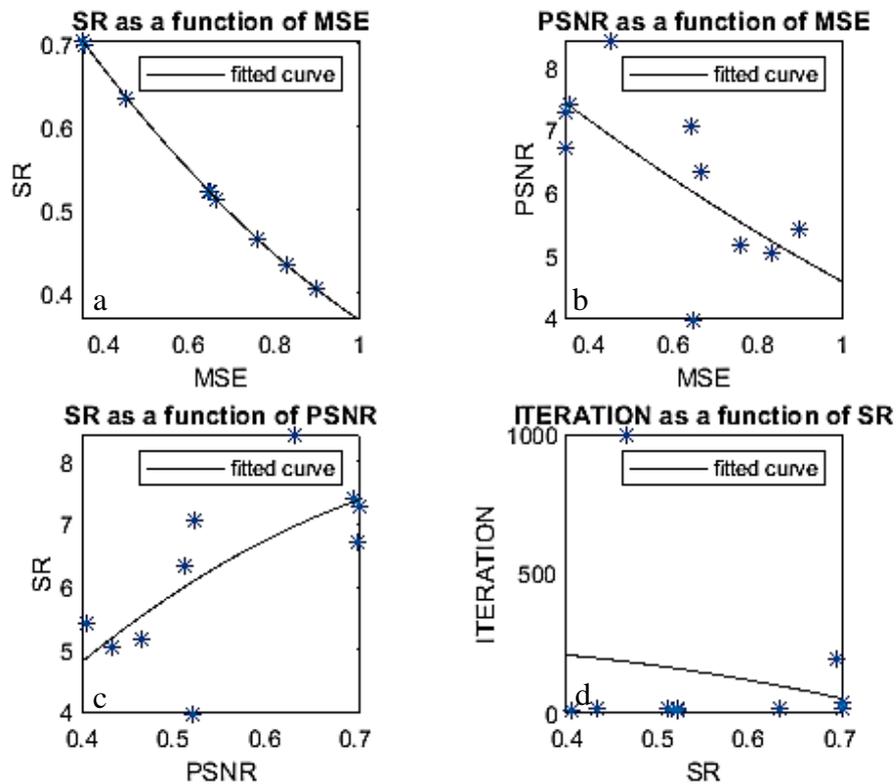


Figure 16: Performance metrics for seventh case.

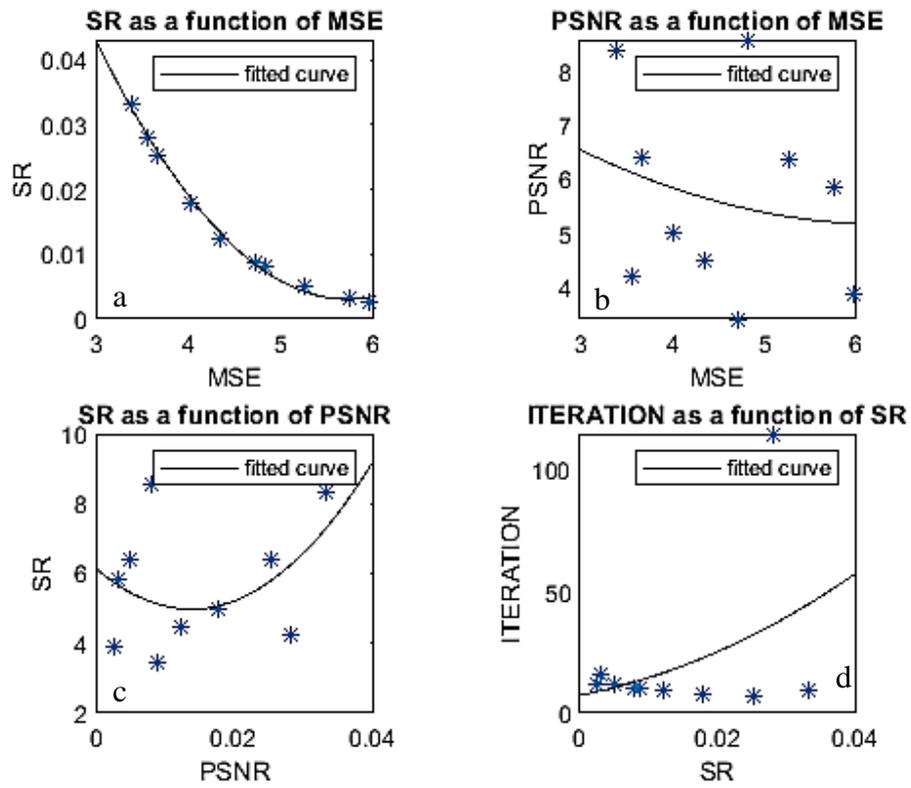


Figure 17: Performance metrics for eightieth case.

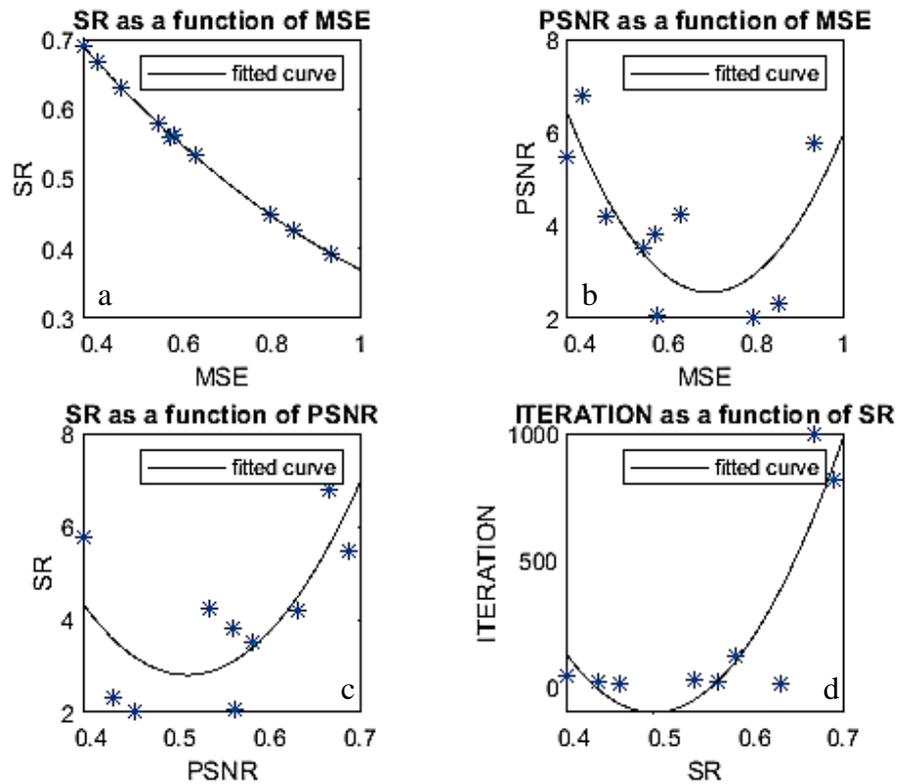


Figure 18: Performance metrics for ninth case.

Using the polynomial curve fits implies that the relationships between the variables under investigation may not adhere strictly to a linear pattern. The subplots (a, b, c, d) in Figs. 10-18 depict the relationships between SR vs MSE, PSNR vs MSE, SR vs PSNR, and iteration vs SR, respectively.

In the first case (Training Ratio=90%, Variance=0.2, Neurons=50), Fig. 10, note that as the MSE decreases, the PSNR and SR generally increase, indicating an inverse relationship between MSE and PSNR & SR. The relationship between iteration and SR displays varying trends, suggesting a complex association between these variables.

In the second case (Training Ratio=60%, Variance=0.2, Neurons=50), Fig. 11, similar trends are observed, with lower MSE values corresponding to higher PSNR and SR. However, the relationship between iteration and SR seems inconsistent.

Moving on to the third case (Training Ratio=30%, Variance=0.2, Neurons=50), Fig. 12, comparable patterns in the relationships between the performance metrics are identified, albeit with some variations in the specific values.

In the fourth case (Training Ratio=90%, Variance=0.6, Neurons=50), Fig. 13, and the fifth case (Training Ratio=60%, Variance=0.6, Neurons=50), Fig. 14, the relationships between the metrics exhibit different characteristics due to the changes in training ratio and variance, leading to distinct trends in the data.

In the sixth case (Training Ratio=30%, Variance=0.6, Neurons=50), Fig. 15, and the seventh case (Training Ratio=90%, Variance=0.6, Neurons=100), Fig. 16, shows further variations in the relationships between the performance metrics, indicating the impact of different parameters on the model's performance.

Finally, in the eighth case (Training Ratio=60%, Variance=0.6, Neurons=100), Fig. 17, and the ninth case (Training Ratio=30%, Variance=0.6, Neurons=100), Fig. 18, diverse trends in the relationships between the metrics are still noted, highlighting the influence of training ratio and neuron count on the model's performance.

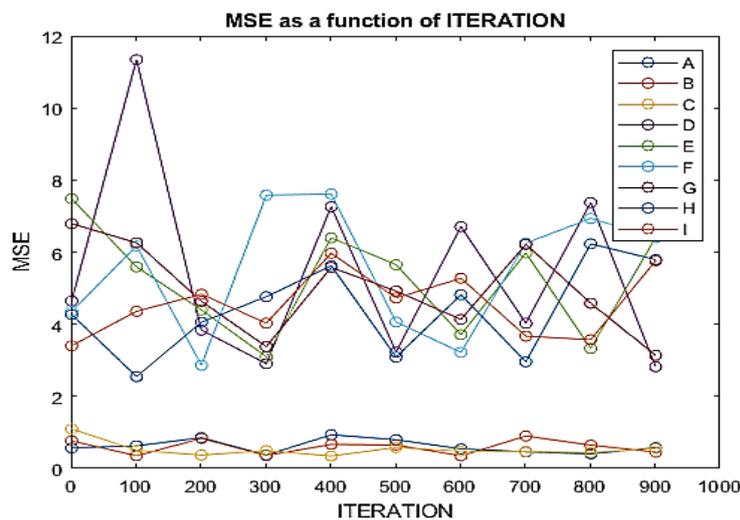


Figure 19: MSE as a function of iteration.

Overall, the analysis of the relationships between SR vs. MSE, PSNR vs. MSE, SNR vs. PSNR, and iteration vs. SR across the different cases reveals the complex and multifaceted nature of these interactions, emphasizing the need for a comprehensive understanding of the underlying factors influencing the performance metrics in the given scenarios.

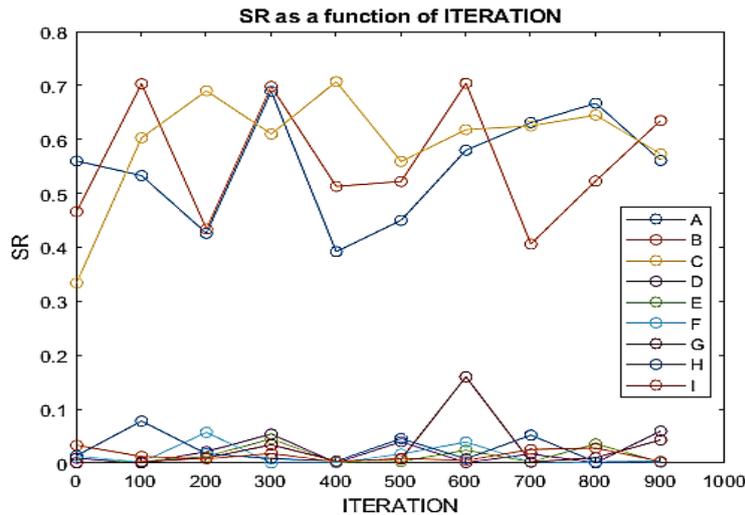


Figure 20: SR as a function of iteration.

To comprehensively assess the effectiveness of NNAO compensation, all cases (the letters A, B, I, in plot’s legend represent the previous cases: 1<sup>st</sup>, 2<sup>nd</sup>, ...9<sup>th</sup>, respectively) can be evaluated collectively by plotting MSE, SR, and PSNR together. By examining these combined plots, one can understand how the MSE, SR, and PSNR values vary and interact with each other under different simulation conditions. This evaluation helps in assessing the overall effectiveness of NNAO compensation for wavefront aberrations.

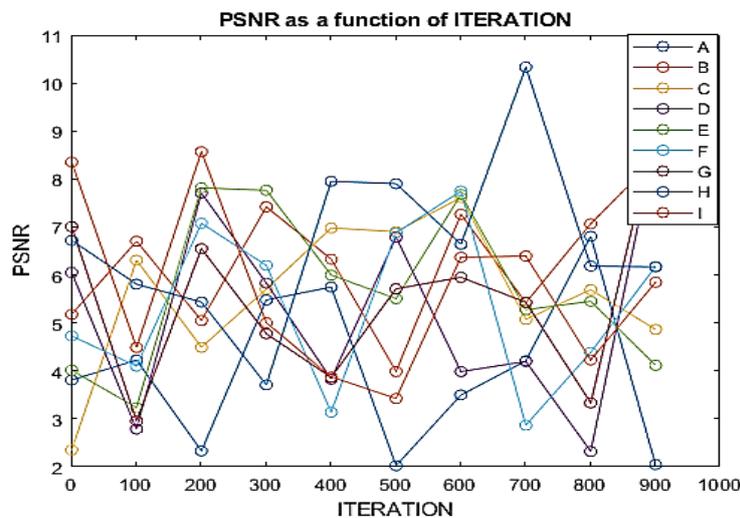


Figure 21: PSNR as a function of iteration.

Through the three above Figs. 19-21, a comparison was made between the performance metrics (MSE, SR, and PSNR) and the highest criterion affected by the related conditions (variance, training ratio and number of neurons). The results showed that the variance of the wavefront aberration was the dominant, as shown in SR, which decreases from an average value of 0.548 for a variance of 0.2 to 0.020 for a variance of 0.6, MSE increases from an average value of 0.613 for a variance of 0.2 to 5.414 for a variance of 0.6, but for PSNR, the effect is not apparent. Also, increasing the number of neurons and the training ratio does not affect the results obtained, but the time consumed is clearly affected.

## 7. Conclusions

In this numerical computer simulation, a neural network model was trained to perform adaptive optics phase compensation based on wavefront sensor measurements. After training the model, a new wavefront sensor measurement was generated, normalized, and used to predict wavefront corrections using the MLP trained neural network. The simulation included additional sections where the performance metrics were calculated for different conditions with varying training ratios, variances, and the number of neurons in the hidden layer. The average performance metrics for each case were computed and plotted against each other to analyze the impact of these parameters on the model's performance. In summary, the variance of the wavefront aberration emerged as the dominant factor, as evidenced by the decrease in SR from an average value of 0.548 for a variance of 0.2 to 0.020 for a variance of 0.6, and the increase in MSE from an average value of 0.613 for a variance of 0.2 to 5.414 for a variance of 0.6. However, the impact on PSNR remains unclear. Furthermore, increasing the number of neurons and the training ratio did not appear to affect the results obtained, but it significantly impacted the time consumed. This suggests that more complex neural network architectures may be necessary to achieve more accurate wavefront corrections.

## Conflict of interest

Authors declare that they have no conflict of interest.

## References

1. S. U. Jasim and R. N. Hassan, *Iraqi J. Phys.* **21**, 9 (2023).
2. M. K. Miridan, R. N. Hassan, and B. Q. Al-Aboodi, *Euro. Schol. J.* **3**, 16 (2022).
3. R. N. Hassan, H. S. Ali, and W. H. Wadee, *Iraqi J. Sci.* **62**, 2463 (2021).
4. M. K. Miridan, R. N. Hassan, and B. Q. Al-Aboodi, *Karbala Int. J. Mod. Sci.* **8**, 306 (2022).
5. L. Abood, S. Al-Hilly, and R. Hassan, *Iraqi J. Sci.* **54**, 222 (2013).
6. K. M. Hampson, R. Turcotte, D. T. Miller, K. Kurokawa, J. R. Males, N. Ji, and M. J. Booth, *Nat. Rev. Meth. Prim.* **1**, 68 (2021).
7. J. R. P. Angel, P. Wizinowich, M. Lloyd-Hart, and D. Sandler, *Nature* **348**, 221 (1990).
8. D. Sandler, T. Barrett, D. Palmer, R. Fugate, and W. Wild, *Nature* **351**, 300 (1991).
9. L. Demailly, E. Gendron, J.-L. Beuzit, F. Lacombe, and N. N. Hubin, *Adaptive Optics in Astronomy (SPIE, 1994)*. p. 867.
10. F. G. Riesgo, S. L. S. Gómez, J. D. S. Rodríguez, C. G. Gutiérrez, Y. M. Hernando, L. M. M. Martínez, A. A. Ramos, M. C. Vera, M. N. Cagigal, and F. J. D. C. Juez, *Opt. Las. Eng.* **158**, 107157 (2022).
11. M. Lloyd-Hart, P. Wizinowich, B. Mcleod, D. Wittman, D. Colucci, R. Dekany, D. Mccarthy, J. Angel, and D. Sandler, *Astrophys. J. Part 2-Lett.* **390**, L41 (1992).
12. S.-J. Chung, MSc. Thesis, Massachusetts Institute of Technology, 2002.
13. M. B. Jorgenson and G. J. Aitken, *Adaptive and Learning Systems (SPIE, 1992)*. p. 113.
14. S. Gigan, O. Katz, H. B. De Aguiar, E. R. Andresen, A. Aubry, J. Bertolotti, E. Bossy, D. Bouchet, J. Brake, and S. Brasselet, *J. Phys. Phot.* **4**, 042501 (2022).
15. U. Wittrock, (Münster, Germany Springer Science & Business Media, 2005). p. 19.
16. R. Landman, S. Y. Haffert, V. M. Radhakrishnan, and C. U. Keller, *Adaptive Optics Systems VII (SPIE, 2020)*. p. 842.
17. A. Wong, B. R. Norris, V. Deo, O. Guyon, P. Tuthill, J. Lozi, S. Vievard, and K. Ahn, *Machine learning for wavefront sensing (SPIE, 2022)*.

18. Y. Guo, L. Zhong, L. Min, J. Wang, Y. Wu, K. Chen, K. Wei, and C. Rao, *Opto-Elect. Advan.* **5**, 200082 (2022).
19. M. Wang, W. Guo, and X. Yuan, *Opt. Expr.* **29**, 3465 (2021).
20. X. Liu, T. Morris, C. Saunter, F. J. De Cos Juez, C. González-Gutiérrez, and L. Bardou, *Month. Not. Roy. Astronom. Soci.* **496**, 456 (2020).
21. C. A. Larry and L. P. Ronald, *Laser Beam Propagation Through Random Media* (Bellingham, Washington, USA, Society of Photo-Optical Instrumentation Engineers (SPIE), 2005).
22. J. H. Churnside and R. J. Lataitis, *JOSA A* **4**, 1264 (1987).
23. I. Toselli, L. C. Andrews, R. L. Phillips, and V. Ferrero, *Atmospheric Propagation IV* (SPIE, 2007). p. 149.
24. S. Basu, J. Osborn, P. He, and A. Demarco, *Month. Not. Roy. Astronom. Soci.* **497**, 2302 (2020).
25. M. Xu, S. Shao, N. Weng, L. Zhou, Q. Liu, and Y. Zhao, *Appl. Sci.* **11**, 10548 (2021).
26. S. Umeyama and T. Yamada, *J. of Biomed. Opt.* **14**, 054038 (2009).
27. A. Y. Shikhovtsev, P. G. Kovadlo, E. A. Kopylov, M. A. Ibrahimov, S. A. Ehgamberdiev, and Y. A. Tillyayev, *Atmosphere* **12**, 1614 (2021).
28. T. A. Bonin, D. C. Goines, A. K. Scott, C. E. Wainwright, J. A. Gibbs, and P. B. Chilson, *Boun. Lay. Meteor.* **155**, 417 (2015).
29. L. F. Otoniel Canuet, MSc. Thesis, Universitat Politècnica de Catalunya, 2015.
30. R. N. Hassan and H. S. Ali, *Karbala Int. J. Mod. Sci.* **9**, 3 (2023).
31. D. G. Ortiz-Suslow, Q. Wang, J. Kalogiros, and R. Yamaguchi, *J. Atmosph. Ocean. Tech.* **37**, 85 (2020).
32. G. G. Katul, M. B. Parlange, and C. R. Chu, *Phys. Flui.* **6**, 2480 (1994).
33. M. Chen, T. Gao, S. Hu, Q. Zeng, L. Liu, and G. Li, *Res. Phys.* **7**, 3596 (2017).
34. D. A. Paulson, PhD. Thesis, University of Maryland, College Park, 2020.
35. M. R. Chatterjee and F. H. Mohamed, *Laser Communication and Propagation through the Atmosphere and Oceans III* (SPIE, 2014). p. 13.
36. F. Wang, W. Du, Q. Yuan, D. Liu, and S. Feng, *Atmosphere* **12**, 1608 (2021).
37. Z. Malikov, *Appl. Mathem. Modell.* **82**, 409 (2020).
38. K. Barkalov, I. Lebedev, M. Usova, D. Romanova, D. Ryazanov, and S. Strijhak, *Mathematics* **10**, 2708 (2022).
39. V. S. R. Gudimetla, R. Holmes, T. Farrell, and J. Lucas, *SPIE Def. Secur. Sens.* **8038**, 803808 (2011).
40. D. Vasylyev, W. Vogel, and F. Moll, *Physical Review A* **99**, 053830 (2019).
41. B. J. Rollick, PhD. Thesis, University of Tennessee, Knoxville, 2022.
42. M. Cubillos and E. Jimenez, *J. Opt. Soci. Amer. A* **39**, 1403 (2022).
43. R. Jiang, K. Wang, X. Tang, and X. Wang, *Photonics* **10**, 832 (2023).
44. E. Klotz, S. Lefebvre, N. Védrenne, C. Musso, T. Fusco, S. Poulénard, L. Coret, and A. Louis, *International Conference on Space Optics* (Dubrovnik, Croatia SPIE, 2023). p. 1277758.
45. H. S. Ali, *Iraqi J. Sci.* **64**, 4579 (2023).
46. J. G. Chen, V. Shah, and L. Liu, *Opt. Lett.* **46**, 2513 (2021).
47. A. P. Wong, B. R. Norris, P. G. Tuthill, R. Scalzo, J. Lozi, S. Vievard, and O. Guyon, *J. Astron. Teles. Instrum. Sys.* **7**, 019001 (2021).
48. S. Almabdy and L. Elrefaei, *Appl. Sci.* **9**, 4397 (2019).
49. G. Aksu, C. O. Güzeller, and M. T. Eser, *Int. J. Assess. Too. Educ.* **6**, 170 (2019).
50. R. D. Haameid, B. Q. Al-Abudi, and R. N. Hassan, *Iraqi J. Sci.* **62**, 5008 (2021).
51. R. Swanson, M. Lamb, C. Correia, S. Sivanandam, and K. Kutulakos, *Adaptive Optics Systems VI* (SPIE, 2018). p. 481.

52. A. Paleyes, R.-G. Urma, and N. D. Lawrence, ACM Comp. Surv. **55**, 1 (2022).

## تعويض طور تشوه الاضطرابات الجوية باستخدام البصريات التكيفية للشبكة العصبية

راند نوفي حسان<sup>1</sup>

قسم الفلك والفضاء، كلية العلوم، جامعة بغداد، بغداد، العراق

### الخلاصة

تُحدث البصريات التكيفية ثورة في وضوحية التلسكوب ولكنها تواجه عقبات تتعلق بالتكلفة والتعقيد والمعايرة. تقدم البصريات التكيفية للشبكة العصبية (NNAO) وعدًا باستخدام الشبكات العصبية لتصميم تصحيحات للتلسكوبات والظروف الجوية، وتجاوز المعايير وأجهزة التحسس. تبحث هذه الدراسة المبنية على MATLAB في تأثير NNAO على جودة الصورة الفلكية، وتكشف عنها كحل فعال من حيث التكلفة يعزز البصريات التكيفية في علم الفلك. وكانت نتائج المحاكاة العددية مشجعة حيث تجاوز معدل التعويض 50% لظروف الرصد الملائمة. تشير النتائج إلى أن العامل المهيمن الذي يؤثر على جودة الصورة هو تباين انحرافات جبهة الموجة. تنخفض نسبة سترهل (SR) من متوسط 0.548 لتباين 0.2 إلى 0.020 لتباين 0.6، بينما يزيد متوسط الخطأ التربيعي (MSE) من متوسط 0.613 إلى 5.414. ومع ذلك فإن التأثير على ذروة نسبة الإشارة إلى الضوضاء (PSNR) غير حاسم علاوة على ذلك وجدنا أن زيادة عدد الخلايا العصبية ونسبة التدريب لا يؤثر بشكل كبير على النتائج التي تم الحصول عليها ولكنه يؤثر بشكل ملحوظ على الوقت الحسابي المطلوب.

**الكلمات المفتاحية:** البصريات التكيفية، الاضطراب الجوي، الشبكة العصبية، الشبكة العصبية متعددة الطبقات (MLP)، نسبة سترهل.