Double-Star Detection Using Convolutional Neural Network in Atmospheric Turbulence

Jafar Bakhtiar Shohani[a], Morteza Hajimahmoodzadeh[b,*], and Hamidreza Fallah[a]

[a]Department of Physics, University of Isfahan, Isfahan, Iran
[b]Quantum Optics Group, Department of Physics, University of Isfahan, Isfahan, Iran

[*]Corresponding author email: M.Hajimahmoodzadeh@sci.ui.ac.ir

Regular paper: Received: Sep. 25, 2022, Revised: Nov. 21, 2022, Accepted: Nov. 22, 2022,
Available Online: Nov. 24, 2022, DOI: 10.52547/ijop.16.2.121

ABSTRACT— In this paper, we investigate the usage of machine learning in the detection and recognition of double stars. To do this, numerous images including one star and double stars are simulated. Then, 100 terms of Zernike expansion with random coefficients are considered as aberrations to impose on the aforementioned images. Also, a telescope with a specific aperture is simulated. In this work, two kinds of intensity are used, one is in-focus and the other is out-of-focus of the telescope. After these simulations, a convolutional neural network (CNN) is configured and designed and its input is simulated intensity patterns. After learning the network, we could recognize double stars at severe turbulence without needing phase correction with a very high accuracy level of more than 98%.

KEYWORDS: Aberration**,** Turbulence, Double Stars, Convolutional Neural Network, Machine Learning.

I. INTRODUCTION

In astronomy, a pair of stars that are close enough together from the point of view on Earth is called a double-star. The recognition of these types of star systems is important, because studying their properties including mass, motions, and other parameters could give us some useful information for further studies. Indeed, double stars are a set of two stars that are located at a very small distance from each other. However, this recognition encounters some challenges. Due to the turbulence, the earth's atmosphere causes a degradation in the quality of images taken from the earth. Various models such as Kolmogorov, Tatarskii, and Von-Karman have been introduced to express the atmospheric turbulence[1]. Atmospheric turbulence is caused by random temperature changes. As a result, it causes random variations of the refractive index of the air. These variations happen both spatially and temporally. Therefore, when optical waves are propagating through the atmosphere, they encounter these variations. Hence, some mathematical models were introduced to investigate the changing refractive index of air. The main equation of light propagation in turbulent environments also follows the wave equation as [1]:

$$[\nabla^2 + k^2 n^2(r)]U(r) = 0, \quad (1)$$

where r represents the radial coordinate, k is wavenumber, n is the refractive index, and U is the electric or magnetic field of light. The turbulence impacts the wavefront of light, therefore it can be said that the wavefront loses its ideal form and involves some aberrations. One of the ways to correct the wavefront and, as a result, get better images is adaptive optics. Primary setups of adaptive optics utilized wavefront sensors. The most famous wavefront sensor is Shack-Hartman. Sensor-based wavefront sensing has some limitations [2], such that in recent years, adaptive optics techniques without using wavefront sensors have been introduced[3]-[8]. In this paper, we have introduced a method based on machine learning to be able to directly detect double

stars without the need to identify the distorted wavefront and methods of adaptive optics.

In the last decade, usage of Machine Learning (ML) as a branch of Artificial Intelligence (AI) has been greatly expanded in various fields. The basic calculations in this area is related to connections of artificial neurons. A single neuron with some input and an output is shown in Fig. 1. The connection of some neurons in some layers is done via some weights and establishes an Artificial Neural Network (ANN) which has an input layer, an output layer, and one or more hidden layers.

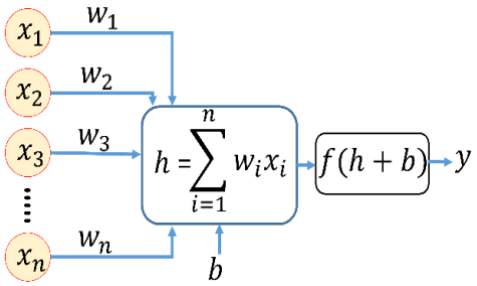

Fig. 1. Input and output of a single artificial neuron.

As can be seen in Fig. 1, input (x_i) and weights (w_i) are multiplied element-wise and the result is passed to a function (f) named "activation function". Therefore, a single neuron has an output (y) and the relationship between input and output is achieved by:

$$y = f(\sum_{i=1}^{n} w_i x_i + b). \quad (2)$$

This output can be considered as one input to the next layer neurons. Fig. 2, shows some neurons connected to each other in layers.

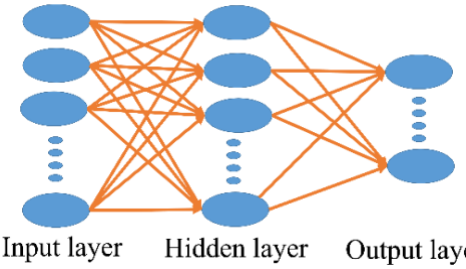

Fig. 2. Input, hidden, and output layer of a typical artificial neural network with some neurons in each layer.

In Fig. 2, a typical artificial neural network with some neurons in each layer is shown. The first layer is usually called the input layer which gets the input data. The final layer is called the output layer which usually has some definite number of neurons depending on the problem. Every layer between the input and output layers is called the hidden layer. This type of neural network is called a Multi-Layer Perceptron (MLP). Peterson *et al.* explained in details the basics and concepts of ANNs [9].

It should be noted that MLP is not suitable for deep layers with a large number of neurons in layers. When the input data to the network are two-dimensional arrays like images, overfitting is a phenomenon that can happen. Overfitting is one of the modeling errors in data science. This error occurs when the model has retained the features of the training data instead of learning it, i.e., it has been overtrained on it; As a result, the model will have the generalization problem and is only useful for the training dataset and not on test data (after learning) that has not yet seen. To avoid this, deep learning models have been introduced which can have deeper layers of neurons. One of the most well-known deep learning models is Convolutional Neural Network (CNN) which has a general configuration as presented in Fig. 3.

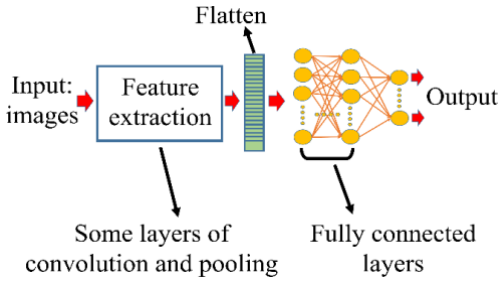

Fig. 3. The typical conceptual scheme of a CNN.

There could be different layers in this configuration, but the two main layers are convolution and pooling. The purpose of these two layers is to extract features and reduce the size of input images. Two-dimensional convolution is defined as:

$$(I * f)(m,n) = \sum_{i,j} I(m-i, n-j) f(i,j), \quad (3)$$

Where I and f represent the image and filter. In the convolution operation, filters slide on image, and the element-wise multiplication is done and the result is an image that is called a feature map.

Another important operation in feature extraction is called pooling which is related to dimension reduction. The dimension of an image after this operation is determined by:

$$\text{size} = \left(\left\lfloor \frac{I_x - p_x}{s_x} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{I_y - p_y}{s_y} \right\rfloor + 1\right), \qquad (4)$$

in which $\lfloor\ \rfloor$ represents a floor function, I_x, I_y are input shape, p_x , p_y are dimension of pooling kernel and s_x, s_y are strides, all in x and y directions respectively. One of the well-known operations in pooling is named "max pooling" in which the maximum value of the array in each block of image is extracted. In Fig. 4, the functionality of max pooling is shown.

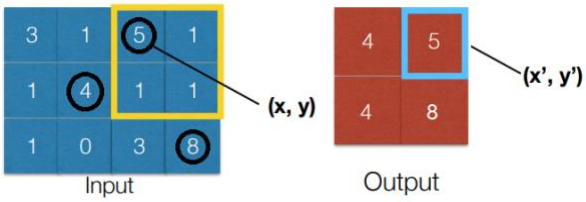

Fig. 4. A typical example of max pooling Operation, with $s_x = 2$, $s_y = 1$.

After feature extraction steps, all of the data are flattened as input to the fully connected layers (are called dense layers as well) which are regular MLP layers. Finally, the last step is considering some neurons as the output layer.

LeCun introduced the CNN and explained its properties including convolution and max pooling operations [10].

Nevertheless, there are many methods to use machine learning in different areas. Each of them may have different uses for different types of data. However, when we are dealing with images, we should use an algorithm that has the potential of feature extraction. The best option for this aim is CNN which is explained above.

II. METHOD AND SIMULATION

In this paper, we have investigated a method to distinguish double stars from single stars using deep learning. In the different conditions of atmospheric turbulence, it is not straightforward to accurately accomplish this kind of recognition in the imaging system without using adaptive optics techniques. But we have introduced a deep learning model that can do that. This method is straightforward and intelligent. The diagnosis can be done with high accuracy without any need to phase correction setups. It needs a big set of images with proper labels. Indeed, each image has a label (or class) which is either "1" or "2". Label "1" indicates that there is one star in the distorted image. Likewise, the label "2" indicates that there are two stars in the distorted image. The inputs are simulated images of stars in turbulence conditions. Therefore, the first step is simulating the ground-truth images (aberration-free) of single and double stars that can be seen in Fig. 5.

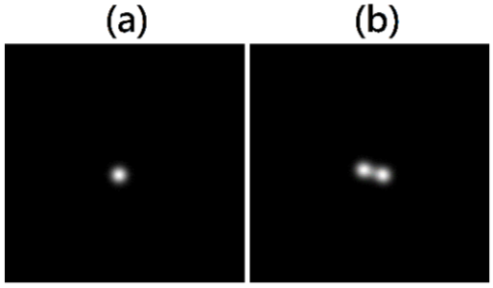

Fig. 5. Simulation of aberration-free images of (a): single star, (b) double-star.

To simulate the double-star systems, their distance from each other must be defined. In this paper, the separation of their maximum pixel intensity is considered as a parameter. This parameter is chosen as a random variable by a condition that their maximum intensity locations should be greater than 3 pixels and fewer than 12 pixels, i.e. $3 < d < 12$. Moreover, it should be noted that the orientation of their position relative to each other is chosen as a random variable too.

Propagation of light through the atmosphere degrades these images. The figure below shows an example relating to the effect of turbulence on the star images in Fig. 6.

Figure 6 (a), is related to a single star, and Fig. 6 (b) represents a double-star in the presence of aberration. The severe turbulence of the atmosphere has made us unable to recognize it in normal conditions. To simulate these intensity patterns, the parameters in Table 1 have been used.

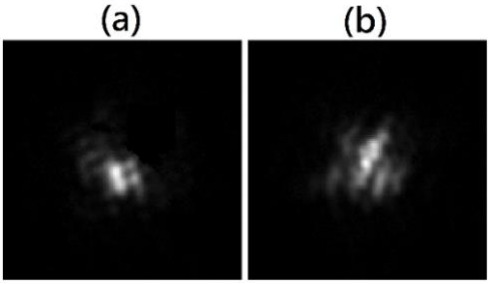

Fig. 6. Simulation of aberrated images of (a): single star, (b): double star, in atmospheric turbulence.

Table 1. Basic parameters in simulation.

Parameter	Dimension	value
Wavelength (λ)	$[\mu m]$	2.2
Telescope diameter (D)	$[m]$	10
Grid size (N)	$[pixel]$	128
Pixel scale	$[arcsecond]$	0.04

Therefore, the first step is simulating a large number of turbulent wavefronts for which we use the Zernike expansion.

$$\varphi(r,\theta) = \sum_i a_i Z_i(r,\theta), \quad (5)$$

in which a_i and $Z_i(r,\theta)$ are the coefficients and polynomials of Zernike expansion, and $\varphi(r,\theta)$ is the turbulent wavefront. These wavefronts are simulated based on Fourier transform [11], and the modified Von-Karman turbulence model **Error! Reference source not found.**.

In many scientific researches, the reconstruction of wavefronts is usually done in terms of Zernike terms. But, since considering Zernike terms in high orders does not cause many changes in the wavefront, they might be disregarded. A criterion that can be considered is the wavefront error. In this work, based on experience and several times of trial and error, we observed that considering terms higher than n=13 (104 terms) does not have a significant effect on the wavefront. However, we added and highlighted explanations in this regard in the text. As an example, the simulation of a distorted wavefront at the focal plane of the telescope is shown in Fig. 7.

But for the learning process to be done well, we need to have two categories of images. For this purpose, a value of the defocus term in Zernike expansion is added to the previous defocus term for each wavefront. This term is called defocus and is defined as:

$$Z_4 = \sqrt{3}(2r^2 - 1). \quad (6)$$

In another word, this is equivalent to shifting the image plane relative to the focal point. In this paper, the amount of defocus is considered as $\frac{\lambda}{8} * Z_4$, where λ is the wavelength. It is better to have more than one set of images as input, because it increases the capability of learning process in this problem. Also, due to the diversity of turbulent wavefronts, the uniqueness of phases in focal plane is not guaranteed. Therefore, another set of images in a defocused location is helpful.

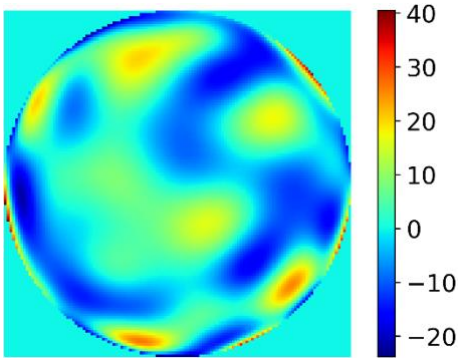

Fig. 7. An example of distorted wavefront by turbulence.

This is shown schematically in Fig. 8, where there are two kinds of image acquisition in the simulation.

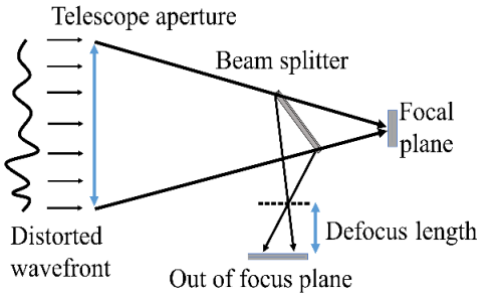

Fig. 8. Getting in-focus and out of focus images.

As can be seen in Fig. 8, the incoming distorted wavefront to the telescope enters the beam splitter and two images could be achieved, one in focal plane and one in out of focal plane.

As an example, the out of focus images corresponding to the images in Fig. 6, and the

corresponding wavefront in Fig. 7, are shown in Fig. 9.

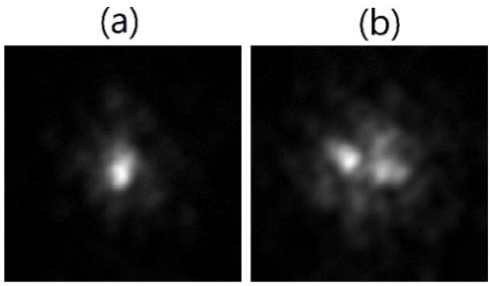

Fig. 9. (a): in-focus, and (b): out of focus, images affected by turbulence.

This procedure is done repetitively and thousands of images are generated. Therefore, the input includes the distorted intensity patterns (images) and the number of stars (labels) corresponding to the intensity patterns. Fig. 10, shows the sequential procedure of the designed CNN.

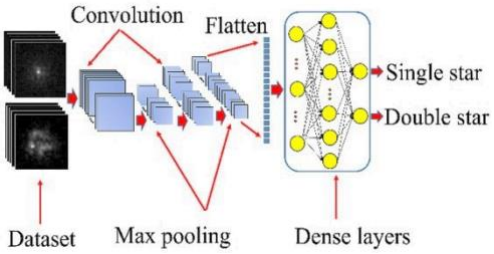

Fig. 10. Design of the CNN for this paper, with two kinds of in-focus and out of focus images as input.

In this configuration, the dataset which are generated based on the method mentioned above, enters the first layer of the network. This dataset includes both the in focus and out of focus intensities. After the feature extraction operations all of the data become flattened and enter the dense layers. This is called a feed forward propagation in which the trainable parameters get some values for the first time. Based on these weights, the optimization algorithm starts to assess the results and some values as output are achieved. This process is called an "epoch". This procedure is repeated and the weights are updated during the next epochs. To this aim, a cost function and an optimizer are needed. The optimization process is looking for parameters that can minimize the cost function. CNN is a branch of supervised learning. Thus, both the images and labels, are inputs to the network. In this paper, Adaptive Moment estimation (ADAM) is used as the optimizer. This optimization algorithm is a modification to the stochastic gradient descent algorithm that has recently been utilized for deep learning models in different areas including computer vision. The updating the weights, based on this algorithm is as follows:

$$w_{t+1} = w_t - \widehat{m_t}\left(\frac{\alpha}{\sqrt{\widehat{v_t}}} + \varepsilon\right), \tag{7}$$

where w_t and w_{t+1} are the weights at times t and $t+1$, respectively. Also,

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}, \tag{8}$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t}, \tag{9}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right], \tag{10}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\left[\frac{\delta L}{\delta w_t}\right]^2. \tag{11}$$

In these equations, α is the learning rate, β_1, β_2 are decay rates of gradients [12].

It is efficient, especially when we are dealing with large datasets and many parameters. In Adam rather than adapting the learning rate in which the average first moment is used, the average of the second moments of the gradients is considered. Also, the exponential moving average of the gradients and square gradients are calculated. Also, the Sparse Categorical Cross-Entropy is used as the cost function which is defined as [14][14]:

$$C = -\sum_{j=1}^{N} y_i \, log(\widehat{y_i}), \tag{12}$$

in which y_i and $\widehat{y_i}$ are the actual and predicted labels, and N is the number of labels. The configuration of designed CNN is presented in Table 2.

As shown in Table 2, size of the images is 128×128 which is selected based on some trial and error to get appropriate images of star systems in the presence of aberration. There are two convolution layers and two max pooling layers. At the first convolution layer, 64 filters

with the size 11×11 are considered which can extract large-scale features. Also, 64 filters are selected to extract as much as convenient different features. The result of this operation is passed to the max pooling layer where the dimension of images can reduce effectively.

Table 2. Layers of the designed CNN.

Layer	Type	Input Size	Filter Size	Kernels
1	Input	128×128×2	-	-
2	Convolution	128×128×2	11×11	64
3	Max pooling	59×59×64	3×3	-
5	Convolution	29×29×64	5×5	192
6	Max pooling	29×29×192	3×3	-
12	Dense	7×7×128	-	-
14	Dense	6272×1	-	-
16	Output	N×1	-	-

The second convolution layer has 192 filters with the sizes of 5×5 which can extract small-scale features. These two operations are repeated and the resulting images enter two dense layers with the determined number of neurons. In the final layer, two neurons must be defined, as there are two labels at most in the images. Indeed, the images belong to either a single or a double-star. Hence, the final layer necessarily has two neurons.

By doing so, the input dataset is built which includes 6000 in-focus and out of focus images.

III. Results and Discussion

The innovation of this work is that there is no need to identify the wavefronts and remove the aberrations by sensor-less adaptive optics, and this diagnosis can be made directly and only by using the intensity images in the focal and non-focal planes. In doing so, the main focus of this paper is configuring a CNN in such a way that after the learning procedure, it could have prediction capability. It means that if a pair of intensity patterns that are affected by turbulence is given to the network, in the shortest possible time it can recognize that this pair of images belong to a single or double-star. Fig. 11, shows the input and output, before and after the learning procedure.

It should be noted that 80% of the images are considered training data and 20% as test data.

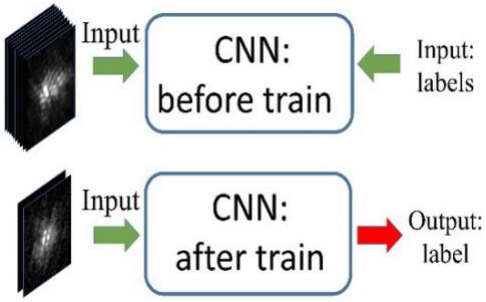

Fig. 11. Comparison of input and output for a deep learning model, before and after the learning process.

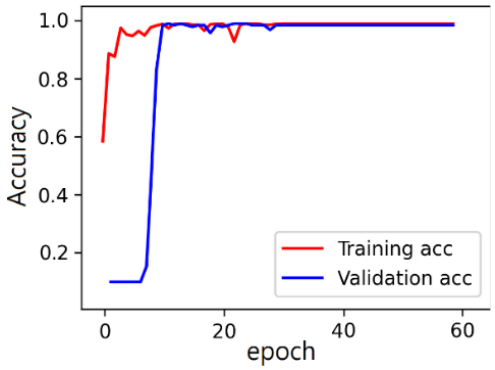

Fig. 12. Accuracy of the CNN in 60 epochs for training and validation data.

Furthermore, 20% of the training data is allocated to validation data. After the learning process which may take some hours, the recognition time was only a few milliseconds for any pair of images. Therefore, it can be said that this method is one of the best possible options to achieve real-time recognition. Learning is done in some iterations, named "epoch". The optimization operation in every epoch is done based on the train data and the cost function. Training accuracy is related to the training data that the network is experiencing, and validation accuracy is the evaluation of the network performance on the validation data during the learning process. These two accuracies are shown in Fig. 12.

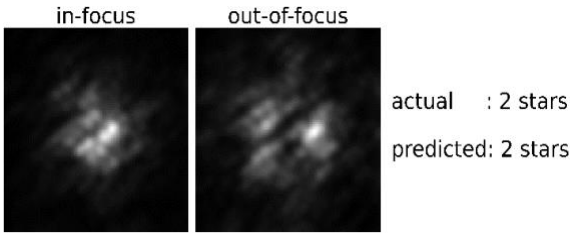

Fig. 13. An example of the CNN evaluation by a test data.

After learning, the evaluation of the network is done on the test data in the dataset. It should be

emphasized that the test data do not participate in learning. The accuracy of this network after the end of the learning process has been 98%, approximately, which shows the power of the designed CNN. Fig. 13, shows an example of the test samples which includes an in-focus and an out of focus images. These two images belong to a double-star which the network could detect it correctly.

As can be seen in Fig. 13, it is not possible to recognize the double-star without using the phase correction methods. But these images are given to the network, and it predict that they belong to a double-star correctly. However, to assess the accuracy of the network after learning, all of the test data should apply to the network. In doing so, a confusion matrix can be made. A confusion matrix is usually used in classification problems. A typical confusion matrix is defined as:

$$C = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}, \quad (13)$$

where *TP*, *FN*, *FP*, and *TN* stand for true positive, false negative, false positive, and true negative, respectively. For two classes (class "1" and class "2"), these parameters are defined simply as follows:

- TP shows the number of samples that belong to class "1" (Positive) and they are correctly predicted by the model to belong to class "1" (True).
- FP shows the number of samples that actually belong to class "1" (Positive), but the model predicted them not to be in class "1" (False).
- TN shows the number of samples that are not in class "1" (Negative) and the model correctly predicted them to be so (True).
- FN shows the number of samples that the model predicted to be in class "1", but did not actually belong to it (False Negative).

These definitions can be generalized for more complex systems, e.g. 3, 4, 5, and 6 stars.

In this work, the resulted normalized confusion matrix is shown in Fig. 14. The main diagonal elements of confusion matrix are related to the true predictions. If these elements approaches one, it can be said that the classification works well which can be seen in Fig. 14.

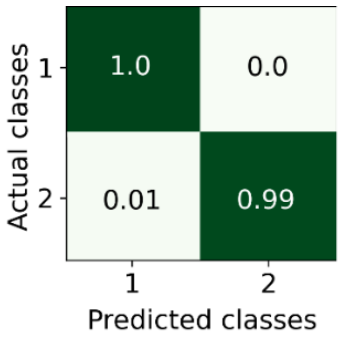

Fig. 14. The resulted confusion matrix.

IV. CONCLUSION

In this paper, we investigated the use of a deep learning network to detect and classify the single and double stars in turbulent atmosphere. Production of single and double stars, simulation of turbulent wavefront, and simulation of intensity pattern in the focal and out of focus planes were done. All of the steps are done in Python programming language. The algorithm of production the images of single and double stars is worked in a "for loop" in python programming language. Therefore, at the same time for every iteration, a pair of focused and defocused images is generated and saved in hard drive in computer. Therefore, a dataset is built which contains various intensity pattern with random kinds of aberrations. The position of stars are based on the random numbers with uniform distribution. After building the dataset, it is enters the CNN algorithm. The CNN can learn to distinguish the star systems based on the experience it achieve by the dataset and learning procedure.

A convolutional neural network was designed. In this network, layers of convolution and pooling are responsible to extract the useful features in the images. 6000 images were generated as the input to the network, and 6000 labels corresponding to the images were defined. The learning process was done well and the accuracy of network approaches to 1. All of the simulations steps in this paper were

done in python programming language. To test the network, we used the test data. Moreover, confusion matrix method was utilized to interpret the true and false predictions for evaluating the network. Finally, the trained network had more than 98% accuracy.

REFERENCES

[1] L.C. Andrews and R.L. Phillips, *Laser beam propagation through random media*, Bellingham, 2nd Ed., pp. 82-91, 2005.

[2] D. Gratadour, E. Gendron, and G. Rousset, "Intrinsic limitations of Shack–Hartmann wavefront sensing on an extended laser guide source," J. Opt. Soc. Amer. A, Vol. 11, pp. 171-181, 2010.

[3] J. Antonello, X. Hao, E.S. Allgeyer, J. Bewersdorf, J. Rittscher, and M.J. Booth, "Sensorless adaptive optics for isoSTED nanoscopy," Adaptive Opt. Wavefront Control Biological Systems IV, Vol. 10502, pp. 11-16, 2018.

[4] D.J. Wahl, R. Ng, M.J. Ju, Y. Jian, and M.V. Sarunic, "Sensorless adaptive optics multimodal en-face small animal retinal imaging," Biomed. Opt. Express, Vol. 10, pp. 252-267, 2019.

[5] A. Camino, P. Zang, A. Athwal, S. Ni, Y. Jia, D. Huang, and Y. Jian, "Sensorless adaptive-optics optical coherence tomographic angiography," Biomed. Opt. Express, Vol. 11, pp. 3952-3967, 2020.

[6] X. Wei, T.T. Hormel, S. Pi, B. Wang, and Y. Jia, "Wide-field sensorless adaptive optics swept-source optical coherence tomographic angiography in rodents," Opt. Lett., Vol. 47, pp. 5060-5063, 2022

[7] Y. Jian, J. Xu, M.A. Gradowski, S. Bonora, R.J. Zawadzki, and M.V. Sarunic, "Wavefront sensorless adaptive optics optical coherence tomography for in vivo retinal imaging in mice," Biomed. Opt. Express, Vol. 5, pp. 547-559, 2014.

[8] R.R. Iyer, J.E. Sorrells, L. Yang, E.J. Chaney, D.R. Spillman, B.E. Tibble, C.A. Renteria, H. Tu, M. Žurauskas, M. Marjanovic, and S.A. Boppart, "Label-free metabolic and structural profiling of dynamic biological samples using multimodal optical microscopy with sensorless adaptive optics," Sci. Rep., Vol 12, pp.1-15, 2022.

[9] H. Kukreja, N. Bharath, C.S. Siddesh, and S. Kuldeep," An introduction to artificial neural network," Int J. Adv Res. Innov. Ideas. Educ., Vol. 1, pp. 27-30, 2016.

[10] Y. LeCun, "Generalization and network design strategies," Connectionism in Perspective Elsevier, Vol. 19, pp. 143-155, 1989.

[11] J.D. Schmidt, "Numerical simulation of optical wave propagation: With examples in MATLAB. SPIE," pp. 169-178, 2010.

[12] L.C. Andrews, R.L. Phillips, *Laser Beam propagation through random media*, Bellingham, 2nd Ed., pp. 66-71, 2005.

[13] P.D. Kingma, J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv: 1412, pp. 69-80, 2014.

[14] P.-T. De Boer, P.K. Dirk, M. Shie, Y.R. Rubinstein, "A tutorial on the cross-entropy method," Ann. Oper. Res., Vol. 134, pp. 19-67, 2005.

Jafar Bakhtiar Shohani, born in Ilam, Iran in 1987, received his BSc from Malek-Ashtar University of Technology in optics and laser engineering in 2010.

He received his MSc from Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan in 2014, and now is PhD student in University of Isfahan, Iran. His research interests include adaptive optics, machine learning and its applications in optics.

Morteza Hajimahmoodzadeh, born in Tehran, Iran (1964), received his BSc and MSc

degrees in Physics from University of Isfahan, Isfahan (1988 and 1992) and PhD degree in Physics from Moscow State University, Moscow, Russia (2005).

He was a lecturer in University of Isfahan since 1992. His fields of interest are Dynamical Systems, Optical Thin Films, and Computational Physics.

Hamid Reza Fallah is a professor of physics at Physics Department, University of Isfahan, Isfahan, Iran.

He received his B.Sc. in Applied Physics from University of Isfahan in 1988, his M.Sc. and Ph.D. in Applied Optics from Imperial College, London in 1993 and 1997, respectively. His research interests include Optical system design, Optical Thin Films, and Applied Optics.

THIS PAGE IS INTENTIONALLY LEFT BLANK.