MALATYA TURGUT OZAL UNIVERSITY JOURNAL of ENGINEERING and NATURAL SCIENCES Volume 5, Issue 2 (2024) 8-15

NATURENGS

E-ISSN 2717-8013

https://dergipark.org.tr/en/pub/naturengs

WF-AlexNet:AlexNet with Automatically Optimized Hyperparameters for Weather Forecasting

Soner KIZILOLUK [a], Eser SERT [a,*]

[a] Malatya Turgut Ozal University, Department of Computer Engineering, Malatya Türkiye - 44210

*Corresponding author

ARTICLE INFO

Received	07.09.2024
Accepted	25.10.2024

Doi: 10.46572/naturengs.1545180

ABSTRACT

Image classification is a critical area of research with widespread applications across various disciplines, including computer vision, pattern recognition, and artificial intelligence. Despite the advancements in Convolutional Neural Networks (CNNs), which have revolutionized the field by providing powerful tools for image classification, many studies have encountered challenges in achieving optimal classification performance. These challenges often arise from the complex nature of CNN architectures and the multitude of hyperparameters that require fine-tuning. Among the CNN models, AlexNet has been widely recognized for its contributions to deep learning, yet there remains significant potential for improvement through the optimization of its hyperparameters. In this study, WF-AlexNET designed to enhance the performance of the AlexNet architecture by optimizing the hyperparameters of its first convolutional layer using the Equilibrium Optimization (EO) algorithm. The EO algorithm, was employed to fine-tune the filter size, filter number, stride, and padding parameters, which are crucial for effective feature extraction. The proposed WF-AlexNET method was rigorously tested on a multi-class weather image dataset to evaluate its classification accuracy and robustness. Experimental results demonstrate that WF-AlexNET significantly outperforms the standard AlexNet model, achieving a 10.5% increase in mean validation accuracy and a 6.51% improvement in test accuracy. Furthermore, the proposed approach was compared against other prominent CNN architectures, including VGG-16, GoogleNet, ShuffleNet, MobileNet-V2, and VGG-19. WF-AlexNET consistently exhibited superior classification performance across multiple metrics, including F1-score and maximum accuracy, highlighting its efficacy in addressing the challenges associated with hyperparameter optimization in CNNs.

Keywords: Image classification; Equilibrium Optimization (EO); Convolutional Neural Networks (CNN); AlexNet

1. Introduction

CNN was first proposed by Fukushima in 1988. However, due to the limited computing capacity of computer hardware, they could not be used to train the network. LeCun and colleagues [1] used a gradient-based learning approach to classify handwritten digits using CNNs and achieved success. Later, various researchers have proposed different CNN architectures to improve CNN performance and successfully applied them to different areas like segmentation, classification, image recognition. VGG-16, GoogleNet, ShuffleNet, AlexNet, MobileNet-V2, and VGG-19 are among the most popular CNN architectures.

In parallel with the developments in CNNs, the number of hyper parameters has also increased remarkably. These hyper parameters can be listed as number of layers, stride, sizes of filter and pooling, and the specifications in each fully-connected layer and so forth [2]. In recent times, various approaches have been proposed in order to optimize these hyper parameters. Similarly, the current work uses the Equilibrium Optimization (EO) technique introduced by Faramarzi et al. [3] to optimize the first convolutional layer hyperparameters of the AlexNet. As a result, the suggested technique was named WF-AlexNET.

The present study mainly focuses on improving the performance of AlexNet and proposing a classifier with a higher performance.

Some algorithms which have been used to optimize CNN in the existing literature until now are as follows:

1- Asynchronous successive halving algorithm [4].

2- Bayesian optimization [4].

3- Equilibrium optimization [5].

* Corresponding author. e-mail address: eser.sert@ozal.edu.tr
ORCID : 0000-0002-8611-701X

4- Evolutionary algorithm [6].

5- Genetic algorithm [7].

6- Gray wolf optimizer algorithm [8].

7- Grid search [4].

8- Particle swarm optimization [9,10].

9- PSF-HS Algorithm [11].

10- Randomized L'evy flight coati optimization [12].

11- Random search [4].

Wojciuk et al. [4] discussed the importance of different hyperparameters and compared them by optimizing the hyperparameters of CNN using asynchronous successive halving algorithm, grid search and random search, Bayesian optimization. The authors also performed fine-tuning of the hyperparameters. As a result, optimizing the hyperparameters of the CNN resulted in an increase of up to 6% in classification accuracy. Emam et al. [12] introduced a method for classifying breast cancer, utilizing the DenseNet architecture to extract features from breast thermal images. In their study, they fine-tuned the model's hyperparameters to achieve optimal classification performance. The authors also applied a randomized L'evy flight to optimize specific hyperparameters within the DenseNet121 architecture, naming their approach LFR-COA-DenseNet121-BC.

Naik et al. [8] introduced a method utilizing a CNN for the identification of primitive Indian paddy grains. In their research, the CNN's hyperparameters were fine-tuned with the gray wolf optimizer, which enhanced the classification performance. In a another study, Nguyen et al. [5] developed a time-series prediction method for traffic transportation, where they employed the meta-heuristic EO algorithm to train the CNN instead of the conventional backpropagation technique. Similarly, Esfahanian and Akhavan [7] applied Genetic Algorithms to optimize CNN architectures. Wang et al. [9] proposed a PSO-based approach to refine the hyperparameters of CNNs for better performance.

Soon et al. [13] developed a CNN architecture that includes hyperparameter optimization for recognizing car brand logos. In the related study, Lee et al. [11] utilized the PSF-HS algorithm, a meta-heuristic optimization technique, to optimize hyperparameters during the feature extraction phase of a CNN. Bochinski et al. [6] developed an evolutionary approach for tuning the hyperparameters of a CNN model. Similarly, Xiao et al. [14] used a variable-length GA to automatically adjust and improve the performance of CNNs by optimizing their parameters.

Yildirim et al. [15] performed the classification of weather images with the hybrid method they proposed using deep learning, SVM classifier, and mRMR feature selection methods. Chen et al.[16] proposed a cascade machine learning forecasting system for 15-day global weather forecasting and named it FuXi. Venkatachalam et al. [17] proposed the weather forecasting approach based on the deep learning model Long Short-Term Memory and the transductive long short-term memory model.

The main contributions of the present study can be summarized as follows:

1) In this study, the hyperparameters of CNN's AlexNet architecture were optimized with EO, one of the new metaheuristic algorithms.

2) In our proposed approach, AlexNet's accuracy value increased significantly at fewer epochs compared to the approaches in the literature.

3) The WF-AlexNET was compared with unoptimized CNN architectures frequently used in the literature, and it was clearly stated that the WF-AlexNET demonstrated superior classification performance.

The organization of the present study is as follows: Section 2 describes the EO, AlexNet and CNN. Section 3 and 4 present the WF-AlexNET and experiments, respectively. Section 5 concludes the study.

2. Material and Method

2.1. Equilibrium optimization algorithm

EO, a cutting-edge meta-heuristic optimization algorithm, was recently introduced by Faramarzi et al. [3]. It draws inspiration from control volume mass balance models, which are used to estimate both dynamic and equilibrium states. In the EO framework, each particle, representing a potential solution, functions as a search agent with its concentration (or position) guiding the search process. The algorithm involves several key steps, which are outlined below. For a comprehensive overview of the EO algorithm, refer to Faramarzi et al. [3]. The pseudo-code for the EO algorithm is illustrated in Figure 1.

```
Set EO parameters as a1=2, a2=1, GP=0.5
Initialize population Pi (i=1,2, 3,...,N)
While (iter<max_iter)
    Calculate the fitness of Pi (i=1,2,3,...,N)
    Build equilibrium pool using four best particles
    Accomplish the memory saving
    Assign t
    For i=1:Number of Particles
        Calculate F, GCP, G, G0
        Update Pi
    End For
    iter=iter+1
End While
```

Figure 1. Pseudo code of EO

2.2. Convolutional neural networks

The development of CNNs was inspired by the visual processing system of animals [18,19]. CNNs have become a popular tool in image and voice processing due to their capability to automatically and adaptively extract features during training [20]. However, to build an effective CNN architecture, it is essential to carefully fine-tune the hyperparameters.

CNN architecture mainly consists of three layers: Convolution, pooling, and fully connected layers. Convolution, which is the layer for feature extraction, is considered as one of the most significant components of a CNN architecture [20]. Specially, given the pivotal role of the initial convolution layer in extracting features from unprocessed data, extant literature indicates that the optimization of hyperparameters can markedly enhance the performance of the initial convolution layer [9,21].

In the present study, WF-AlexNET approach was proposed using EO algorithm in order to optimize hyper parameters such as filter size, filter number, stride and padding in the first convolution layer of AlexNet architecture [22]. The weather dataset was chosen as a test data to evaluate the performance of the WF-AlexNET in image processing. Additionally, the results of the WF-AlexNET were compared with those of five distinct CNN: VGG-19, GoogleNet, ShuffleNet, MobileNet V2, and VGG-16. The MobileNet CNN model comprises 53 layers and accepts images of 224x224 dimensions as input [23]. The ShuffleNet, introduced by Zhang et al. [24], comprises 50 layers and accepts images of dimensions 224x224. GoogleNet, introduced by Szegedy et al. [25], has a depth of 22 layers and an image input dimension of 224x224. Finally, VGG-16 and VGG-19, introduced by Simonyan and Zisserman [26], are CNN models with a depth of 16 and 19 layers, respectively, and an image input dimension of 224x224.

2.3. AlexNet

In [22], Krizhevsky and colleagues proposed the AlexNet model, which accepts images with dimensions of 227 x 227. The network comprises 5 convolution layer and 3 fully connected layers. Furthermore, following the convolution 1, 2 and, 5, there are 3 additional maximum pooling layers with a kernel size of 3x3. Figure 1 shows the main layers of AlexNet. Table 1 shows the filter size, filter number, stride and padding values of the convolutional layers.

Table 1. The hyperparameter settings of AlexNet's convolution layers.

	Filter Size	Filter Number	Stride	Padding
Convolution1	11x11	96	4	0
Convolution2	5x5	256	1	2
Convolution3	3x3	384	1	1
Convolution4	3x3	384	1	1
Convolution5	3x3	256	1	1

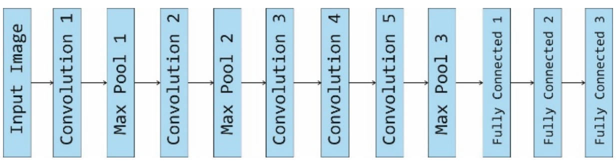

Figure 2. AlexNet architecture

3. Optimizing hyper-parameters of AlexNet using the equilibrium optimization algorithm

3.1. Preprocessing

Wiener filter is a statistical approach used to improve a noisy image or signal. Its main purpose is to minimize the effect of noise by making the best estimate of the original signal. The Wiener filter is used to both reduce random noise and correct the distorted signal. It was aimed to purify the images from noise by passing the images in the Multi-class Weather [27] dataset used in this study through the Wiener filter. Thus, it is ensured that the proposed method works with superior performance in terms of accuracy.

3.2. Optimizing AlexNet's Hyperparameters and Classification Process

It is of great consequence to calibrate the hyperparameters of CNN in an optimal manner to enhance the efficacy of CNN performance. The present work introduces the WF-AlexNET strategy, which uses the EO to optimize hyperparameters of the initial convolution layer in the AlexNet CNN model. Therefore, the present study aims to achieve the lowest Accuracy Error (ACCerr) value in Equation (1) during the training process of AlexNet, in other word reaching the highest Validation Accuracy.

$$ACC_{err} = 100 - Validation\ Accuracy \quad (1)$$

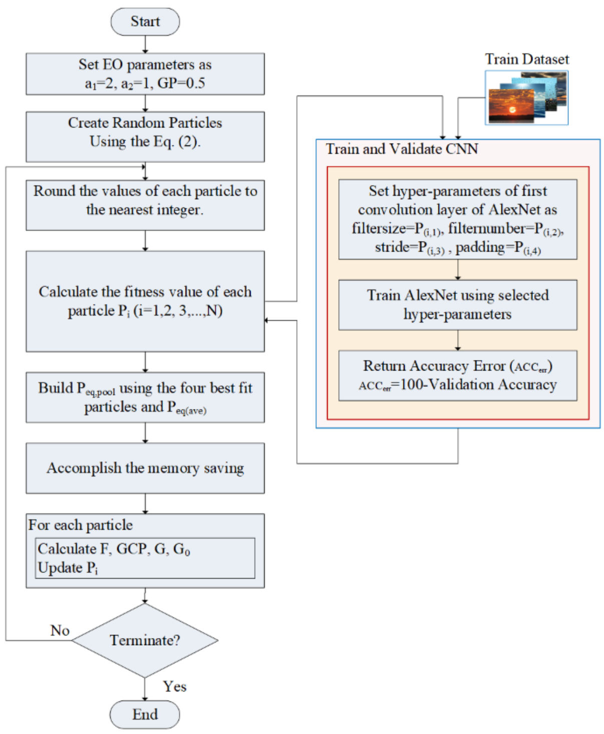

Figure 3. Flowchart of WF-AlexNET

The flow chart in Figure 3 depicts the essential processes in the proposed WF-AlexNET technique. First, GP, a2, and a1 parameters of the EO is adjusted.

Subsequently, the algorithm generates an initial population of particles, each of which is assigned a random set of values. These particles are four-dimensional, representing the filter size, number of filters, stride, and padding parameters in AlexNet's convolutional layer. The current study's upper and lower bounds for these parameters are detailed in Table 2. Following these limits, each particle in the initial population is generated using equation (2).

$$P_{(i,j)} = LB_{(j)} + rand_{(i,j)}\left(UB_{(j)} - LB_{(j)}\right), \quad i = 1,2,\dots,N, \; j = 1,\dots,4 \tag{2}$$

Table 2. Hyper-parameters to optimize and their ranges.

Parameter	LB	UB
Filter size	1	11
Filter Number	20	120
Stride	2	5
Padding	0	4

The parameters of the convolution layer are given integer values. In the initial population and subsequent EO steps, particles possess real values. Consequently, the next phase of the proposed method involves rounding each particle's value to the nearest integer.

In order to calculate the fitness values of a particle, the following steps are undertaken: First, the particle's values, for which the fitness value is to be computed are allocated to the parameters of the filter size, filter number, stride, and padding in Alexnet's first convolutional layer. The CNN is then taught using training data sets based on a given epoch. Once the training process is complete, the ACCerr value is calculated using the equation (1), and this value is returned as the fitness value.

Following this, an equilibrium pool and particle storage are established using four particles that exhibit the best fitness (with the lowest ACCerr) and a fifth particle that represents the arithmetic mean of these four. The G0, G, F, and GCP values for each particle are then computed and updated accordingly. Once the termination criteria are satisfied, the particle with the best fitness value is chosen as the final solution.

4. Experimental results

This work presents an EO based WF-AlexNET technique for optimizing the AlexNet CNN model's first convolution layer hyperparameters in order to improve classification performance. The suggested approach's image classification efficacy was tested using the weather dataset as a benchmark. In addition, the suggested approach's results were compared to those of five unique CNN are given in Section 2.2. This section contains the picture dataset, experiment design, and comparative experimental results from the current investigation.

4.1. Image datasets

In the present study, Multi-class Weather [24 27] was selected as image classification dataset. Multi-class Weather dataset consists of four classes and 1124 images. 920 and 204 images were randomly selected for training and testing, respectively. In addition, during the training, 70% of the training data in these three datasets were used for the training process, while the remaining 30% were used for the training validation. Testing data were used for testing after the training process. Figure 4 shows samples from this dataset.

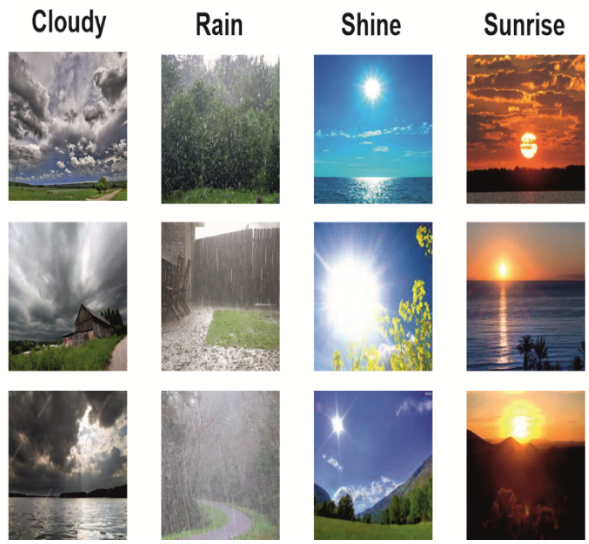

Figure 4. Sample images from the dataset

4.2. Experimental setup

All experiments were conducted using MATLAB R2024a. For the WF-AlexNET method, the parameters GP, a2, and a1 were set to 0.5, 1, and 2, respectively. The initial population size and the maximum number of iterations were both set to 10. An initial learning rate of 0.0005 was used for the WF-AlexNET as well as for other CNN models. The training consisted of 10 epochs, and each model was evaluated over ten runs on the dataset. The models were compared based on their standard deviation, F1 score, maximum accuracy, and overall accuracy.

4.3. Experimental results

First, the results of only these two methods were compared in order to see how the proposed WF-AlexNET affects the success of the standard AlexNet model. Mean train validation accuracy and test accuracy values obtained from 10 independent runs on all datasets are given in Table 3. In addition, confusion matrices obtained 10 independent runs on Weather dataset are shown in Figure 5 and 6, respectively.

Table 3. WF-AlexNET and AlexNet mean accuracy values

	Train validation	Test
AlexNet (%)	80.90	85.00
WF-AlexNET (%)	91.40	91.51
Increment (%)	10.5	6.51

As indicated in Table 3, in the training, the proposed approach has resulted in a 10.5% increase for AlexNet. Likewise, there has been a 6.5% increase in the mean accuracy values for testing.

When analyzing Figure 6, AlexNet yielded a lower classification accuracy rate in three other classes except sunrise. On other hand, WF-AlexNET's accuracy rate increased for sunrise compared to AlexNet, but more for the other three classes.

AlexNet — TRAIN

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	57	10	6	0	0,7808
rain	15	40	4	0	0,6779
shine	1	3	51	6	0,8360
sunrise	1	0	1	82	0,9761
recall	0,7702	0,7547	0,8225	0,9318	0,8303

AlexNet — TEST

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	43	8	0	1	0,8269
rain	9	27	0	1	0,7297
shine	2	4	46	0	0,8846
sunrise	0	0	0	63	1
recall	0,7962	0,6923	1	0,9692	0,8774

EO-AlexNet — TRAIN

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	68	2	6	0	0,8947
rain	2	50	0	0	0,9615
shine	3	1	55	5	0,8593
sunrise	1	0	1	83	0,9764
recall	0,9189	0,9433	0,8870	0,9431	0,9241

EO-AlexNet — TEST

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	52	1	0	1	0,9629
rain	1	33	2	0	0,9166
shine	1	5	44	0	0,88
sunrise	0	0	0	64	1
recall	0,9629	0,8461	0,9565	0,9846	0,9460

Figure 5. Sample images from the dataset

AlexNet — TRAIN

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	60,6	20,9	7,4	0,5	0,6778
rain	7,7	28,2	0,5	0	0,7747
shine	4,9	3,4	53,1	5,3	0,7961
sunrise	0,8	0,5	1	82,2	0,9727
recall	0,8189	0,5320	0,8564	0,9340	0,8090

AlexNet — TEST

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	42,3	10,1	2,9	0,3	0,7607
rain	8,6	24,7	0,7	0,6	0,7138
shine	2,4	4,2	42,4	0,1	0,8635
sunrise	0,7	0	0	64	0,9891
recall	0,7833	0,6333	0,9217	0,9846	0,85

EO-AlexNet — TRAIN

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	67,9	5,1	3,8	0	0,8841
rain	1,8	45,1	0	0,1	0,9595
shine	3,2	2,3	57,2	4,9	0,8461
sunrise	1,1	0,5	1	83	0,9696
recall	0,9175	0,8509	0,9225	0,9431	0,9140

EO-AlexNet — TEST

Output Class	Target Class				
Classes	cloudy	rain	shine	sunrise	precision
cloudy	45,6	1,1	1,1	0,2	0,95
rain	5,9	34	2	0,4	0,8037
shine	2,1	3,9	42,7	0	0,8767
sunrise	0,4	0	0,2	64,4	0,9907
recall	0,8444	0,8717	0,9282	0,9907	0,9151

Figure 6. Sample images from the dataset

Table 4. Results for all models

Model	Validation				Test			
	Mean Acc.	Max Acc.	F1-Score	Std.	Mean Acc.	Max Acc.	F1-Score	Std.
MobileNet-V2	74.58	77.61	71.54	3.60	76.86	81.86	74.29	3.20
ShuffleNet	84.58	89.16	83.16	3.24	88.33	93.13	87.13	3.06
GoogleNet	83.53	87.36	82.04	4.11	86.37	90.20	84.81	2.66
VGG-16	78.73	81.58	76.25	1.75	82.25	85.29	80.12	**2.01**
VGG-19	79.81	83.39	78.20	2.99	84.21	89.21	82.78	3.77
AlexNet	80.90	83.03	78.77	1.56	85.00	87.74	83.04	2.32
WF-AlexNET	**91.40**	**92.41**	**91.03**	**0.65**	**91.51**	**94.60**	**90.57**	2.46

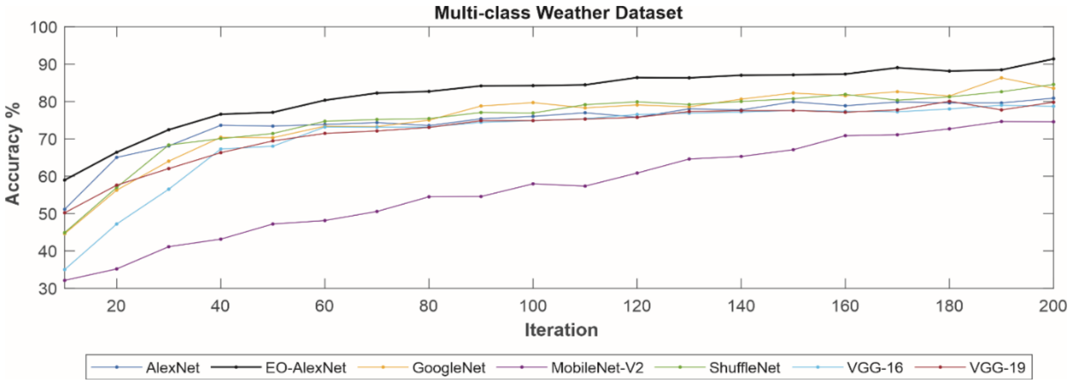

Figure 7. Mean validation accuracy curves for all models in experiments on the Multi-class Weather dataset

Standard deviation, F1-score, maximum accuracy, and mean accuracy of the all models obtained from 10 independent runs are given in Table 4.

It can be understood from Weather dataset findings that F1-score, maximum accuracy, and mean accuracy of WF-AlexNET for train were calculated as 91.03, 92.41, and 91.40, respectively. This performance was followed by ShuffleNet with 83.16, 89.16, and 84.58. Conversely, the lowest values were produced by MobileNet-V2 with 71.54, 77.61 and, 74.58. WF-AlexNET was the most accurate at 91.51%, followed by ShuffleNet at 88.33% and VGG-16 at 82.25%. The lowest standard deviation value belonged to VGG-16 with 2.01.

Mean validation accuracy curves of all models obtained from Weather dataset are shown in Figure 7. It can be clearly seen that the highest performance was displayed by WF-AlexNET, while the lowest performance was displayed by MobileNet-V2. It can also be observed that accuracy value of AlexNet increased from 35% to 70% during the first 60 iterations, while displaying a more horizontal performance in the remaining iterations. Other models displayed performances closer to each other and yielded similar results.

In Table 5, the image classification accuracy of WF-AlexNet was compared with several state-of-art methods in terms weather dataset. The maximum accuracy of WF-AlexNet in 10 independent runs is 94.60%. According to these results, in Table 5, the proposed approach outperformed the other weather dataset models in image classification accuracy.

Table 5 Accuracy Comparison of proposed WF-AlexNet approach with other state-of-art methods for weather prediction

Source	Models/Methods	Number of classes	Overall Acc (%)
Oluwafemi et al. [28]	Ensemble Method	4 (cloudy, rainy, sunshine, sunrise)	86
Tian et al. [29]	Spiking neural network	4 (cloudy, rain, sunshine, sunrise)	93.5
Proposed	WF-AlexNet	4 (cloudy, rain, shine, sunrise)	94.60

5. Conclusion

In this research, we introduced WF-AlexNET, an innovative method to optimize the hyperparameters of the initial convolutional layer in the AlexNet CNN model using the EO algorithm. The primary objective was to enhance the classification performance of the AlexNet architecture, particularly in the context of weather data analysis. The experimental results indicate that the proposed WF-AlexNET approach significantly improves the accuracy and robustness of the AlexNet model compared to its standard, unoptimized version and other prominent CNN architectures, such as VGG-19, VGG-16, GoogleNet, ShuffleNet, and MobileNet-V2.

Our approach's success can be attributed to the precise optimization of critical hyperparameters—filter size, filter number, stride, and padding—in the initial convolutional layer, which plays a pivotal role in feature extraction. By employing the EO algorithm, we effectively fine-tuned these parameters, resulting in a marked improvement in both validation and test accuracy. Specifically, WF-AlexNET achieved a 10.5% increase in mean validation accuracy and a 6.51% improvement in test accuracy over the standard AlexNet, highlighting the efficacy of the proposed method. Furthermore, WF-AlexNET consistently outperformed the comparative CNN models across multiple evaluation metrics, including F1-score, maximum accuracy, and standard deviation, reinforcing its superiority in image classification tasks. The results also demonstrated that our method required fewer epochs to achieve higher accuracy, making it not only more accurate but also more efficient in terms of computational resources and training time.

The significance of this work lies in its contribution to the ongoing efforts to optimize CNN architectures for better performance. By focusing on the initial convolutional layer—a critical component of CNNs—this study provides insights into the importance of hyperparameter optimization in improving model accuracy. The success of WF-AlexNET underscores the potential of EO and similar meta-heuristic algorithms in enhancing CNN performance, offering a valuable tool for researchers and practitioners working on image classification and other related fields.

In conclusion, the proposed WF-AlexNET approach offers a high-performance solution for weather image classification, with potential applications extending to other image classification tasks. Future research could explore the application of the EO algorithm to optimize other layers within CNN architectures or extend the approach to different CNN models and datasets. Additionally, integrating WF-AlexNET with other optimization techniques could further refine its performance, paving the way for even more robust and efficient image classification solutions.

References

[1] **Lecun, Y.**, Bottou, L., Bengio, Y., and Haffner, P., (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278-324. https://doi.org/10.1109/5.726791

[2] **Andonie, R.,** and Florea, A.-C., (2020). Weighted Random Search for CNN Hyperparameter Optimization. *International Journal of Computers Communications & Control*, 15. https://doi.org/10.15837/ijccc.2020.2.3868

[3] **Faramarzi, A.**, Heidarinejad, M., Stephens, B., and Mirjalili, S., (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191, 105190. https://doi.org/10.1016/j.knosys.2019.105190

[4] **Wojciuk, M.**, Swiderska-Chadaj, Z., Siwek, K., and Gertych, A., (2024). Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization. *Heliyon*, 10, e26586. https://doi.org/10.1016/j.heliyon.2024.e26586

[5] **Nguyen, T.**, Nguyen, G., and Nguyen, B.M., (2020). EO-CNN: An Enhanced CNN Model Trained by Equilibrium Optimization for Traffic Transportation Prediction. *Procedia Computer Science*, 176, 800-9. https://doi.org/10.1016/j.procs.2020.09.075

[6] **Bochinski, E.**, Senst, T., and Sikora, T., (2017). Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, Beijing. s. 3924-8. https://doi.org/10.1109/ICIP.2017.8297018

[7] **Esfahanian, P.**, and Akhavan, M., (2019). GACNN: Training Deep Convolutional Neural Networks with Genetic Algorithm. arXiv:1909.13354. https://arxiv.org/abs/1909.13354.

[8] **Naik, N.K.**, Sethy, P.K., Devi, A.G., and Behera, S.K., (2024). Few-shot learning convolutional neural network for primitive indian paddy grain identification using 2D-DWT injection and grey wolf optimizer algorithm. *Journal of Agriculture and Food Research*, 15, 100929. https://doi.org/10.1016/j.jafr.2023.100929

[9] **Wang, Y.**, Zhang, H., and Zhang, G., (2019). cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. *Swarm and Evolutionary Computation*, 49, 114-23. https://doi.org/10.1016/j.swevo.2019.06.002

[10] **Wang, Q.**, Jiang, H., Ren, J., Liu, H., Wang, X., and Zhang, B., (2024). An intrusion detection algorithm based on joint symmetric uncertainty and hyperparameter optimized fusion neural network. *Expert Systems with Applications*, 244, 123014. https://doi.org/10.1016/j.eswa.2023.123014

[11] **Lee, W.-Y.**, Park, S.-M., and Sim, K.-B., (2018). Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik*, 172, 359-67. https://doi.org/10.1016/j.ijleo.2018.07.044

[12] **Emam, M.M.**, Houssein, E.H., Samee, N.A., Alohali, M.A., and Hosney, M.E., (2024). Breast cancer diagnosis using optimized deep convolutional neural network based on transfer learning technique and improved Coati optimization algorithm. *Expert Systems with Applications*, 255, 124581. https://doi.org/10.1016/j.eswa.2024.124581

[13] **Soon, F.C.**, Khaw, H.Y., Chuah, J.H., and Kanesan, J., (2018). Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition. *IET Intelligent Transport Systems*, 12, 939-46. https://doi.org/10.1049/iet-its.2018.5127

[14] **Xiao, X.**, Yan, M., Basodi, S., Ji, C., and Pan, Y., (2020). Efficient Hyperparameter Optimization in Deep Learning Using a Variable Length Genetic Algorithm. arXiv:2006.12703. https://arxiv.org/abs/2006.12703.

[15] Yildirim, M., Çinar, A., & CengIl, E. (2022). Classification of the weather images with the proposed hybrid model using deep learning, SVM classifier, and mRMR feature selection methods. Geocarto International, 37(9), 2735-2745. https://doi.org/10.1080/10106049.2022.2034989

[16] Chen, L., Zhong, X., Zhang, F., Cheng, Y., Xu, Y., Qi, Y., & Li, H. (2023). FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. npj Climate and Atmospheric Science, 6(1), 190. https://doi.org/10.1038/s41612-023-00512-1

[17] Venkatachalam, K., Trojovský, P., Pamucar, D., Bacanin, N., & Simic, V. (2023). DWFH: An improved data-driven deep weather forecasting hybrid model using Transductive Long Short Term Memory (T-LSTM). Expert systems with applications, 213, 119270. https://doi.org/10.1016/j.eswa.2022.119270

[18] **Guo, T.**, Dong, J., Li, H., and Gao, Y., (2017). Simple convolutional neural network on image classification. *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, IEEE, Beijing, China. s. 721-4. https://doi.org/10.1109/ICBDA.2017.8078730

[19] **Nhat-Duc,** H., Nguyen, Q.-L., and Tran, V.-D., (2018). Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Automation in Construction*, 94, 203-13. https://doi.org/10.1016/j.autcon.2018.07.008

[20] **Yamashita, R.**, Nishio, M., Do, R.K.G., and Togashi, K., (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9, 611-29. https://doi.org/10.1007/s13244-018-0639-9

[21] **Sinha, T.**, Haidar, A., and Verma, B., (2018). Particle Swarm Optimization Based Approach for Finding Optimal Values of Convolutional Neural Network Parameters. *2018 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, Rio de Janeiro. s. 1-6. https://doi.org/10.1109/CEC.2018.8477728

[22] **Krizhevsky, A.**, Sutskever, I., and Hinton, G.E., (2012). ImageNet Classification with Deep Convolutional Neural Networks. *In Proceedings of the 25th International Conference on Neural Information Processing Systems* (pp. 1097–1105).

[23] **Sandler, M.**, Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT. s. 4510-20. https://doi.org/10.1109/CVPR.2018.00474

[24] **Zhang, X.**, Zhou, X., Lin, M., and Sun, J., (2018). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT. s. 6848-56. https://doi.org/10.1109/CVPR.2018.00716

[25] **Szegedy, C.**, Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Boston, MA, USA. s. 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[26] Simonyan, K., and Zisserman, A., (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556. https://arxiv.org/abs/1409.1556.

[27] Ajayi, G., (2018). Multi-class Weather Dataset for Image Classification. Mendeley Data, v1. http://dx.doi.org/10.17632/4drtyfjtfy.1.

[28] Oluwafemi, A. G., and Wang Z., (2019). Multi-class weather classification from still image using said ensemble method. *2019 IEEE Southern African universities power engineering conference/robotics and mechatronics/pattern recognition association of South Africa (SAUPEC/RobMech/PRASA)*,135-140. https://doi.org/10.1109/RoboMech.2019.8704783

[29] Tian, M., Chen, X., Zhang, H., Zhang, P., Cao, K., and Wang, R., (2021). Weather classification method based on spiking neural network. *2021 IEEE International Conference on Digital Society and Intelligent Systems (DSInS)*, 134-137. https://doi.org/10.1109/DSInS54396.2021.9670557