JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 13(1), MARCH, 2025
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

# Nomadic Pastoralist Optimization Algorithm for Enhanced Convolutional Neural Networks in Face Recognition

[1,2]Abdulrazaq, A. A., [2]Abdulrazaq, M. B, [2]Ibrahim, Y., [2]Adedokun, E. A
[1]*Computer Engineering Department, University of Maiduguri, Borno, Nigeria*
[2]*Computer Engineering Department, Ahmadu Bello University, Zaria*

## ABSTRACT

*Face recognition (FR) remains a prominent focus in computer vision research, with convolutional neural networks (CNNs) widely recognized for their efficiency in this domain. Achieving optimal performance from CNNs requires fine-tuning their parameters, which can be challenging and time-consuming when done manually. To address this, population-based metaheuristic methods offer an automated approach to hyperparameter optimization, significantly reducing manual effort while enhancing model efficiency. This study employs the Nomadic Pastoralist Optimization Algorithm (NPOA) to optimize CNN hyperparameters for face recognition. Key parameters such as learning rate, batch size, the number of filters, dropout rate and dense units are fine-tuned using NPOA. Experiments conducted on a standard FR dataset demonstrate that the proposed NPOA-CNN model achieves superior accuracy compared to existing CNN models. Through iterative optimization, NPOA effectively minimizes the objective function by selecting the best combination of hyperparameters. As a result, the proposed model achieves an impressive face recognition accuracy of 98.82%.*

**ARTICLE INFO**
*Article History*
Received: October, 2024
Received in revised form: December, 2024
Accepted: February, 2025
Published online: March, 2025

**KEYWORDS**
Nomadic Pastoralist Optimization Algorithm, Convolutional Neural Networks & Face Recognition

## INTRODUCTION

In recent years, the need for high security has become a critical concern across various domains, from military and governmental institutions to civilian applications such as financial transactions, secure access to sensitive facilities, and public surveillance(Zulfiqar et al., 2019). With the rapid evolution of technology, the need for robust, reliable, and efficient security systems has intensified. Among various biometric systems, face recognition has emerged as a promising solution due to its non-invasive, user-friendly nature and the availability of facial data in numerous applications, such as social media and surveillance footage. Face recognition systems leverage the uniqueness of facial features to identify individuals. These systems have demonstrated significant utility in applications like law enforcement, fraud prevention, and personalized customer services. However, despite extensive research and development in automatic face recognition, achieving accurate and consistent identification remains a formidable challenge, particularly in unconstrained environments. Such environments often involve variations in lighting, pose, expression, age, and occlusions, all of which can degrade the performance of face recognition models (Agarwal et al., 2019; Kortli et al., 2020).

Convolutional Neural Networks (CNNs), a class of deep learning models, have revolutionized the field of computer vision and have shown exceptional performance in face recognition tasks (Hu et al., 2015). CNNs are particularly well-suited for extracting and learning hierarchical features from facial images, enabling them to achieve high accuracy even in challenging scenarios (Alzubaidi et al., 2021; Ilyas et al., 2019; Nurhopipah & Larasati, 2021). However, deploying CNNs for face recognition in real-world environments still encounters significant challenges, including the reliance on large-scale

*Corresponding author: Abdulrazaq, A. A.*
ibnrazaaq@unimaid.edu.ng
*Computer Engineering Department, University of Maiduguri, Borno, Nigeria.*

labeled datasets, and the complexity of tuning hyperparameters to achieve optimal performance and generalization in diverse conditions.(Mikolaj et al., 2024)

This study explores the optimization of CNN model for face recognition, emphasizing their ability to address challenges associated with unconstrained environments. In Convolutional Neural Networks (CNNs), hyperparameters include variables such as the number of hidden layers, padding, learning rate, batch size, and others, which must be defined before training begins (Wang et al., 2019). Finding the optimal configuration for these parameters is challenging due to their large number and complex interactions. Hyperparameters play a crucial role in shaping the performance of the training process (Wu et al., 2019). Studies have shown that selecting appropriate hyperparameters leads to significantly better results (Vose et al., 2019). However, hyperparameter tuning is a nonconvex optimization problem characterized by nonlinear interactions (Yang & Shami, 2020), and the search space grows exponentially, often leading to solutions that are trapped in local optimal (Wang et al., 2019). Identifying the right hyperparameter values remains one of the most complex tasks in CNN development.

Researchers typically employ either manual or automatic methods for CNN hyperparameter optimization. Manual approaches often rely on trial-and-error or experiential knowledge to initialize hyperparameters. While commonly used, this method is inefficient, time-consuming, and prone to suboptimal results. Effective tuning requires configurations that minimize loss or maximize model accuracy. In contrast, automatic methods, such as grid search, systematically evaluate all possible combinations of hyperparameters. While this approach can be effective for small-scale problems, it becomes computationally expensive as the number of parameters increases, suffering from the curse of dimensionality(Bergstra & Bengio, 2012). Random search, another automatic technique, selects hyperparameter combinations randomly from the search space. Within a fixed time and resource constraint, it often outperforms grid search by exploring a broader range of configurations. Given that hyperparameter optimization is a difficult problem, intelligent search strategies are necessary. Metaheuristic optimization methods are particularly useful when exact solutions are computationally infeasible or unnecessary. Although these methods do not guarantee finding the global optimum, they are adept at identifying good solutions within reasonable time and resource limits. As exact optimization approaches are often impractical in machine learning due to high computational costs, metaheuristic strategies have emerged as a preferred alternative (Karlupia et al., 2023).

Hyperparameter optimization in CNNs is a complex combinatorial problem, often addressed effectively using metaheuristic methods like Nomadic pastoralist Optimization Algorithm (NPOA), Genetic Algorithms (GAs) and Bacterial Foraging Optimization (BFO). These algorithms leverage exploration and exploitation to solve multi-objective optimization challenges. In this work, a hybrid framework integrates NPOA with CNNs for optimizing hyperparameters in face recognition tasks is proposed. Key parameters tuned include number of filters, batch size, dropout rate and dense units. This study highlights the applicability of nature-inspired algorithms, particularly NPOA in improving CNN performance for face recognition. Hence, the main contribution of this work is the proposed hybrid approach by integrating the evolutionary technique NPOA with CNNs for hyperparameter optimization.

**Nomadic Pastoralist Optimization Algorithm (NPOA)**

The Nomadic Pastoralist Optimization Algorithm (NPOA) is a nature-inspired optimization technique that mimics the adaptive behaviors of nomadic pastoralist communities in search of optimal grazing areas. NPOA iteratively refines candidate solutions (pastoralists) by balancing exploration (searching new regions) and exploitation (refining promising solutions), aiming to find the best solution (Abdullahi *et al*., 2019; Abdullahi *et al*., 2021)

In NPOA, each pastoralist represents a potential solution in hyperparameter tuning such

*Corresponding author: Abdulrazaq, A. A.*
ibnrazaaq@unimaid.edu.ng
*Computer Engineering Department, University of Maiduguri, Borno, Nigeria.*

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 13(1), MARCH, 2025
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

as model hyperparameters optimization for a neural network like a CNN or represent an optimal location in a utility placement Problem. The algorithm proceeds in cycles where each pastoralist's position is evaluated based on a "fitness" score, guiding movements toward higher-scoring areas. Through two main strategies, *exploration* (searching new areas) and *exploitation* (focusing on promising areas), the algorithm refines these solutions iteratively. The end goal is to identify the best-performing solution across the hyperparameter space (Abdullahi *et al*., 2021).

The NPOA algorithms follows the following steps:

i. *Start*
ii. *Initialize NPOA parameters*
iii. *Initialize the cultural algorithm Belief Space (situational, normative and domain knowledge components)*
iv. *Select 25% of pastoralist as scout pastoralist randomly from population of pastoralists and initialize scout location*
v. *Evaluate the fitness of each scout, sort and select best accepted 20% scouters for scout belief space knowledge adjustment*
vi. *Adjust scout situational and normative knowledge components*
vii. *If the maximum scouting rate is reached, the next scout locations are generated using the influence of the updated SK and NK components*
viii. *Select best camping location based on the best scout fitness and move pastoralist to camp for herding*
ix. *Evaluate the fitness of each herd pastoralist, sort and select best accepted 20% herders for herd belief space knowledge adjustment*
x. *Adjust herders SK and NK components*
xi. *Split pastoralist to different locations within camp, evaluate the fitness of each pastoralist, update the NK and update SK*
xii. *Repeat step xi until maximum splitting rate is reached. For each split, update the camp best (Cbest) location and fitness*
xiii. *If all regions within the search space have not been explored (maximum iteration not reached), select 25% pastoralist as scout and change their location. Repeat steps v to xii and update the global camp best pastoralist Gbest.*
xiv. *Else, return the global best-found pastoralist Gbest,*
xv. *Stop*

**CNN Architecture**

In CNNs, feature extraction is accomplished through a series of convolutional and pooling layers. The initial convolutional layers capture basic, low-level features such as horizontal, vertical, and diagonal edges. As we move to deeper layers in the network, these simple features are combined to form more complex shapes, like curves and small geometric patterns, which can serve as discriminative features essential for accurate classification (Bacanin et al., 2020). In a typical CNN, a flattening layer reshapes the output of the convolutional layers into a single vector before classification. This vector is then passed through one or more fully connected (FC) layers, where the classification process takes place.

The final output layer, which uses a softmax or sigmoid function, calculates the probability of each class. The class with the highest probability is chosen as the network's predicted category for the input. To evaluate its performance, CNNs use various loss functions that compute the error between predicted and actual values. This error is then minimized through a backpropagation algorithm, which iteratively adjusts the network's weights and biases to improve accuracy (Mittal & Patel, 2023).

The core architecture includes different types of layers such as convolution, pooling, ReLU, normalization, dropout, fully connected, and softmax, each performing a specific role. Figure 1 depicts a standard CNN model typically employed for classification purposes.

*Corresponding author: Abdulrazaq, A. A.*
ibnrazaaq@unimaid.edu.ng
*Computer Engineering Department, University of Maiduguri, Borno, Nigeria.*

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 13(1), MARCH, 2025
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

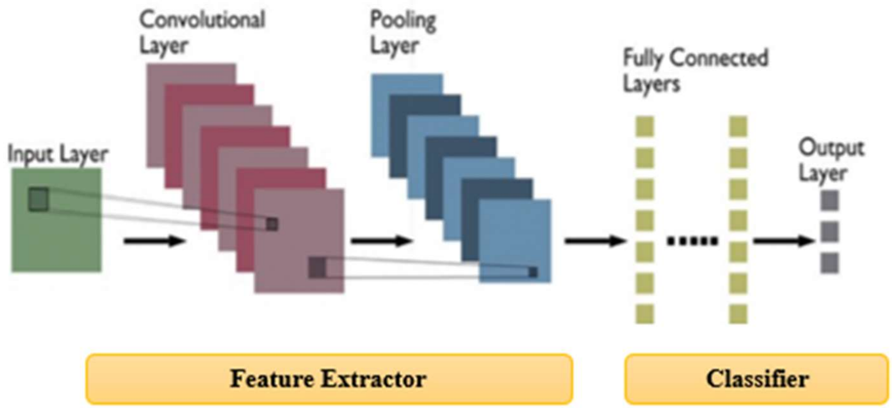


Figure 1: A typical architecture of the CNN model

The convolutional layer produces a set of feature maps. For a 2-D input image, denoted as $I(i, j)$, and a 2-D convolutional kernel, K, with dimensions $m \times n$, the convolutional operation can be expresses by;

$$C\ (i,j) = \sum_{m}\sum_{n} I\ (i+m-1, j+n-1)\ K\ (m,n) \qquad (1)$$

The normalized activation is given by:

$$\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma^2\beta + \epsilon}} \qquad (2)$$

Where $x_i$ are inputs over mini-batches, $\mu_\beta$ and $\sigma^2$ represent the mean and variance respectively. Also, $\epsilon$ represent numerical stability for the low mini-batch variance. After shifting the input by a learnable offset β and scaling it using the learnable factor γ, the activation is given by;

$$y_i = \gamma\hat{x}_i + \beta \qquad (3)$$

Where $\hat{x}_i$ is the normalized value obtained. A dropout layer can be incorporated to set certain input elements to zero based on a defined probability. The activation probability for each node is determined by a binomial distribution, which decides whether a node is active or dropped (Karlupia et al., 2023).

**The Proposed Model**

A highly efficient method for image classification using the CNN-NPOA model is proposed in this work. It leverages the NPOA to optimize the hyperparameters of the CNN model. The analyzed parameters include kernel size (filter size), the number of filters, batch size, dense units and the number of hidden layers, which collectively determine the depth of the CNN structure. While shallow networks lack the capacity to handle complex image processing tasks, excessively deep networks are prone to overfitting.

The selection of appropriate hyperparameters is essential to ensure that the network is capable of effectively capturing and representing the most important features from the image. Proper tuning of the model's parameters significantly enhances the CNN's ability to extract and highlight key texture features, ensuring that the entire image distribution is represented accurately. Without this careful optimization, the network may fail to converge efficiently or might miss out on critical feature patterns, impacting overall model performance. The NPOA was employed to address these challenges by systematically searching the hyperparameter space to find the best combination for improved network performance.

The Nomadic Pastoralist Optimization Algorithm (NPOA) was chosen for its ability to continuously explore new regions of the search space, avoiding dependence on previously

Corresponding author: Abdulrazaq, A. A.
ibnrazaaq@unimaid.edu.ng
Computer Engineering Department, University of Maiduguri, Borno, Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 13(1), MARCH, 2025
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

identified solutions. This helps prevent local minima and premature convergence, increasing the likelihood of finding the global optimum. NPOA's dynamic exploration ensures thorough coverage of the search space, making it ideal for optimizing CNN hyperparameters for enhancing model performance.

**Hyperparameter Space Definition**

In the process of optimizing convolutional neural networks (CNNs), Table 1 Depicts the search space for the hyperparameter that was used for the training process CNN model to learn from the data.

**Table 1:** Hyper-parameters CNN search space

| | Parameter | Value |
|---|---|---|
| 1 | Learning rate | (1e-5, 1e-2) |
| 2 | Dropout rate | (0.2, 0.5) |
| 3 | Number of Filters | (8,128) |
| 4 | Dense Units | (10,500) |
| 5 | Batch size | (16, 64) |

A flowchart for implementing of the optimized CNN hyperparameter using Nomadic Pastoralist Optimization Algorithm (NPOA) is presented in figure 2. NPOA iteratively adjusts the position of the pastoralist representing the CNN hyperparameter sets to find the best-performing configuration. Starting with randomly initialized sets of hyperparameters, NPOA evaluates each set's performance (fitness) on the CNN model. It then updates positions based on fitness scores, where better-performing configurations guide others.

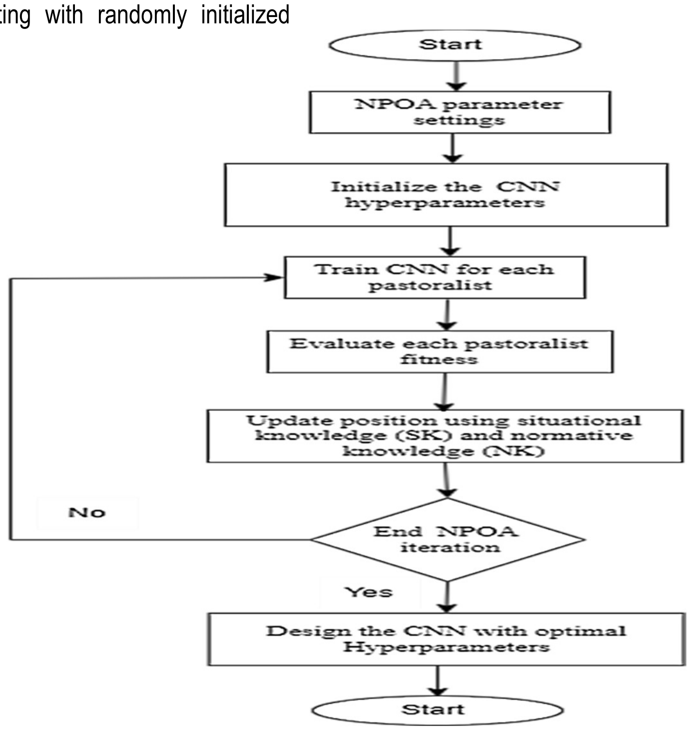


Figure 2: Flowchart for implementing NPOA for CNN Hyperparameter Optimization

Corresponding author: Abdulrazaq, A. A.
ibnrazaaq@unimaid.edu.ng
Computer Engineering Department, University of Maiduguri, Borno, Nigeria.

The initialization parameter of NPOA is set as shown in table 2. The initialization parameters in NPOA are set to defines the starting points for the search process. These parameters determine the initial population's diversity, impacting the algorithm's ability to explore the search space effectively.

Table 2: NPOA Parameter Initialization

| | Parameter | Value |
|---|---|---|
| 1 | Number of Pastoralists | 30 |
| 2 | Number of max_iteration | 1000 |
| 3 | Scout Rate | 5 |
| 4 | Split Rate | 5 |
| 5 | Camp radius | 0.01 |
| 6 | Number of Runs | 10 |

The optimization begins by setting up a population of pastoralists, each representing a candidate configuration of CNN hyperparameters (e.g., learning rate, filter size, number of filters, and dropout rate). The configurations of the CNN hyperparameter are also initialized as presented in table 3

Table 3: CNN Hyperparameter Configuration

| Parameter | Value(s) |
|---|---|
| Optimizer | Adam |
| Activation function | ReLU |
| Loss function | Cross entropy |
| Epoch Number | [10, 50] |
| Number of conv. Layers | [2, 3, 4] |
| Metrics | Accuracy |

Certain parameters are fixed while others are set within the range to reduce the number of parameters (weights) and computational burden. Further, we utilized the ReLU (rectified linear unit) activation function and employed the Adam optimizer. Cross-entropy loss is used due to its effectiveness in measuring the dissimilarity between predicted and actual class probabilities. The objective during training is to minimize this loss function, thereby enhancing prediction accuracy. As the CNN iteratively updates its weights and biases in response to the calculated loss, it gradually learns to make more accurate predictions. The loss function is given by:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i;\theta), y_i) \qquad (5.12)$$

Where:
$N$ is the number of samples in the training set.
$x_i$ is the input image.
$y_i$ is the true label for the image $x_i$.
$f(x_i;\theta)$ is the output of CNN model, which depends on input image $x_i$ and the parameters $(\theta)$.
$\ell(\cdot)$ is the loss function, often cross-entropy for classification tasks.

## RESULTS AND DISCUSSION

### Training and Validation Result

The training and validation accuracy and loss results are essential for evaluating how well your CNN model has learned after optimization using the Nomadic Pastoralist Optimization Algorithm (NPOA).

Figure 3 depicts the training and validation accuracy results. This plot is used to track the accuracy of the model during training and on unseen validation data across epochs.

Corresponding author: Abdulrazaq, A. A.
ibnrazaaq@unimaid.edu.ng
Computer Engineering Department, University of Maiduguri, Borno, Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 13(1), MARCH, 2025
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

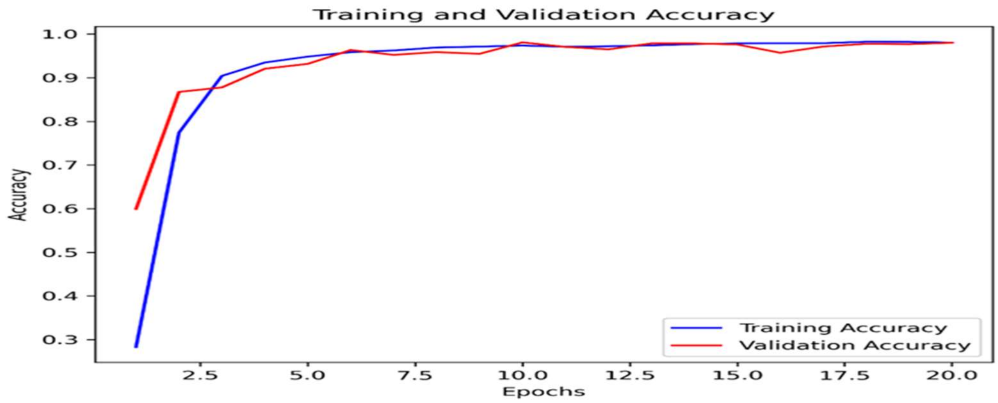


Figure 3: Training and validation accuracy results against numbers of epochs

Based on the results shown in Figure 6.13, which illustrates training and validation accuracy across epochs, we observe a clear progression. The training was conducted over 20 epochs using the training dataset. At the end of the first epoch, training accuracy and validation accuracy were at 25.0% and 60.0%, respectively. By the fifth epoch, these values rose to 94% for training accuracy and 91% for validation accuracy. Finally, by the 20th epoch, training accuracy reached 98.1%, and validation accuracy reached 98%. The steady increase in validation accuracy suggests that the model is effectively learning meaningful patterns from the data. Similarly, the rise in training accuracy without signs of overfitting indicates that the model is generalizing well. Thus, the consistent improvement in both training and validation accuracy reflects that the NPOA has successfully optimized the hyperparameters, enhancing the model's overall performance and its ability to generalize across unseen data.

The training and validation loss results the loss (error) during training and validation. Loss is a measure of how well the model's predictions match the actual outcomes, with lower values indicating better predictions. Figure 4 depicts training and validation loss plots.

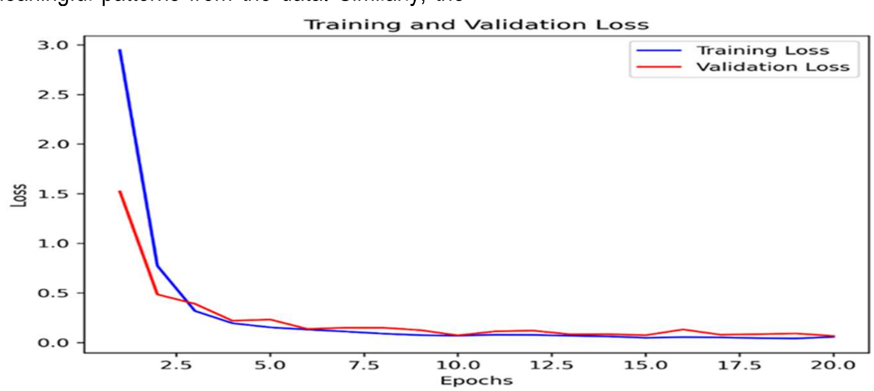


Figure 4: Training and validation loss results against numbers of epochs

According to the results in Figure 4, training was conducted over 20 epochs using the training dataset. At the end of the first epoch, the training loss and validation loss were 2.80 and 1.45, respectively. By the fifth epoch, these values had decreased to 0.24 for training loss and 0.50 for validation loss. Finally, by the 20th epoch, training loss reached 0.21, and validation loss

Corresponding author: Abdulrazaq, A. A.
ibnrazaaq@unimaid.edu.ng
Computer Engineering Department, University of Maiduguri, Borno, Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 13(1), MARCH, 2025
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

declined to 0.21. These results indicate a consistent decrease in both training and validation losses, suggesting that the model is learning efficiently without signs of overfitting. The progressive reduction in validation loss confirms that the hyperparameter optimization by NPOA has resulted in a model that not only fits the training data well but also generalizes effectively to new data.

**NPOA-CNN Best Parameter Space after Optimization**

The optimized values of the CNN-hyperparameters considered is presented in Table 4 using 4-convolutional layers to show how the model is configured to balance learning speed, complexity, and generalization ability.

Table 4: Best Hyperparameters obtained after Training

| s/n | Best Hyperparameter | |
|---|---|---|
| 1 | Learning rate | 0.00046 |
| 2 | Number filters | 32.0 |
| 3 | Dropout rate | 0.2558 |
| 4 | Dense units | 226 |

The small learning rate of 0.00046 indicates a careful and incremental approach to updating the model weights during training. This ensures stability and prevents overshooting the minima of the loss function, especially in deeper architectures. Such a low value is ideal for achieving precise weight updates, avoiding convergence issues like oscillations or divergence. The number of convolutional filters determines the model's ability to capture features at each layer. A value of 32 is a moderate choice, providing a balance between computational efficiency and feature extraction capability. This ensures the network captures sufficient patterns in the input images without excessive computational overhead or risk of overfitting.

The dropout rate of 0.2558 which is about 25.6% effectively mitigates overfitting by randomly deactivating a fraction of neurons during training. This value is optimal for deeper models, allowing them to generalize better by reducing reliance on specific nodes. It supports robust feature learning without overly weakening the network. The dense layer of 226 units reflects an optimal capacity for capturing and combining high-level features learned from the convolutional layers. This value ensures sufficient representational power to classify the images effectively while maintaining computational efficiency.

The implication of this results interprets that the optimized CNN architecture ensures robust performance for image classification tasks while minimizing computational resources and training instability. The model is well-suited for datasets that require capturing intricate patterns without the risk of overfitting such as age-invariant face recognition.

**Comparative Analysis of Developed POA-CNN with Different Models**

A comparison of the NPOA-CNN method with existing CNN techniques to include VGG-16, CNN without optimization and optimized genetic algorithm-CNN are shown in Table 6.6

Table 5: Performance Comparison NPOA-CNN Different CNN Models

| | Method | Accuracy (%) |
|---|---|---|
| 1 | VGG-16 (Sikha and Bharath, 2022) | 85.002 |
| 2 | CNN without optimization | 60.00 |
| 3 | GA-CNN (Karlupia *et al*., 2023) | 94.50 |
| 4 | **Proposed NPOA-CNN** | **98.82** |

*Corresponding author: Abdulrazaq, A. A.*
*ibnrazaaq@unimaid.edu.ng*
*Computer Engineering Department, University of Maiduguri, Borno, Nigeria.*

Table 5 shows that the proposed NPOA-CNN outperforms other methods in terms of accuracy. This difference in performance highlights the impact of optimized CNN hyperparameters, which enable more precise tuning and improved model effectiveness. For the VGG-16 model, it achieved an accuracy of 85% on age-invariant feature extraction. While VGG-16 with known deep architecture generally supports detailed feature extraction, it may not be fully suited for facial recognition task without specific tuning, as its standard architecture is not tailored towards directly to face recognition task.

Also, the base CNN model without any optimization yielded the lowest accuracy of 60%. This suggests that without optimized hyperparameters, the model lacks the capacity to focus on age-invariant features effectively. Age-invariant facial recognition is inherently challenging, and the unoptimized CNN struggles to generalize across varying age representations. However, the Genetic Algorithm (GA)-optimized CNN reached a significantly higher accuracy of 94.5%, showing that GA-driven hyperparameter tuning enabled the CNN to capture age-invariant features more effectively. GA's iterative selection process helped in finding settings that improved the model's robustness to age variation, leading to notable improvements in performance. On the other hand, the NPOA-optimized CNN achieved the highest accuracy of 98.82%, reflecting excellent performance in age-invariant feature extraction. This high accuracy indicates that NPOA's hyperparameter search strategy, balancing exploration and exploitation, was particularly effective in finding parameter settings that enhance face representation. The 98.82% accuracy demonstrates that NPOA was able to fine-tune the model's learning process, allowing it to focus on salient features across different facial representations.

## CONCLUSION

This study implements the Nomadic Pastoralist Optimization Algorithm (NPOA) for hyperparameter optimization of a Convolutional Neural Network (CNN) model applied to face recognition. As the complexity of identifying the optimal parameter set increases with the number of hyperparameters, metaheuristic optimization techniques are employed to efficiently explore the search space and determine the best parameter combinations. The research provides an in-depth experimental analysis of an NPOA-optimized CNN, demonstrating its effectiveness in enhancing the face recognition system. The results reveal that the proposed approach achieves a reliable and optimized CNN-based face recognition system, with performance evaluations indicating a significant improvement compared to existing methods. Future work could explore the integration or comparison with alternative metaheuristic algorithms to further refine optimization and improve system performance.

## REFERENCES

Abdullahi, I. M., Mu'azu, M. B., Olaniyi, O. M., & Agajo, J. (2019). A novel cultural evolution-based nomadic pastoralist optimization algorithm (NPOA): the mathematical models. 2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf),

Abdullahi, I. M., Mu'azu, M. B., Olaniyi, O. M., & Agajo, J. (2021). Pastoralist Optimization Algorithm (POA): A Culture-Inspired Metaheuristic for Uncapacitated Facility Location Problem (UFLP). Hybrid Intelligent Systems: 20th International Conference on Hybrid Intelligent Systems (HIS 2020), December 14-16, 2020,

Agarwal, R., Jain, R., Regunathan, R., & Pavan, K. C. S. (2019). Automatic attendance system using face recognition technique. Proceedings of the 2nd International Conference on Data Engineering and Communication Technology: ICDECT 2017,

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Laith Farhan, L.

Corresponding author: Abdulrazaq, A. A.
ibnrazaaq@unimaid.edu.ng
Computer Engineering Department, University of Maiduguri, Borno, Nigeria.

(2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal on Big Data*, *3*(53), 1-72. https://doi.org/10.1186/s40537-021-00444-8

Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., & Tuba, M. (2020). Optimizing CNN hyperparameter by enhanced swarm intelligence metaheuristics. *Algorithms*, *13*(67), 1-33. https://doi.org/10.3390/a13030067

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).

Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S. Z., & Hospedales, T. (2015). When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. Proceedings of the IEEE international conference on computer vision workshops,

Ilyas, B. R., Mohammed, B., Khaled, M., &, & Miloud, K. (2019). Enhanced face recognition system based on deep CNN. 2019 6th International Conference on Image and Signal Processing and their Applications (ISPA),

Karlupia, N., Mahajan, P., Abrol, P., & Lehana, P. K. (2023). A genetic algorithm based optimized convolutional neural network for face recognition. *International Journal of Applied Mathematics and Computer Science*, *33*(1).

Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. *Sensors*, *20*(2), 342.

Mikolaj, W., Zaneta, S., Gertych, S. K., & Arkadiusz, G. (2024). Improving classification accuracy of fine turned CNN model impact of hyperparameter Optimization. *Heliyon 10*, 1-11. https://doi.org/10.1016/j.heliyon.2024.e26586

Mittal, S., & Patel, S. (2023). Age Invariant Face Recognition Techniques: A Survey on the Recent Developments, Challenges and Potential Future Directions. *International Journal of Engineering Trends and Technology 71* (5), 435-460. https://doi.org/https://doi.org/10.14445/22315381/IJETT-V71I5P243

Nurhopipah, A., & Larasati, N. A. (2021). CNN Hyperparameter optimiztion using random grid coarse to fine seach for face classification. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *6*(1), 19-26. https://doi.org/10.22219/kinetik.v6i1.1185

Vose, A., Balma, J., Heye, A., Rigazzi, A., Siegel, C., Moise, D., Robbins, B., & Sukumar, R. (2019). Recombination of artificial neural networks. *arXiv preprint arXiv:1901.03900*.

Wang, Y., Zhang, H., & Zhang, G. (2019). cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. *Swarm and Evolutionary Computation*, *49*, 114-123.

Wu, J., Chen, X., Zhang, H., Xiong, L., Lei, H., & Deng, S. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26-40.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295-316.

Zulfiqar, M., Syed, F., Khan, M. J., & Khurshid, K. (2019). Deep face recognition for biometric authentication. 2019 international conference on electrical, communication, and computer engineering (ICECCE).

*Corresponding author: Abdulrazaq, A. A.*
*ibnrazaaq@unimaid.edu.ng*
*Computer Engineering Department, University of Maiduguri, Borno, Nigeria.*

*Corresponding author: Abdulrazaq, A. A.*
*ibnrazaaq@unimaid.edu.ng*
*Computer Engineering Department, University of Maiduguri, Borno, Nigeria.*