International Journal of Computer Sciences and Engineering
Vol.13, Issue.8, pp.30-41, August 2025
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

Research Article

A Convolution Neural Network, Particle Swarm Optimization Hybrid Model for Scripting Language Handwritten Character Recognition

Saeed Ur Rahman[1], Muneeb Ullah[2*], Nisar Ullah[3], Muhammad Faizan Khan[4], Irzum Shafique[5]

[1]School of Computer Science and Technology Xidian University Xi'an, Shaanxi, China
[2]School of Electronic Engineering, Xidian University, Xi'an, Shaanxi, China
[3]School of Optoelectronic Engineering, Xidian University, Xi'an, Shaanxi, China
[4]School of Artificial Intelligence Xidian University Xi'an, Shaanxi, China
[5]School of Computer Science and Technology, Xidian University, Xi'an, China

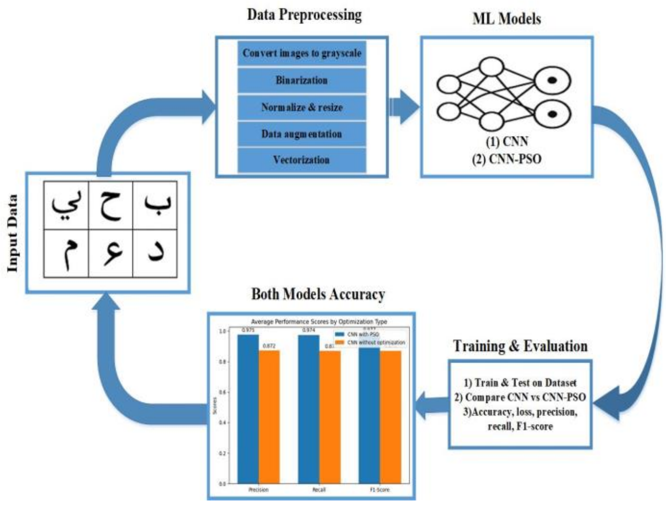
**Corresponding Author:*

Received: 26/Jun/2025; **Accepted**: 28/Jul/2025; **Published**: 31/Aug/2025. **DOI**: https://doi.org/10.26438/ijcse/v13i8.3041

Abstract: In the realm of character recognition, the availability of comprehensive and relevant datasets is essential for developing accurate and robust models. In this research, we address the dearth of datasets for Pashto handwritten character recognition by introducing a novel and extensive dataset, previously unavailable in the field. The absence of a Pashto dataset posed a significant challenge to developing effective recognition models, making this dataset a valuable contribution to the research community. To optimize the recognition process, we put forward a CNN-PSO (Convolutional Neural Network–Particle Swarm Optimization) based hybrid approach. The CNN component is utilized for extracting features, while PSO is applied for parameter optimization. By incorporating PSO, the aim is to strengthen the model's capacity to accurately recognize and classify handwritten Pashto characters. To validate this approach, we compare the CNN-PSO framework with a standard CNN baseline. The outcomes clearly indicate that the hybrid design surpasses the performance of the CNN-only model in Pashto handwritten character recognition. The proposed research findings reveal the potential of hybrid models in character recognition tasks and underline the significance of utilizing the Pashto dataset in advancing research in this domain. This study aids in building recognition systems for Pashto that are more precise and effective, with possible applications in OCR technologies and language processing for Pashto-speaking regions.

Keywords: Data analysis, machine learning, deep learning, data science, data compression, data mining

Graphical Abstract- This graphical abstract representation delineates a comparative machine learning process for image categorization. The procedure commences with data preprocessing steps such as grayscale conversion, binarization, normalization, resizing, augmentation, and vectorization, then followed by modeling with both a conventional CNN and a hybrid CNN-PSO (Particle Swarm Optimization) model. The results juxtapose the mean performance scores of both models, presenting metrics including accuracy, precision, recall, and the F1-score. The assessment underscores the disparity in performance between the optimized (CNN-PSO) and non-optimized CNN models.



1. Introduction

Character recognition, a fundamental aspect within computer vision and NLP, has significantly advanced with the rise of deep learning techniques and the availability of diverse datasets [1],[2]. Accurate character recognition models are crucial for applications ranging from optical character recognition systems to language processing in specific linguistic contexts [3],[4]. However, the performance of these systems is strongly dependent on both the quality and variety of the training datasets [5],[6]. This research addresses a critical gap in character recognition, particularly for Pashto handwritten characters. Pashto, an Indo-European language spoken by millions in Afghanistan and Pakistan, has a distinct script that requires specialized recognition tools [7]. The absence of a comprehensive Pashto handwritten character dataset has hindered the development of accurate recognition systems. To fill this gap, we introduce a novel and extensive dataset for Pashto handwritten characters. To optimize the recognition process, we propose a hybrid CNN-PSO (Convolutional Neural Network-Particle Swarm Optimization) model. This model combines CNNs' feature extraction capabilities with PSO's optimization strengths to enhance accuracy in recognizing Pashto handwritten characters. We evaluate the hybrid model’s performance against a standard CNN, demonstrating the superiority of the CNN-PSO hybrid model in this domain [8],[9]. This hybrid model strategically utilizes CNNs for extracting features while employing PSO techniques to optimize the network, enhancing the ability to identify and classify Pashto handwritten characters accurately. The central objective of the proposed research is to rigorously assess the efficacy of the proposed CNN-PSO hybrid model. We plan to conduct a comparative analysis, pitting this hybrid model against a simpler CNN model. Through extensive experimentation and rigorous evaluation, the proposed study’s aim is to confirm the improved results achieved by the CNN-PSO hybrid model in Pashto handwritten recognition. This research holds wide-reaching implications, as accurate Pashto handwritten character recognition has practical applications in optical character recognition, document processing, and language-related tasks in Pashto-speaking regions. The study serves as a pivotal contribution, enhancing the development of more accurate and efficient recognition systems for Pashto characters, ultimately benefiting various fields by facilitating effective communication and data processing in Pashto-scripted content. The main contributions of this research can be outlined as:

1. Novel Pashto Handwritten Character Dataset: This research introduces a novel and extensive dataset of Pashto handwritten characters. Prior to this study, there was a dearth of comprehensive Pashto character datasets, hindering the development of accurate recognition systems for this unique script. This dataset provides a valuable resource to the research community, filling a critical gap in character recognition datasets and offering researchers the essential materials to develop accurate Pashto character recognition systems.

2. Hybrid Model for Character Recognition: The paper presents a novel CNN-PSO (Convolutional Neural Network-Particle Swarm Optimization) hybrid model specifically designed for character recognition tasks, with a focus on Pashto handwritten characters. This model employs the feature extraction strength of CNNs along with the optimization capacity of PSO. The integration of PSO represents a creative method to boost recognition accuracy and efficiency, particularly in the context of Pashto script.

3. Comparative Analysis: This work performs a detailed evaluation between the proposed CNN-PSO hybrid model and a simpler CNN model. Through extensive experimentation and thorough evaluation, the research demonstrates that the hybrid CNN-PSO achieves higher recognition accuracy in the recognition of Pashto handwritten characters. This comparative analysis highlights the effectiveness of hybrid models in character recognition tasks and highlights the efficacy of the model introduced in this paper.

4. Practical Implications: Accurate Pashto handwritten character recognition is of significant practical importance. It finds applications in optical character recognition systems, document processing, and language-related tasks in Pashto-speaking regions. The paper’s contribution facilitates effective communication and data processing in Pashto-scripted content, benefiting various fields, including linguistics, data analysis, and language technology.

2. Related Work

The area of machine learning, especially in handwritten character recognition, has seen significant advances through the use of Convolutional Neural Networks (CNNs) together with various optimization algorithms. This chapter reviews the literature surrounding the application of CNNs in handwriting recognition and their enhancement through hybridization with optimization techniques such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA). CNNs have become the leading method in recognizing handwritten cursive scripts, recommended by numerous researchers for their effectiveness [10]-[11]. These models can recognize characters without prior knowledge of the language's grammatical structure, showcasing their versatility in both image based and language-processing tasks [12]-[13]. However, Pashto character recognition presents unique difficulties arising from the script complexity of the language and the lack of comprehensive datasets. Previous research efforts have primarily focused on character recognition for languages like Arabic and Urdu, which share some similarities with Pashto [14]. For instance, LSTM-based re-current neural networks (RNNs) have demonstrated remarkable recognition capabilities for Arabic and Urdu languages [15]. Yet, these approaches cannot be directly transferred to Pashto because of differences in script and character forms. Specific attempts to recognize Pashto characters include in [35] assessment of the BBN Byblos OCR system. In [36] development of a Pashto ligature recognition system combining SIFT and Principal Component

Analysis (PCA). Despite these efforts, the absence of a dedicated Pashto dataset has limited the progress in this field. Our proposed research addresses these challenges by introducing a novel Pashto handwritten character dataset, addressing the dearth of resources in this domain. The dataset's development involved a meticulous crowdsourcing approach, ensuring its comprehensiveness and diversity. Additionally, we propose a hybrid CNN-PSO model, leveraging CNNs' strength in feature extraction and PSO's optimization capabilities. This integration uniquely addresses the optimization of the number of filters in CNN layers, enhancing the model's accuracy and efficiency in recognizing Pashto characters. Through a comparative analysis with a simpler CNN model, we demonstrate the superior performance of our hybrid approach, highlighting its potential to advance Pashto handwritten character recognition.

In recent years, recurrent neural networks (RNNs) built on LSTM units have shown impressive recognition performance in the context of Arabic and Urdu scripts. Rashid et al. [37] proposed a framework that could handle printed Arabic text across different fonts, sizes, and even low-resolution inputs, making use of a multidimensional RNN architecture. This system achieved recognition accuracies exceeding 99% at both the character and word levels. Likewise, similar LSTM-driven approaches have been applied to Urdu recognition [38], exploring two distinct scenarios. In the first scenario, each character received a unique class label based on its relative placement in a ligature or word. In the second case, characters were assigned a single class label, independent of their visual shape or position. The resulting error rates were recorded at 13.57% and 5.15% for these scenarios, respectively. Additionally, a Feed Forward Neural Network (FFNN) was employed for Urdu script classification, trained and validated on attributes such as second-order moments, hole count, solidity, aspect ratio, normalized segment length, eccentricity, and curvature [39][40]. This resulted in recognition rates of 100% for one case and 70% for the other. Pal et al. [41] presented a scale-invariant technique for Urdu script identification using concepts of water reservoirs and contour topology, achieving 97.8% accuracy in recognizing basic characters and numerals. Furthermore, Sabbour and Shafiat et al. [42] explored the application of contour-based shape context for Arabic and Urdu recognition, employing a k-nearest neighbor (k-NN) model. Their method, trained with over 10,000 samples, reached a word-level accuracy of 91%.

Optimization algorithms like GA and PSO have been employed to refine CNN architectures, particularly in the context of handwriting recognition. Genetic Algorithms have been applied to tune the structure of the network and its hyperparameters, such as learning rates and filter sizes, leading to enhanced classification accuracies [16]. Similarly, PSO has been effective in adjusting CNN parameters and configurations, particularly in deciding the filter count and kernel dimensions within convolutional layers [17] [18] [19]. The literature indicates varied applications of these methodologies. For instance, region-based models with optimization algorithms have shown better classification accuracy in identifying similarities in Quranic content compared to traditional models [20]. Furthermore, supervised learning techniques based on zoning models have been developed for Arabic letter recognition [21] [22], while recurrent Neural Networks (RNN) were utilized effectively for Urdu Nastaliq text detection [23] [24]. In handwritten character recognition, the combination of CNN with PSO (CNN-PSO) models has been a focus of recent studies. The hybrid model uses PSO to determine the most suitable number of filters for each CNN convolutional layer, leading to models that are both efficient and effective in recognizing a wide range of characters [25-[27]. This approach has shown promising results, especially in languages with complex scripts like Urdu, Persian, and Arabic [21]. While the integration of CNNs with optimization algorithms offers enhanced performance, it also introduces challenges like increased computational complexity and the need for ensuring model generalization [15] [16]. Future research is directed toward developing hybrid algorithms that balance optimization efficiency with computational feasibility [17].

PSO, first introduced in [28], draws inspiration from natural phenomena such as the flocking of birds, swarming of fish, and the behavior of bee colonies, which has been adapted into algorithms that support metaheuristic optimization in real-life scenarios [29]. Since its inception, the PSO algorithm has undergone numerous modifications, resulting in a variety of enhanced versions [30]. In recent years, several dynamic and adaptive models have been introduced, specifically tailored to improve decision-making optimization [30]. Among these, certain iterations have emerged as 'standard' owing to their exceptional performance in achieving optimization results in practical applications [30]. In the PSO framework, particles adjust their trajectories as they respond to environmental shifts [29]. They navigate through the solution space, realigning towards the most favorable outcomes by referencing the best-known positions [29]. Furthermore, reference [30] highlights experiments with a limited set of global optimizations addressing human-centric issues. These experiments utilized various topologies to structure particle interactions, enhancing information dissemination across the group. Such configurations enable individuals within the swarm to benefit from the successes of the highest performers [30]. Although PSO and similar metaheuristic algorithms have recently proliferated, there remain numerous opportunities for further refinement and exploration of optimization techniques. For instance, the Pendulum Search Algorithm (PSA) [32] represents another approach in the realm of population-based optimization strategies, with recent studies suggesting that PSA may surpass PSO ineffectiveness [29]. The research outlined in [31] introduced a dynamic multi-swarm (DMS-PSO) based approach designed to enhance the selection of critical attributes for diagnosing heart diseases through medical analysis. The integration of fuzzy logic with DMS-PSO proved effective in increasing the precision and reliability of diagnostic outcomes. Experimental results suggest that DMS-PSO outperforms current systems used in healthcare and manual diagnostics, and it is anticipated to deliver more dependable outcomes in practical applications. Additionally, reference [33] discusses the development of an enhanced swarm optimization method that incorporates Newtonian mechanics, called Centripetal Accelerated Particle Swarm Optimization (CAPSO). This approach is designed to speed up both learning and convergence in classifiers. Evaluations on nine medical diagnosis benchmarks demonstrated its superiority in terms of faster convergence and higher classification accuracy. Within the PSO framework, each particle operates autonomously, navigating the solution space through personal experience and collective interactions, which mirrors the behavior of animals in nature [30]. It is essential to evaluate the appropriateness of the algorithm for particular tasks. PSO is known for its robustness, flexibility across various application settings, and powerful distributed capabilities that facilitate rapid convergence towards optimal solutions [34]. Recent enhancements in computational algorithms have improved the efficiency, effectiveness, and resilience of PSO. Consequently, this has opened up a range of potential applications that could exploit PSO for enhanced outcomes. The literature review highlights the significant advancements in handwriting recognition through the use of CNNs and the further enhancement of these models through optimization algorithms like PSO and GA. These hybrid models not only demonstrate superior performance in traditional tasks but also show promise in handling more complex, script-specific recognition tasks. Continuous exploration in this domain is vital to build handwriting recognition systems that are increasingly efficient, accurate, and dependable.

3. Materials and Methods

In this section, we outline the approach illustrated in Figure 1 for this study, covering data gathering, data preprocessing, and the architecture of our suggested CNN-PSO framework.

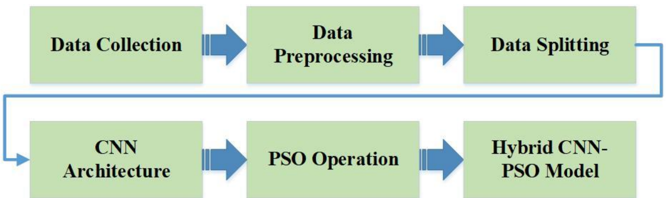


Figure 1. Research Flow diagram

Figure 2. Sample form of Data Collection

3.1. Data Collection

The process of gathering data for this research involved a systematic approach to collect handwritten Pashto characters from a diverse group of individuals.

3.2. Proposed Dataset

The dataset created for this research project is an innovative contribution to Pashto language studies, addressing a longstanding need for comprehensive resources in this field. Comprised of 15,050 individual character images, this dataset is the culmination of extensive collaboration involving 350 university students proficient in Pashto. Through a meticulous crowdsourcing approach, each participant was tasked with handcrafting the 43 distinct Pashto characters on standardized A4-sized papers, as shown in Figure 2. Subsequently, these handwritten pages underwent precise digitization, facilitated by high-resolution scanning at 200 dpi. The images were then meticulously processed using GIMP 2 software, employing sophisticated segmentation techniques to isolate each character. This meticulous approach ensured the dataset's richness and diversity, capturing the nuances of Pashto handwriting styles with unparalleled accuracy.

3.3. Data Preprocessing

Following the phase of collecting data, the next important step focused on preparing the data. Data preparation is an essential technique used to refine data, eliminate noise, and select the most suitable samples. For this purpose, we utilized MATLAB and Python, along with the OpenCV and Pillow libraries. The data points, represented as images, underwent the following preprocessing procedures:

Initially, the images, varying in size and captured in a three-channel RGB format, underwent dimension reduction and color channel reduction to achieve a uniform, single-channel format. This process was crucial to prevent computational overhead from complex network architectures. Further, the images were converted to binary using MATLAB's binaries () method, with an established threshold value of 120, to distinguish between black and white pixels. Additionally, the image pixel values were converted to double-precision numbers, as neural networks work best when input values range from 0 to data augmentation, another critical step, was employed to account for natural variations in handwritten characters. Using OpenCV and Pillow libraries, the images were randomly rotated by up to 10°, resulting in a diverse dataset of 43,000 images. Each image was then transformed into a vector representation, creating a two-dimensional matrix dataset. A corresponding label matrix was also generated, with each label representing the class of the image. This comprehensive dataset, normalized to a fixed dimension of 224 × 224 pixels, comprised 4,300 images in a matrix sized 4,300 × 784, alongside a label matrix sized 4,300 × 43. This dataset creation was a significant step towards achieving the first research objective of developing a Pashto handwritten dataset, previously unavailable in this format.

3.4. Data Splitting

We partitioned the dataset into training and test subsets using a 70-30 ratio, providing an adequate amount of data for model training and assessment.

3.5. Hybrid CNN-PSO Architecture

The proposed research methodology focuses on developing a hybrid Convolutional Neural Network (CNN) guided by Particle Swarm Optimization (PSO) for the task of handwritten character recognition. The proposed model, the CNN-PSO hybrid, is a novel approach to character recognition. It combines the power of Convolutional Neural Networks (CNNs) alongside Particle Swarm Optimization (PSO) for feature extraction and model optimization. The integration of a CNN with a PSO-based algorithm offers a new strategy within machine learning, particularly for complex tasks like handwritten character recognition with 43 classes. In this architecture, the CNN is used for feature extraction and classification, while PSO is applied to fine-tune the number of filters in the CNN layers, improving the model's overall performance.

3.5.1. Convolutional Neural Network (CNN)

This architecture represents a CNN model designed for image classification. It takes 28x28-pixel grayscale images as input and passes them through convolutional, max-pooling, and fully connected layers to make predictions for different classes. The parameters in each layer are determined by the filter dimensions and the number of neurons in that layer. This architecture is commonly used for tasks Pashto handwritten character recognition. The architectural details are summarized in Table 1 and figure 3.

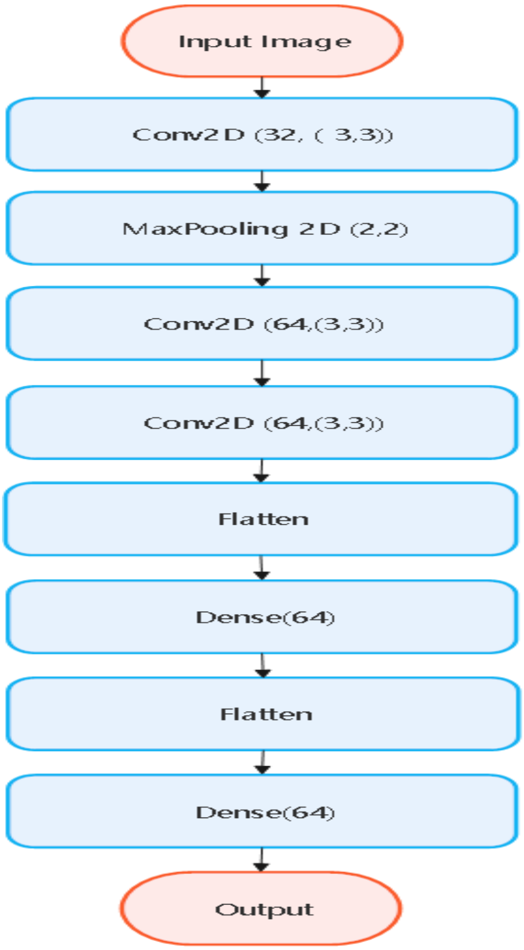


Figure 3. CNN layer Architecture

Table 1. CNN Architectural Details

Layer (type)	Output Shape	Param #
Input Image	(28, 28, 1)	0
Conv2D (32, (3, 3))	(26, 26, 32)	320
MaxPooling2D (2, 2)	(13, 13, 32)	0
Conv2D (64, (3, 3))	(11, 11, 64)	18496
MaxPooling2D (2, 2)	(5, 5, 64)	18496
Flatten	1600	0
Dense (64)	(64)	0
Dense (Number of Classes)	(Number of Classes)	102464

3.5.2. Input Image

This is the initial layer that takes an image with dimensions of 28 pixels in width, 28 pixels in height, and 1 channel (grayscale).

3.5.3. Conv2D

The initial convolutional layer uses 32 filters of 3x3 dimensions on the input image. It produces an output feature map with dimensions 26x26 and a total of 32 feature channels. This layer has 320 parameters to be learned.

3.5.4. MaxPooling2D

The first max-pooling layer decreases the feature map size by selecting the maximum value within each 2x2 block. It produces an output feature map with dimensions 13x13 and still has 32 feature channels. No parameters are learned in this layer.

3.5.5. Conv2D

The second convolutional layer utilizes 64 filters of size 3x3 on the output from the preceding max-pooling layer. It produces an output feature map with dimensions 11x11 and a total of 64 feature channels. This layer has 18,496 parameters to be learned.

3.5.6. MaxPooling2D

The second max-pooling layer further shrinks the feature map size. It functions similarly to the first max-pooling layer but processes the 64 feature channels. It produces an output feature map with dimensions 5x5 and still has 64 feature channels. No parameters are learned in this layer.

3.5.7. Flatten

The flatten layer converts the 2D feature map into a 1D array. Here, it changes the 5x5x64 feature map into a vector of 1,600 elements. This layer does not have any parameters.

3.5.8. Dense

The first dense layer consists of 64 units and applies the Rectified Linear Unit (ReLU) activation function. It facilitates feature learning and provides non-linearity to the network. This layer has 102,464 parameters to be learned.

3.5.9. Input Image

This is the initial layer that takes an image with dimensions of 28 pixels in width, 28 pixels in height, and 1 channel (grayscale).

3.6. Hybrid CNN-PSO Optimization

Particle Swarm Optimization (PSO) is a computational technique that iteratively refines candidate solutions according to a defined quality metric, called the fitness function. Within Convolutional Neural Networks (CNN), PSO is used to identify the best number of filters for each convolutional layer. In this case, the fitness function measures the CNN's classification accuracy on a validation dataset, which evaluates the effectiveness of each potential filter configuration represented by the particles in the swarm.

The process starts with setting up the PSO, where a collection of particles—each corresponding to a distinct filter configuration in the CNN layers—is initialized. The starting range of filters, as shown in Table 2, is based on empirical evidence and heuristic understanding..

3.6.1. Filter Sizes
The chosen filter sizes (e.g., 3x3) are based on common practices in CNN architecture, proven effective for capturing local image patterns. These sizes strike an equilibrium between computational cost and the capacity to extract meaningful features.

3.6.2. Number of Layers
The convolutional layers are determined to gradually capture higher-level features from the input images. Early layers detect basic elements such as edges and textures, while deeper layers recognize more intricate structures.

3.6.3. Initial Filter Range
The initial range for the number of filters (e.g., [16, 32, 64, 96]) is based on previous studies and practical experience, which suggest that adding more filters in the deeper layers allows the network to capture finer details.

Each particle's performance is then evaluated by training the CNN using the training data and assessing its accuracy on the validation dataset. Based on these results, the particles adjust their configurations by updating their velocities and positions, aiming to converge toward the optimal filter arrangement. This iterative process of evaluation and adjustment repeats until a predetermined stopping condition is reached, such as a specific number of iterations or an acceptable performance level. The configuration of the best-performing particle at the end of these iterations is adopted as the ideal number of filters in each CNN layer. Subsequently, the CNN is structured according to this optimal configuration and trained on the dataset of handwritten characters, ensuring that the model is both computationally efficient and capable of effectively extracting and learning features.

Table 2. Initial Range for Number of Filters in Each Layer

Layer	Hyper parameter	Range
1st Convolutional Layer	Number of Filters ($Filters_1$)	[16, 32, 64, 96]
2nd Convolutional Layer	Number of Filters ($Filters_2$)	[48, 64, 96, 128]
3rd Convolutional Layer	Number of Filters ($Filter_3$)	[64, 96, 128]

By using PSO to optimize the number of filters, CNN can potentially maintain an optimal trade-off between computational cost and effective feature extraction, resulting in enhanced performance for tasks like image classification, object detection, and more. The PSO-driven approach to optimizing hyperparameters like the number of filters demonstrates a practical application of evolutionary computation techniques in fine-tuning deep learning models. The proposed model Pseudo Code are given in below table 3.

Table 3. Proposed Model Pseudo Code

Pseudocode
Take the input image.
Initialize CNN model parameters: weights (W) and biases (B).
Apply the first convolution operation (C1) to the input:
$Z1 = W1 \times X + B1$,
where X is the input image, W1 is the filter weights, and B1 is the bias.
Perform max-pooling to reduce spatial dimensions.
Apply the second convolution operation (C2) to the output of C1:
$Z2 = W2 \times A1 + B2$,
where A1 is the output of the first convolutional layer.
Perform max-pooling again.
Apply Rectified Linear Unit (ReLU) activation function to replace negative pixel values with zero:
$A2 = \max(0, Z2)$
Flatten the output from the previous layer.
Connect the flattened output to fully connected layers with weights (W_fc) and biases (B_fc).
Apply activation functions (e.g., ReLU) to the fully connected layers.
Output the result of the final layer.
Randomly initialize CNN model parameters: weights (W) and biases (B).
Evaluate model performance using a fitness function.
While the termination condition is not met:
a. If fitness(xi) > fitness(gbest)
gbest = xi
b. If fitness(xi) > fitness(pbest)
pbest = xi
c. Update particle velocity using the formula:
$Vi = Vi + c1 \times rand() \times (pbest - xi) + c2 \times rand() \times (gbest - xi)$
d. Update particle position using the formula:
$xi = xi + Vi$
End while

3.7. GA-based CNN
The weights from convolutional filters and fully-connected layers are transformed into matrices. These matrices are then converted into one-dimensional vectors to create the initial population for genetic algorithms (GA). These vectors are designated as the starting population for the GA. Each potential solution is evaluated using a fitness function to determine the most effective network configuration. The fitness level of each solution is determined by its error rate during successive iterations. In an effort to enhance performance, crossover and mutation processes are applied to chosen solutions. A single-point crossover is utilized to produce offspring from one generation to the next, while mutations are made by altering a single gene in every offspring, increasing in frequency throughout successive

generations. A gene is randomly selected following a random distribution, and this selection and mutation process continues until the model achieves the optimal weight configuration with the minimum classification error [35].

4. Result and discussion

In the following discussion, we examine the results obtained from applying PSO to a CNN, an important application in machine learning and deep learning. This analysis focuses on identifying the tangible impact of PSO on the performance metrics of CNN, specifically tracking the progression in terms of loss and accuracy over a set of epochs. Through this comparative study, we aim to highlight how optimization methods improve the predictive performance of neural networks, offering useful insights into their real-world applications and potential enhancements in various computational tasks.

The experiments were performed on a high-end system to ensure accurate and efficient results. The system utilized was a Lenovo model 82UU, powered by an AMD Ryzen 9 6900HS Creator Edition processor with 16 cores running at around 3.3GHz. The OS was Windows 11 Pro 64-bit (Build 22631), with English as the regional setting. This setup included 16GB of RAM, providing ample memory for complex computations. Additionally, the system had a substantial page file capacity, with 40,886MB used and 6,029MB available, ensuring smooth handling of data-intensive processes. These robust specifications facilitated a comprehensive and practical assessment of the model's performance. To rigorously evaluate our proposed model, the dataset was split into 70% for training and 30% for testing, providing sufficient data for the training process while retaining a significant portion for unbiased evaluation. The training set was further partitioned into training and validation subsets, enabling us to track the model's performance and fine-tune parameters to prevent overfitting. This approach ensures that our model is tested on previously unseen data, giving a clear measure of its generalization and overall effectiveness.

In Table 4, the comparative analysis of CNN models with and without PSO reveals the significant impact of optimization techniques in deep learning. The non-optimized CNN shows a starting loss of 0.3430 and an accuracy of 70.79%, which over 20 epochs, steadily improves to a loss of 0.0045 and accuracy of 96.13%, reflecting the model’s better alignment with the data and its improved classification capabilities, as depicted in Figure 4 which illustrates the loss and accuracy progression over all epochs, where each epoch represents a full pass through the training dataset, emphasizing the iterative learning process. Conversely, the PSO-optimized CNN presents an intriguingly different trajectory. Beginning with a higher initial loss of 0.8574, it swiftly decreases to 0.0113, while the accuracy remarkably escalates to 99% by the end of the 20 epochs.

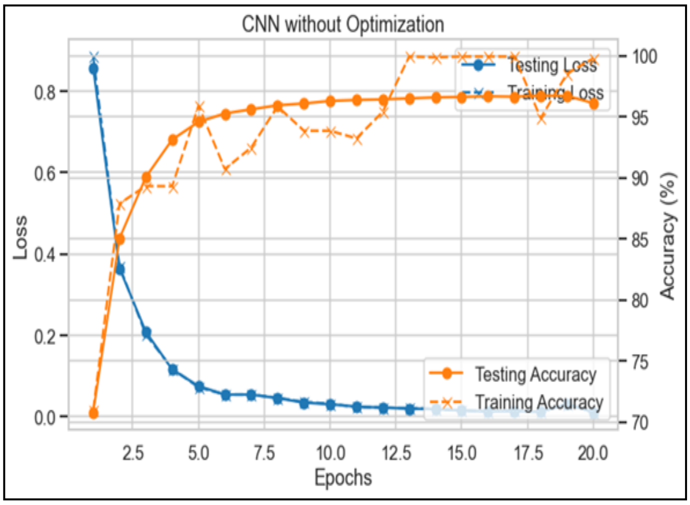


Figure 4. CNN loss and accuracy progression without PSO over the epochs

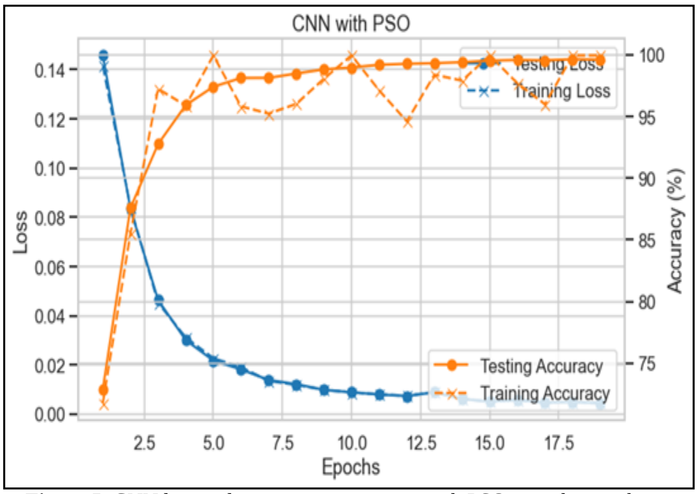


Figure 5. CNN loss and accuracy progression with PSO over the epochs

Table 5 and figure 6 the comparison between Convolutional Neural Networks (CNN) optimized with Particle Swarm Optimization (PSO) and their non-optimized counterparts across multiple metrics—Precision, Recall, and F1-Score—provides valuable insights into how optimization methods impact machine learning models. The data clearly demonstrates that CNNs enhanced with PSO optimization consistently outperform those without optimization, achieving near-perfect or perfect scores in precision, recall, and F1-Score throughout different evaluations. This shows that the PSO-driven optimization process notably enhances the model’s ability to correctly classify positive cases while reducing false positives and negatives. In contrast, the non-optimized CNN models exhibit more variability in their performance, with generally lower scores across all metrics. This suggests a higher incidence of misclassification or failure to identify all positive samples accurately, highlighting potential challenges in model generalization and sensitivity to dataset specifics. The stark difference in performance emphasizes the value of optimization strategies such as PSO in achieving higher accuracy, reliability, and overall performance in machine learning applications. Thus, the results advocate for the integration of optimization processes in the model development phase to enhance predictive capabilities and ensure more robust outcomes.

Table 4. CNN with and without PSO Accuracy and Loss Results

Epochs	CNN result without optimization	CNN result with PSO
1	Loss: 0.8574, Accuracy: 70.79%	Loss: 0.1458, Accuracy: 72.90%
2	Loss: 0.3644, Accuracy: 85.05%	Loss: 0.0834, Accuracy: 87.59%
3	Loss: 0.2086, Accuracy: 90.07%	Loss: 0.0464, Accuracy: 92.76%
4	Loss: 0.1160, Accuracy: 93.14%	Loss: 0.0301, Accuracy: 95.92%
5	Loss: 0.0752, Accuracy: 94.59%	Loss: 0.0218, Accuracy: 97.41%
6	Loss: 0.0542, Accuracy: 95.28%	Loss: 0.0182, Accuracy: 98.12%
7	Loss: 0.0544, Accuracy: 95.62%	Loss: 0.0139, Accuracy: 98.13%
8	Loss: 0.0456, Accuracy: 95.95%	Loss: 0.0122, Accuracy: 98.48%
9	Loss: 0.0347, Accuracy: 96.10%	Loss: 0.0099, Accuracy: 98.82%
10	Loss: 0.0306, Accuracy: 96.31%	Loss: 0.0089, Accuracy: 98.97%
11	Loss: 0.0247, Accuracy: 96.39%	Loss: 0.0080, Accuracy: 99.19%
12	Loss: 0.0222, Accuracy: 96.44%	Loss: 0.0074, Accuracy: 99.27%
13	Loss: 0.0200, Accuracy: 96.52%	Loss: 0.0089, Accuracy: 99.32%
14	Loss: 0.0186, Accuracy: 96.60%	Loss: 0.0062, Accuracy: 99.40%
15	Loss: 0.0154, Accuracy: 96.64%	Loss: 0.0051, Accuracy: 99.53%
16	Loss: 0.0127, Accuracy: 96.69%	Loss: 0.0057, Accuracy: 99.58%
17	Loss: 0.0142, Accuracy: 96.66%	Loss: 0.0047, Accuracy: 99.55%
18	Loss: 0.0118, Accuracy: 96.75%	Loss: 0.0049, Accuracy: 99.64%
19	Loss: 0.0306, Accuracy: 96.72%	Loss: 0.0045, Accuracy: 99.61%
20	Loss: 0.0113, Accuracy: 96.13%	Loss: -

Table 5. CNN with and without PSO Optimization: Accuracy and Loss Results

Classes	CNN without optimization			CNN with PSO optimization		
	Precision	Recall	F1score	Precision	Recall	F1score
01	0.98	0.99	0.98	1.00	1.00	1.00
02	0.87	0.93	0.90	0.99	1.00	1.00
03	0.93	0.89	0.91	1.00	1.00	1.00
04	0.88	0.86	0.87	1.00	0.98	0.99
05	0.87	0.91	0.89	0.99	1.00	1.00
06	0.93	0.87	0.90	1.00	0.99	1.00
07	0.75	0.84	0.80	0.87	0.96	0.92
08	0.93	0.86	0.89	1.00	0.98	1.00
09	0.81	0.86	0.84	0.93	0.98	0.96
10	0.81	0.77	0.79	0.93	0.89	0.91
11	0.78	0.85	0.81	0.90	0.97	0.93
12	0.85	0.74	0.79	0.97	0.86	0.91
13	0.93	0.92	0.93	1.00	1.00	1.00
14	0.92	0.94	0.93	1.00	1.00	1.00
15	0.81	0.91	0.85	0.93	1.00	0.97
16	0.91	0.87	0.89	1.00	0.99	1.00
17	0.89	0.79	0.84	1.00	0.91	0.96
18	0.91	0.90	0.90	1.00	1.00	1.00
19	0.90	0.92	0.91	1.00	1.00	1.00
20	0.82	0.86	0.84	0.94	0.98	0.96
21	0.89	0.89	0.89	1.00	1.00	1.00
22	0.86	0.82	0.84	0.98	0.94	0.96
23	0.85	0.80	0.83	0.97	0.92	0.95
24	0.82	0.83	0.83	0.94	0.95	0.95
25	0.86	0.89	0.88	0.98	1.00	1.00
26	0.84	0.91	0.87	0.96	1.00	0.99
27	0.79	0.82	0.80	0.91	0.94	0.92
28	0.85	0.80	0.83	0.97	0.92	0.95
29	0.86	0.87	0.87	0.98	0.99	0.99
30	0.89	0.85	0.87	1.00	0.97	0.99
31	0.86	0.89	0.88	0.98	1.00	1.00
32	0.89	0.86	0.88	1.00	0.98	1.00
33	0.94	0.95	0.95	1.00	1.00	1.00
34	0.97	0.96	0.97	1.00	1.00	1.00
35	0.85	0.83	0.84	0.97	0.95	0.96
36	0.85	0.86	0.85	0.97	0.98	0.97
37	0.92	0.92	0.92	1.00	1.00	1.00
38	0.89	0.92	0.90	1.00	1.00	1.00
39	0.94	0.88	0.91	1.00	1.00	1.00
40	0.83	0.84	0.83	0.95	0.96	0.95
41	0.85	0.87	0.86	0.97	0.99	0.98
42	0.85	0.81	0.83	0.97	0.93	0.95
43	0.85	0.87	0.86	0.97	0.99	0.98

Furthermore, we evaluated the performance of CNN, CNN-PSO with another optimization algorithm, CNN-based GA [35] in classifying different handwritten characters. Table 6 presents the results of these models across four evaluation metrics. It is evident from Table 6 that CNN-PSO outperforms the other two models on all metrics. The CNN-PSO model surpasses the CNN and CNN-based GA models in that the classification accuracy of all models achieves better performance, and the combined CNN-PSO model has the best overall accuracy of 0.99. All the recognition models can recognize 43 handwritten characters well but the combined CNN-PSO model shows a certain advantage in overall performance.

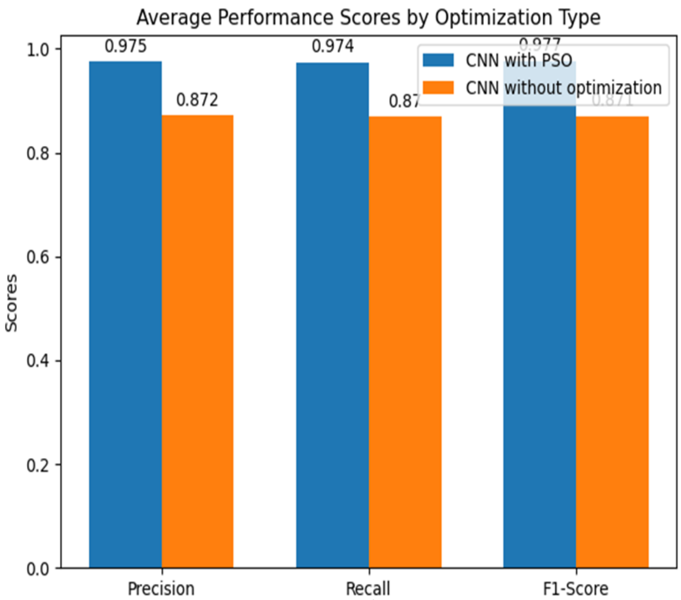


Figure 6. Average performance score

Table 6. CNN Models Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score	Additional Precision
CNN	0.87	0.86	0.87	0.85	0.87
CNN-GA	0.97	0.95	0.97	0.96	0.97
CNN-PSO	0.99	0.99	0.98	0.99	0.99

We further evaluated the computational efficiency of the three models, measured as training time in Table 7. The

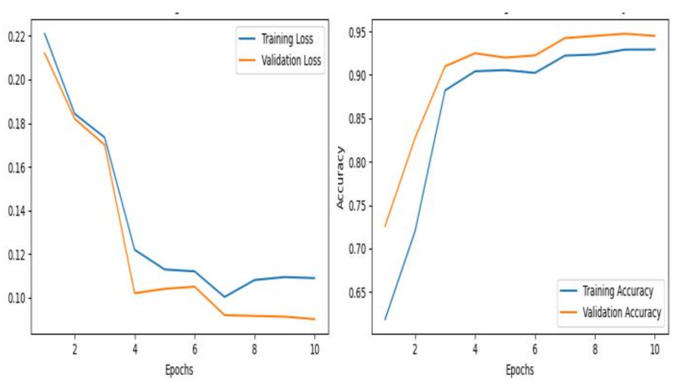
CNN-PSO model is more computationally beneficial than the others when the network has more layers. These models were run on a CPU platform in Python 3.7. The training time for the CNN-PSO model is slightly longer than that of CNN and CNN-GA. Although the inference minutes per data sample using the CNN models with PSO and without PSO are relatively low, it is only 3.99×10^{-3} s to train 13 of a second. The CNN-PSO model does a good job of balancing performance and complexity. The complementary implementation of the CNN-PSO model improves performance while increasing complexity and longer training.

Table 7. Computational time of CNN and CNN-PSO

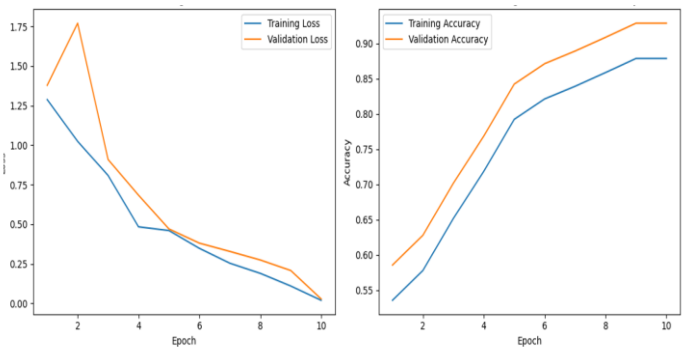
Models	Training time per data sample	Interference time
CNN	0.14 s	2.54×10^{-3} S
CNN-GA	0.16 s	3.12×10^{-3} S
CNN-PSO	0.18 s	3.99×10^{-3} S

Figure 6 shows the convergence curves, accuracy measurements, and confusion matrices to provide a detailed analysis of the classification outcomes of the publicly available handwritten Arabic character dataset [36]. By modifying parameters such as the number of epochs, batch sizes, and hidden units, the performance of deep learning models is critically assessed. Figure 7 showcases the progression of accuracy and loss convergence for both the training and validation phases within these models. Initial findings show moderate accuracy levels that steadily improve as the number of epochs increases. Simultaneously, improvements in the loss convergence trends for both training and validation phases are noted, with CNN-PSO achieving higher accuracy and the lowest loss compared to the other two models.

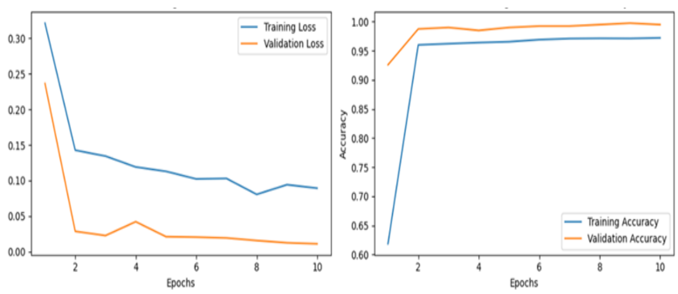
Table 8 shows the performance of CNN, CNN-PSO with other optimization algorithms CNN-based GA [3] in classifying Arabic handwritten characters. It presents the results of these models across four evaluation metrics. It is clear from Table 8 that CNN-PSO surpasses the other two models on all metrics. The CNN-PSO model clearly outperforms CNN and CNN-based GA in classification accuracy, achieving the best overall accuracy of 99.12%. All the recognition models can recognize Arabic handwritten characters well but the combined CNN-PSO model shows a certain edge in overall performance.



(a)



(b)



(c)

Figure 7. (a) CNN loss and accuracy graph, (b) CNN-GA loss accuracy graph, (c) CNN-PSO Loss and accuracy.

Table 8. Comparison of Precision and Accuracy Metrics

Model	Accuracy	Precision	Recall	F1-Score
CNN	92.87%	95.31%	84.14%	89.38%
CNN-GA	97.3%	97.42%	93.79%	95.57%
CNN-PSO	99.12%	99.35%	96.25%	97.78%

Table 9 illustrates the comparison between our proposed model and existing methods used for character recognition. In this table, the performance of our model, which integrates a CNN combined with PSO, is evaluated against other established techniques. Our CNN-PSO model stands out by achieving higher accuracy and lower error rates, indicating a significant improvement over the traditional methods currently used in character recognition. This comparison highlights the effectiveness of our method in improving the accuracy and reliability of character recognition systems.

Table 9. Comparison of Different Methods for Script Recognition

References	Method	Accuracy/loss
[38]	LSTM-based systems for Urdu recognition (two cases)	error rates of 13.57% and 5.15%
[39]	FFNN as classifier for Urdu script, using various features	74%
[41]	Scale-invariant method based on water reservoir principles for Urdu recognition	97.8%
[42]	Shape context and contours with k-nearest neighbor for Arabic and Urdu scripts	91% word recognition rate
[35]	BBN Byblos OCR system for Pashto recognition using left-to-right HMM	error rates: 2.1% to 26.3%,

[36]	SIFT and PCA based classifier for Pashto ligature recognition	73% scale and rotation invariant recognition accuracy
Proposed	CNN-PSO	99.64%

4.1. Comparison with Deep Neural Networks (DNN)

In this section, we assess the performance of the proposed CNN-PSO method against several deep neural networks, including VGGnet [42], Resnet [43], Nasnet [44], Mobilenet [45], Inception [47], and Xception [46]. Given the well-documented benefits of DNNs in machine learning, particularly in image classification, they have been widely adopted in recent studies for specific domain tasks. Table 10 presents a comparative analysis between the CNN-PSO algorithm and these DNNs.

From Table 8, it is evident that although DNNs typically achieve high recognition ac-curacy, they still fall short compared to the proposed approach. Specifically, the DNNs achieve recognition accuracies of 91% and 93%, whereas the CNN-PSO algorithm at-trains a higher accuracy of 96%.

Table 10. Comparison with Deep Neural Network

Model	Accuracy (%)	Recall	Precision
VGGnet	91	100	91
Resent	53	74	85
Inception	86	95	96
Exception	93	96	87
Mobile net	93	95	100
Proposed	99	99	96

4.2. Limitations and challenges

This research has several limitations and challenges. First, utilizing the PSO algorithm with CNN architecture, complex computations with a longer training time were required. Second, the parameter optimization of the PSO algorithm was challenging as it depended on how the model was tuned or optimized, which provided the best performance without over utilizing the computational resources. Finally, the performance completely relies on CAD and CUDA-based high computing technology like Nvidia GPU.

5. Conclusion and Future Work

This study explored the significant effects of Particle Swarm Optimization (PSO) on a CNN, analyzing key metrics such as loss and accuracy across multiple epochs. Initially, the optimized CNN showed progressive improvement, highlighting its growing classification capabilities. However, the integration of PSO significantly transformed the CNN's learning trajectory, as the PSO-optimized CNN rapidly decreased its loss and achieved remarkable accuracy, vastly outperforming the non-optimized model. The comparative analysis further underscored the PSO-optimized CNN's superior efficiency in reducing loss and faster achievement of high accuracy. Detailed examinations of precision, recall, and F1 scores across various classes revealed significant enhancements, particularly in classes where the unoptimized model was less effective. The stark improvement in overall accuracy, derived from average F1 scores, attests to PSO's effectiveness. This study conclusively affirms PSO's efficacy as a potent optimization tool in deep learning, enhancing neural network models to adapt more quickly and with greater precision, and opens new avenues in computational tasks requiring intricate data interpretation, like handwritten character recognition, thereby contributing significantly to the advancement of machine learning.

Future work could explore other optimization approaches, such as GA or Differential Evolution (DE), to further improve the CNN-PSO hybrid model. Increasing the dataset with a wider variety of handwriting samples would enhance the model's robustness and ability to generalize. Additionally, applying the hybrid model to other complex scripts beyond Pashto and implementing transfer learning to leverage pre-trained models are promising directions. Finally, integrating the hybrid model into real-time optical character recognition (OCR) systems and language translation tools could demonstrate its practical utility and further refine its capabilities.

Author's statements

Disclosures. We confirm that this research was conducted without any conflicting interests, external funding, or institutional bias that could affect the integrity of the findings or interpretations.

Acknowledgements. The authors extend heartfelt appreciation to all individuals and institutions who contributed indirectly to this research. We are especially thankful to the developers and maintainers of the open-access dataset hosted on Kaggle, which served as a valuable foundation for our analysis. The authors also acknowledge the support of their respective institutions for enabling access to essential facilities and resources crucial to this study. Furthermore, we appreciate the insightful input and constructive feedback shared by our colleagues throughout the research process.

Funding Source. None

Authors' Contributions.

Author 1: Designed the PSO-CNN hybrid framework and led the experimental analysis of optimization impacts. Author 2 Implemented the CNN architecture and conducted performance evaluations (accuracy/loss metrics). Author 3 Developed data pre-processing/augmentation methods and validated model robustness. Author 4 Optimized PSO parameters and interpreted results through Grad-CAM visualizations. Author 5 Reviewed and critically evaluated the manuscript for clarity, coherence, and technical accuracy. All authors actively participated in drafting and revising the manuscript and have given their approval for the final submitted version.

Conflict of Interest. The authors declare that they have no financial or commercial interests that might be perceived as influencing the results or conclusions of this study.

Data Availability. The dataset developed for this research is publicly available, providing an invaluable resource for the research community to advance character recognition technologies.[Handwritten dataset (kaggle.com)]. All data were obtained from publicly available sources and are governed by Kaggle's terms of use as outlined by the original dataset contributors. No special permissions or restrictions apply. The authors are available to assist with dataset access or usage inquiries, if needed.

References

[1] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, "Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends," Natural Language Processing Journal, Vol.**4**, pp.**100026, 2023.** DOI: 10.1016/j.nlp.2023.100026

[2] J. Torregrosa, G. Bello-Orgaz, E. Martínez-Cámara, et al., "A survey on extremism analysis using natural language processing: definitions, literature review, trends and challenges," Journal of Ambient Intelligence and Humanized Computing, Vol.**14**, pp.**9869–9905, 2023**. DOI: 10.1007/s12652-021-03658-z

[3] F. Meng and B. Ghena, "Research on Text Recognition Methods Based on Artificial Intelligence and Machine Learning," Advances in Computer and Communication, Vol.**4**, No.**5**, pp.**340–344, 2023.** DOI: 10.26855/acc.2023.10.014

[4] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," Multimedia Tools and Applications, Vol.**82**, No.**3**, pp.**3713–3744, 2023.**

[5] Y. Yang, A. K. Singh, M. Elhoushi, A. Mahmoud, K. Tirumala, F. Gloeckle, B. Rozière, C.-J. Wu, A. S. Morcos, and N. Ardalani, "Decoding data quality via synthetic corruptions: Embedding-guided pruning of code data," arXiv preprint arXiv:2312.02418, **2023**.

[6] H. Wang, Z. Yue, Q. Xie, Q. Zhao, Y. Zheng, and D. Meng, "From rain generation to rain removal," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.**14791–14801, 2021.**

[7] S. Khan, S. Nazir, H. U. Khan, and A. Hussain, "Pashto characters recognition using multi-class enabled support vector machine," International Journal/Conference (details missing), **2021**.

[8] Y. N. Pawan, K. B. Prakash, S. Chowdhury, and Y. C. Hu, "Particle swarm optimization performance improvement using deep learning techniques," Multimedia Tools and Applications, Vol.**81**, No. **19**, pp.**27949–27968, 2022.**

[9] T. Lawrence, L. Zhang, C. P. Lim, and E. J. Phillips, "Particle swarm optimization for automatically evolving convolutional neural networks for image classification," IEEE Access, Vol.**9**, pp.**14369–14386, 2021.**

[10] M. Abd Elaziz et al., "Advanced metaheuristic optimization techniques in applications of deep neural networks: A review," Neural Computing and Applications, pp.**1–21, 2021.**

[11] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, Vol.**8**, pp.**1–74, 2021.**

[12] Y. B. Hamdan and A. Sathesh, "Construction of statistical SVM based recognition model for handwritten character recognition," Journal of Information Technology, Vol.**3**, No.**2**, pp.**92**–**107**, **2021**.

[13] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)," IEEE Access, Vol.**8**, pp.**142642**–**142668, 2020.**

[14] F. Asadi-zeydabadi, A. Afkari-Fahandari, A. Faraji, E. Shabaninia, and H. Nezamabadi-pour, "IDPL-PFOD2: A New Large-Scale Dataset for Printed Farsi Optical Character Recognition," arXiv preprint arXiv:2312.01177, **2023**.

[15] S. Naz et al., "Urdu Nastaliq recognition using convolutional recursive deep learning," Neurocomputing, Vol.**243**, pp.**80–87**, **2017**.

[16] F. Alotaibi et al., "Optical character recognition for Quranic image similarity matching," IEEE Access, Vol.**6**, pp.**554–562, 2018.**

[17] C. Boufenar, M. Batouche, and M. Schoenauer, "An artificial immune system for offline isolated handwritten Arabic character recognition," Evolving Systems, Vol.**9**, No.**1**, pp.**25–41, 2018.**

[18] M. I. Razzak, "Urdu nastaliq text recognition system based on multidimensional recurrent neural network and statistical features," Neural Computing and Applications, Vol.**28**, No.**2**, pp.**219–231, 2017.**

[19] S. Naz et al., "Statistical features extraction for character recognition using recurrent neural network," Pakistan Journal of Statistics, Vol.**34**, No.**1**, pp.**47–53, 2018.**

[20] M. Rabi, M. Amrouch, and Z. Mahani, "A survey of contextual handwritten recognition systems based HMMs for cursive Arabic and Latin script," International Journal of Computer Applications, Vol.**160**, No.**21, 2017.**

[21] S. Islam et al., "A comprehensive survey on applications of transformers for deep learning tasks," Expert Systems with Applications, pp.**122666, 2023.**

[22] A. V. Geetha, T. Mala, D. Priyanka, and E. Uma, "Multimodal emotion recognition with deep learning: advancements, challenges, and future directions," Information Fusion, Vol.**105**, pp.**102218, 2024.**

[23] F. Zeng, W. Gan, Y. Wang, N. Liu, and P. S. Yu, "Large language models for robotics: A survey," arXiv preprint arXiv:2311.07226, **2023**.

[24] J. Wang et al., "Review of large vision models and visual prompt engineering," Meta-Radiology, pp.**100047**, **2023**.

[25] J. Lin et al., "M6: A Chinese multimodal pretrainer," arXiv preprint arXiv:2103.00823, **2021**.

[26] K. Han et al., "A survey on visual transformer," arXiv preprint arXiv:2012.12556, **2020**.

[27] M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep learning for sign language recognition: Current techniques, benchmarks, and open issues," IEEE Access, Vol.9, pp.**126917–126951, 2021.**

[28] J. Zheng, Y. Gao, H. Zhang, Y. Lei, and J. Zhang, "OTSU multi-threshold image segmentation based on improved particle swarm algorithm," Applied Sciences, Vol.**12**, No.**22**, pp.**11514, 2022.**

[29] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview," Soft Computing, Vol.**22**, pp.**387–408, 2018.**

[30] T. Blackwell and J. Kennedy, "Impact of communication topology in particle swarm optimization," IEEE Transactions on Evolutionary Computation, Vol.**23**, pp.**689–702, 2018.**

[31] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," Archives of Computational Methods in Engineering, Vol.**29**, pp.**2531–2561, 2022.**

[32] N. A. A. Aziz and K. A. Aziz, "Pendulum search algorithm: An optimization algorithm based on simple harmonic motion and its application for a vaccine distribution problem," Algorithms, Vol.**15**, pp.**214, 2022.**

[33] A. K. Paul, P. C. Shill, R. I. Rabin, A. Kundu, and A. H. Akhand, "Fuzzy membership function generation using DMS-PSO for the diagnosis of heart disease," Proceedings of the 2015 18th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, pp.**456**–**461**, **2015**.

[34] Z. Beheshti, S. M. H. Shamsuddin, E. Beheshti, and S. S. Yuhaniz, "Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis," Soft Computing, Vol.**18**, pp.**2253–2270**, **2013**.

[35] J. Li, T. Liu, X. Wang, and J. Yu, "Automated asphalt pavement damage rate detection based on optimized GA-CNN,"

Automation in Construction, Vol.**136**, pp.**104180, 2022.**

[36] A. Mloey, "AHCD1: Arabic handwritten characters dataset," Dataset Publication (details missing), **2022**.

[37] S. Poornima and M. Pushpalatha, "Prediction of rainfall using intensified LSTM based recurrent neural network with weighted linear units," Atmosphere, Vol.**10**, No.**11**, pp.**668, 2019.**

[38] M. F. Bashir et al., "Context-aware emotion detection from low-resource Urdu language using deep neural network," ACM Transactions on Asian and Low-Resource Language Information Processing, Vol.**22**, No.**5**, pp.**1–30, 2023.**

[39] V. N. Boddeti, "Advances in correlation filters: vector features, structured prediction and shape alignment," Ph.D. Thesis, Carnegie Mellon Univ., USA, **2012**.

[40] R. Ahmad et al., "Scale and rotation invariant OCR for Pashto cursive script using MDLSTM network," Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp.**1101–1105**, Aug. **2015**.

[41] F. M. Nashwan et al., "A holistic technique for an Arabic OCR system," Journal of Imaging, Vol.**4**, No.**1**, pp.**6**, **2017**.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, **2014.**

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.**770–778**, **2016**.

[44] Google Research Blog, "AutoML for large scale image classification and object detection," Nov. **2017**.

[45] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, **2017**.

[46] F. Chollet, "Exception: Deep learning with depth wise separable convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.**1251**–**1258**, **2017**.

[47] C. Szegedy et al., "Going deeper with convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.**1–9**, **2015**.

AUTHORS PROFILE

Saeed Ur Rahman received the B.Sc. degree in Computer Science from Shaheed Benazir Bhutto University (SBBU), Pakistan, in 2023. He is currently pursuing the Master's degree in Computer Science and Technology at Xidian University, Xi'an, Shaanxi, China. His current research interests include privacy-preserving machine learning, homomorphic encryption, secure multi-party computation (MPC), federated learning, cryptography, and cloud computing security.

Muneeb Ullah received the B.Sc. degree in computer science from the University of Peshawar, Peshawar, Pakistan, in 2016, and the master's degree from the Agriculture University of Peshawar, Peshawar, in 2019. He is currently pursuing the Ph.D. degree with Xidian University, Xi'an, Shaanxi, China. His research interests include image processing and using SDRF technology for health monitoring and disease detection

Nisar Ullah received the Bachelor's degree in Physics from Hazara University, Pakistan, and the Master's degree in Spectroscopical Studies from the School of Physics, Xidian University, Xi'an, Shaanxi, China. He is currently pursuing the Ph.D. degree at Xidian University, with research focused on image processing and image enhancement. His current research interests include optical imaging, high-resolution imaging, and inverse problems in computational imaging, optical design using deep learning, and machine learning and computer vision applications.

Muhammad Faizan Khan received his M.S. in Computer Science from the School of Artificial Intelligence, Xidian University, China, in 2025. He will pursue his Ph.D. in Computer Science at Xi'an Jiaotong University. His research interests include automatic modulation classification, deep learning, and AI for wireless communication. He has published papers in UCom, ICECE, and ICMMT, and has one paper accepted for publication in an IEEE conference.

Irzum Shafique is currently pursuing a Master of Engineering in Computer Science at Xidian University, China, where he also earned his Bachelor's degree. His research focuses on deep reinforcement learning for robotic path planning, intelligent autonomous drones for agriculture and urban firefighting, and the application of artificial intelligence for social good.

He has published several research papers in peer-reviewed journals and international conference proceedings, covering topics such as autonomous systems, machine learning, and intelligent robotics. His work reflects a strong emphasis on both theoretical development and real-world application.

Irzum is a co-inventor of the utility model patent "Autonomous Agricultural Drone with Adaptive Fertilizer Spraying Mechanism," recognized for its innovation in precision agriculture. He has received numerous awards for academic and entrepreneurial excellence, including a gold medal in the China International University Student Innovation Competition (national finalist), third place at the provincial level, and third place in the Kyoto University International Student Entrepreneurship Challenge.

