UDC 004.855.5

ENHANCING ARABIC CHARACTER RECOGNITION VIA FEATURE ENGINEERING AND PSO

Sarra ROUABHI, Redouane TLEMSANI
(Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, Oran, Algeria)

Abstract. *Accurate recognition of handwritten Arabic characters poses significant challenges, especially for non-native learners. With the increasing adoption of digital teaching and distance learning, there is a pressing need for efficient and robust automatic recognition systems for Arabic characters. This work proposes a novel approach to address this challenge. First, handwritten Arabic character images are processed to extract features using two pre-trained deep learning models: EfficientNet B2 and DenseNet 201. The extracted features from these models are then concatenated to form a comprehensive feature set. Subsequently, the Particle Swarm Optimization (PSO) algorithm is employed to identify the most relevant features from this concatenated set through an optimized feature selection process. Finally, the selected features are fed into a classical classifier for character recognition. The proposed approach achieves an accuracy exceeding 90%, demonstrating its effectiveness in recognizing handwritten Arabic characters.*

Keywords: *Arabic handwriting recognition, feature extraction, feature selection, Particle Swarm Optimization.*

Introduction. Millions of people in the Middle East and North Africa speak Arabic, a Semitic language. It is one of the six official languages of the United Nations and the official language of 26 other nations. One of the world's oldest and most prominent languages, Arabic has a rich literary and cultural history [1].

The Arabic alphabet is written from right to left, it has 28 letters, with each letter having up to four different shapes depending on its position in the word [2, 3] This means that there are many variations of each letter that a writer must learn and master to write the Arabic alphabet correctly.

Like any other language, Arabic can be written in different styles or scripts by different people. The variation in writing styles can depend on several factors [4], including the author's level of education, cultural background, geographical location, personal preferences, and the context in which the writing is produced. However, Arabic script is generally considered difficult to learn and master, especially for non-native speakers. Nowadays, digital teaching is being replaced, which may force children or adults who need to study Arabic to do so remotely via the Internet. In addition, it is very difficult to distinguish Arabic script, especially when it comes to assessing students' skills in writing Arabic characters, which requires scientific study [5]. The identification of Arabic scripts using deep learning and machine learning methods is an intriguing topic.

Several researchers are dedicated to enhancing the recognition performance of Arabic handwritten characters utilizing convolutional neural networks (CNNs). Here's a refined summary:

In 2017 [3] El-Sawy et al. proposed a straightforward CNN architecture comprising two convolutional layers, two pooling layers, and one fully connected layer to categorize Arabic handwritten characters. Their model attained a classification error rate of 5.1% on their proprietary database of 16,800 images 32 x 32 pixels categorized into 28 classes.

Alkhateeb assessed his CNN model on three databases: AHCR (89.9% accuracy), AHCD (95.4% accuracy), and Hijja (92.5% accuracy), showcasing the efficacy of the model in recognizing isolated Arabic handwritten characters [6].

In 2021, Balaha et al. [7] introduced the HMBD database and proposed two deep learning architectures, HMB1 and HMB2, achieving remarkable testing accuracies on the HMBD database and other databases. Alrobah and Albahli developed a hybrid architecture comprising two CNN feature extractors and three classifiers (SVM, Fully Connected Layer, and XGBoost) to classify Arabic handwritten characters from the Hijja dataset, achieving test accuracies of 96.3% (SVM), 89% (FCL), and 95.7% (XGBoost).

Another study in 2021 [8] evaluated 14 CNN architectures on the HMBD dataset, with the highest testing accuracy recorded at 91.96%. They also proposed a classification approach utilizing pre-trained models (VGG16, VGG19, MobileNetV2) with genetic optimization, achieving a testing accuracy of 92.88%.

In 2022 another research [9] introduced by Nayef et al. introduced an optimized Leakage Rectified Linear Unit (ReLU) activation function integrated into a CNN architecture with six layers and four fully connected layers. Their model achieved testing accuracies of 87.45% (ReLU), 89.6% (Leaky ReLU), and 95% (Optimized Leaky ReLU) on their self-collected dataset.

This body of work showcases diverse CNN architectures, optimization techniques, and activation functions applied to enhance the recognition performance of Arabic handwritten characters across various databases. However, it's noteworthy that only small 32×32 datasets were utilized, which aids in classification.

Our study aims to develop systems based on particle swarm optimization capable of classifying Arabic manuscripts using the HMBDv1 database, which is notably more complex compared to the databases typically utilized in the literature.

This document is organized into several sections. Section 1 provides an overview of the research objectives and presents recent work in Arabic feature recognition. Section 2 details the proposed approach, including feature extraction, concatenation, and feature selection techniques, hybridization methods, and classification mechanisms. Next, Section 3 presents experimental results, assessing the efficiency and accuracy of the proposed approach. Finally, Section 4 summarizes the results and suggests avenues for future research in the field of Arabic handwriting feature recognition.

1. Methodology

This section will outline the comprehensive procedures and techniques employed throughout our investigative endeavor.

1.1. Features extraction

Convolutional neural networks (CNNs) are popular methods for extracting features from images, including handwritten character images [10]. A CNN is a deep learning model designed to automatically recognize hierarchical features in input images. A CNN can be trained to recognize features such as edges, curves [11], and character shapes in the case of handwriting recognition. We extracted 1408 features using DenseNet-201 and 1920 features using EfficientNet-B2 from the Arabic handwritten character images. We then concatenated these feature representations to enhance the recognition rate as shown in Figure 1.

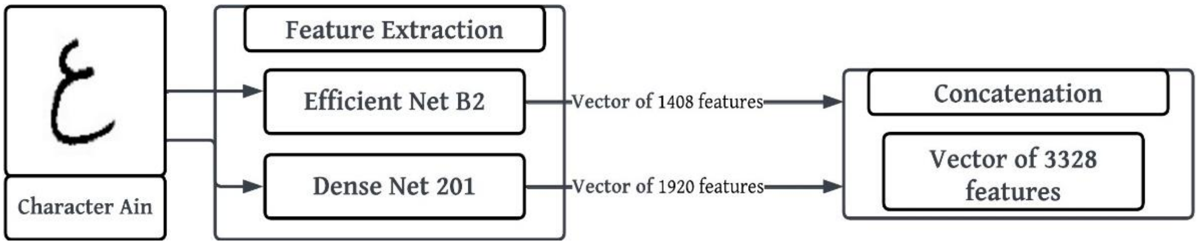

Figure 1.– Feature Extraction Architecture

DenseNet-201 is a densely connected convolutional neural network architecture that consists of 201 layers. It introduces dense blocks and dense connections, enabling feature reuse and alleviating the vanishing gradient problem. In our case, the DenseNet-201 model extracted 1408 features from the input images [12].

EfficientNet-B2, on the other hand, is a variant of the EfficientNet family, which employs a compound scaling method that uniformly scales the network's width, depth, and resolution with a fixed ratio. EfficientNet-B2 is a balanced model that achieved good accuracy and efficiency. In our experiment, the EfficientNet-B2 model extracted 1920 features from the input images [13].

To leverage the strengths of both architectures and capture a more diverse set of features, we concatenated the feature representations obtained from DenseNet-201 and EfficientNet-B2. Concatenation is a technique in which the output feature vectors from multiple models are combined by stacking them along a new dimension, creating a single, higher-dimensional feature representation. Mathematically, let's denote the feature vector extracted from DenseNet-201 as f_dense (with 1408 dimensions) and the feature vector from EfficientNet-B2 as f_eff (with 1920 dimensions). The concatenated feature vector, denoted as f_concat, is obtained by:

$$f\_concat = \left[f\_dense, f\_eff \right], \tag{1}$$

where $[,]$ represents the concatenation operation along a new dimension, typically the channel dimension for convolutional features or the feature dimension for fully connected layers.

By concatenating the 1408 features from DenseNet-201 and the 1920 features from EfficientNet-B2, we created a higher-dimensional feature representation with a total of 3328 features (1408 + 1920). This combined representation aimed to capture complementary information from both architectures, potentially leading to improved performance in the Arabic handwritten character recognition task.

1.2. Feature Selection

After concatenating the features extracted from DenseNet-201 and EfficientNet-B2, we employed the bio-inspired Particle Swarm Optimization (PSO) algorithm for feature selection to identify the most relevant and informative features.

Particle Swarm Optimization (PSO) [14, 15, 16] is a population-based metaheuristic optimization algorithm inspired by the social behavior of bird flocking or fish schooling. In the context of feature selection, PSO can be used to search for an optimal subset of features that maximizes a given objective function, such as classification accuracy or other performance metrics.

1.3. Classification

The Support Vector Classifier (SVC) is a supervised machine learning algorithm used for classification tasks. It constructs a hyperplane or a set of hyperplanes in a high-dimensional space to separate different classes. The hyperplane is chosen to maximize the margin between the classes, making the SVC robust to high-dimensional data and effective in handling non-linear decision boundaries [17].

The SVC algorithm works by finding the optimal hyperplane that separates the classes with the largest margin. The data points closest to the hyperplane, called support vectors, are used to define the decision boundary. The SVC can handle non-linear decision boundaries by using kernel functions to map the input data into a higher-dimensional feature space [18, 19].

In our work, we employed the SVC algorithm for the task of recognizing Arabic handwritten characters. The SVC was trained on the selected feature subset obtained from the PSO-based feature selection process, and its parameters were optimized to achieve the best possible recognition performance.

2. Result and discussion

In this section, we delve into the outcomes of our experiment. The initial stage involves combining the features extracted from DenseNet 201 and EfficientNet B2.

Subsequently, we elucidate the efficacy of employing PSO for selecting the most pertinent Arabic feature attributes. Finally, we employ SVC to classify them.

In this contribution, we have incorporated the Particle Swarm Optimization (PSO) algorithm to enhance the feature selection process and classification performance using a Support Vector Classifier (SVC) classifier as shown in Figure 2 .

The PSO algorithm efficiently explores the feature space and identifies a subset of features that leads to better classification performance. The selected features are then utilized to train the SVM model, which leverages the discriminative power of these features to make accurate predictions on new data instances.

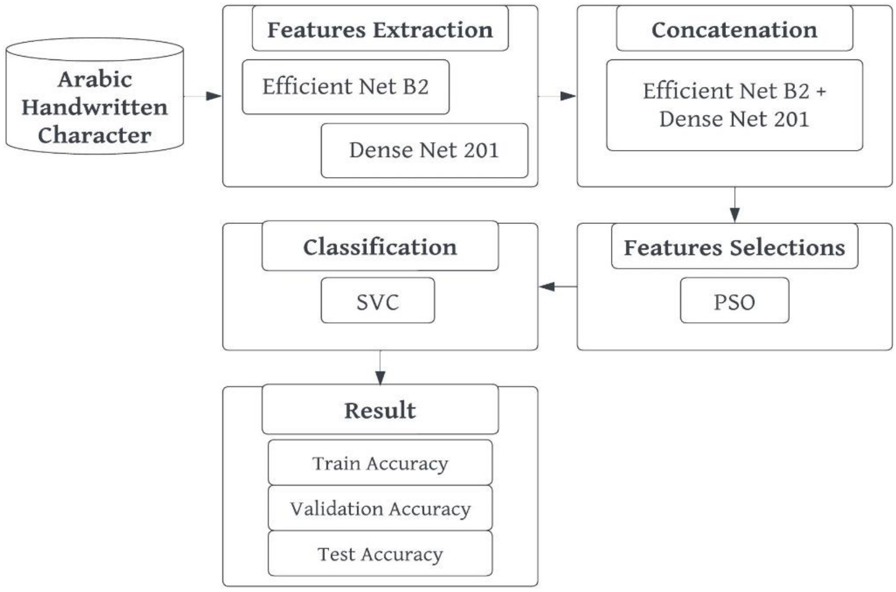

Figure 2. – Proposed Arabic Handwritten Character Recognition System Architecture

The table determines hyperparameters such as the behavior and performance of the Particle Swarm Optimization (PSO) algorithm for feature selection. The number of particles defines the size of the swarm, and the number of iterations determines how many times the particles will be updated. The number of features represents the total number of features in the input dataset, while the number of features selected determines the desired percentage of features to be selected.

Table 1. – The Hyperparameters for PSO-SVC hybridization

Hyperparameter	Description	Value
num-particles	Number of particles in the PSO algorithm	20
num-iterations	Number of iterations for the PSO algorithm	100
num-features	Number of features in the dataset	3328
test-size	Proportion of the dataset used for testing	0.2
random-stat	Random seed for reproducibility	42
inertia-weight	Inertia weight for velocity update	0.9
cognitive-weight	Cognitive weight for velocity update	1.5
social-weight	Social weight for velocity update	1.5

The algorithm refer to Figure 3 begins with particle initialization, where the positions and velocities of the particles are randomly initialized. Then, for each iteration, the following steps occur:

– Particle Velocity and Position Update: The velocities and positions of the particles are updated according to their current values, the influence of their personal best position, and the global best position.

– Suitability Evaluation: The suitability (fitness) of each particle is evaluated based on its current position.

– Personal Best Position and Suitability Update: If a particle's current position yields a better suitability value than its previous personal best position, the particle updates its personal best position and suitability value.

– Global Best Position and Suitability Update: The global best position and suitability are updated by comparing the suitability values of all particles. The global best position represents the best combination of features found so far.

– Feature Selection: Features are selected based on the global best position, which represents the optimal subset of features.

– Support Vector Classifier (SVC) Classifier Training: The SVC classifier model is trained using the selected features.

– Performance Evaluation: The performance of the trained SVC model is evaluated using a set of test data.

– Convergence Visualization: The convergence behavior of the algorithm is plotted on a convergence graph.

This loop represents the main iterative process of the Particle Swarm Optimization (PSO) algorithm for feature selection and classification using the SVC.

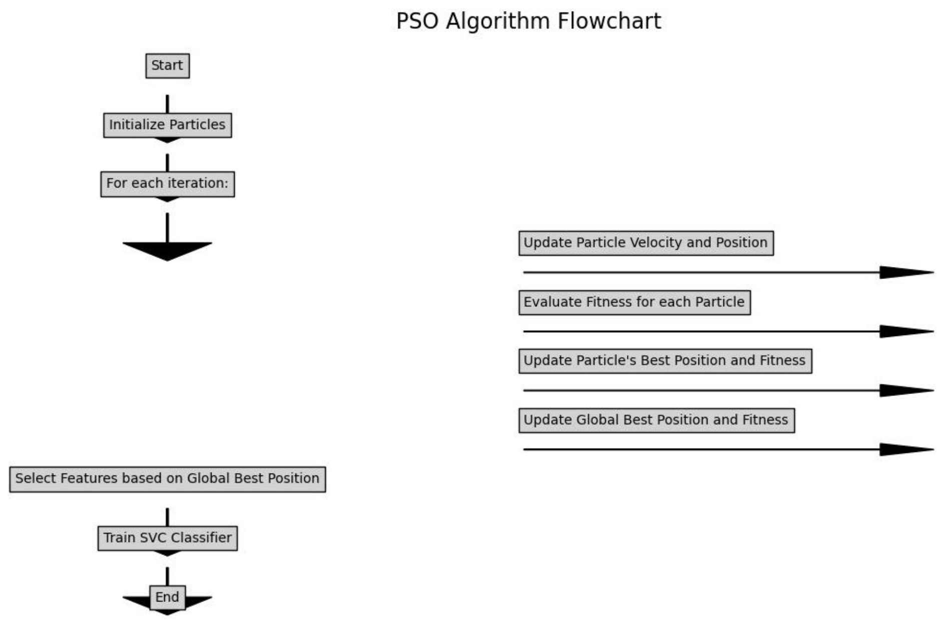

Figure 3. – Flowchart represents the steps of the PSO-SVC algorithm

The results of the third experiment, which involved the integration of the Particle Swarm Optimization (PSO) algorithm for feature selection with a SVC classifier, are summarized in Table 2. These results demonstrate the performance of the system in terms of accuracy, precision, recall, and F1-score [20, 21].

Table 2. – Test results for accuracy, macro and weighted avg

Metrics	EfficientNet B2 + DenseNet 201
	PSO-SVC
Test accuracy	90.20%
Macro avg Precision	90.00%
Macro avg Recall	91.00%
Macro avg F1-score	90.00%
Weighted avg Precision	91.00%
Weighted avg Recall	90.00%
Weighted avg F1-score	90.00%

The test accuracy achieved by the PSO-SVM algorithm is 90.20%, indicating a high level of accuracy in recognizing isolated handwritten Arabic characters. This result signifies the effectiveness of the feature selection process using the PSO algorithm in enhancing the overall performance of the system.

The macro-averaged precision, recall, and F1-score are 90.00%. The macro-average considers the equal contribution of each class, providing an overall evaluation of the system's ability to classify Arabic characters across all classes. These metrics indicate a balanced performance, with consistent precision, recall, and F1-score values.

Similarly, the weighted-average precision, recall, and F1-score are also 90.00%. The weighted average takes into account the class distribution, giving more weight to classes with a larger number of samples. This result indicates that the system performs well across different classes, considering their varying sizes and distributions.

Table 3. – Classification report

ArabicCharacters	Precision	Recall	F1-score
Ain_Isolated	0.69	0.92	0.79
Alf_Hamza_Above_Isolated	0.91	1.00	0.95
Alf_Hamza_Under_Isolated	1.00	0.85	0.92
Alf_Isolated	0.95	0.95	0.95
Baa_Isolated	1.00	1.00	1.00
Daad_Isolated	1.00	0.83	0.91
Dal_Isolated	1.00	1.00	1.00
Faa_Isolated	0.88	0.88	0.88
Gem_Isolated	1.00	0.85	0.92
Gen_Isolated	1.00	0.77	0.87

ArabicCharacters	Precision	Recall	F1-score
Ha_Isolated	1.00	0.95	0.98
Haa_Isolated	0.79	0.92	0.85
Hamza_Isolated	0.89	1.00	0.94
Kaf_Isolated	0.87	0.93	0.90
Khaa_Isolated	0.92	0.86	0.89
Lam_Alf_Hamza_Isolated	0.64	0.82	0.72
Lam_Alf_Isolated	0.84	0.94	0.89
Lam_Alf_Mad_Isolated	0.87	0.68	0.76
Lam_Isolated	0.94	1.00	0.97
Mem_Isolated	0.93	0.93	0.93
Non_Isolated	1.00	0.90	0.95
Qaf_Isolated	0.75	0.82	0.78
Raa_Isolated	0.92	1.00	0.96
Saad_Isolated	0.95	0.95	0.95
Shen_Isolated	0.85	0.92	0.88
Sin_Isolated	1.00	1.00	1.00
Taa_Isolated	0.88	0.67	0.76
Tah_Isolated	0.94	0.94	0.94
Thaa_Isolated	0.77	0.89	0.83
Waw_Hamza_Isolated	0.87	0.76	0.81
Waw_Isolated	0.94	0.94	0.94
Yaa_Dot_Isolated	1.00	1.00	1.00
Yaa_Isolated	0.87	1.00	0.93
Zah_Isolated	0.83	1.00	0.91
Zal_Isolated	0.93	0.93	0.93
Zin_Isolated	0.88	1.00	0.94

The results in Table 3 demonstrate the effectiveness of the "Efficient Hybridization B2+Dense201-PSO-SVC" approach in recognizing a wide range of isolated Arabic characters. Most characters achieve high precision, recall, and F1 scores, indicating accurate classification. For instance, characters such as "Baa-Isolated", "Dal-Isolated", "Sin-Isolated", and "Yaa-Dot-Isolated" achieve perfect precision, recall, and F1 scores of 1.00. This indicates that the system correctly recognizes all instances of these characters. Other characters, such as "Alif-Hamza-Above-Isolated", "Alif-Isolated", "Lam-Isolated", and "Zah-Isolated", also exhibit high precision, recall, and F1 scores, ranging between 0.94 and 1.00. These results demonstrate the system's ability to accurately classify a diverse set of isolated Arabic characters.

While most characters achieved high scores, some performed relatively lower. For example, the character "Taa-Isolated" has a precision of 0.88, a recall of 0.67, and an F1 score of 0.76. This suggests that the system may encounter difficulties in accurately recognizing certain instances of this character due to potential similarities in the handwriting of some characters.

The proposed approach shows promising results in recognizing isolated handwritten Arabic characters, with most characters achieving high classification performance. However, there is still room for improvement, particularly for characters that exhibit lower scores, which could be addressed by further refining the feature selection and classification techniques.

Additionally, a convergence graph refer to Figure 4 was plotted to visualize the classification accuracy of the Support Vector Classifier (SVC) classifier as a function of the number of iterations during the Particle Swarm Optimization (PSO) algorithm. This graph illustrates the improvement in accuracy as the PSO algorithm iteratively selects the most relevant features for the classification task. It depicts the convergence behavior of the algorithm and provides valuable insights into the efficiency and effectiveness of the feature selection process.

The convergence graph allows for the analysis of how the classification accuracy evolves over the iterations of the PSO algorithm. It helps to understand the rate at which the algorithm converges to an optimal solution and the stability of the solution once convergence is achieved. This information can be useful for tuning the algorithm parameters and assessing the overall performance of the proposed approach.

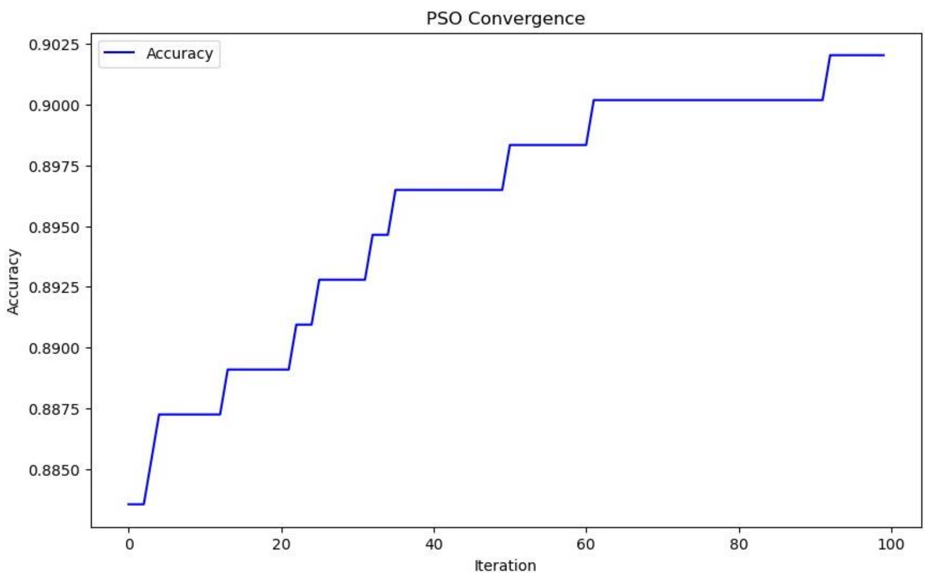

Figure 4. – PSO convergence graph ”accuracy as a function of iterations

3. Conclusion

In this study, we proposed a novel approach for recognizing isolated handwritten Arabic characters by combining feature extraction from two deep learning models (EfficientNet B2 and DenseNet 201), followed by feature selection using the Particle Swarm Optimization (PSO) algorithm, and classification with a Support Vector Classifier (SVC).

The concatenation of features from EfficientNet B2 and DenseNet 201 aimed to capture complementary information and create a comprehensive feature representation. The PSO algorithm proved effective in identifying the most relevant features, leading to improved classification performance.

The experimental results demonstrated the efficacy of the proposed approach, achieving an overall test accuracy of 90.20% in recognizing isolated handwritten Arabic characters from the HMBDv1 dataset. The macro-averaged precision, recall, and F1-score of 90.00% further validated the system's balanced performance across different classes. Most characters attained high classification scores, with several achieving perfect precision, recall, and F1-scores of 1.00, indicating accurate recognition. However, some characters exhibited relatively lower performance, suggesting the need for further refinement and improvements.

The convergence analysis illustrated the progressive improvement in classification accuracy as the PSO algorithm iteratively selected the optimal feature subset. This visualization provided valuable insights into the algorithm's convergence behavior and the effectiveness of the feature selection process.

REFERENCES

1. Comrie, B. (Ed.). (2003). The Major Languages of South Asia, the Middle East, and Africa. Routledge.
2. Lorigo, L. M., & Govindaraju, V. (2006). Offline Arabic handwriting recognition: a survey. IEEE transactions on pattern analysis and machine intelligence, 28(5), 712–724.
3. El-Sawy, A., Loey, M., & El-Bakry, H. (2017). Arabic handwritten characters recognition using convolutional neural network. WSEAS Transactions on Computer Research, 5(1), 11–19.
4. Marcelli, A., Parziale, A., Santoro, A.: Modeling handwriting style: a preliminary investigation. In: 2012 International Conference on Frontiers in Handwriting Recognition, p. 411–416 (2012). IEEE.
5. Parwej, Shams, M., Elsonbaty, A., ElSawy, W., et al.: Arabic handwritten character recognition based on convolution neural networks and support vector machine. arXiv preprint arXiv:2009.13450 (2020).
6. Alkhateeb, J.H.: An effective deep learning approach for improving off-line arabic handwritten character recognition. International Journal of Software Engineering and Computer Systems 6(2), 53–61 (2020).
7. Balaha, H.M., Ali, H.A., Saraya, M., Badawy, M.: A new arabic handwritten character recognition deep learning system (ahcr-dls). Neural Computing and Applications 33, 6325–6367 (2021).
8. Alrobah, N., Albahli, S.: A hybrid deep model for recognizing arabic handwritten characters. IEEE Access 9, 87058–87069 (2021).
9. Nayef, B.H., Abdullah, S.N.H.S., Sulaiman, R., Alyasseri, Z.A.A.: Optimized leaky relu for handwritten arabic character recognition using convolution neural networks. Multimedia Tools and Applications, 1–30 (2022).
10. Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R., Mittal, A.: Pneumonia detection using cnn based feature extraction. In: 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), p. 1–7 (2019). IEE.
11. Munien, C., Viriri, S.: Classification of hematoxylin and eosin-stained breast cancer histology microscopy images using transfer learning with efficientnets. Computational Intelligence and Neuroscience 2021 (2021).

12. Zhu, Y., & Newsam, S. (2017, September). Densenet for dense flow. In 2017 IEEE international conference on image processing (ICIP) (p. 790–794). IEEE.
13. Koonce, B., & Koonce, B. (2021). EfficientNet. Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization, 109–123.
14. Juneja, M., Nagar, S.: Particle swarm optimization algorithm and its parameters: A review. In: 2016 International Conference on Control, Computing, Communication and Materials (ICCCCM), p. 1–5 (2016). IEEE.
15. Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. Ieee Access, 10, 10031–10061.
16. Krisshna, N.A., Deepak, V.K., Manikantan, K., Ramachandran, S.: Face recognition using transform domain feature extraction and pso-based feature selection. Applied Soft Computing 22, 141–161 (2014).
17. Giacinto, G., Perdisci, R., Del Rio, M., Roli, F.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Information Fusion 9(1), 69–82 (2008).
18. Lau, K., Wu, Q.: Online training of support vector classifier. Pattern Recognition 36(8), 1913–1920 (2003).
19. Nishat, M.M., Faisal, F., Hasan, T., Karim, M.F.B., Islam, Z., Shagor, M.R.K.: An investigative approach to employ support vector classifier as a potential detector of brain cancer from mri dataset. In: 2021 International Conference on Electronics, Communications and Information Technology (ICECIT), pp. 1–4 (2021). IEEE.
20. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings 19, pp. 1015–1021 (2006). Springe.
21. Chicco, D., T¨otsch, N., Jurman, G.: The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData mining 14(1), 1–22 (2021).