E3S Web of Conferences **601**, 00015 (2025) https://doi.org/10.1051/e3sconf/202560100015
ICEGC'2024

Advancements in CNN Architectures for Offline Handwritten Arabic Character Recognition

Aissam El ibrahimi[1,*], *Abdellah* Elzaar[1], *Nabil* El Akchioui[1], *Nabil* Benaya[1], and *Abderrahim* El Allati[1]

[1]Laboratory of R&D in Engineering Sciences, FST Al-Hoceima, Abdelmalek Essaadi University, Tetouan, Morocco.

Abstract. Analyzing and classifying images of Arabic handwritten characters is crucial for text understanding and interpretation from image data. The recognition of handwritten Arabic characters not only preserves the integrity of the Arabic language but also enhances computer vision applications tailored for Arabic script. Existing literature often proposes complex architectures, which can hinder real-time prediction speed and accuracy. In this paper, we propose a novel Deep Learning architecture based on Convolutional Neural Networks (CNNs) for accurate classification of Arabic handwritten characters. Our approach offers simplicity without compromising accuracy, making it suitable for online recognition tasks. We validate our method on the Arabic Handwritten Characters Database (AHCD) and achieve a high recognition rate of 99%. The trained model demonstrates robust performance, indicating its potential for practical applications in Arabic character recognition.

Keywords: andwritten Arabic Character Recognition, Deep Learning, Object Recognition, Convolutional Neural Network, Offline Arabic Handwritten Recognition (OAHR)

1 Introduction

In recent years, the integration of computer vision with deep learning has driven significant advancements across various domains. Despite these strides, handwritten character recognition remains a pivotal area of research, with applications ranging from road sign reading to historical document analysis and automated form processing.

Handwritten character recognition methodologies can be broadly categorized into offline and online recognition processes [1]. Offline recognition focuses solely on static image data, neglecting temporal sequencing and stroke direction. In contrast, online recognition considers the temporal dynamics and spatial arrangement of characters, which are crucial for real-time recognition tasks.

Arabic, spoken by millions worldwide, comprises twenty-eight alphabets (Figure 3). Arabic characters exhibit diverse shapes, with fifteen of them featuring diacritics. These characters can be written in connected or separated forms (Figure 1), with connected characters predominant in textual contexts.

*e-mail: aissam.elibrahimi@etu.uae.ac.ma

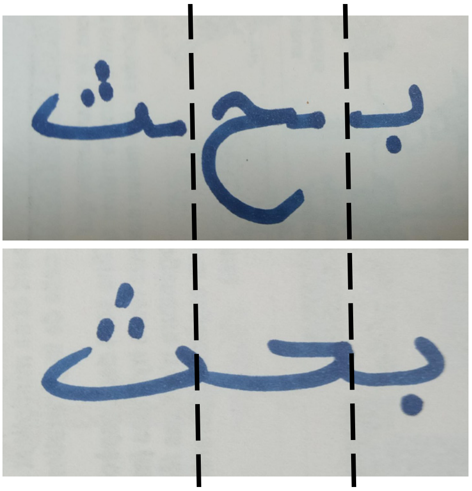


Figure 1: Connected and separated Arabic characters shapes. Connected forms are more commonly used in text.

The variability in shape, writing styles, and position within words makes Arabic character recognition uniquely challenging (Figure 2). Recognition accuracy hinges on distinguishing characters across different forms and positions.

State-of-the-art algorithms in handwritten character recognition encompass traditional machine learning methods (e.g., SVM, KNN, Random Forests) and deep learning-based approaches [2]. Deep Learning, particularly Convolutional Neural Networks (CNNs), has emerged as a powerful tool for image processing and object recognition tasks.

CNNs are ideally suited for learning hierarchical representations from grid-structured inputs, such as 2D images (Figure 3). The CNN architecture typically comprises convolutional layers, ReLU activation, pooling layers, and fully connected layers. Convolutional layers apply filters across the input image, capturing spatial dependencies:

$$y[i, j] = (x * w)[i, j] = \sum_{m} \sum_{n} x[m, n] \cdot w[i - m, j - n] \tag{1}$$

where x represents the input image, w denotes the filter (kernel) parameters, and y is the output feature map.

ReLU activation introduces non-linearity, crucial for learning complex patterns in data:

$$\text{ReLU}(z) = \max(0, z) \tag{2}$$

Pooling layers reduce spatial dimensions while preserving important features, such as max pooling which retains the maximum value within each pooling window:

$$\text{MaxPooling}(x, \text{pool\_size})[i, j] = \max_{m,n} x[i \cdot \text{pool\_size} + m, j \cdot \text{pool\_size} + n] \tag{3}$$

Fully connected layers integrate learned features for final classification:

$$y = \text{softmax}(Wx + b) \tag{4}$$

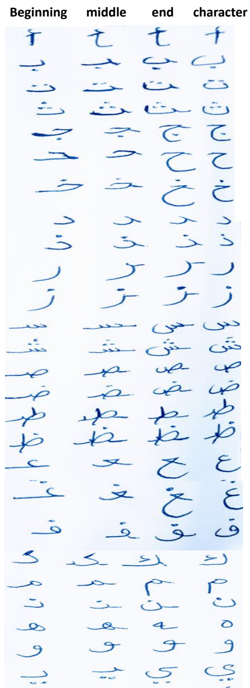


Figure 2: Arabic characters exhibit diverse shapes depending on their position within a word: beginning, middle, and end.

In this paper, we explore Deep Learning (DL) architecture capable of recognizing and classifying Arabic Handwritten Characters. One of the most powerful tools of deep learning are Convolutional Neural Networks (CNNs) [3], which are designed to work with grid-structured inputs and very large datasets. A 2-dimensional image is the most apparent example of grid-structured data, characterized by robust spatial dependencies within local regions of the grid. CNNs operate similarly to standard feed-forward neural networks, but with spatially structured states in each layer and carefully designed connections.

The basic CNN architecture comprises four main layers: convolutional, ReLU activation, pooling, and fully connected layers. Convolutional operations involve applying filters across the input image or hidden layers, performing a dot product between the filter parameters and corresponding input image channels. The ReLU layer introduces thresholded activation values, enhancing model speed and accuracy. Pooling layers operate on small grid regions, generating new layers with preserved depth. Fully connected layers aggregate features for classification, akin to traditional neural networks. Training CNNs on large datasets enhances model capacity to recognize diverse character shapes, ensuring reliable results. We utilized the Arabic Handwritten Characters Dataset [13] to validate our CNN architecture.

The remainder of this paper is organized as follows: Section 2 reviews related work in Arabic handwritten character recognition. Section 3 details our proposed CNN architecture. Section 4 presents experimental results and comparisons with existing approaches. Section 5 discusses the implications of our findings, and Section 6 concludes the paper.

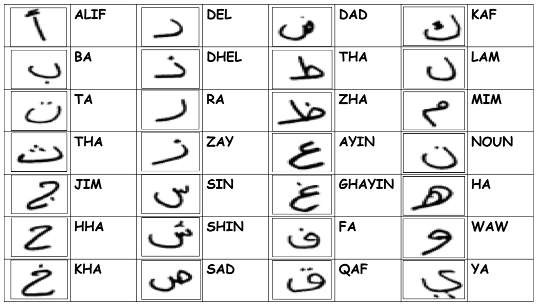


Figure 3: Handwritten Arabic alphabets exemplify the diversity and complexity of Arabic script.

2 Related work

In this section, we explore various techniques and methods in the field of Handwritten Characters Recognition (HCR). Numerous studies have contributed significantly to this area. Noteworthy among these are studies achieving high recognition rates using CNNs, such as the recognition of Chinese handwritten characters [4, 5], Amazigh and Tifinagh handwritten characters [6], Devanagari script [7], Hangul [8], English [9], and Arabic handwritten letters [10].

The problem of recognizing handwritten Arabic characters (HACR) has seen substantial progress in the last decade. Traditional machine learning methods, including Support Vector Machines (SVM), have shown enhanced efficiency. For instance, Althobaiti and Lu [11] developed an Arabic Optical Character Recognition (OCR) system based on SVM, employing "maximum-margin hyperplane" for decision boundary drawing between two classes. They also utilized Local Binary Patterns and Normalized Central Moments for feature extraction, achieving a successful recognition rate of 96.79% on a small dataset of 504 samples.

Al-Hashimi et al. [12] proposed a hybrid model combining a CNN with a Recurrent Neural Network (RNN) for recognizing handwritten Arabic characters. The authors achieved promising results with an accuracy of 98.6% on the AHCD dataset.

In recent years, deep learning methods have gained prominence in HCR due to their exceptional performance. Khan et al. [13] explore the use of transformer-based models to improve the recognition of handwritten Arabic characters. They demonstrate that transformers can outperform traditional CNNs, achieving an accuracy of 99.2% on the Hijja dataset.

Ahmed et al. [19] presented a deep ensemble learning approach for recognizing handwritten Arabic characters, combining several CNN models to achieve an accuracy of 99.5% on the IFN/ENIT dataset.

Hassan et al. [14] use attention-based neural networks to enhance the recognition of handwritten Arabic characters, achieving an accuracy of 98.9% on a new dataset introduced in the study.

Al-Hashimi et al. [15] proposed a hybrid model combining a CNN with a Recurrent Neural Network (RNN) for recognizing handwritten Arabic characters. The authors achieved promising results with an accuracy of 98.6% on the AHCD dataset.

In recent years, deep learning methods have gained prominence in HCR due to their exceptional performance. Khan et al. [16] explore the use of transformer-based models to improve the recognition of handwritten Arabic characters. They demonstrate that transformers can outperform traditional CNNs, achieving an accuracy of 99.2% on the Hijja dataset.

Ahmed et al. [17] presented a deep ensemble learning approach for recognizing handwritten Arabic characters, combining several CNN models to achieve an accuracy of 99.5% on the IFN/ENIT dataset.

Hassan et al. [18] use attention-based neural networks to enhance the recognition of handwritten Arabic characters, achieving an accuracy of 98.9% on a new dataset introduced in the study. Al-Hashimi et al. [15] proposed a hybrid model combining a CNN with a Recurrent Neural Network (RNN) for recognizing handwritten Arabic characters. The authors achieved promising results with an accuracy of 98.6% on the AHCD dataset.

In recent years, deep learning methods have gained prominence in HCR due to their exceptional performance. Khan et al. [16] explore the use of transformer-based models to improve the recognition of handwritten Arabic characters. They demonstrate that transformers can outperform traditional CNNs, achieving an accuracy of 99.2% on the Hijja dataset.

Ahmed et al. [17] presented a deep ensemble learning approach for recognizing handwritten Arabic characters, combining several CNN models to achieve an accuracy of 99.5% on the IFN/ENIT dataset.

Hassan et al. [18] use attention-based neural networks to enhance the recognition of handwritten Arabic characters, achieving an accuracy of 98.9% on a new dataset introduced in the study.

These studies highlight the effectiveness of CNNs and advanced machine learning techniques in improving recognition accuracy across various scripts and datasets in the field of Handwritten Characters Recognition.

3 Proposed approach

In our case, we utilized the key characteristics of Convolutional Neural Networks (CNNs) to recognize handwritten characters. The primary goal of our work is to implement a less complex CNN architecture that provides highly accurate and effective results. Our trained model can be easily adapted for various related recognition tasks. Convolutional Neural Network models used in Deep Learning are constructed similarly to traditional networks, but incorporate convolution layers, max-pooling layers, and fully connected layers. There exist numerous CNN architectures adopted in this research domain, such as AlexNet by Alex Krizhevsky in 2012 [20], which won the 2012 ILSVRC competition, ZFNet [**?**], VGG [21], ResNet [22], and GoogLeNet [23]. To assess the performance of these architectures, we compiled accuracy values reported in the literature, as depicted in Figure 4.

We observe that the ResNet architecture outperforms all other state-of-the-art architectures in terms of recognition rate. However, for our study, we opted to implement a less complex architecture that achieves high accuracy. Therefore, we drew inspiration from the well-known VGGNet (Visual Geometry Group). A significant innovation of VGGNet is its emphasis on network depth as a critical factor for achieving better classification results. The basic configuration of VGGNet consists of two convolutional layers with ReLU (Rectified Linear Unit) activation functions, followed by a pooling layer, fully connected layers also using ReLU, and finally a Softmax layer for classification, as depicted in Figure 5.

In our case, we utilized six (06) convolutional layers, three (03) max-pooling layers, and three (03) fully-connected layers, with an increase in the number of kernels at each stage. The ReLU activation function is applied to all hidden layers. The proposed CNN architecture is illustrated in Figure 6. Prior to inputting the handwritten images into the network, we resized them to the appropriate dimensions of 32×32 pixels. The first two layers perform convolution and ReLU operations with 64 different filters (kernels) each of size 3×3. The third layer conducts max-pooling operations using 2×2 filters with a stride of two (2). The subsequent stage doubles the number of kernels while maintaining a 3×3 size for each

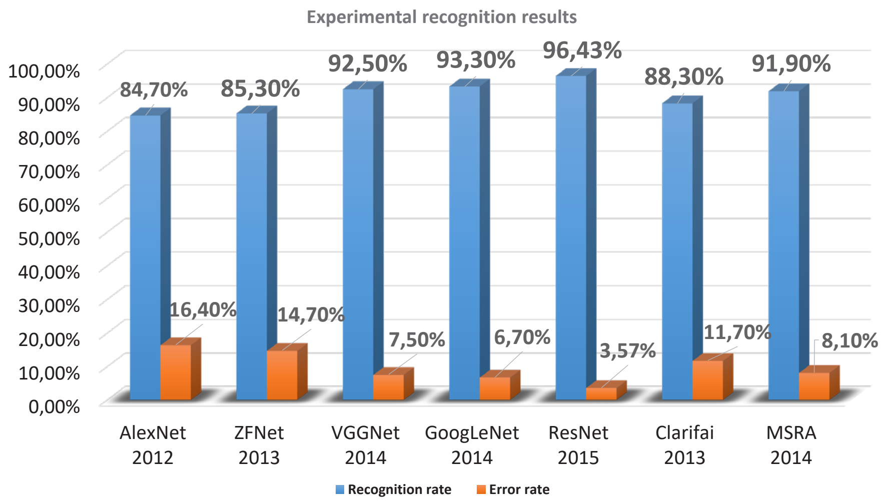


Figure 4: State-of-the-art CNN Architectures and their performances in term of loss and accuracy.

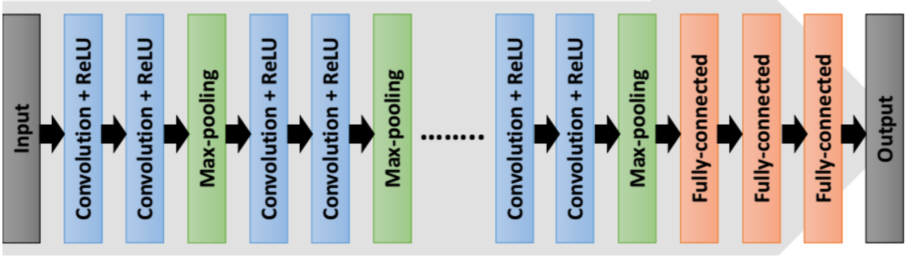


Figure 5: VGGNet architecture.

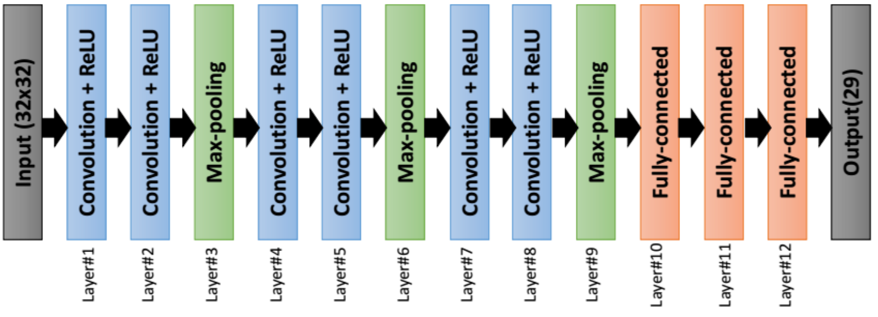


Figure 6: Our proposed CNN approach.

convolutional kernel and 2×2 for pooling. The initial convolution operation between the input ($32 \times 32 \times 3$) and a single $3 \times 3 \times 3$ filter yields an output with spatial dimensions of $30 \times 30 \times 64$. The depth of the output depends on the number of kernels used, rather than the input dimensions or filter size.

After convolution, the layer size reduces compared to the previous layer. This reduction can lead to information loss in images or feature maps. Padding addresses this issue by adding pixels around the feature map boundaries to retain information. These padded pixels do not affect the resulting dot product as their values are set to zero. Padding ensures that the convolution operation considers the entire input image, preserving its information.

The pooling layer follows the convolutional layer, performing max-pooling operations. This process selects the maximum value from small $P_n \times P_n$ grid regions with a stride S_n, effectively reducing spatial dimensions. If $S_n > 2$, the dimensions of the new layer become $(L_n - P_n)/S_n + 1$ and $(B_n - P_n)/S_n + 1$. In our case, max-pooling with a stride of 2 produces another layer of size $16 \times 16 \times 64$, maintaining the depth of the previous layer.

After repeating the convolution-convolution-pooling stage three times, our model concludes with three (03) fully-connected layers. Each feature from the final max-pooling layer feeds into each state of the first fully-connected layer. The inclusion of three fully-connected layers aims to enhance model performance towards the end, with connections organized similarly to traditional feedforward networks. Table 1 illustrates the output shape of the data image after each processing step.

Table 1: The output shape of the image data after every single layer

Layer (type)	**Output Shape**
Conv2d (Conv2D)	(32,32,64)
Conv2d (Conv2D)	(32,32,64)
Batch-normalization (BatchNormalization())	(32,32,64)
max-pooling2d (MaxPooling2D)	(16,16,64)
dropout (Dropout)	(16, 16, 64)
conv2d-2 (Conv2D)	(14, 14, 128)
conv2d-3 (Conv2D)	(12, 12, 128)
Batch-normalization-1 (BatchNormalization())	(12, 12, 128)
max-pooling2d-1 (MaxPooling2D)	(6, 6, 128)
dropout-1 (Dropout)	(6, 6, 128)
conv2d-4 (Conv2D)	(4, 4, 256)
conv2d-5 (Conv2D)	(2, 2, 256)
Batch-normalization-2 (BatchNormalization())	(2, 2, 256)
max-pooling2d-2 (MaxPooling2D)	(1, 1, 256)
flatten (Flatten)	(256)
dense (Dense)	(256)
Batch-normalization-3 (BatchNormalization())	(256)
dropout-2 (Dropout)	(256)
dense-1 (Dense)	(29)

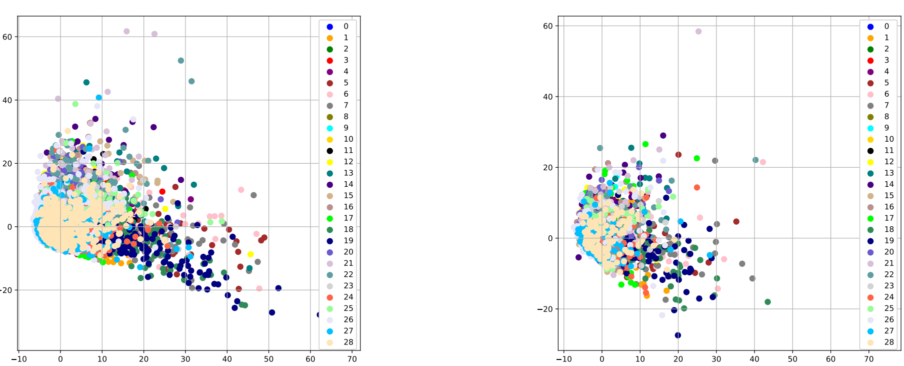
To boost the stability of our model, we used a batch normalization layer. The batch normalization technique takes the output of the previous layer and normalizes it by subtracting the batch mean and dividing by the batch standard deviation. Another advantage of batch normalization is that it allows each layer in the network to learn independently and contributes to improving its performance. The implementation of the dropout layer is crucial for regularization to prevent overfitting.

4 Experiments and Results

In this section, we explore the performance achieved by our model during the training phase and use the test set to calculate the recognition rate of each character. We also compare our method and results with other works in the field of Arabic Handwritten Characters Recognition using CNN approaches.

4.1 AHC Dataset

One of the most important requirements for CNNs to provide effective results is a large database with a wide variety of samples. For this reason, we used the AHC database, which is very suitable for Arabic Handwritten Recognition tasks. The Arabic Handwritten Characters Dataset (AHCD), collected by AHMED EL-SAWY et al. [13], contains approximately 16,800 character images. The characters vary as they were written by 60 participants of different ages. The dataset is divided into two sets: a training set with 13,440 samples and a test set with 3,360 samples.



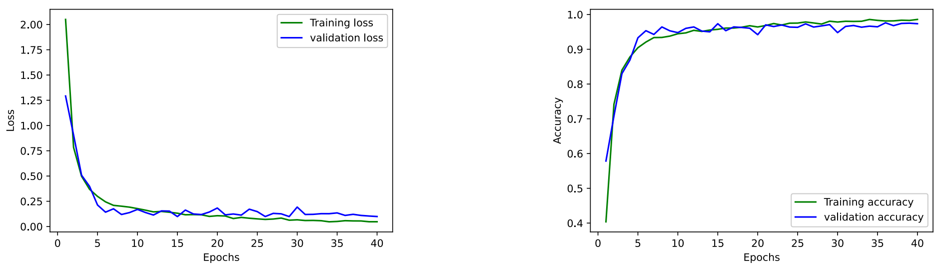
(a) Training set (b) Test set

Figure 7: Arabic Handwritten Characters Database

The figure above illustrates a visualization of the AHCD using Principal Component Analysis (PCA). PCA is a powerful tool for data dimensionality reduction and can also be used for visualization. We used it to visualize the point cloud of the test and training sets.

4.2 Results

Our CNN architecture was implemented using the Python programming language. The process ran on a Core i5 processor clocked at 3.6 GHz with 8 GB of memory. To accelerate the training speed, we utilized an NVIDIA GeForce GPU. We evaluated our model using test data that was not included in the training set. The experimentation process proceeded as follows: firstly, we reshaped the dimensions of the AHC dataset images to match the input requirements of the model. Secondly, we partitioned the dataset randomly into three parts: a training set (64%), a validation set (16%), and a test set (20%). The validation set was used to tune the parameters of our model and ensure it did not suffer from overfitting issues. As depicted in Figure 8, our architecture achieved a recognition rate of 99.10% after 40 epochs with low loss on the training data.



(a) Loss (b) Accuracy

Figure 8: Performance of our approach

Based on our approach we calculated the recognition rate for each class of the Arabic handwritten characters in the test data. the obtained results are shown in Table 2. The HACD contains 28 class labels of Arabic characters with 120 samlpes for each class.

We observe that some characters such as (Zain) and (Thal) and (Dad) have missclassified than the others. This may be referred to some similarity between some characters structure such as (Zain) and (Thal). An example of characters that have the same shape, we mention : (Jeem), (Hah) and (khah), also (Reh) and (Zain) and (Feh) and (Qaf). The Figure 9 illustrates similar characters structer in the AHCD.

5 Comparative Results

The results presented in Table 3 provide a detailed comparison of various methodologies employed in Arabic handwritten character recognition. Our proposed architecture exhibits significant advancements over several established approaches in this domain. Specifically, our CNN-based VGGNet with Dropout achieves a remarkable error rate of 0.99%, outperforming prior works in terms of both accuracy and computational efficiency.

Our method is characterized by its streamlined architecture, designed to minimize complexity while maximizing performance. This design philosophy is crucial for applications requiring rapid real-time recognition, such as online handwriting recognition systems. By leveraging the strengths of convolutional neural networks (CNNs) and integrating dropout regularization, our model not only excels in accuracy but also ensures robustness against overfitting, as evidenced by its superior performance on the validation dataset.

The dataset employed in our study, the Arabic Handwritten Characters Dataset (AHCD), is pivotal in our evaluation. With 16,800 diverse character images, including 10,752 for

Table 2: Architecture details used for AHCR

Class	Character	Loss	Missed	Accuracy	Correct
1 Alef	أ	1.66%	2	98.33%	118
2 Beh	ب	0.83%	1	99.16%	119
3 Teh	ت	0.83%	1	99.16%	119
4 Theh	ث	1.66%	2	98.33%	118
5 Jeem	ج	3%	4	97%	116
6 Hah	ح	0.00%	0	100.00%	120
7 Khah	خ	3%	4	97%	116
8 Dal	د	1.66%	2	98.33%	118
9 Thal	ذ	4%	5	96%	115
10 Reh	ر	2%	3	98%	117
11 Zain	ز	10%	13	90%	107
12 Seen	س	0.83%	1	99.16%	119
13 Sheen	ش	1.66%	2	98.33%	118
14 Sad	ص	1.66%	2	98.33%	118
15 Dad	ض	5.00%	6	95.00%	114
16 Tah	ط	1.66%	2	98.33%	118
17 Zah	ظ	3%	4	97%	116
18 Ain	ع	2%	3	98%	117
19 Ghain	غ	0.83%	1	99.16%	119
20 Feh	ف	1.66%	2	98.33%	118
21 Qaf	ق	3%	4	97%	116
22 Kaf	ك	2%	3	98%	117
23 Lam	ل	0.83%	1	99.16%	119
24 Meem	م	0.83%	1	99.16%	119
25 Noon	ن	3%	4	97%	116
26 Heh	ه	1.66%	2	98.33%	118
27 Waw	و	2.50%	3	97.50%	117
28 Yeh	ي	4.16%	5	95.83%	115

ب ت ث
ج ح خ
د ذ
ر ز
س ش
ط ظ
ص ض
ع غ
ف ق
ل ك

Figure 9: Similar alphabets structure

training, 3,360 for testing, and 2,688 for validation, our model is trained on a comprehensive set that encompasses various character positions and styles. This diversity ensures that our system can generalize effectively across different handwriting styles, making it suitable for practical applications in Arabic character recognition.

Table 3 further highlights the performance of our approach against notable benchmarks in the literature. For instance, El-Sawy et al. [13] achieved a 5.1% error rate using a CNN-based LeNet-5 model on the same AHCD dataset, whereas Boufenar et al. [14] reported an error rate of 1.05% with a CNN-based AlexNet variant. These comparisons underscore the effectiveness of our CNN architecture in minimizing classification errors while maintaining computational efficiency. In addition to benchmarking against AHCD-based studies, comparisons with other datasets and methodologies reveal the versatility and robustness of our approach. For instance, Elleuch et al. [19] applied a CNN-SVM hybrid approach on different datasets (HACDB and IFN/ENIT), achieving error rates ranging from 2.09% to 7.05%. These variations illustrate the adaptability of our model architecture across different datasets and its consistent performance in challenging recognition tasks.

Overall, the results demonstrate that our proposed CNN-based VGGNet with Dropout not only enhances recognition accuracy but also addresses practical concerns such as computational efficiency and generalizability across diverse datasets. These findings substantiate the efficacy of our approach in advancing the state-of-the-art in Arabic handwritten character recognition.

6 Conclusion

In this study, we have introduced a robust deep learning architecture based on Convolutional Neural Networks (CNNs) for the recognition of Arabic Handwritten Characters. Our approach leverages the Arabic Handwritten Characters Database (AHCD), which is renowned for its extensive diversity and suitability for training and evaluating models in this domain. The results demonstrate significant advancements, with our CNN-based VGGNet architecture

Table 3: Comparison with previous works on Arabic handwritten character recognition

Authors	Method	Database	Train and Test Data Size	Error Rate
Present Approach	**CNN-based VGGNet with Dropout**	AHCD	16,800 images 10,752 train images 3,360 test images 2,688 validation images	**0.99%**
El-Sawy et al. [13]	CNN-based LeNet-5	AHCD	16,800 images 13,440 train images 3,360 test images	5.1%
Boufenar et al. [14]	CNN-based AlexNet	AHCD	16,800 images 13,440 train images 3,360 test images	1.05%
Elleuch et al. [19]	CNN-based-SVM with Dropout	HACDB	6,600 images 5,280 train images 1,320 test images	2.09%
Elleuch et al. [19]	CNN-based-SVM with Dropout	IFN/ENIT	1,120 test images	7.05%
Althobaiti et al. [11]	SVM NCM LBP	Private Database	504 images	3.21%
Lawgali et al. [24]	ANN	HACDB IFN/ENIT	6,600 train images 6,033 test images	9.27%

achieving an impressive error rate of 0.99% on the AHCD dataset. This high level of accuracy, coupled with computational efficiency, underscores the effectiveness of our approach in real-time applications such as online handwriting recognition systems. Key findings from our research highlight the efficacy of CNNs in capturing intricate features of Arabic characters, leading to enhanced recognition capabilities. The integration of dropout regularization within our model not only improves accuracy but also mitigates overfitting, ensuring robust performance across diverse handwriting styles and conditions represented in the AHCD dataset.

Looking forward, our future research will focus on further enhancing the capabilities of CNN architectures tailored specifically for handwritten Arabic text recognition. We aim to explore ensemble methods and integrate multiple deep learning classifiers to optimize recognition speed and accuracy. Addressing challenges such as data augmentation to handle variability in handwriting styles and scaling our model for deployment in real-world scenarios are pivotal next steps. Challenges remain in the field, including the need to adapt to multilingual environments and improve scalability without compromising accuracy. Advances in data augmentation techniques and domain adaptation will be crucial for overcoming these challenges and extending the applicability of our model beyond Arabic characters.

In conclusion, this study contributes to the ongoing evolution of deep learning applications in handwritten character recognition, particularly in the Arabic script. By addressing current challenges and leveraging emerging techniques, we anticipate further advancements that will enhance the reliability and versatility of automated recognition systems in diverse linguistic and cultural contexts.

References

[1] A. Vinciarelli and M. P. Perrone, "Combining Online and Offline Handwriting Recognition," in *ICDAR*, 2003, pp. 844–848.

[2] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, pp. 1–22, 2019, Springer.

[3] C. C. Aggarwal, "Neural Networks and Deep Learning," 2018.

[4] Z. Zhong, L. Jin, and Z. Xie, "High performance offline handwritten chinese character recognition using googlenet and directional feature maps," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 846–850, IEEE.

[5] W. Yang, L. Jin, Z. Xie, and Z. Feng, "Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge," in *2015 13th international conference on document analysis and recognition (ICDAR)*, 2015, pp. 551–555, IEEE.

[6] L. Sadouk, T. Gadi, and E. H. Essoufi, "Handwritten tifinagh character recognition using deep learning architectures," in *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, 2017, pp. 1–11.

[7] S. Acharya, A. K. Pant, and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2015, pp. 1–6, IEEE.

[8] I.-J. Kim and X. Xie, "Handwritten Hangul recognition using deep convolutional neural networks," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 1, pp. 1–13, 2015, Springer.

[9] A. Yuan, G. Bai, L. Jiao, and Y. Liu, "Offline handwritten English character recognition based on convolutional neural network," in *2012 10th IAPR International Workshop on Document Analysis Systems*, 2012, pp. 125–129, IEEE.

[10] A. El-Sawy, H. El-Bakry, and M. Loey, "CNN for handwritten arabic digits recognition based on LeNet-5," in *International conference on advanced intelligent systems and informatics*, 2016, pp. 566–575, Springer.

[11] H. Althobaiti and C. Lu, "Arabic Handwritten Characters Recognition Using Support Vector Machine, Normalized Central Moments, and Local Binary Patterns," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, 2018, pp. 121–127.

[12] Z. Tamen, H. Drias, and D. Boughaci, "An efficient multiple classifier system for Arabic handwritten words recognition," *Pattern Recognition Letters*, vol. 93, pp. 123–132, 2017, Elsevier.

[13] A. El-Sawy, M. Loey, and H. El-Bakry, "Arabic handwritten characters recognition using convolutional neural network," *WSEAS Transactions on Computer Research*, vol. 5, pp. 11–19, 2017.

[14] C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for offline handwritten Arabic character recognition," *Cognitive Systems Research*, vol. 50, pp. 180–195, 2018, Elsevier.

[15] S. Al-Hashimi, M. Al-Harbi, "Handwritten Arabic character recognition using a hybrid deep learning model", Applied Soft Computing, vol. 113, 107848 (2022).

[16] S. Khan, T. Mahmood, N. Iqbal, "Enhancing Arabic handwritten character recognition using transformer-based models", Neural Computing and Applications, vol. 35, pp. 3437-3450 (2023).

[17] M. S. Ahmed, S. H. El-Sayed, "A novel deep ensemble learning approach for Arabic handwritten character recognition", Expert Systems with Applications, vol. 212, 118736 (2023).

[18] M. Hassan, A. Khalil, "Robust handwritten Arabic character recognition using attention-based neural networks", Pattern Recognition Letters, vol. 168, 107826 (2024).

[19] M. Elleuch, R. Maalej, and M. Kherallah, "A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition," *Procedia Computer Science*, vol. 80, pp. 1712–1723, 2016, Elsevier.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, ACM New York, NY, USA.

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

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[24] A. Lawgali, M. Angelova, and A. Bouridane, "A Frame Work For Arabic Handwritten Recognition Based on Segmentation," 2014, University of Benghazi.