Baghdad Science Journal

Volume 23 | Issue 3 Article 29

3-25-2026

Recognition of Arabic Derived Handwritten Letters using a VGG-like Convolutional Neural Network

Mouhssine El Atillah
Computer Systems Engineering, Mathematics and Applications (ISIMA), Polydisciplinary Faculty of Taroudant, University Ibn Zohr, Taroudant, Morocco, m.elatillah@uiz.ac.ma

How to Cite this Article
El Atillah, Mouhssine (2026) "Recognition of Arabic Derived Handwritten Letters using a VGG-like Convolutional Neural Network," *Baghdad Science Journal*: Vol. 23: Iss. 3, Article 29.
DOI: https://doi.org/10.21123/2411-7986.5254

BAGHDAD SCIENCE JOURNAL 2026;23(3):1084–1099

RESEARCH ARTICLE

Recognition of Arabic Derived Handwritten Letters using a VGG-like Convolutional Neural Network

Mouhssine El Atillah

Computer Systems Engineering, Mathematics and Applications (ISIMA), Polydisciplinary Faculty of Taroudant, University Ibn Zohr, Taroudant, Morocco

ABSTRACT

Due to its complexity, the Arabic language and its extensions build a fertile field of research in the field of artificial intelligence in general and optical character recognition (OCR) specifically. There are several languages that use the Arabic alphabet in their manuscripts. These languages innovated new letters to pronounce sounds not found in the Arabic language. These letters are called 'Arabic-derived letters'. To enrich the Arabic language, we can use these letters to know the true pronunciation of intrusive words in the Arabic language. This article deals with the Arabic-derived letters (ADL) dataset. It is a new dataset that consists of 55440 scanned images of papers written by 30 participants of different ages, with a data augmentation technique to increase the number of images. This study aims to evaluate and compare the effectiveness of different convolutional neural network architectures for ADL recognition, focusing on accuracy, robustness, and generalization capability. Three architectures were implemented: LeNet, a simplified ResNet model with residual blocks, and a deep VGG-Like network. Training was limited to 40 epochs with early stopping after 5 epochs without improvement. Experimental results show that the VGG-Like model achieves the best performance with 99.61% accuracy in validation, closely followed by ResNet with an accuracy of 98.98%. In contrast, LeNet performs less efficiently by 96.43%. These results clearly demonstrate that modern and deep architectures provide better accuracy and robustness for the classification of handwritten characters.

Keywords: Arabic handwritten, Convolutional neural network, Derived Arabic letters, Optical character recognition, VGG-Like, ResNet, LeNet

Introduction

In computer vision, handwritten character recognition is a key task that plays a main role in the development of handwriting recognition systems.[1] While significant progress has been made in recognizing characters from the original Arabic alphabet[2], the derived letters – ڠ, پ, چ, ژ, ڤ, گ – used in languages such as Persian, Urdu, Pashto, Kurdish, and Sindhi[3] are poorly studied. These derived letters are not simply extensions of the Arabic alphabet, but they play a crucial role in representing phonemes absent in Arabic.[4] The integration of these letters into Arabic characters allows for the proper pronunciation of intrusive words in the Arabic language. This plays a crucial role in enriching the Arabic language and also in distinguishing its original words.[5] The stability of Arabic writing over the years enhances its usability in various languages. In addition to this, the flexibility of the Arabic language highlights its adaptability and cultural importance compared to other languages. Despite the importance of these letters in the field of optical character recognition, studies that have dealt with these letters are almost negligible,[6] This is due to their limited use in particular linguistic contexts and also to the lack of a dataset for these letters. The recognition of these Arabic letters in general is difficult because of the particular shapes that present variations according to the location in the words. This is the case for the derived letters, which also have the

Received 29 March 2025; revised 16 July 2025; accepted 29 July 2025.
Available online 25 March 2026

E-mail address: m.elatillah@uiz.ac.ma (M. El Atillah).

https://doi.org/10.21123/2411-7986.5254

same shapes as the originals with small modifications. Therefore, the recognition of these letters plays an important role in the development of OCR systems for languages that use letters, and also for the Arabic language, which can use these letters to distinguish sounds coming from other languages.

This study proposes a VGG-Like[7] approach for the classification of the Arabic-derived letters dataset, which is a convolutional neural network (CNN) to differentiate complex characteristics present in visual images[8] and efficiently extracts and classifies features,[9] thus ensuring robust classification across different handwriting characters. This method not only emphasizes the importance of these derived letters, but also positions them as adding value to the Arabic script.[10] The proposed VGG-like architecture is directly inspired by the modular design of VGG-16, notably through the repeated use of 3 × 3 filters, conv-pool blocks, and a "deep but simple" structure. However, it has been slimmed down to better adapt to the reduced size of images, the limited number of classes, and to avoid overfitting on a smaller dataset than that of ImageNet. These adaptations fully justify the "VGG-like" name. This study provides an advanced classification technical solution, highlighting how derived letters can overcome linguistic gaps.[11] The integration of these letters in the Arabic language allows for a more flexible and inclusive system that can accommodate a wider variety of linguistic expressions. In addition to closing an important research gap, this study also opens the way for researchers in the OCR field to explore other less-studied manuscripts and their potential integration into broader linguistic frameworks. All this is to preserve and promote the linguistic heritage of the communities that use these unique letters by highlighting the importance of recognizing handwriting in these letters. The main contributions of this work are as follows:

- The establishment of a handwritten dataset dedicated to letters derived from the Arabic alphabet.
- The introduction of the VGG-Like model prior to classification of the ADL dataset.
- An experimental analysis demonstrating the performance of the proposed approach compared to other approaches (LeNet, ResNet).
- A contribution to the Arabic OCR literature by highlighting the shortcomings related to the processing of derived letters.

Related work

The remarkable advancement in deep learning techniques has played a major role in developing the field of optical character recognition (OCR) by becoming one of the most successful applications of those techniques.[12,13] However, the use of deep learning models in the recognition of Arabic manuscripts still insufficient.[14] Arabic is a rich language and one of the most widely spoken in the world.[15] Recently, the Arabian researchers have made significant advances in this area, spatially in Arabic letters and numerals.[16,17] The advanced techniques of deep learning algorithms are a fertile field to overcome the challenges related to the complexity and diversity of Arabic handwriting.

The paper in[6] also proposes an architecture that combines two methods to address the recognition of intrusive characters in Arabic manuscripts. Morphological gradient techniques are used in this method to identify letter contours using a multi-layer perceptron (MLP) network regularized by batch normalization. Designed for this study, this method achieved a classification accuracy of 96% on the validation set. However, this study has several constraints. It is limited to three derived letters: gaf (گ), ve (ڨ), and pe (پ), without extending to other derived letters such as چ, ژ, and ڠ. Also, this study uses a simple MLP and does not consider using another strong deep learning architecture, such as convolutional neural networks (CNNs), which have had remarkable success in extracting the features and classifying the handwritten characters. In contrast, the studies based on auto extraction deeper algorithms for Arabic character classification have proven their ability to identify the diverse and complex variations present in spoken Arabic handwritten texts. Such as the studies based on the IFN/ENIT dataset. Awni M, et al.,[18] propose a sequential transfer learning approach with the ResNet18 model for handwritten Arabic word recognition, exploring the effectiveness on the AlexU-W and IFN/ENIT datasets. Their method achieved recognition accuracy of up to 96.11% on the full dataset. Maalej R, et al.,[19] propose new architectures based on MDLSTM (with Dropout, ReLU و and Maxout) and data augmentation for offline Arabic handwriting recognition. Their MDLSTM-CTC-Maxout model achieved an accuracy of 92.59% on the IFN/ENIT dataset, improved to 93.46% with data augmentation. The challenge lies in the still-limited results of deep learning for offline Arabic handwriting, justifying the need to improve existing architectures. In this context, numerous studies have been conducted in the field of optical character recognition (OCR), including the study of Ghadekar P, et al.,[20] which proposes an automated system for handwritten exam correction integrating OCR and machine learning, achieving an accuracy of 95% for answer mapping, 92% for diagram matching, and 85% for grammar and context checking. This system is also accurate in the handwritten words recognition, which always exceeds 98%. Minor fluctuations in results are the only limitation of this study because of the eligible handwriting manuscript, which demonstrates the need for continuous enhancement of this model. A hybrid model for the recognition of Arabic air-written letters is proposed by Nahar KM, et al.,[21] using feature extraction, machine learning (ML), deep learning models, and OCR, which results in 88.8% accuracy using VGG16 and a neural network (NN). The fluid nature of writing (air-written letters), and the precise application of segmentation techniques for precise delineation of each word are the mere challenges raised here. A comparative study of deep convolutional neural networks (DCNN) for offline Arabic handwriting recognition is proposed by Ghanim TM, et al.,[22] showing a multi-stage cascaded system adopting a hierarchical clustering technique for reduced computational complexity and better classification outcomes. The system accuracy proves better results, though not mentioned, compared to existing systems, especially after using only 11% of the dataset. The use of CNN as an extractor and AlexNet as a classifier results in an 95.6% accuracy using the IFN/ENIT dataset, as mentioned in this study. The similarity of characters and writing style is the only challenge. Minoofam SAH, et al.,[23] suggest a DB-QM, an analytical framework for the qualitative evaluation of Persian and Arabic OCR datasets. They provide quantitative and qualitative evaluation criteria and categorize these datasets. The vital contribution of the paper is, thus, to offer a perspective on the various stages of both monolingual and multilingual character recognition and to target the gaps. The accuracy or performance of the proposed method is not highlighted, as it is an evaluation framework and not a recognition method. Therefore, the lack of categorization and identification of the datasets for the purpose of evaluation is the challenge faced in this regard. Sahlol AT, et al.,[24] suggest a hybrid approach for handwritten Arabic OCR, adopting a binary whale optimization algorithm and neighborhood fuzzy sets for feature selection. The proposed model accuracy is 96% for recognizing hand-written Arabic characters with 50% feature reduction, as implemented on the CENPARMI dataset. The proposed system limitation is the complexity of the Arabic character set, as it varies in shape, points, and writing style. Bouibed ML, et al.,[25] proposed an innovative hybrid system for author extraction in hand-written documents. This system integrates handcrafted features (MO-HOT) with deep features extracted from neural networks such as VGG-16 and MobileNetV2, using SVM-based dissimilarity learning for classification. In terms of accuracy, their approach outperforms the state-of-the-art by improving the TOP-2 score by up to 2% and the MAP score by at least 3.6% on diverse datasets such as CVL and ICDAR (2011, 2013, 2017). However, challenges include the complexity of efficiently aggregating complementary features and the difficulty of generalizing the system to a wide variety of writing styles. Aabed S, et al.,[26] propose an end-to-end, segmentation-free Arabic handwriting recognition model using a DCNN for feature extraction, a BLSTM for sequence recognition, and the CTC loss function on the KHATT database. The model achieves a recognition rate of 84% at the character level and 71% at the word level. Challenges include the cursive nature of Arabic, which their segmentation-free approach aims to overcome. Meddeb O, et al.,[27] propose a hybrid model for offline Arabic handwriting recognition, combining Hidden Markov Models (HMMs) and Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs). The model achieves an accuracy of 88%. The system is presented as very interesting and compared favorably to similar works. Typically, the system limitations for Arabic handwriting recognition encompass the complexity of ligatures and the variability of writing style. Meddeb O, et al.,[28] adopt the system of handwritten Arabic script recognition based on multi-features, the concept of fine segmentation up to the grapheme, and the hybrid LSTM-RNN + HMM classifier with linguistic post-processing. 91.77% accuracy is shown in the system for the IFN/ENIT database, but it still faces challenges in robustness with regard to complexity. The system of machine learning for Arabic handwriting recognition is proposed by Ritonga M, et al.,[29] including segmentation, image processing, and classification with the VGG16 CNN and SVM, displaying 99.33% classification accuracy with the VGG16 CNN. The cursive nature and variability of characters, and the complexity of the character segmentation process, are the challenges faced. Alqahtani FA,[30] proposes the Optimized Deep Learning Framework for Handwritten Arabic Character Recognition (ODLF-AHCRCOA). The method makes use of the Sobel filter for the image preprocessing, the modified LeNet-5 method for the feature extraction, and the LSTM classifier with the optimized hyperparameters by the Coati Optimization Algorithm (COA). The suggested method achieved a superior accuracy level of 98.60%, as demonstrated in the experimental validation process. The recognition of the Arabic script challenge lies in the different shapes that characters take depending on their location. The comparative analysis, conducted by Svendsen B, et al.,[31,32] on the different machine learning models, including SVM, KNN, and CNN, on the Norwegian Sign Language data set they created. Their results indicate that the SVM and CNN models

Table 1. Overview of state-of-the-art methods.

Study	Method	Accuracy
Mouhssine EL, et al. 2020 [6]	Morphological gradient + MLP	96%
Awni M, et al. 2022 [18]	Sequential Transfer Learning (ResNet18)	96.11%
Maalej R, et al. 2022 [19]	MDLSTM-CTC-Maxout	93.46%
Ghadekar P, et al. 2025 [20]	OCR + ML (RAG) hydride	95% : answer mapping. 92% : diagram matching. 85% : grammar and context checking. Consistently above 98% : handwritten word recognition.
Nahar KM, et al. 2023 [21]	ML + CNN + OCR (VGG16)	88.8%
Ghanim TM, et al. 2020 [22]	CNN + AlexNet	95.6%
Sahlol AT, et al. 2020 [24]	WOA + NRS (feature selection)	96%
Bouibed ML, et al. 2024 [25]	Hybrid (MO-HOT + DCNN + SVM)	Outperforms the state-of-the-art by improving the TOP-2 score by up to 2% and the MAP score by at least 3.6%
Aabed S, et al. 2024 [26]	DCNN + BLSTM + CTC (without segmentation)	84% : for characters 71% : for words
Meddeb O, et al. 2017 [27]	LSTM + RNN	88%
Meddeb O, et al. 2016 [28]	hybrid LSTM-RNN + HMM	91.77%
Ritonga M, et al. 2025 [29]	CNN (VGG16) + SVM	99.33%
Alqahtani FA. 2024 [30]	ODLF-AHCRCOA (LeNet-5 + LSTM optimized by COA)	98.60%
Svendsen B, et al. 2023 [31,32]	Comparative Analysis (SVM/CNN)	99.9%

were the most effective, achieving 99.9% accuracy with high computational efficiency. The challenges highlighted concern the inherent complexity of sign language gesture recognition. Table 1 shows a comparison summary of the reviewed methods in the literature.

As mentioned before, only one study deals with three Arabic-derived letters, while the others focus exclusively on the original Arabic alphabet. It is important to head towards this side in order to close this gap, because the derived letters not only expand the phonetic palette of the Arabic language, but also play a crucial role in adapting the script to the linguistic and cultural diversity of our languages. It is vital to expand optical character recognition (OCR) researchers towards derived letters to improve the inclusiveness and versatility of Arabic-based writing systems. The graphic and contextual variability of these letters, such as چ, ژ, and ڠ, present particular challenges due to their different forms and their use in handwritten words. These special letters request a robust and advanced system to recognize them, and that can be generalized to different writing styles and contexts. By highlighting these letters, future studies can help develop more comprehensive OCR systems that can handle a wider variety of languages and writing systems, thereby improving accessibility and language preservation.

Method

Dataset description

To address the lack of publicly available datasets for Arabic-derived letters, we developed a new dataset by collecting handwritten samples from participants using pre-designed forms. It includes the letters: گ(Gaf), ڤ (Ve), ژ (Zhe), چ (Che), پ (Pe), and ڠ (Ngain), which are used in different languages: Persian, Pashto, Urdu, Kurdish, and Sindhi, respectively. The papers were collected from 30 participants, 10 images for each range (aged 15–20, 20–30 and over 30), which ensured a variety of handwriting styles. Each participant wrote each letter five times in different forms: isolated, initial, median, and final, according to the rules of Arabic writing, see Table 2, with an additional

Table 2. The main forms of Arabic-derived letters.

Name	Gaf			Pe	Ve	Ngain	Zhe	Che
isolated	گ	ڭ	ݣ	پ	ڤ	ڠ	ژ	چ
Initial	گـ	-	ݣـ	پـ	ڤـ	ڠـ	-	چـ
Median	ـگـ	-	ـݣـ	ـپـ	ـڤـ	ـڠـ	-	ـچـ
Final	ـگ	ـڭ	ـݣ	ـپ	ـڤ	ـڠ	ـژ	ـچ

Fig. 1. A participant scanned image.

printed version for machine writing. This resulted in six times per letter per participant, as you see in Fig. 1.

After collecting the sheets, the handwritten images were scanned at a resolution of 300 ppi and subjected to several pre-tuning steps. To build the ADL (Arabic Derived Letters) dataset, a comprehensive and automated image processing pipeline is designed to transform handwritten pages into a set of standardized, training-ready images of isolated characters. The process begins with the detection of the main contour using the Canny algorithm, followed by orientation correction by rotation based on the bounding box angle. Each image is then segmented row by row, then column by column, to extract individual letters. To enhance the robustness of the model, a data augmentation algorithm has been included as part of the above process. The images are subjected to various transformations such as a slight rotation (by $\pm 5^{\circ}$), a shift (up to 8% horizontally and vertically), a zoom (up to 5%), and a fill mode (nearest). Each original image is augmented 11 times with the basic structure intact. These images are then resized to 32 $\times$ 32 pixels. The images are inverted to ensure that a grayscale character is visible on a black background. The dataset is arranged into six classes based on the following derived Arabic characters: Che, Gaf, Ngain, Pe, Ve, and Zhe. Each class is stored separately. The data is divided into two parts: one for training and the other for validation. Algorithm 1 focuses on the key steps in the preprocessing. Fig. 2 displays some samples from the data set. The latest version of the dataset comprises 55,440 images, evenly distributed between the training phase (80%) and the validation phase (20%), as shown in Fig. 3. Each image file is represented as a CSV file with two columns: the first represents the class labels, and the second represents the pixel values in the images. The representation of the data in this form enables it to be easily integrated with machine learning while maintaining the structure and detail in the images. Fig. 4 gives an overview of the file structure by taking an image sample in its original format (32 $\times$ 32).

Method description

Image preprocessing

The images are processed using a VGG-Like convolutional neural network. Fig. 5 illustrates the model architecture used in this study. The dataset images represent the input to the convolutional network, where each image has a size of 32 $\times$ 32 $\times$ 1 (32 pixels in height, 32 pixels in width, and a single depth layer representing the grayscale pixel intensity).

In the context of handwritten character recognition from small images (32 $\times$ 32 pixels, in grayscale), we opted for a simplified VGG-Like convolutional architecture. Table 3 shows the common elements between the original VGG-16 and the VGG-like used in this study. This choice is justified by the need to efficiently capture the fine local structures (lines, curves, points) characteristic of handwritten letters, while maintaining a model of reasonable complexity, adapted to the size of the dataset. This architecture allows the application of successive operations of feature extraction (via convolutions), dimension reduction (via pooling), and learning of discriminative representations. It thus constitutes a structured and modular basis for the implementation of the supervised learning process, which is based on two main steps: forward propagation for prediction, and backward propagation for parameter adjustment.

Forward propagation

A convolutional neural network (VGG-Like) is used to extract features and classify processed images. The convolution and pooling layers constitute the CNN. Each convolution layer uses multiple filters to extract image properties. Each filter F_1 is convolved with the input image I to obtain a new image I' using the operation explained in Eq. (1).[33]

$$I'_1(i, j) = \sum_{n=-1}^{1} \sum_{m=-1}^{1} F_1(n, m) \times I(i+n, j+m) \tag{1}$$

The convolution layers used the nonlinear activation function ReLU (Rectified Linear Unit), which is defined in Eq. (2). This function makes the model

Algorithm 1: Dataset Creation

Input: Source images containing the 30 sheets of derived Arabic handwritten letters
Output:Final structured, pre-processed, augmented, and resized to a 32 × 32 dataset
1. Detection and Straightening:
    a. Detect edges using the Canny algorithm.
    b. Find the largest edge and calculate the rotation angle.
    c. Apply a rotation to straighten the image.
2. Spatial Clipping:
    a. Clip the image into rows using a regular vertical step.
    b. Extract the columns from each row to isolate each character.
3. Data Augmentation:
    a. For each image, generate 10 new variants via:
        – upload image in grayscale
        – random rotation (±5°),
        – horizontal shift (≤8%),
        – vertical shift (≤8%),
        – slight zoom (≤5%),
        – fill_mode = 'nearest'
    b. Save the augmented images in the respective folders per class.
4. Resizing and Inverting:
    a. Resize each image to 32 × 32 pixels.
    b. Invert the colors to obtain a white character on a black background.
5. Final Structuring:
    a. Organize the images into 6 folders corresponding to the letters: Che, Gaf, Ngain, Pe, Ve, Zhe.
    b. Split the dataset into training and validation sets.
    d. Create final CSV files for training and validation datasets

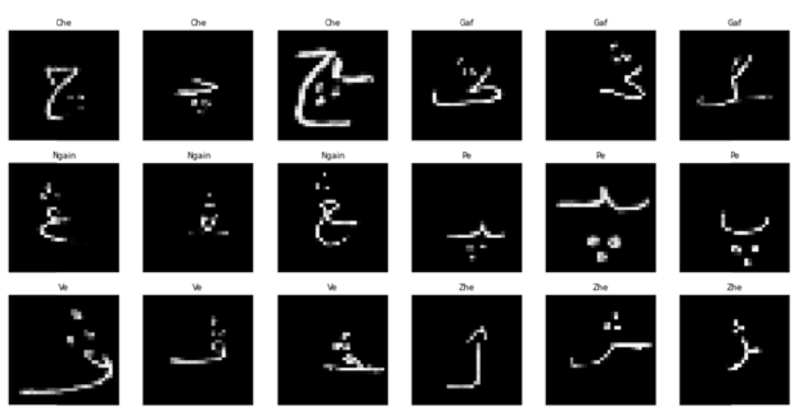


Fig. 2. Dataset samples.

nonlinear, which gives the ability to detect the complex relationship between the image's pixels.

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

The reduction of images' shapes or the spatial dimensions of the feature maps is a key issue in the CNN, for that a Max-Pooling layer is used to feed the ReLU results with a 2 × 2. This step helps to reduce the complexity of computational systems and avoid overfitting. The pooling operation is defined in Eq. (3).

$$Pool\,(i, j) = max\left(I'\left(i + n, j + m\right)\right),\; For\;(n, m) \in \{0, 1\} \tag{3}$$

After the convolution and pooling phases, the feature maps of size 2 × 2 × 128 are reduced to a one-dimensional vector of 512. The fully connected

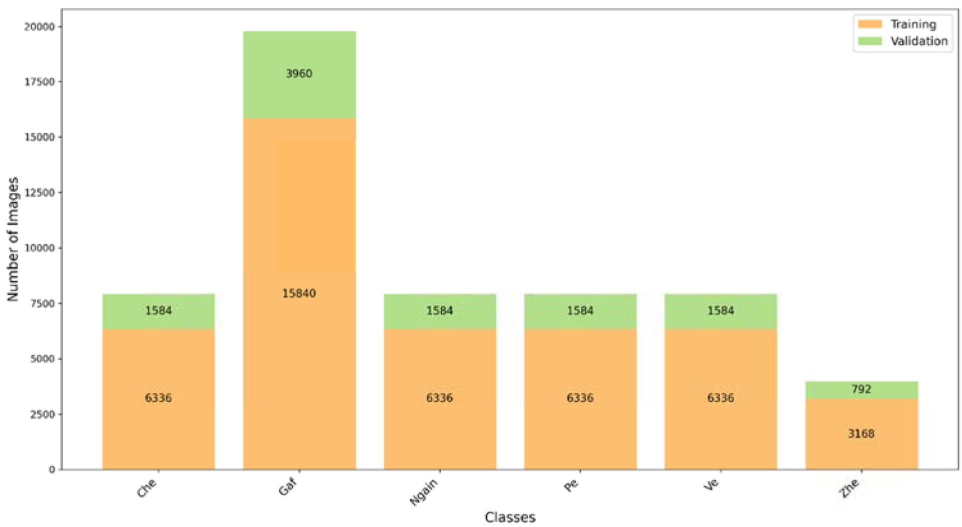


Fig. 3. Distribution of images per class for training and validation datasets.

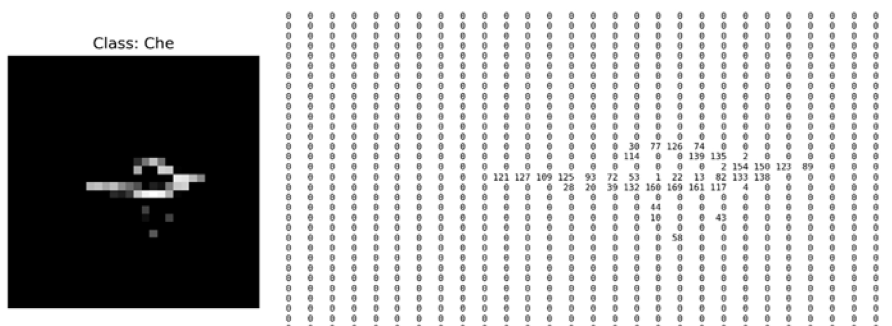


Fig. 4. Sample of training dataset CSV file flattened as 32 × 32.

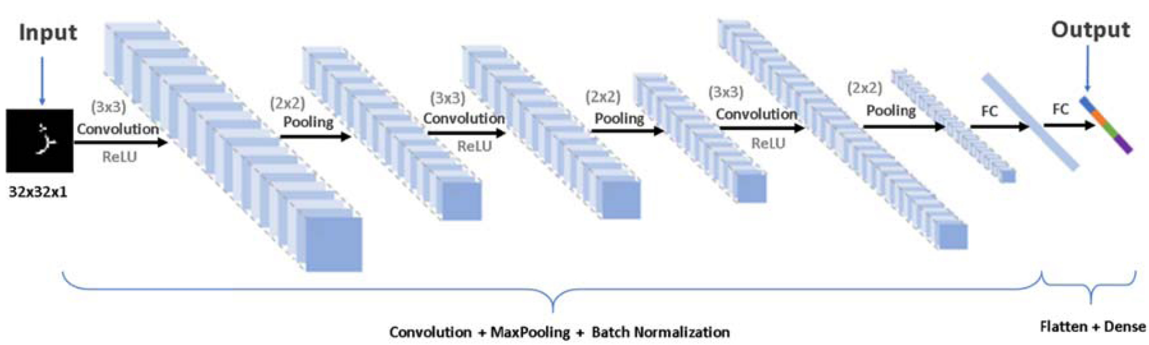


Fig. 5. CNN (VGG-Like) architecture.

layers use this flattened vector to perform the final classification. The linear transformation is applied to the first fully connected layer, followed by a ReLU activation. In mathematics, we obtain the output of this layer using Eq. (4).

$$h = ReLU\left(Wv + b\right) \tag{4}$$

where v represents the flattened input vector of size 512, $W \in R^{128\times512}$ is the weight matrix, $b \in R^{128}$. The vector of bias, and ReLU is the activation function applied. To reduce the overfitting rate, a dropout layer with a rate of 0.5 randomly deactivates 50% of neurons during training. The output layer maps the 128-dimensional output from the first fully connected layer to C neurons corresponding to the number of classes (6 neurons for our dataset target classes). The output is computed as demonstrated in Eq. (5).

$$Out = Softmax\left(W_{out}h + b_{out}\right) \tag{5}$$

Table 3. A comparative overview of VGG-16 and the VGG-like model used.

Element	Original VGG-16	VGG-like	Remark
Input size	224 × 224 × 3 (RGB images)	32 × 32 × 1 (grayscale images)	Adapted to the nature of the data
Number of convolutional layers	13 convolutions distributed over 5 blocks	3 convolutional layers distributed over 3 blocks	Structural inspiration
Filter size	3 × 3 in all layers	3 × 3 in all layers	Conforms to the VGG design
Activation	ReLU	ReLU	Identical
Batch Normalization	No (absent in original VGG-16)	Yes, after each convolutional layer	Modern improvement
Pooling	MaxPooling 2 × 2 after each block	MaxPooling 2 × 2 after each block	Identical
Fully Connected Layer	(Dense) 3 layers (4096 – 4096 - 1000)	2 layers (256 - 6)	Simplified for a reduced dataset
Dropout	Yes (between FC layers)	Yes (0.5 before output)	Present
Output Layer	1000 neurons (ImageNet) with softmax	6 neurons (character classes) with softmax	Adapted to the targeted problem
Network Depth	Very deep (16 layers)	Moderately deep	Justified simplification
Application domain	General vision (ImageNet)	ADL recognition	Specific adaptation

Where $W_{out} \in \mathrm{R}^{C\times 128}$ represents the weight matrix, $b_{\mathrm{out}} \in R^{C}$ is the vector of bias, and the softmax function defines as $\mathrm{Softmax}(o_i) = \frac{e^{o_i}}{\sum_{j=1}^{C} e^{o_j}}$ ensures the outputs represent class probabilities. This architecture effectively transforms the input feature maps into a probability distribution over the C classes for classification. Table 2 describes CNN layers and the number of parameters used in each layer.

The convolutional neural network architecture adopted in this study is hierarchically structured to enable progressive extraction of visual features from input images of size 32 × 32 × 1. The first block comprises a convolutional layer with 64 filters of size 3 × 3, followed by Batch Normalization and down-sampling by MaxPooling (2 × 2), reducing the dimensions to 16 × 16. The second block applies convolution with 128 filters, followed by the same normalization and pooling operations, reducing the size to 8 × 8. The third block further increases the depth with 256 convolutional filters, also followed by normalization and pooling, giving a final output of 4 × 4 × 256. At the output of the convolutional blocks, the feature maps are flattened, generating a vector of 4096 neurons, then passed to a dense layer of 256 units with ReLU activation function. A Dropout regularization mechanism (rate 0.5) is applied at this level to limit overfitting. Finally, the output layer is a dense layer of 6 neurons (corresponding to the target classes), activated by a softmax function to produce a probability distribution. The model has a total of 1,421,830 parameters, of which 1,420,934 are trainable. This structure makes it possible to efficiently learn discriminative representations adapted to variations in handwritten characters. Table 4 shows the details of the model architecture.

Table 4. Model architecture and parameters.

Layer	Output shape	Parameters
Input (grayscale images)	32 × 32 × 1	-
Conv(64 Filters, Size = 3)+ Relu	32 × 32 × 64	640
Batch Normalization	32 × 32 × 64	256
Max pool2D (Size = 2, Stride=2)	16 × 16 × 64	0
Conv (128 Filters, Size = 3)+ Relu	16 × 16 × 128	73856
Batch Normalization	16 × 16 × 128	512
Max pool2D (Size = 2, Stride=2)	8 × 8 × 128	0
Conv(256 Filters, Size = 3)+ Relu	8 × 8 × 256	295168
Batch Normalization	8 × 8 × 256	1024
Max pool2D (Size = 2, Stride = 2)	4 × 4 × 256	0
Flatten	4096	0
Dense + Relu	256	1048832
Dropout	256	0
Dense + Softmax	6	1542

Backpropagation

Neural networks are trained using the backpropagation algorithm. Regular updating of the network's weights and biases is essential to minimize the error between the model's predictions and the real values. The model layers are adjusted during the backpropagation phase to minimize the prediction error on the N = 55,440 images in the dataset. These adjustments are performed using gradient descent, based on Eq. (6).[34]

$$W^{(t+1)} = W^{(t)} - \rho \frac{\partial LF}{\partial W}, \quad b^{(t+1)} = b^{(t)} - \rho \frac{\partial LF}{\partial b} \tag{6}$$

Where $\rho = 0.0001$ is the initial learning rate, $\frac{\partial \mathrm{LF}}{\partial \mathrm{W}}$ and $\frac{\partial \mathrm{LF}}{\partial \mathrm{b}}$ represent the gradients of the loss function with respect to the weights and biases, respectively. Categorical cross-entropy illustrated in Eq. (7), is used as a loss function, defined for the K = 6 derived letter target classes.

$$LF = -\frac{1}{55440}\sum_{I=1}^{55440}\sum_{K=1}^{6} y_{i,k}\, log\left(\hat{y}_{i,k}\right) \tag{7}$$

Where $y_{i,k}$ is the true image label and $\hat{y}_{i,k}$ is the predicted probability for the class k.

The Adam (Adaptive Moment Estimation) optimizer increases convergence by dynamically adjusting the learning rate for each parameter.[35] It combines the advantages of gradient descent with momentum and adaptive learning rates. Parameter updates are performed according to Eq. (8) (updates gradient averages), Eq. (9) (bias correction for $m^{(t)}$ and $v^{(t)}$), and Eq. (10) (updates parameters).

$$\begin{aligned} m^{(t)} &= \beta_1 m^{(t-1)} + (1-\beta_1)\, g_{rad}{}^{(t)}, \\ v^{(t)} &= \beta_2 m^{(t-1)} + (-\beta_2)\ \left(g_{rad}{}^{(t)}\right)^2 \end{aligned} \tag{8}$$

$$\hat{m}^{(t)} = \frac{m^{(t)}}{1-\beta_1^t},\ \hat{v}^{(t)} = \frac{v^{(t)}}{1-\beta_2^t} \tag{9}$$

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\rho}{\sqrt{\hat{v}^{(t)}}+\varepsilon}\hat{m}^{(t)} \tag{10}$$

Where $g_{rad}{}^{(t)}$ is the loss function gradient in step t, $m^{(t)}$ and $v^{(t)}$ are the averages of gradients and squared gradients, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are hyperparameters for controlling the decay rate of the moving averages, and $\varepsilon = 10^{-8}$ is a stability measure to avoid division by zero. Convergence is accelerated by these dynamic adjustments, which are essential for data as diverse as this dataset, where the derived letters are written by participants of different ages and handwriting styles.

The model is based on the Adam optimizer, which is characterized by its ability to manage inertia and the learning rate, which is set to 0.0001. Adam enables deep learning by eliminating abrupt weight changes. Optimization relies on categorical cross-entropy as the cost function, with one-hot encoding outputs adapted for classifying multiple classes. To enhance the model's value, accuracy is emphasized. The architecture also uses the EarlyStopping recall function as a regularization mechanism to mitigate overfitting. This mechanism stops training after 5 epochs without loss of improvement.

Results and discussions

The ADL dataset with 55440 images evaluates the proposed model (VGG-Like), where the training dataset displays 44352 while the validation dataset presents 11088. The processing is done with the help of the Keras and TensorFlow python libraries. The optimizer is Adam with categorical cross-entropy as the loss function to enhance the accuracy of the model by minimizing the prediction error. The training of the model is done for a maximum of 40 epochs with an early stopping mechanism based on the performance of the validation data. The training will be stopped by the early stopping mechanism if no improvement is seen in the validation loss for 5 epochs. The above mechanism will optimize the performance of the model while preventing it from getting into the state of overfitting. The laptop configuration used for the experiment is an Intel (R) Core i58265U CPU with 1.80 GHz, 8 GB internal graphics card, and 16 GB RAM.

Two more architectures of convolutional neural networks were examined in this research in order to evaluate the performance of the above models on the ADL classification problem. The first one is a modified version of the well-known LeNet-5 network, which includes two convolutional layers with the activation function of the hyperbolic tangent, followed by AveragePooling.The extracted features are then passed through three fully connected layers to classify the images. The last layer is also the output layer with softmax as its activation function. The aforementioned network is of a relatively small structure and is best suited for simple image classification problems requiring low computational resources. The second architecture is a simplified version of the ResNet. This type of model is based on residual block networks. Like CNNs, this network begins with a convolutional layer with batch normalization and ReLU as the activation function. The model then successively uses three residual blocks with increasing filters per convolutional layer (32, 64, 128), followed by MaxPooling to reduce image size. The network architecture concludes with a GlobalAveragePooling operation followed by two fully connected layers. To make the network more robust, residual connections are implemented to mitigate gradient degradation during the formation of deep networks.

For the evaluation of the models' performance (VGG-like, ResNet, LeNet), the evolution of two global indicators is observed during the training process: accuracy and the loss function. Accuracy corresponds to the percentage of correctly classified examples, both on the training data and the validation data. The loss function quantifies the difference between the model's predictions and the true labels, and its evolution allows us to monitor the quality and stability of the training. The VGG-Like model stands out for the best overall performance, achieving 99.83% accuracy on training data and 99.61% in validation, with losses of 0.5% and 1.72%

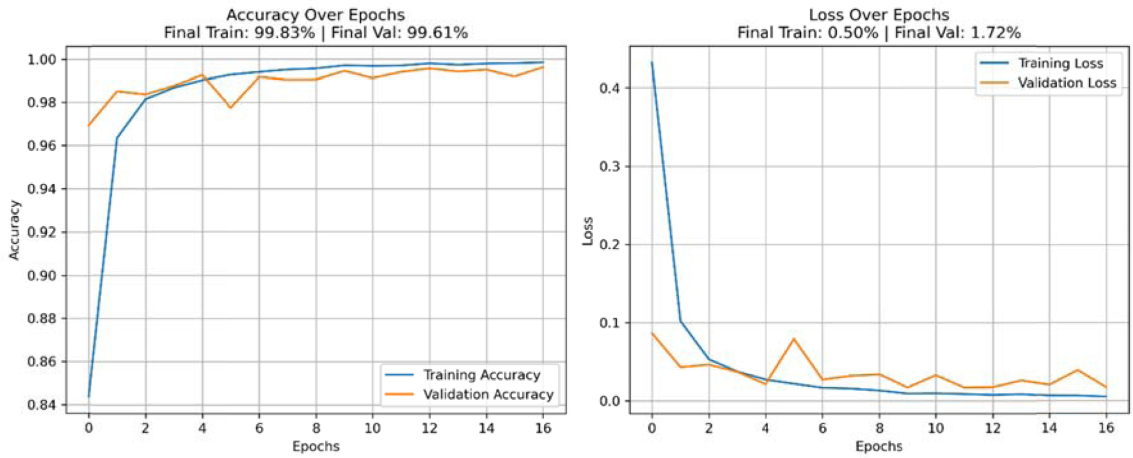


Fig. 6. Accuracy and loss variations of VGG-Like model.

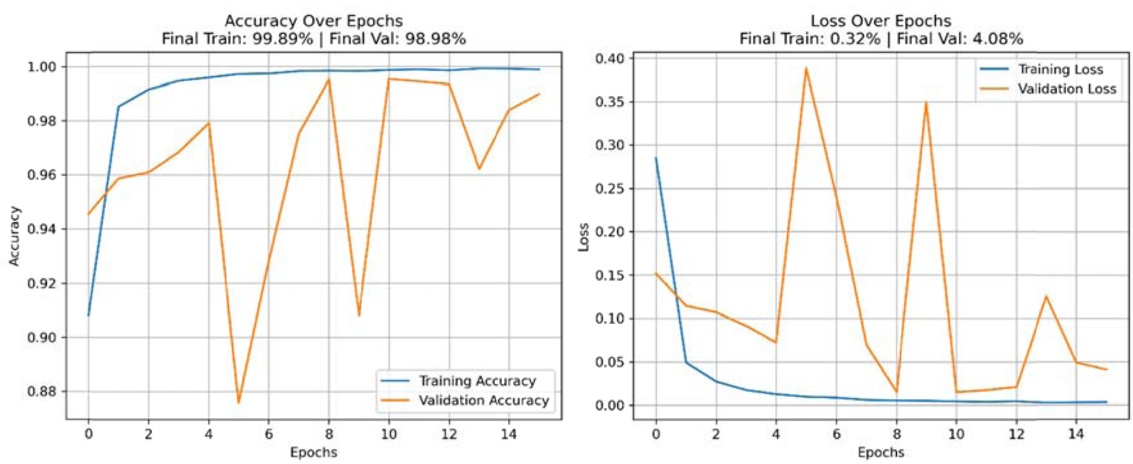


Fig. 7. Accuracy and loss variations of ResNet model.

respectively, all in just 17 epochs as shown in Fig. 6. The Simplified ResNet also delivers excellent results, with 99.89% accuracy in training and 98.98% in validation, accompanied by slightly higher losses (0.32% and 4.08%) and fast convergence as early as the 16th epoch as illustrated in Fig. 7. In comparison, the older and shallower LeNet model has a more modest accuracy (98% in training and 96.43% in validation), with higher losses (5.8% and 9.99%) and a need of 40 epochs to reach convergence as shows in Fig. 8. Considering these results, the VGG-Like model appears to be the most efficient and balanced, offering excellent generalization capacity, low loss, and fast convergence, which justifies its preference in the context of this study for the task of Arabic derived letters classification. This preference over ResNet also comes from the stability of VGG-Like during the learning process, the validation dataset, which normally gives the robustness of the model, has shown a remarkable destabilization for RestNet, unlike VGG-Like, where the learning curve has increased in a homogeneous and stable way.

Beyond the overall indicators, a detailed performance analysis is performed using the confusion matrix and the classification report. The confusion matrix identifies, for each class, the number of correctly or incorrectly classified examples. From this matrix, the following metrics are calculated for each class *i*:

- True positives (TP_i): number of examples from class *i* correctly predicted.
- False positives (FP_i): number of examples from other classes incorrectly predicted as belonging to class *i*.
- False negatives (FN_i): number of examples from class *i* was wrongly predicted in another class.

For the models' performance evaluation (VGG-like, ResNet, LeNet), the evolution of two global indicators

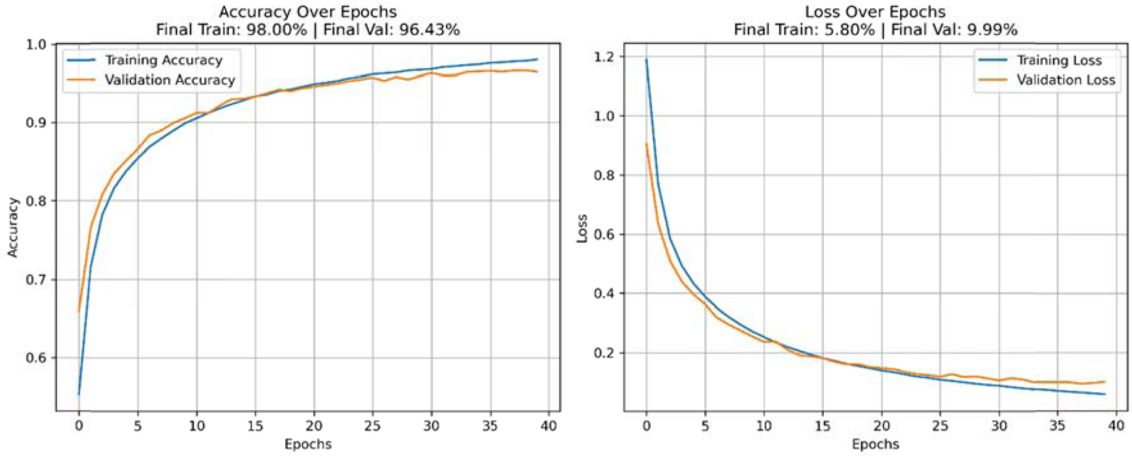


Fig. 8. Accuracy and loss variations of LeNet model.

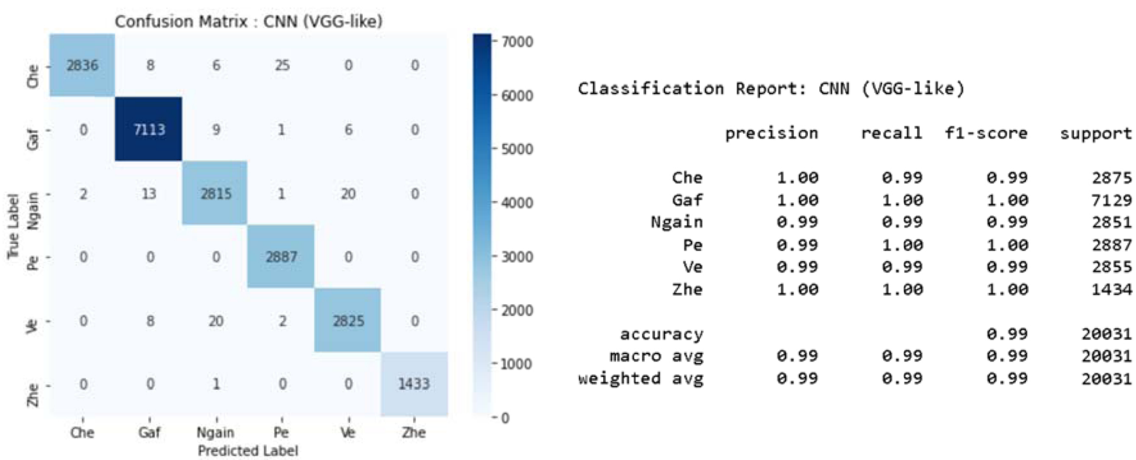


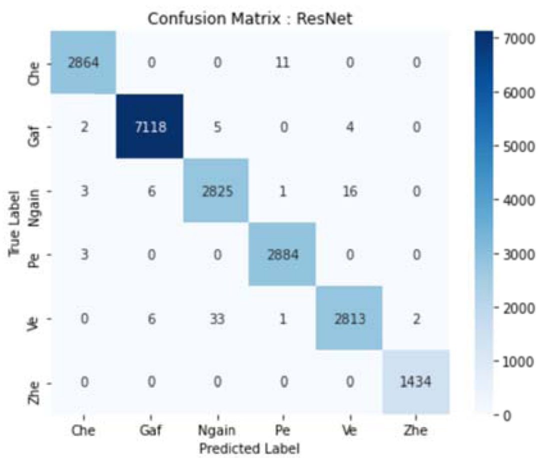
Fig. 9. Confusion Matrix and Classification report of VGG-Like model.

is observed during the training process: accuracy and the loss function. These values are used to compute the precision and the recall for each class, as indicated in Eq. (11). Precision is a measure of the proportion of correct predictions made for class i, while the recall measures the ability of the model to capture all instances of class i. The F1-score is therefore the harmonic mean of the precision and the recall. The F1-score is computed as indicated in Eq. (12). The F1-score is a combination of the two previous scores and is particularly vital in the event showing the classes imbalance. The above measurements provide an integral view of the performance of the models under test, including the accuracy and loss curves.

$$Precision_i = \frac{TP_i}{TP_i + FP_i},\ Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (11)$$

$$F1 - score_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (12)$$

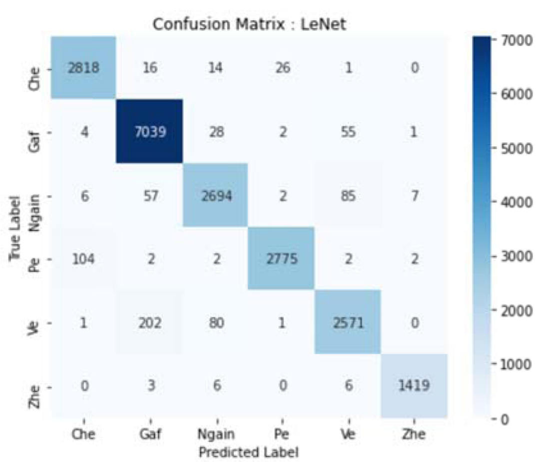
From the analysis of the classification reports and the confusion matrices, we can further refine the comparison of the three models. The VGG-Like and ResNet architectures, as presented in Figs. 9 and 10, are remarkable for the similarity and the very high level of performance they reach, where the precision and F1-score values are around 99.9%, even reaching 100% in the majority of the classes, which proves a very good ability to generalize. This stability is confirmed by the confusion matrices, which reveal 122 classification errors for VGG-Like compared to 93 errors for ResNet, a relatively small difference, indicating almost equivalent performances. In contrast, the LeNet model shows significantly lower results, with an F1-score of only 92% on letter Ve, and a total of 715 misclassified images, as shown in Fig. 11. These results illustrate the limitations of a simpler and shallower model for a fine classification task, particularly in a context where certain letters share significant visual similarities. Thus, VGG-Like and ResNet appear



Classification Report: ResNet

	precision	recall	f1-score	support
Che	1.00	1.00	1.00	2875
Gaf	1.00	1.00	1.00	7129
Ngain	0.99	0.99	0.99	2851
Pe	1.00	1.00	1.00	2887
Ve	0.99	0.99	0.99	2855
Zhe	1.00	1.00	1.00	1434
accuracy			1.00	20031
macro avg	0.99	0.99	0.99	20031
weighted avg	1.00	1.00	1.00	20031

Fig. 10. Confusion Matrix and Classification report of ResNet model.



Classification Report: LeNet

	precision	recall	f1-score	support
Che	0.96	0.98	0.97	2875
Gaf	0.96	0.99	0.97	7129
Ngain	0.95	0.94	0.95	2851
Pe	0.99	0.96	0.97	2887
Ve	0.95	0.90	0.92	2855
Zhe	0.99	0.99	0.99	1434
accuracy			0.96	20031
macro avg	0.97	0.96	0.96	20031
weighted avg	0.96	0.96	0.96	20031

Fig. 11. Confusion Matrix and Classification report of LeNet model.

to be the most reliable and best-suited models in this study, offering both high accuracy and robustness to the variability of handwritten characters.

In the state of the art (see Related work section), Study[6] was conducted only on three Arabic-derived letters (Gaf, Ve, and Pe). It was a hybrid method of the mathematics morphological gradient algorithm to identify letter edges and a multi-layer perceptron (MLP) as a classifier. The highest accuracy was 96%. Compared to this study, the proposed methods in this study provide high accuracy, even though they considered six Arabic-derived letters (Che, Pe, Ve, Ngain, Zhe, and Gaf). The comparison of the accuracies of the validation datasets between the models is presented in Table 5 To obtain concrete results, we use a graphical user interface (GUI). With it, we have the possibility to create a letter and indicate its class using the parameters of our model. The test results are illustrated in Fig. 12.

Although experimental results confirm the effectiveness of deep architectures such as VGG-Like and ResNet for handwritten letter classification, some limitations must be highlighted. On the one hand, the dataset used is limited to isolated letters, which does not necessarily reflect the complexity of handwritten cursive texts in real-world contexts, where letters are often linked or distorted. On the other hand, although recent models such as Visual Transformers (ViT) are promising, they generally require much larger data volumes to be fully effective. In the context of this study, marked by a relatively small dataset, the use of too deep models would have led to overfitting and unjustified computational overhead. Thus, the choice of CNN models adapted to the size and nature of the dataset remained relevant. These limitations do not affect the validity of the results obtained, but highlight the challenges to be addressed for a broader and more generalizable application.

Table 5. Methods comparison.

Model		Class treated	Accuracy
Model in [6]	MG-MLP	Gaf, Pe and Ve	96%
Models used in this study	LeNet	Che, Gaf, Ngain, Pe, Ve and Zhe	96.43%
	ResNet-Like		98.98%
	VGG-Like		**99.61%**

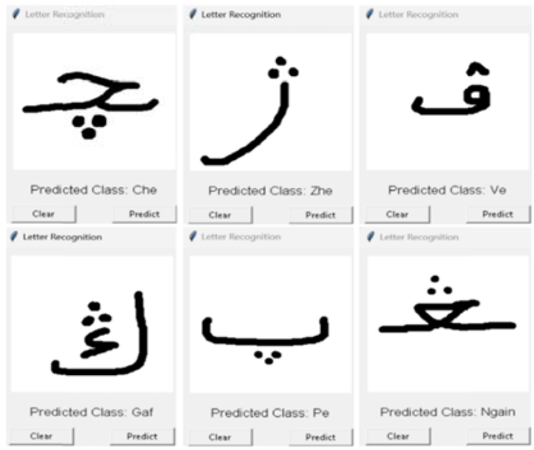


Fig. 12. Real testing examples.

Conclusion

The Arabic-derived letters are extensions of the Arabic language alphabet with specificities in terms of forms and pronunciation. Although these letters are foreign to the Arabic language, they can help it to pronounce well intrusive words that come from other languages because of their absence in the original Arabic language. These words are new, linked to scientific innovations, so they have been integrated into the Arabic language while preserving their original pronunciation.

In this study, the ADL dataset of the Arabic-derived letters has processed. The recognition process is carried out by a VGG-Like convolutional neural network. This produced a solid accuracy of 99.61% and a loss of 1.72%. The deep structure of the VGG-Like model proved capable of producing robust results on a dataset composed of low-resolution handwritten characters (32 × 32 pixels). In order to evaluate the robustness of the proposed model, two other architectures were implemented for comparison. The first is a simplified ResNet, specifically adapted to the characteristics of the studied dataset, while the second is a classic version of LeNet. Both models demonstrate interesting results, especially the ResNet model, which reached 98.98% accuracy, thus getting close to the results obtained with the VGG-Like model. The LeNet model, on the other hand, obtained a lower accuracy, with an accuracy score of 96.43%, which confirms the limits that exist with the complexity of some handwritten letters.

The present study has some limitations despite the good results obtained. The main challenge lies in the fact that the study deals with the classification of individual letters, which limits the application in the context of word writing. Moreover, there are more recent models, even if their complexity and depth are not suited for the modest size of the data set used in the present study. These are the improvements that could be made in the context of further research, aiming at adapting the models to more varied and realistic scenarios, while maintaining the good balance between the models' complexity and their accuracy.

Acknowledgements

The author expresses deep gratitude and appreciation to every participant for their invaluable assistance in completing the alphabet sheets.

Author's Declaration

- Conflicts of Interest: None.
- I hereby confirm that all the Figures and Tables in the manuscript are mine. Furthermore, any Figures and images that are not mine have been included with the necessary permission for republication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- The ethical guidelines for human subjects' research were followed in the conduct of this study.

Prior to data collection, all participants gave informed consent, which ensured that they were fully aware of the purpose of the study.
- Ethical Clearance: The project was approved by the local ethical committee at University Ibn Zohr.

Data availability

The dataset used in this study is available upon request.

References

1. Alhamad HA, Shehab M, Shambour MKY, Abu-Hashem MA, Abuthawabeh A, Al-Aqrabi H, Daoud MS, Shannaq FB. Handwritten recognition techniques: A comprehensive review. Symmetry. 2024;16(6):681. https://doi.org/10.3390/sym16060681.
2. Parvez MT, Mahmoud SA. Arabic handwriting recognition using structural and syntactic pattern attributes. Pattern Recognit. 2013;46(1):141–154. https://doi.org/10.1016/j.patcog.2012.07.012.
3. Doctor R, Gutkin A, Johny C, Roark B, Sproat R. Graphemic normalization of the Perso-Arabic script. 2022. https://doi.org/10.48550/arXiv.2210.12273.
4. Bakalla MH. Arabic culture: through its language and literature. 2023. Taylor & Francis. https://doi.org/10.4324/9781003402503.
5. Al-Jarf, R. Derivation from foreign words and acronyms borrowed in Arabic. Lingua. Language and Culture. 2021;20(2):52–77.
6. El Atillah M, El Fazazy K. Recognition of intrusive alphabets to the arabic language using a deep morphological gradient. Rev. d'Intelligence Artif. 2020;34(3):277–284. https://doi.org/10.18280/ria.340305.
7. Ding X, Zhang X, Ma N, Han J, Ding G, Sun J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021:13733–13742. https://doi.org/10.48550/arXiv.2101.03697.
8. Kattenborn T, Leitloff J, Schiefer F, Hinz S. Review on convolutional neural networks (CNN) in vegetation remote sensing. ISPRS J Photogramm Remote Sens. 2021;173:24–49. https://doi.org/10.1016/j.isprsjprs.2020.12.010.
9. Liu Y, Pu H, Sun DW. Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. Trends Food Sci Technol. 2021;113:193–204. https://doi.org/10.1016/j.tifs.2021.04.042.
10. Rehman A, Naz S, Razzak MI. Writer identification using machine learning approaches: a comprehensive review. Multimed Tools Appl. 2019;78:10889–10931. https://doi.org/10.1007/s11042-018-6577-1.
11. Agathos L, Avgoustis A, Kryelesi X, Makridou A, Tzanis I, Mouratidis D, *et al.* Bridging linguistic gaps: Developing a greek text simplification dataset. Information. 2024;15(8):500. https://doi.org/10.3390/info15080500.
12. Nguyen TTH, Jatowt A, Coustaty M, Doucet A. Survey of post-OCR processing approaches. ACM Comput Surv. 2021;54(6):1–37. https://doi.org/10.1145/3453476.
13. Long S, He X, Yao C. Scene text detection and recognition: The deep learning era. Int J Comput Vis. 2021;129:161–184. https://doi.org/10.1007/s11263-020-01369-0.
14. Almanea M. Deep learning in written Arabic linguistic studies: A comprehensive survey. IEEE Access. 2024;12:172196–172233. https://doi.org/10.1109/ACCESS.2024.3488553.
15. EL ATILLAH M, EL FAZAZY K, Riffi J. Classification of Arabic alphabets using a combination of a convolutional neural network and the morphological gradient method. Baghdad Sci. J. 2024;21(1):0252–0252. https://doi.org/10.21123/bsj.2023.7877.
16. Faizullah S, Ayub MS, Hussain S, Khan MA. A survey of OCR in Arabic language: Applications, Techniques, and Challenges. Appl Sci. 2023;13(7):4584. https://doi.org/10.3390/app13074584.
17. El Atillah M, El Fazazy K, Riffi J. Recognition of Arabic digits using a convolutional neural network. In 2022 International Conference on Intelligent Systems and Computer Vision (ISCV). 2022:1–5. IEEE. https://doi.org/10.1109/ISCV54655.2022.9806129.
18. Awni M, Khalil MI, Abbas HM. Offline Arabic handwritten word recognition: A transfer learning approach. J King Saud Univ Comput Inf Sci. 2022;34(10):9654–9661. https://doi.org/10.1016/j.jksuci.2021.11.018.
19. Maalej R, Kherallah M. New MDLSTM-based designs with data augmentation for offline Arabic handwriting recognition. Multimed Tools Appl. 2022;81(7):10243–10260. https://doi.org/10.1007/s11042-022-12339-8.
20. Ghadekar P, Khanvilkar O, Kharat J, Joshi K, Kulkarni T, Kulkarni Y. Automating the grading of handwritten examinations through the integration of optical character recognition and machine learning algorithms. J Integr Sci Technol. 2025;13(6):1128–1128. https://doi.org/10.62110/sciencein.jist.2025.v13.1128.
21. Nahar K M, Alsmadi I, Al Mamlook RE, Nasayreh A, Gharaibeh H, Almuflih AS, Alasim F. Recognition of Arabic air-written letters: Machine learning, convolutional neural networks, and optical character recognition (OCR) techniques. Sensors. 2023;23(23):9475. https://doi.org/10.3390/s23239475.
22. Ghanim TM, Khalil MI, Abbas HM. Comparative study on deep convolution neural networks DCNN-based offline Arabic handwriting recognition. IEEE Access. 2020;8:95465–95482. https://doi.org/10.1109/ACCESS.2020.2994290.
23. Minoofam SAH, Bastanfard A, Keyvanpour MR. DB-QM: A comparative quality measurement and its prospective on Persian/Arabic databases for OCR. ACM Trans. Asian Low-Resour. Lang Inf Process. 2025;24(7):1–25. https://doi.org/10.1145/3711714.
24. Sahlol AT, Abd Elaziz M, Al-Qaness M A, Kim S. Handwritten Arabic optical character recognition approach based on hybrid whale optimization algorithm with neighborhood rough set. IEEE Access. 2020;8:23011–23021. https://doi.org/10.1109/ACCESS.2020.2970438.
25. Bouibed ML, Nemmour H, Arab N, Chibani Y. Improved writer retrieval in handwritten documents using hybrid combination. Multimed. Tools Appl. 2024;83(26):68671–68695. https://doi.org/10.1007/s11042-023-17841-1.
26. Aabed S, Khairaldin A. An end-to-end, segmentation-free, Arabic Handwritten Recognition Model on KHATT. arXiv preprint arXiv. 2024:2406–15329. https://doi.org/10.48550/arXiv.2406.15329.
27. Meddeb O, Maraoui M, Aljawarneh S. Hybrid modelling of an off line Arabic handwriting recognition system: results and evaluation. Int J Intell Enterp. 2017;4(1-2):168–189. https://doi.org/10.1504/IJIE.2017.087017.

28. Meddeb O, Maraoui M, Aljawarneh S. Hybrid modeling of an offline arabic handwriting recognition system AHRS. In 2016 International Conference on Engineering & MIS (ICEMIS). IEEE. 2016:1–8. https://doi.org/10.1109/ICEMIS.2016.7745319.
29. Ritonga M, Bangare ML, Bangare PM, Bangare SL, Vanjire SS, Moholkar K, *et al.* Optimized convolutional neural network deep learning for Arabian handwritten text recognition. Bull Electr Eng Inform. 2025;14(2):1497–1506. https://doi.org/10.11591/eei.v14i2.7696.
30. Alqahtani FA. Optimized deep learning framework for Arabic handwritten character recognition using coati optimization algorithm. In 2024 International Conference on Engineering and Emerging Technologies (ICEET), Dubai, United Arab Emirates. 2024:1–7. https://doi.org/1109/ICEET65156.2024.10913566.
31. Svendsen B, Kadry S. Comparative analysis of image classification models for norwegian sign language recognition. Technologies. 2023;11(4):99. https://doi.org/10.3390/technologies11040099.
32. Svendsen B, Kadry S. A dataset for recognition of norwegian sign language. Int. J. Math. Stat. Comput. Sci. 2023;2:1–3. https://doi.org/10.59543/ijmscs.v2i.8049.
33. Jie HJ, Wanda P. RunPool: A dynamic pooling layer for convolution neural network. Int J Comput Intell Syst. 2020;13:66–76. https://doi.org/10.2991/ijcis.d.200120.002.
34. Zhang J. Gradient descent-based optimization algorithms for deep learning models training. arXiv preprint arXiv: 1903.03614. 2019. https://doi.org/10.48550/arXiv.1903.03614.
35. Kim KS, Choi YS. HyAdamC: A new adam-based hybrid optimization algorithm for convolution neural networks. Sensors. 2021;21(12):4054. https://doi.org/10.3390/s21124054.

التعرّف على الحروف العربية المُشتقّة المكتوبة بخط اليد باستخدام شبكة عصبية التفافية شبيهة بـ VGG

محسن العاطي الله

هندسة أنظمة الكمبيوتر، الرياضيات والتطبيقات (ISIMA)، الكلية متعددة التخصصات بتارودانت، جامعة ابن زهر، تارودانت، المغرب.

الخلاصة

نظرًا لتعقيد اللغة العربية وامتداداتها، فإنها تُشكّل مجالًا خصبًا للبحث في مجال الذكاء الاصطناعي عمومًا، وفي مجال التعرّف الضوئي على الحروف (OCR) على وجه الخصوص. توجد عدة لغات تستخدم الأبجدية العربية في مخطوطاتها، وقد ابتكرت هذه اللغات حروفًا جديدة لتمثيل أصوات غير موجودة في اللغة العربية. وتُسمّى هذه الحروف بـ«الحروف العربية المُشتقّة». ومن أجل إثراء اللغة العربية، يمكن الاستفادة من هذه الحروف لمعرفة النطق الصحيح للكلمات الدخيلة على اللغة العربية. يتناول هذا البحث مجموعة بيانات الحروف العربية المُشتقّة (ADL)، وهي مجموعة بيانات جديدة تتكوّن من 55440 صورة ممسوحة ضوئيًا لأوراق كتبها 30 مشاركًا من أعمار مختلفة، مع استخدام تقنية تعزيز البيانات لزيادة عدد الصور. تهدف هذه الدراسة إلى تقييم ومقارنة فاعلية بُنى مختلفة من الشبكات العصبية الالتفافية في التعرّف على الحروف العربية المُشتقّة، مع التركيز على الدقة، والمتانة، والقدرة على التعميم. تم تنفيذ ثلاث بُنى: نموذج LeNet، ونموذج ResNet مُبسّط يعتمد على كتل متبقية، وشبكة عميقة شبيهة بـ VGG. اقتصر التدريب على 40 تكرار (Epoch) مع تطبيق الإيقاف المبكر بعد 5 تكرارات دون تحسّن. أظهرت النتائج التجريبية أن نموذج VGG-Like حقق أفضل أداء بدقة بلغت 99.61% في مرحلة التحقق، تلاه نموذج ResNet بدقة 98.98%. في المقابل، كان أداء LeNet أقل كفاءة بنسبة 96.43%. وتُظهر هذه النتائج بوضوح أن البُنى العميقة والحديثة توفّر دقة ومتانة أفضل في تصنيف الحروف المكتوبة بخط اليد.

الكلمات المفتاحية: الكتابة العربية اليدوية، الشبكات العصبية الالتفافية، الحروف العربية المُشتقّة، التعرّف الضوئي على الحروف، LeNet، ResNet، VGG-Like.