

DOI: <https://doi.org/10.21123/bsj.2023.7877>

Classification of Arabic Alphabets Using a Combination of a Convolutional Neural Network and the Morphological Gradient Method

*Mouhssine EL ATILLAH** 

Khalid EL FAZAZY 

Jamal Riffi 

Computer Science, Signals, Automation and Cognitivism Laboratory (LISAC), FSDM, USMBA, Fes, Morocco

*Corresponding author: mouhssine.elatillah@usmba.ac.ma

E-mails addresses: khalid.elfazazy@usmba.ac.ma, jamal.riffi@usmba.ac.ma.

Received 3/10/2022, Revised 18/3/2023, Accepted 19/3/2023, Published Online First 20/5/2023,
Published 01/1/2024



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Abstract:

The field of Optical Character Recognition (OCR) is the process of converting an image of text into a machine-readable text format. The classification of Arabic manuscripts in general is part of this field. In recent years, the processing of Arabian image databases by deep learning architectures has experienced a remarkable development. However, this remains insufficient to satisfy the enormous wealth of Arabic manuscripts. In this research, a deep learning architecture is used to address the issue of classifying Arabic letters written by hand. The method based on a convolutional neural network (CNN) architecture as a self-extractor and classifier. Considering the nature of the dataset images (binary images), the contours of the alphabets are detected using the mathematical algorithm of the morphological gradient. After that, the images are passed to the CNN architecture. The available database of Arabic handwritten alphabets on Kaggle is utilized for examining the model. This database consists of 16,800 images divided into two datasets: 13,440 images for training and 3,360 for validation. As a result, the model gives a remarkable accuracy equal to 99.02%.

Keywords: Handwritten Arabic characters, Deep learning, Morphological gradient, Convolutional neural network, Optical character recognition.

Introduction:

Optical character recognition (OCR) is a technique that transforms several types of documents into editable and useable formats, including scanned paper documents, PDF files, and digital photographs. Depending on the type of writing, printed or handwritten, OCR offers various methodologies. Static recognition, also known as "offline," which operates on a snapshot of digital ink (on an image), and dynamic "online" recognition, where the symbols are identified as they are written by hand, are two separate domains that are taken into consideration¹. Text transcription automation applied to inherited documents is among the application areas of OCR, taking into consideration the irregular and complex nature of writing². The researches that take the classification of Arabic manuscripts are insufficient compared to the studies which deal with the other languages³.

Arabic is the 5th most widely spoken language in the world, with an estimated population of 420 million speakers globally. It is the official language of 26 countries, mainly in the Middle East and North

Africa region, but is also spoken in some countries in Central Asia and Sub-Saharan Africa^{4, 5}. Classification of Arabic alphabets using deep learning involves the use of machine learning techniques, specifically deep neural networks, to classify Arabic letters based on their features. This involves feeding an algorithm with data in the form of images or text representing Arabic letters and training the algorithm to recognize patterns and distinguish between different letters. The algorithm then uses this knowledge to accurately classify new, unseen data. Deep learning methods have been shown to be effective for Arabic alphabet classification due to their ability to learn hierarchical representations of data, making them well-suited for the complex patterns present in Arabic script⁶.

The Arabic language is composed of 28 distinct alphabets illustrated in Table 1⁷, and in recent years, researchers in the field of handwriting classification have made great strides in using deep learning algorithms to identify not only these alphabets, but also Arabic numerals. This is an

exciting development in the world of handwriting classification ⁸, and it demonstrates the potential of deep learning algorithms to help us better understand and categorize written text. By using these algorithms, researchers can more accurately identify different handwriting styles and fonts types, which can be extremely useful for a variety of applications, such as improving handwriting recognition in mobile devices ⁹, or even helping to digitize historical documents. The advancements in this field have paved the way for even more exciting developments in the future, and continued growth and improvement in the accuracy of deep learning algorithms can be expected as they are applied to the recognition of Arabic script and numerals. ¹⁰. Here are several methods used for Arabic handwriting classification: Convolutional Neural Networks (CNNs): These are deep learning algorithms that can be trained to recognize patterns and features in handwriting images. Recurrent Neural Networks (RNNs): These are other types of deep learning algorithms that are

well suited for sequential data, such as handwriting, where the order of the strokes is important. Support Vector Machines (SVMs): These are machine learning algorithms that can be trained to classify handwritten data based on its features, such as the shape, orientation, and size of the strokes. Dynamic Time Warping (DTW): This is a time-series analysis method that can be used to compare the similarity of handwriting sequences over time, and can be used in combination with other methods for improved accuracy. HMM-based: These models use Hidden Markov Models to recognize the underlying structure of the handwriting ¹¹, which can be useful in classifying handwriting styles and variations ¹². These methods can be used either individually or in combination to improve the accuracy of Arabic handwriting classification, and different methods may be more suited to different types of handwriting and recognition tasks.

Table 1. Arabic Alphabets.

alef	bae	tae	thae	jeem	hae	khae
ا	ب	ت	ث	ج	ح	خ
dal	thal	rae	zay	Seen	sheen	sad
د	ذ	ر	ز	س	ش	ص
dad	Tae	zae	ain	ghain	fae	qaf
ض	ط	ظ	ع	غ	ف	ق
kaf	lam	meem	noun	hae	waw	yae
ك	ل	م	ن	ه	و	ي

This research study focuses on the recognition of Arabic characters using the datasets of Ahmed El-Sawy et al, which can be accessed on the Kaggle platform. This dataset was created by 60 individuals who fall within the age range of 19 to 40 years old ¹³. The combination of CNN and the Morphological Gradient method can bring improved accuracy, better handling of variability, robustness, and speed in the classification of Arabic alphabets.

Related work:

Classification of Arabic alphabets is a popular topic in the field of machine learning and computer vision, and there have been many papers published on this topic. Convolutional neural networks are the primary focus of the majority of this field's study. This is the case for the studies:

Ahmed El-Sawy et al ¹⁴, create the Arabic handwritten characters database 16,800 images. They add a regularization parameter to the loss function in order to reduce the problem of overfitting. They use also Relus as activation

functions for the hidden layers which are the efficiency activation functions used recently comparing to sigmoid and hyperbolic tangent functions. ReLU function prevents the problem of vanishing gradient which is common when using sigmoid function. This problem is related to the tendency of neuron gradients to approach zero at high input values. This convolutional neural network gives 5.1% as loss and an accuracy of 94.9% in the validation dataset. The authors highlight the benefits of using CNNs, including their ability to learn feature hierarchies and their robustness against overfitting. The paper only focuses on classifying Arabic handwritten characters and does not consider printed Arabic characters, which may have different characteristics.

The study in ¹⁵ dealt with the AIA9k and AHCD databases. The paper introduced a sophisticated deep neural network for the purpose of recognizing handwritten Arabic characters. The network is based on Convolutional Neural Network (CNN) models and incorporates regularization

techniques such as batch normalization to ensure robust performance and prevent overfitting. The authors tested the effectiveness of their proposed solution on two datasets: AIA9k and AHCD. The results indicated that the deep neural network achieved remarkable classification accuracy, with a score of 94.8% and 97.6% respectively on the two datasets. Furthermore, the authors conducted additional studies on the network's performance using the EMNIST dataset and a form-based AHCD dataset to provide further insights and support the analysis. These results demonstrated the potential of deep neural networks to provide robust solutions for handwritten Arabic character recognition problems.

It is impossible to discuss the classification of Arabic letters in handwriting without mentioning Arabic numerals. In this regard, the study in ¹⁶ focused on introducing a novel architecture that leveraged a deep implementation of the Restricted Boltzmann Machine (RBM) as a feature extractor and a Convolution Neural Network (CNN) as a classifier. The RBM algorithm is considered to be sophisticated, as it has the ability to extract meaningful insights from raw and unprocessed data. The research made use of handwritten Arabic numerals for evaluation, and results indicated that this new architecture achieved a higher accuracy rate compared to when using the CMATERDB 3.3.1 database. This implies that the integration of RBM and CNN proves to be a promising approach for improving the accuracy of pattern recognition and classification tasks.

These papers demonstrated the progress that has been made in the field of Arabic alphabet recognition and the effectiveness of deep learning approaches, such as Convolutional Neural Networks, in achieving high accuracy rates.

**Method:
Motivation**

The Arabic language is very rich at the level of words is the styles of writing and also the linguistic compositions. Despite this, the number of studies that agree on this language is still insufficient. In recent years, Arab researchers are moving towards the recognition of Arabic characters ^{7, 15, 17}. In this side the best value of accuracy obtained tends towards 97%. This value is remarkable, but the development that knows the field of deep learning can override this value. Also, the nature of the images plays a very important role in this aspect, and since the images are binary, the possibility of obtaining very high precision using deep learning methods is logical. This is what leads us to aim for this area.

Dataset

The dataset of Arabic handwritten characters consists of 16,800 alphabets. Sixty participants of ages 19 to 40 contributed in writing of that dataset. 90% of the participants were right-handed. As shown in Fig 1, all the alphabets were written ten times by each of the participants. The forms' resolutions were 300dpi. The database was divided into two sets: a training set (13440 characters x 480 images per class) and a validation set (3360 characters x 120 images per class). The creators of training sets and validation sets were exclusive. The order of authors in the validation set was randomized so that the creators of the validation set did not belong to a single institution (to see the variability of the validation set) ¹⁴.

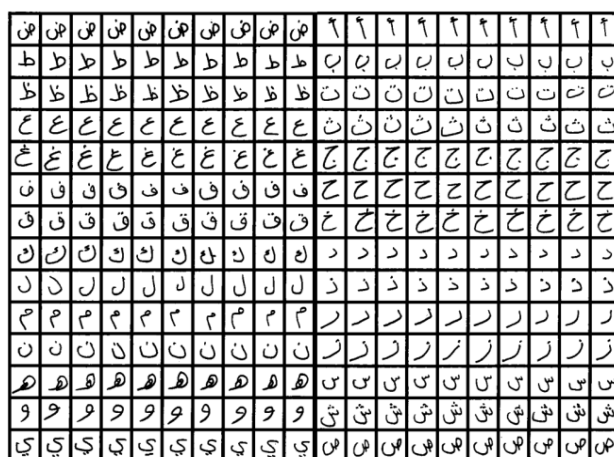


Figure 1. Contribution of a participant ¹⁴.

Architecture

This section will cover the proposed approach, which is visually represented in Fig 2. As seen in this figure, our method was divided into two parts: the first is the mathematical morphological gradient method. Because the processed images are binary, this approach is used, it allows us to highlight the contours of the alphabets in the images according to the increase in thickness of the alphabets. The principle of the method is the use of a succession of dilation and erosion. The dilatation increases the contours of the alphabets using a structuring element (A kernel consisting of ones with a size of 3x3 is utilized). After that came the erosion method to reduce the contours using the same structuring element. So, the difference between these two methods gives an inverted binary image which contains the outlines of the alphabet in white. The second part took the product of this phase and transitions it to a CNN (Convolutional Neural Network) architecture. CNNs in general are massive methods that give remarkable results by playing with its parameters. They use filters to extract features from images and fully connected layers to classify

images. So, they are self-extractors and classifiers at the same time ¹⁷.

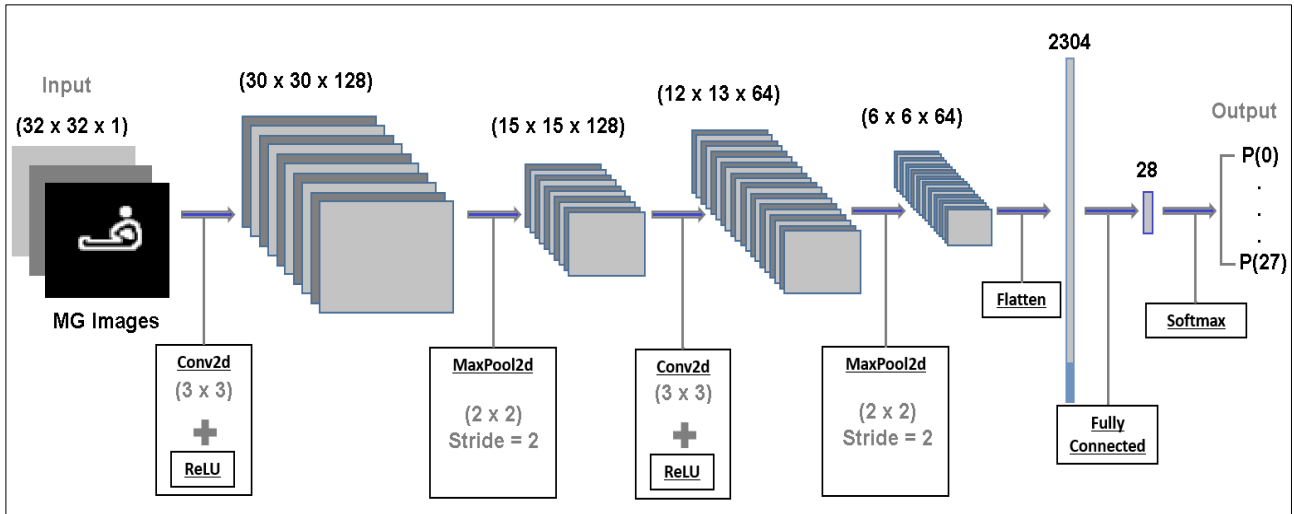


Figure 2. Morphological gradient convolutional neural network method.

Morphological Gradient Architecture

The first phase in the method involved the Morphological gradient, which can be expressed mathematically as shown in Eq. 1. The morphological gradient product highlighted the outlines of the alphabets based on the distinction between the mechanisms of dilatation and erosion.

$$G_{ES}(I) = (I \oplus Es) - (I \ominus Es) \quad 1$$

Where $I: Ec \rightarrow R$ an image in grayscale, Ec is a discrete grid that included in Z^2 or R^2 , Es is a structuring element that browses the image.

The operator \oplus indicate the dilatation. It writes as followed:

$$\begin{aligned} I \oplus Es &= \bigcup_{b \in Es} Ig_b \\ &= \bigcup_{I \in Ig} Es_b \\ &= \{I + b, I \in Ig, b \in Es\} \end{aligned} \quad 2$$

The operator \ominus denote the erosion and write as followed ¹⁸:

$$\begin{aligned} I \ominus Es &= \bigcap_{b \in Es} Ig - b \\ &= \{p \in Ec, Es_p \subseteq Ig\} \end{aligned} \quad 3$$

The dilation plays the role of a constructor according to the increase in the sizes of the alphabets using the structuring element. It can merge pixels when the distance between them is less than the size

of the structuring element. Conversely, erosion removes pixels from an object if it is small from the structuring element, which gives the possibility of also keeping certain connected objects. Fig 3, shows some images after the application of morphological gradient algorithm.

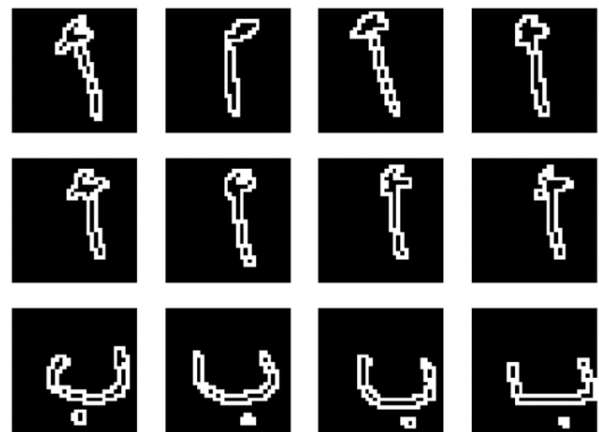


Figure 3. Samples of morphological gradient results.

Convolutional Neural Network (CNN)

Convolutional layer: Convolution consists in calculating the value of a given pixel of a starting image based on the value of the pixel itself and on the value of the pixels surrounding the pixel to be recalculated. A convolution is associated with a kernel (kernel which is a matrix which defines the weightings to be considered to calculate the convolution product). The result of the convolution phase called a feature map as shown in Fig 4. Eq 4 it shows the mathematical application of the first convolutional layer.

$$\text{Convolution}[Ig^{(lr-1)}, F^{(m)}]_{s,y} = 4$$

$$A^{(lr)} \left[\sum_{i=1}^{m_H^{(lr-1)}} \sum_{j=1}^{m_W^{(lr-1)}} \sum_{k=1}^{m_C^{(lr-1)}} F_{i,j,k}^{(m)} Ig_{x+i-1, y+j-1, k}^{(lr-1)} b_m^{(lr)} \right]$$

$Ig^{(lr-1)}$ is the input layer with size $(m_H^{(lr-1)}, m_W^{(lr-1)}, m_C^{(lr-1)})$, $Ig^{(0)}$ denote the image in the input. $F^{(m)}$ is a filter with dimension: $(f^{(lr)}, f^{(lr)}, m_C^{(lr-1)})$ and $m_C^{(lr)}$ is the number of filters. $b_n^{(lr)}$ is the bias of the m^{th} convolution. $A^{(l)}$ is the activation function, $Ig^{(lr)}$ denote the output with size $(m_H^{(lr)}, m_W^{(lr)}, m_C^{(lr)})$ and $m \in [1, 2, \dots, m_C^{(lr)}]$.

One of the most popular activation functions in contemporary deep learning networks is the rectified linear unit (ReLU). In artificial neural networks (ANN), the activation function serves as a part of an artificial neuron and is in charge of analyzing weighted inputs and assisting in producing an output. When the ReLU is used as the activation function, the function only returns positive values when the input is positive; otherwise, it returns zero (see Eq 5) ¹⁹.

$$\omega(X_i^l) = \text{Max}(0, X_i^l) \quad 5$$

ReLU activations are glaringly the powerful non-linear functions used in neural networks. Its derivative gives the simplest 1 when the input is positive. So, the fashionable propagation bug of the sigmoid function has no compression effect. Researches have proven that ReLUs bring about a great deal quicker training for large networks ^{20, 21}. On its hidden layers, the deep learning framework (TensorFlow) makes it smooth to apply ReLU, this eliminates the need to manually impose them. ²².

Two convolutional layers were employed in the model. The first layer (which has 128 filters of size (3×3) , valid padding, and stride = 1) gave an input morphological gradient image of size $(32 \times 32 \times 1)$. It generated large images $(30 \times 30 \times 128)$. The output pictures of the first max pooling layer, which has a size of $(15 \times 15 \times 128)$, were sent through the second convolutional layer, which has 64 filters of size (3×3) , valid padding, and stride = 1. It produced images of size $(13 \times 13 \times 64)$. Small filter sizes were used to retain the delicate details in the images.

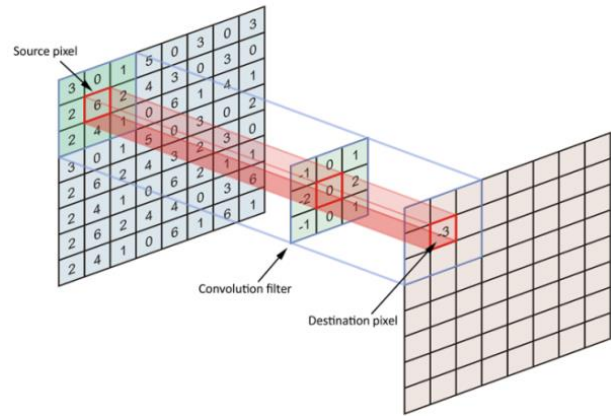


Figure 4. Filtering in a convolutional layer.

Max-Pooling layer: A down-sampled version of an image is the goal of pooling. The input image was divided into a series of squares of m pixels and non-overlapping sides (pooling). It is possible to display each square as a tile. The values that the individual squares' pixels produce was used to define the tile output signal. Pooling reduced the spatial size of the intermediate image ²³, which lowered the network's requirement for parameters and computations. To prevent overlearning, in a CNN model, it is usual to sporadically add a pooling layer between two adjacent convolutional layers. A translational invariance format is also created during the pooling process ²⁴.

The pooling phase is expressed mathematically as shown in Eq 6.

$$Ig_{x,y,z}^{(l)} = \text{pooling}(Ig^{(l-1)})_{x,y,z} \quad 6$$

$$= A^{(l)} [(Ig_{x+i-1, y+k-1, z}^{(l-1)})_{(i,k) \in [1, 2, \dots, f^l]}]^2$$

$Ig^{(l-1)}$ denote the input layer of size $(m_H^{(l-1)}, m_W^{(l-1)}, m_C^{(l-1)})$, $Ig^{(0)}$ indicates the image in the input, f^l denotes the size of pooling filter, $A^{(l)}$ is the activation function, $Ig^{(l)}$ indicate the output of size $(m_H^{(l)}, m_W^{(l)}, m_S^{(l)})$.

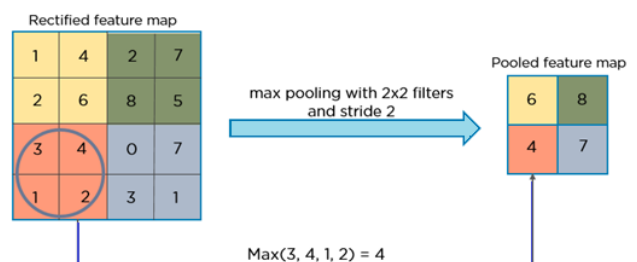


Figure 5. Max pooling process.

Two max pooling layers were utilized in the model (Fig 5). The first max pooling layer (128

filters with a size of (2 x 2) and a stride of = 2) received the output images from the first convolutional layer, which have dimensions of 30 x 30 x 128, and returns images of (15 X 15 x 128). The second convolutional layer's output images (13 x 13 x 64) were passed through 64 filters with a size of (2 x 2) and a stride of 2, and they returned as images with a size of (6 x 6 x 64). To preserve greater pixel densities and extract useful information, max pooling was used.

Fully connected layer: After a few layers of convolution and maxpooling, the neural network performed the high-level inference process through the fully connected layers. The neurons in the fully linked layer were coupled with all of the outputs from the layer above (as seen Fig 6). As a result, matrix multiplication and polarization shift could be used to determine their activation function ²⁵. In general, considering the j^{th} node of the i^{th} layer, Eq 7 was obtained.

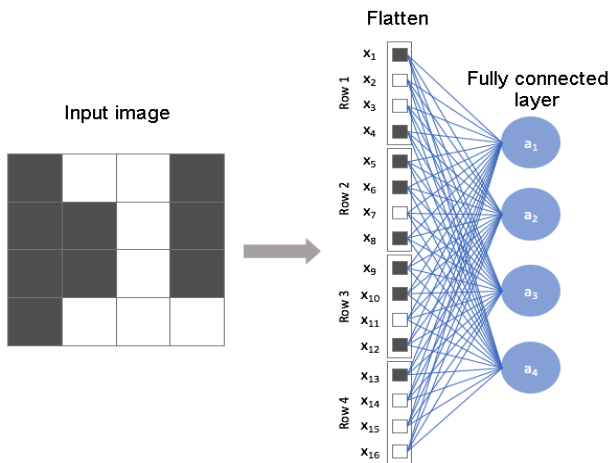


Figure 6. Fully connected layer architecture.

$$FC_j^{(i)} = \sum_{l=1}^{m_{i-1}} w_{j,l}^{(i)} I g_l^{(i-1)} + b_j^{(i)} \quad 7$$

The input $I g^{(i-1)}$ was the result of the max pooling layer, with the dimensions $(m_H^{(i-1)}, m_W^{(i-1)}, m_C^{(i-1)})$. The tensor was flattened into a 1D vector $(m_H^{(i-1)}, m_W^{(i-1)}, m_C^{(i-1)}, 1)$ to be able to plug it into the fully connected layer, thus: $m_{i-1} = m_H^{(i-1)} * m_W^{(i-1)} * m_C^{(i-1)}$. Parameters at the l^{th} layer were Weights $W_{j,l}$ with $m_{l-1} * m_l$ and bias with m_l parameters.

Like the sigmoid function, the softmax function limits each neuron's output to a value between 0 and 1, making the sum of all outputs 1. A probability distribution by category was the output of the softmax function. It displayed the likelihood that at least one of the target classes is accurate. Mathematically, the softmax function ²⁶ was written as shown in Eq 8.

$$Softmax(s)_j = \frac{e^{s_j}}{\sum_{k=1}^K e^{s_k}} \quad 8$$

If there were 28 output units, then there would be 28 items in s . The input vector of the FCL was s (the number of target classes in our case) ²⁷. The output units were indexed by j , so $j = 1, 2, \dots, K$.

CNNs were terminated by fully connected layers. They served as the model's classifiers. It was then flattened to a 1D vector and given a feature map of the second max pooling layer. In accordance with the number of Arabic alphabets, it has 28 neurons. This layer, employing the softmax activation function, should transform the neuron findings into a categorical probability of 0 to 1, indicating that one of the target classes is accurate.

Experience, results and discussion

The dataset of 16,800 images of Arabic alphabets, was divided into two separate databases (13440 for training and 3360 for validation), and used to process the suggested model. Fifteen iterations were used to build the model. The dataset was processed using a batch consisting of 32 images. The optimization function employed was the Adam technique. The loss was calculated using the mean squared logarithmic error. The backend Tensorflow libraries and Keras were used in Python-based experiments. The test-bed computer had an Intel(R) Core (TM) i5-8265U CPU was running at 1.60GHz with 16 GB of RAM and had an inbuilt graphics card with 8GB of memory.

Our method produced remarkable results. Fig 7 displays the training dataset's accuracy. This typically ends up being 99.96%, which is unusual for images recognition. Starting with a high value of 97.30% on the first iteration of the method, this procedure increases throughout time to attain the final accuracy (99.96% on the 15th iteration).

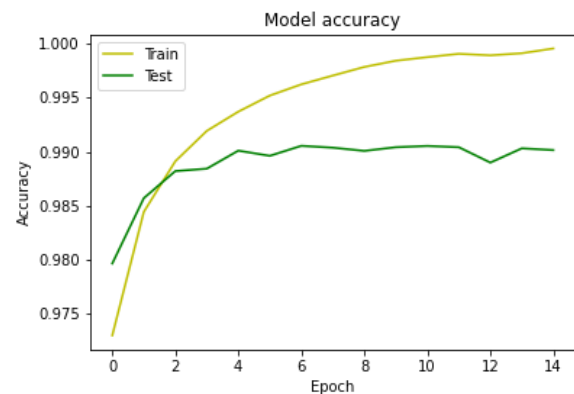


Figure 7. Accuracy of our model.

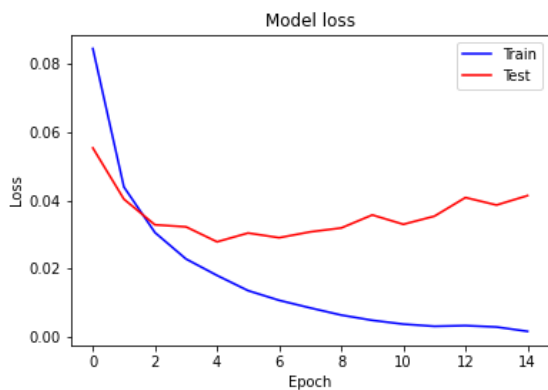


Figure 8. Loss of our model.

The number of iterations demonstrated the efficiency of our model which reached final accuracy in 15 iterations. The validation sample's results were equally impressive, starting at 97.97% on the first try and stabilizing at 99.02%. This demonstrates the capability to comprehend fresh images that are not part of the training collection. Figure 8 shows how our model failed (training and validation databases). The loss started at a low value of 8.46% and stabilized at a too low value of 0.16% after 15 rounds of the training database execution procedure. The loss for the validation database began at 5.55% and steadied at 4.15% by the time the algorithm had finished running.

The creators of the Arabic characters database also used a CNN to process their database. Their model gave remarkable results tending towards 94% as precision with a small loss of 5.1%. Despite these results, our method was able to overcome them using a morphological gradient algorithm. Table 2, shows a comparison between the two methods.

Table 2. Comparison between the original method and ours

Method	Dataset	Accuracy (%)	Loss (%)
Original method	Arabic Handwritten characters	94.9	5.1
Our method	(16,800 image)	99.02	4.15

To demonstrate the practical applications of our model, A graphical user interface (GUI) was employed, which enabled the input of a handwritten alphabet and the specification of its class based on the model parameters. The results of some tests conducted using this interface are illustrated in Fig 9. The GUI enabled us to present the results in a visually appealing manner and assess the performance of our model in a real-world.

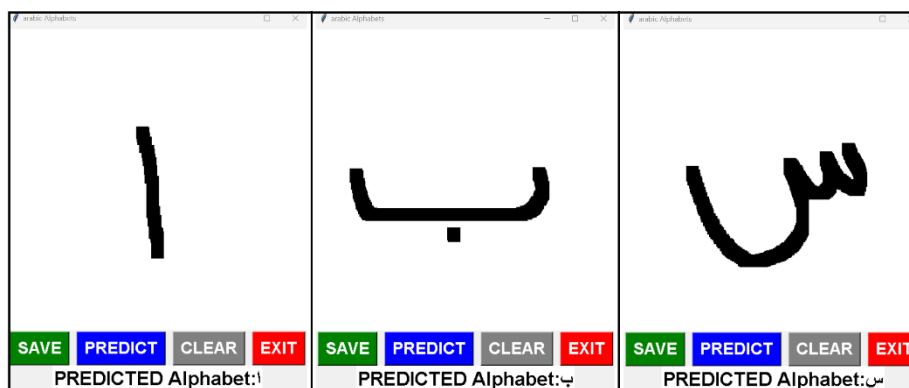


Figure 9. Real graphical user interface results.

Conclusion and Perspectives:

The improvement of automated software systems utilized in significant applications across various industrial sectors was possible with Arabic character recognition. In this study, a mathematical morphological gradient was used to introduce a convolutional neural network. This model recognized Arabic alphabets handwriting dataset with an accuracy of 99.02%. Batch normalization was a regularization technique used by the model to increase efficiency and accuracy. The Keras framework and Tensorflow were used to create the system.

As a future work, deeper learning techniques will explore on larger and more diverse datasets. This will be achieved by combining multiple datasets that recognize different elements of the Arabic language, including letters, numerals, and intrusive characters. The goal is to create a more advanced network that is capable of recognizing a total of 41 different classifications, including 28 Arabic letters, 10 Arabic numerals, and 3 Arabic intrusive characters. By working with a more comprehensive dataset, the model is likely to achieve better results and accuracy.

Authors' Declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in USMBA, Morocco.

Authors' Contributions Statement:

K. EL. F. gave the idea of the article and supervised the general work. J. R. followed up the work step by step and gave remarks to improve the scientific and linguistic value. M. EL. A. implemented the source code and wrote the paper.

References:

1. Singh H, Sharma RK, Singh VP. Online handwriting recognition systems for Indic and non-Indic scripts: a review. *Artif Intell Rev.* 2021; 54: 1525–1579. <https://dx.doi.org/10.1007/s10462-020-09886-7>.
2. Ríos-Vila A, Esplà-Gomis M, Rizo D, Ponce de León PJ, Iñesta JM. Applying Automatic Translation for Optical Music Recognition's Encoding Step. *Appl Sci.* 2021; 11(9): 3890. <https://dx.doi.org/10.3390/app11093890>.
3. USHIODA E. The Impact of Global English on Motivation to Learn Other Languages: Toward an Ideal Multilingual Self. *Mod Lang J.* 2017; 101: 469-482. <https://dx.doi.org/10.1111/modl.12413>.
4. IPAC Team. Quelles sont les 10 langues les plus parlées au monde. IPAC . 2018 Jun. <https://www.ipac-traductions.com/blog/10-langues-plus-parlees>.
5. Andrea Z. Developing a Heritage Database for the Middle East and North Africa. *J Field Archaeol.* 2018; 43:sup1 S9-S18. <https://dx.doi.org/10.1080/00934690.2018.1514722>.
6. Chaabane S, Chaabna K, Abraham A, Mamtani R, Cheema S. Physical activity and sedentary behaviour in the Middle East and North Africa: An overview of systematic reviews and meta-analysis. *Sci Rep* 10. 2020; 9363. <https://dx.doi.org/10.1038/s41598-020-66163-x>.
7. Abdalkafor AS. Survey for Databases on Arabic Off-line Handwritten Characters Recognition System. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS). 2018; 1-6. <https://dx.doi.org/10.1109/CAIS.2018.8442001>.
8. A AL-Saffar, S Awang, W Al-Saiagh, S Tiun, A S Al-khaleefa. Deep learning algorithms for arabic handwriting recognition. *IJET (UAE)* . 2018; 7: 344-353. <https://dx.doi.org/10.14419/ijet.v7i3.20.19271>.
9. Dargan S, Kumar M. Gender Classification and Writer Identification System based on Handwriting in Gurumukhi Script. *Int Conf comp commun intell Syst.* 2021; 388-393. <https://dx.doi.org/10.1109/ICCCIS51004.2021.9397201>.
10. AL-Shareeda M, Manickam S, Saare M. DDoS attacks detection using machine learning and deep learning techniques: analysis and comparison. *Bull Electr Eng Inform.* 2023; 12(2): 930-939. <https://dx.doi.org/10.11591/eei.v12i2.4466>.
11. Al-Mekhlafi Z G, Al-Shareeda M A, Manickam S, Mohammed B A, Qtaish A. Lattice-Based Lightweight Quantum Resistant Scheme in 5G-Enabled Vehicular Networks. *Mathematics.* 2023; 11(399). <https://dx.doi.org/10.3390/math11020399>.
12. Mahmood A, Al-Shareeda M, Selvakumar M, Murtaja Ali, Navaneethan C A. Proposed security mechanism for preventing fake router advertisement attack in IPv6 link-local network. *Indones. J Electr Eng Comput.* 2023; 518-526. <https://dx.doi.org/10.11591/ijeecs.v29.i1.pp518-526>.
13. Loey Mohamed. Arabic Handwritten Characters Dataset. Kaggle. 2017 jun. <https://www.kaggle.com/datasets/mloey1/ahcd1>.
14. El-Sawy A, Loey M, El-Bakry H. Arabic handwritten characters recognition using convolutional neural network. *WSEAS Trans Comput.* 2017 [cited 2022 Oct. 23]; 5(1): 11-19. <https://wseas.com/journals/articles.php?id=3300>.
15. Younis K S. Arabic handwritten character recognition based on deep convolutional neural networks. *Jordanian J Comput Inf Technol.* 2017; 3(3): 186. <https://dx.doi.org/10.5455/jjcit.71-1498142206>.
16. Alani Ali A. Arabic Handwritten Digit Recognition Based on Restricted Boltzmann Machine and Convolutional Neural Networks. *Information.* 2017; 8(4): 142. <https://dx.doi.org/10.3390/info8040142>.
17. Asroni A, Ku-Mahamud KR, Damarjati C, Slamati HB. Arabic Speech Classification Method Based on Padding and Deep Learning Neural Network. *Baghdad Sci J.* 2021 Jun. 20 ;18(2(Suppl.): 0925. [https://doi.org/10.21123/bsj.2021.18.2\(Suppl.\).0925](https://doi.org/10.21123/bsj.2021.18.2(Suppl.).0925).
18. David Sh, Sebastien L, Jonathan W. La morphologie mathématique binaire pour l'extraction automatique des b atiments dans les images THRS. *Revue internationale de Géomatique.* 2007; 17(3-4): 333-352. <https://dx.doi.org/10.3166/geo.17.333-352>.
19. El Atilah M, El fazazy K. Deep Morphological Gradient for Recognition of Handwritten Digits. *Association for Computing Machinery. BDIoT'19: Proc 4th Int Conf Big Data Int Things.* 2019. <https://dx.doi.org/10.1145/3372938.3372993>.
20. Thakare A, Gondane S, Prasad N, Chigale S. A Machine Learning-Based Approach to Password Authentication Using Keystroke Biometrics. *Deep Learning and Computational Intelligence for Wireless Communication. Lecture Notes in Electrical Engineering Machine Learning.* Singapore. 2022; 749. <https://dx.doi.org/10.1007/978-981-16-0289-430>.
21. Siddiqui N, Dave R, Vanamala M, Seliya N. Machine and Deep Learning Applications to Mouse Dynamics for Continuous User Authentication. *Mach Learn Knowl Extr.* 2022; 4(2): 502-518. <https://dx.doi.org/10.3390/make4020023>.
22. Gholamalinezhad H, Khosravi H, S L. Pooling Methods in Deep Neural Networks, a Review *CoRR* .

- 2020 arXiv:2009.07485.
<https://arxiv.org/abs/2009.07485> .
23. Nagi J, Ducatelle F, Di Caro G A, Cireşan D, Meier U, Giusti A, et al. Max-pooling convolutional neural networks for vision-based hand gesture recognition. IEEE Int Conf Signal Image Processing Appl. 2011; 342-347.
<https://dx.doi.org/10.1109/ICSIPA.2011.6144164> .
24. Basha S S, Dubey S R, Pulabaigari V, Mukherjee S. Impact of fully connected layers on performance of convolutional neural networks for image classification. Neurocomputing. 2020; 378: 112-119.
<https://dx.doi.org/10.1016/j.neucom.2019.10.008> .
25. Al-Shareeda M A, Manickam S, Laghari S A, Jaisan A. Replay-Attack Detection and Prevention Mechanism in Industry 4.0 Landscape for Secure SECS/GEM Communications. Sustainability. 2022; 14(23): 15900.
<https://dx.doi.org/10.3390/su142315900> .
26. Al-Shareeda M A, Manickam S. COVID-19 Vehicle Based on an Efficient Mutual Authentication Scheme for 5G-Enabled Vehicular Fog Computing. Int J Environ Res Public Health. 2022; 19(23): 15618.
<https://dx.doi.org/10.3390/ijerph192315618> .
27. Hussien Z, Dhannoon Ban. Anomaly Detection Approach Based on Deep Neural Network and Dropout. Baghdad Sci J . 2020 Jun. 23; 17(2) : 0701.
[https://doi.org/10.21123/bsj.2020.17.2\(SD\).0701](https://doi.org/10.21123/bsj.2020.17.2(SD).0701)

تصنيف الأبجديات العربية باستخدام مزيج من شبكة عصبية تلافيفية وطريقة التدرج المورفولوجي

جمال ريفي

خالد الفزازي

محسن العاطي الله*

مختبر علوم الكمبيوتر والإشارات والأتمتة والمعرفة (LISAC)، كلية العلوم ظهر المهرز، جامعة سيدي محمد بن عبد الله بفاس، المغرب

الخلاصة:

مجالات التعرف الضوئي على الحروف (OCR) هي عملية تحويل صورة نصية إلى تنسيق نصي يمكن قراءته آلياً. يعتبر تصنيف المخطوطات العربية بشكل عام جزءاً من هذا المجال. في السنوات الأخيرة، شهدت معالجة قواعد بيانات الصور العربية بواسطة آليات التعلم العميق تطوراً ملحوظاً. ومع ذلك، لا يزال هذا غير كافٍ لتلبية الثروة الهائلة للمخطوطات العربية. في هذا السياق يتعامل هذا البحث مع بنية تعلمية عميقة لحل مشكلة تصنيف الحروف العربية المكتوبة بخط اليد. تعتمد هذه الطريقة على بنية شبكية عصبية تلافيفية (CNN) كمستخرج ومصنف ذاتي. مع الأخذ بعين الاعتبار الطبيعة الثنائية للصور المعالجة. تكتشف ملامح الأبجديات بواسطة الخوارزمية الرياضية للتدرج المورفولوجي قبل تمريرها إلى بنية CNN. تستخدم قاعدة بيانات الحروف الهجائية العربية المكتوبة بخط اليد المتوفرة على Kaggle لتدريب هذا النموذج. تتكون قاعدة البيانات هذه من 16800 صورة (13440 صورة لمجموعة بيانات التدريب و3360 صورة لمجموعة بيانات التحقق). ونتيجة لذلك، يعطي النموذج دقة ملحوظة تساوي 99.02%.

الكلمات المفتاحية: الحروف العربية المكتوبة بخط اليد، التعلم العميق، الشبكة العصبية التلافيفية، التدرج المورفولوجي، التعرف الضوئي على الأحرف.