Offline Arabic Handwriting Recognition Using a Deep Neural Approach

Benbakreti Samir [1[0000-0001-6587-6626]], Benouis Mohamed [2[0000-0002-9107-9329]] and Benkaddour Mohammed Kamel [3[0000-0003-1895-7540]]

[1] National Institute of Telecommunication and ICT, Oran, Algeria,
[2] Computer Science Department, University of M'sila, M'sila, Algeria,
[3] Department of Computer Science and Information Technology, FNTIC faculty, University Kasdi Marbah, Ouargla, Algeria.

Abstract. Arabic handwritten recognition systems face several challenges such as the very diverse scripting styles, the presence of pseudo-words and the position-dependent shape of a character inside a given word. These characteristics complicated the task of features extraction. The paper presents a deep neural approach for the handwritten recognition of Arabic words. This work is focusing on the offline recognition, thereby, the processed information represents an image. We chose the CNN method, which is one of the deep architectures which permits to remove several steps from the recognition process, including preprocessing and feature extraction. The used database is NOUN v3 contained images represented the Algerian cities. A CNN architecture was trained and then tested on the database to accomplish this task. The advantage of a CNN is that it can extract specific features from each image while compressing it to lower its initial size. Our experimental study, gives a satisfactory word recognition rate.

Keywords: Arabic Handwriting, Offline Recognition, Deep learning, CNN.

1 Introduction

Handwriting recognition (HWR) is an essential playground for new learning algorithms and remains a real scientific and technical challenge. It is strongly involved in several technological sectors (digital publishing, e-commerce, heritage studies, postal sorting, document security, etc.). Concerning the script itself, the increasingly sophisticated classification techniques seem to largely reduce the gap that existed between the performances obtained on the printed word and those obtained on the unconstrained manuscript, which is reputed to be more difficult. Especially since good recognition results for "clean" printed script cannot be generalized to handwriting, because handwriting poses other types of problems such as: recognition of characters with variable shapes and areas of handwriting. In addition, some handwriting [1] remains difficult to decipher by an algorithm, or even by a human, like the prescription.

There are different ways to proceed in handwriting recognition. The latter can be done "online" [2], the handwriting is thus captured in real time and can be represented in a sequence of chronologically ordered points that give valuable indications on the dynamics of the line, the velocity and the morphology of the handwriting. While "off-line" writing is obtained through an image by scanning an existing text [3], using a scanner or a camera. In this case, time information is not available.

Many image pre-processing steps were necessary to normalize the writing: filtering and removal of background noise (e.g. squares, marks due to aging of the media), conversion to a binary image, line detection, segmentation into words and characters (see below). Once these pre-processing steps were completed, features were calculated on the writing elements at the line, character or word level. Then pattern recognition algorithms were used to classify each of the elements of the script [4, 5].

The advent of deep neural networks has made it possible to get rid of several blocks, such as the pre-processing and extraction of relevant features from the image [6], as well as the use of the non-segmentation approach which reproduces the human reading mode, where the brain processes data on several levels simultaneously.

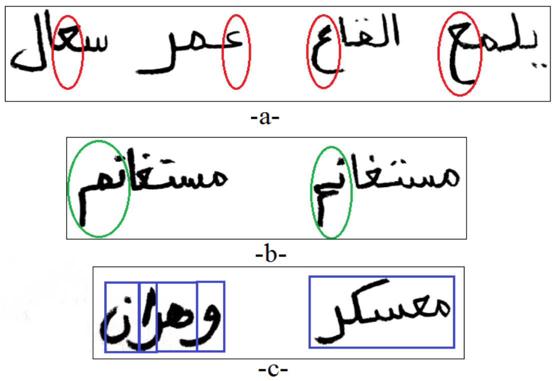

Fig. 1. Handwritten Arabic script. - a- The character shape changes according to its position in the word, -b- The vertical ligature is present on the right and does not appear on the left for the same word, -c- The word on the left is composed of 4 pseudo-words while the other word has no notion of pseudo-words.

In this paper, we will use CNN to perform off-line recognition of Arabic handwriting (words). The proposed approach represents a particular algorithm belonging to the family of deep neural networks,.This approach without segmentation allows to take the word in its entirety, therefore ligatures (links between characters) and pseudowords are no longer a problem for recognition.

2 Related Work

Although there are only 28 different characters in Arabic script, variations based on position lead to nearly 100 different character shapes. Moreover, the intra-character variations are quite large as shown in the figure 1. We can observe that the character shapes are visually very different from each other.

The authors propose in [7] an automated model of handwriting recognition based on convolutionary neural networks (CNN). A 97% output is provided by the training of the model on the Arabic Handwritten Character Dataset (AHCD).

In [8], Sana et al. propose an Arabic handwriting recognition scheme based on multiple combination architectures of BLSTM-CTC. The experiments carried out on the KHATT database showed that compared to the other fusion strategies, the high-level combination approach significantly enhances the recognition rate.

In [9], for the recognition of handwritten Arabic characters, the authors model a deep learning architecture (DLA). The DLA trained and checked a database containing 16800 handwritten Arabic characters in the supervised mode. Using CNN leads to major changes in various classification algorithms for machine learning. In the test results, the proposed CNN had a classification rate of 94.9 %.

Riaz Ahmed et al. suggested in [10] the implementation of a deep learning approach focused on Multi-Dimensional Long Short-Term Memory (MDLSTM) and Connectionist Temporal Classification networks (CTC). By acquiring 80.02 percent Character Recognition (CR) over 75.08 percent as a basis, the data increase with a deep learning approach proves to achieve a stronger and promising improvement in performance.

The authors used Convolutional Deep Belief Networks (CDBN) in [11] to test textual image data containing Arabic handwritten script (AHS) on two separate databases. The authors demonstrated experimentally that the extracted characteristics are productive for the identification of handwritten characters and demonstrate very good efficiency. Yet when applied to IFN/ENIT and HACDB databases, the proposed CDBN architectures achieved promising precision rates of 91.55 % and 98.86 % using Dropout.

A multi-step cascading recognition method based on the application of the Random Forest (RF) classifier to create a decision tree forest is the proposed approach in [12]. Based on the most discriminating geometric features, the decision trees broke the database into several sets. RF matches one of the data mining sets with each test image. Then using the Pyramid Histogram of Gradients and the Kullback-Leibler-based ranking algorithm, the corresponding classes are sorted against the test image. The proposed method was checked on the Arabic database of IFN-ENIT.

3 Proposed Model

Most of the literature cited above without considering word recognition, deals with the issue of character recognition, a much more challenging task and which is the main objective of this paper.

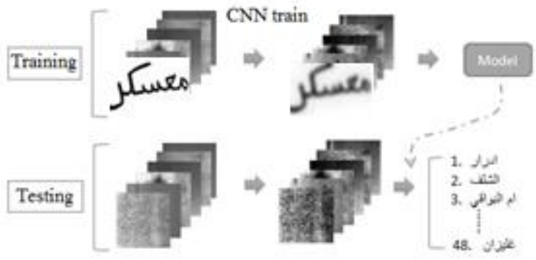

Fig. 2. Proposed system with CNN architecture.

The benefit of deep architecture is its ability to learn to hierarchically reflect the world. There is no need to create a feature extractor by hand, because all layers are trainable. The illustration of the training process is shown in Figure 2. The detailed description of the model is given in table 1.

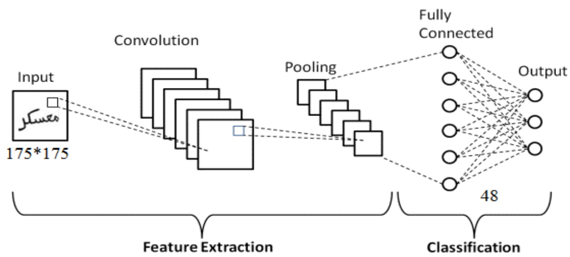

Fig. 3. CNN architecture.

Contrary to the classic MLP model (Multi-Layer Perceptron), the CNN architecture through the inclusion of the convolutionary component (illustrated in figure 3) allows a "characteristic map" or "CNN code" whose dimensions are smaller than those of the original image to be obtained as an output, which will have the benefit of significantly reducing the number of parameters to be calculated in the model.

3.1 CNN (Convolutional Neural Network)

CNN is a kind of anticipatory neural network composed of a set of convolutional layers and subsampling. CNN's architecture is based on research on the visual cortex of the cat by Hubel and Wisiel [13]. The visual cortex contains an arrangement of cells, which act as input space filters. The number of layers and their sizes are determined by the complexity and type of the problem to be treated.

The convolution layer is the first layer that is used to extract the various features from the input images. The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map.

A more detailed overview of CNN layers is described below.

- Layers with convolutional filter: a product is created between a filter (a two-dimensional table whose weights will be adjusted during learning) and the input of the layer. This filter is generally of dimension 3×3 or 5×5. The multiplication of these layers within the network will allow the extraction of more and more complex features that will ultimately allow the prediction of a class of membership for the item present in the image.
- Layers with pooling: they allow an undersampling that will compress the image dimension and reduce the computational cost of the following layers. In general, a maximum or average function is used.
- ReLU (Rectified Linear Units) is the non-linear real function defined by ReLU(x)=max(0,x). The ReLU correction layer therefore replaces all negative values received as inputs with zeros. It acts as an activation function.
- Fully Connected Layer (FC): This layer connects the output of the last layer. The last layer applies a softmax function to determine the class of the image among the ten categories. These last layers are reminiscent of the structure of a multi-layer perceptron.

The training phase is performed using K-fold cross validation where, using 2 folds for the training set. Then, the data enhancement is applied using the images obtained from the pre-processing. By applying rotations and scales, the data set is multiplied by four. The CNN is formed when these operations are applied to the dataset, by setting the input size of the image to 175*175.

The structure of our CNN as table1 shows is trained on a NOUN Database v3 to word recognition task. The network architecture includes number of layers. The input of CNN is gray face image of size 175 by 175pixels and is given to six convolution layers that compose the network with Conv 1 consist of 32 feature map, C2 consist of 64 feature map, C3 consist of 128 feature map…, as shown in table I. Six pooling layers are used, Layer S1, S2 … and S6 are subsampling layers, whose number of feature maps is equal to the maps number of their previous convolution. After the pre-processing layers, comes to the full connected layers FC with 48 neurons, each one connects to all the maps of last pooling layer followed by the output layer is also a full connected layer. The architecture of CNN is summarized in Table I.

Table 1. Detailed architecture of our CNN.

Name	Type	Description of output size
Input layer	Input data	1*175*175
Conv 1	Convolution+Relu	32_8*8 filters
S1	Max pooling	3*3, stride 2
Conv 2	Convolution+Relu	64_3*3 filters
S1	Max pooling	2*2, stride 2
Conv 3	Convolution+Relu	128_5*5 filters
S3	Max pooling	3*3, stride 2
Conv 4	Convolution+Relu	256_5*5 filters
S4	Max pooling	3*3, stride 2
Conv 5	Convolution+Relu	512_5*5 filters
S5	Max pooling	3*3, stride 2
Conv 6	Convolution+Relu	1024_5*5 filters
S6	Max pooling	3*3, stride 2
Fc	Fully connected	48 units 1*1

4 NOUN Dataset

Keep in mind that the purpose of the model proposed in this paper is to recognize Arabic words. These words come from the latest edition of our NOUN-DATABASE v3 hybrid database. The database included 2800 Arabic characters and 4800 words acquired from the complex entry of the plot in the acquisition tablet (representing the 48 names of Algerian cities). The extracted data represented the files from which we stored the written signal information online [2]. The novelty of this database is that we again asked the 20 scribers to capture the same cities in addition to the 7600 previous files, but this time the stored files are 175*175 resolution images. Each scripter enters the city five times, giving 4800 pictures.

5 Experiments and results

In this section, we will explore the experimental study conducted to investigate the performance of our system. The experiments were conducted using MATLAB 2019b, an Intel I Core I i7-2670QM processor computer with a clock speed of 2.20 GHz and a RAM of 6 GB. Through experiments, our proposed system, was tuned in order to find the optimal parameters. Then its various parameters (i.e. the percentage of training, LR and FV) were tuned in the experiments 1, 2 and 3 respectively. The optimal combination of these parameters was found through a heuristic study in the experiment 4.

The learning steps and CNN configurations are clarified in this section. The configuration of the CNN parameters is established based on the results obtained in each experiment to obtain the best classification results. The softmax layer is formed to produce 48 output classes representing Algerian city names.

Table 2. Initial parameters of the CNN

Parameter	Value
% of training (TR)	0.75
Activation function	Sgdm
Momentum	0.9
Mini Batch Size	64
Learning Rate (LR)	0.01
Max epochs	25
Validation Frequency	30

When:

- SGDM: Solver for training network, it use the stochastic gradient descent with momentum
- Momentum is the contribution to the current iteration of stochastic gradient descent with momentum of the parameter update stage of the previous iteration, defined as the comma-separated pair consisting of 'Momentum' and a scalar of 0 to 1.
- Mini Batch size: at each iteration the stochastic gradient descent algorithm evaluates the gradient and updates the parameters using a subset of the training data. A different subset, called a mini-batch, is use at each iteration to evaluate the gradient of loss function and update the weights.
- Learning rate used for training, specified as the comma-separated pair consisting of 'InitialLearnRate' and a positive scalar. The default value is 0.01 for the 'sgdm' solver and 0.001 for the 'rmsprop' and 'adam' solvers.
- The Validation Frequency value is the number of iterations between evaluations of validation metrics.

5.1 Experiment 1 (Impact of % training on classification results)

We will use a wide range of values for the % of training, the values attempted and the results obtained are shown in Table 3.

Table 3. Results of the variation of the % training parameter

TR	0.6	0.65	0.7	075	0.8	**0.85**	**0.9**
Acc (%)	89.06	92.41	91.67	90.72	91.67	**95.92**	**95.83**
Time (sec)	542	541	547	794	780	**815**	**815**

Discussion: The execution of the model showed an accuracy of 95.92%, which corresponds to the recognition rate of well classified Arabic words in an estimated classification time of 815 seconds. In the following experiments, we will optimize this recognition rate through parameters tuning of the CNN. It is also important to note that after each experiment, we will retain the optimal value for each parameter, in this case TR=85%.

5.2 Experiment 2 (Impact of learning rate value on classification results)

In this experiment, we will try to observe the impact of the variation of Learning rate (LR) on the previous results. The attempted values are shown in Table 4.

Table 4. Results of the variation of the LR parameter

LR	**0.005**	0.01	**0.015**	0.02	0.025	0.03
Acc (%)	**97.96**	95.92	**95.92**	87.76	85.71	82.3
Time (sec)	**789**	815	**789**	788	787	785

Discussion: We note that the best results are obtained with the parameters LR 0.005 and 0.015. The time classification is the same. Nevertheless, we note that the results obtained for a small value (0.005, 0.01 and 0.015) of LR are very interesting, the more we increase this parameter, the more the results worsen. In the following experiment, we will set the value of this parameter to 0.005.

5.3 Experiment 3 (Impact frequency validation value on classification results)

Now, we will see the impact of the FV variation on the previous results. The attempted values are shown in the table 5.

Table 5. Results of the variation of the FV parameter

FV	10	**15**	20	**25**	30	35	40
Acc (%)	87.76	**93.88**	83.67	**97.96**	89.9	87.67	85.71
Time (sec)	766	**761**	760	**774**	759	794	798

Discussion: We note that the better results are obtained by using FV values of 15 and 25 respectively.

5.4 Experiment 4 (Combination of the optimal parameters)

In this experiment, we will study the performance of our system regarding the different combination of the three parameters (i.e. %TR, LR and FV). Indeed, we will proceed to the variation of each parameter between its first and second best values i.e. %TR= 85 or 90, LR = 0.005 or 0.015 and FV = 15 or 25, which will give us 8 possible combinations. The results obtained are mentioned in Table 6.

Table 6. Results of the optimal parameters combinations

Combination	TR (%)	LR	FVV	CR	CT (sec)
1	85	0.005	15	95.92	773
2	85	0.005	25	93.88	773
3	85	0.015	15	89.90	771
4	**85**	**0.015**	**25**	**97.96**	**774**
5	90	0.005	15	95.83	775
6	90	0.005	25	91.67	771
7	90	0.015	15	91.67	770
8	90	0.015	25	97.92	768

Discussion: We note that in addition to the previously predicted combination (TR, LR, FV) = (85%, 0.015, 25), most of the combinations gave satisfactory results but never exceeded the previous recognition rate which was 97.96%.

6 Conclusion

In this paper, we have demonstrated the effectiveness of using CNN for the Arabic handwritten words offline recognition. The fact that the CNN belong to the deep learning allows to manage large inputs without performing any pre-processing or features extraction phase (e.g. raw images). It is the CNN itself which performs these operations by summarizing the image features during the reconstruction phase. This approach is very efficient for dealing with a multi-writer database and especially the Arabic handwriting and its problematic characteristics.

Our experimental study, shows that the optimal (%TR, LR, FV) parameters for our system were (85, 0.015, 25). We have get a significant improvement in classification time and the obtained recognition rate reaching 97.96% which was very encouraging.

References

1. Aini Najwa Azmi, Dewi Nasien, Siti Mariyam Shamsuddin. A review on handwritten character and numeral recognition for Roman, Arabic, Chinese and Indian scripts. International Journal Computer Science Engineering, 2(1). Pp: 1-8. (2013).
2. Benbakreti Samir, Aoued Boukelif, New approach for online Arabic manuscript recognition by deep belief network, Acta Polytechnica 58(5):297–307. (2018).
3. Ghadhban H.Q., Othman M., Samsudin N.A., Ismail M.N.B., Hammoodi M.R. Survey of Offline Arabic Handwriting Word Recognition. In: Ghazali R., Nawi N., Deris M., Abawajy J. (eds) Recent Advances on Soft Computing and Data Mining. SCDM 2020. Advances in Intelligent Systems and Computing, vol 978. Springer, Cham. (2020)
4. Atallah AL-Shatnawi, Faisal Al-Saqqar Mustafa A Holistic Model for Recognition of Handwritten Arabic Text Based on the Local Binary Pattern Technique. International Journal of Interactive Mobile Technologies (iJIM), 14(16). (2020).
5. Abu Sayeed A. Huquea, Mainul Haque, Haidar A. Khan, Abdullah Al Helal, Khawza I. Ahmed, Comparative Study of KNN, SVM and SR Classifiers in Recognizing Arabic Handwritten Characters based on Feature Fusion, Signal and Image Processing Letters. 1(2), pp. 41-49. (2019).
6. Joan Pastor-Pellicer, María José Castro-Bleda, Salvador España-Boquera, Francisco Zamora-Martínez, Handwriting recognition by using deep learning to extract meaningful features, Ai Communications 32(11):1-12. (2018).
7. Altwaijry, N., Al-Turaiki, I. Arabic handwriting recognition system using convolutional neural network. Neural Comput & Applic (2020).
8. S. K. Jemni, Y. Kessentini, S. Kanoun and J. Ogier, "Offline Arabic Handwriting Recognition Using BLSTMs Combination," 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, 2018, pp. 31-36, (2018)
9. H. M. Najadat, A. A. Alshboul and A. F. Alabed, "Arabic Handwritten Characters Recognition using Convolutional Neural Network," 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2019, pp. 147-151. (2019).
10. Riaz Ahmad, Saeeda Naz, Muhammad Afzal, Sheikh Rashid, Marcus Liwicki, and Andreas Dengel, A Deep Learning based Arabic Script Recognition System: Benchmark on KHAT, The International Arab Journal of Information Technology, 17(3), pp: 299-305. (2020)
11. Elleuch, M., & Kherallah, M. Boosting of Deep Convolutional Architectures for Arabic Handwriting Recognition. International Journal of Multimedia Data Engineering and Management (IJMDEM), 10(4), 26-45. (2019)
12. Ghanim, T., Khalil, M. and Abbas, H, Multi-stage Off-line Arabic Handwriting Recognition Approach using Advanced Cascading Technique, In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, 2019, pages 532-539. (2019)
13. D. H. Hubel, T. N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, J. Physiol, vol. 195, pp. 215-243, 1968.