



Handwritten Tifinagh Characters Recognition Using Deep Convolutional Neural Networks

Mohamed Benaddy¹ · Othmane El Meslouhi² · Youssef Es-saady³ · Mustapha Kardouchi⁴

Received: 31 March 2018 / Revised: 6 March 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Handwritten characters recognition is a challenging task due to the diversity of writing styles. In the present paper, a new offline recognition system based on deep convolutional neural networks (CNNs) is designed. It uses CNNs to extract features from raw pixels which makes the proposed technique more flexible than the other conventional ones that require a set of additional preprocessing steps to extract the desired invariants features. The proposed system is tested on the AMHCD data set and achieved the best recognition accuracy (99.10%) compared to the state-of-the-art results.

Keywords Pattern recognition · Machine learning · Deep learning · Deep convolutional neural networks · Amazigh handwritten characters recognition · Amazigh Tifinagh

✉ Mohamed Benaddy
m.benaddy@uiz.ac.ma

Othmane El Meslouhi
o.elmeslouhi@uiz.ac.ma

Youssef Es-saady
y.essaady@uiz.ac.ma

Mustapha Kardouchi
mustapha.kardouchi@umoncton.ca

¹ LabSIE Laboratory, Polydisciplinary Faculty of Ouarzazate, Ibn Zohr University, Ouarzazate, Morocco

² LabSIE Laboratory, Polydisciplinary Faculty of Ouarzazate, Ibn Zohr University, Ouarzazate, Morocco

³ IRF-SIC Laboratory, FSA & FPT of Ibn Zohr University, Taroudant, Morocco

⁴ Département d'Informatique, Université de Moncton, Moncton, Canada

1 Introduction

Amazigh languages are a family of languages in the Afro-Asiatic language phylum [1]. They are spoken by people living in scattered areas in a large region of northern Africa situated between Egypt's Siwa Oasis and Mauritania [2].

The used writing system of the Amazigh language is called *Tifinagh* [3], derived from the “*Libyco-Berber*” inscriptions used since the sixth century before Christ era by the populations of North Africa, the Sahel, and the Canary Islands [3, 4] (Fig. 1).

Ancient Tifinagh scripts are engraved in the stones and tombs of some historic sites in Algeria, Morocco, Tunisia, the Tuareg areas and Canary Islands [5]. Figure 2 shows an image of an ancient Tifinagh script found in the Dougga site in Tunisia.

Tifinagh has been modified from its origins to its present form, from Libyque to Neo-Tifinagh, passing through the Saharan and Tuareg Tifinagh [2, 4] Libyque is the oldest one used in the Mediterranean coasts from Kabylie, Constantine, Aurès (Algeria) to Morocco, Tunisia and in the Canary Islands (Spain) [3]. Saharan Tifinagh also called Libyco-Berber or the old Tuareg was used to transcribe the old Tuareg inscriptions. Neo-Tifinagh, based on the Tuareg Tifinagh, is designed for writing Amazigh dialects of the Maghreb (Morocco and Algeria) [3, 4].

Further information about the ancient and the modern Tifinagh can be found in the book [4] by Ameur et al. It also provides a history of the alphabet, its origin, its different variants and their decryption.



Fig. 1 Medallion of Carthage [5]

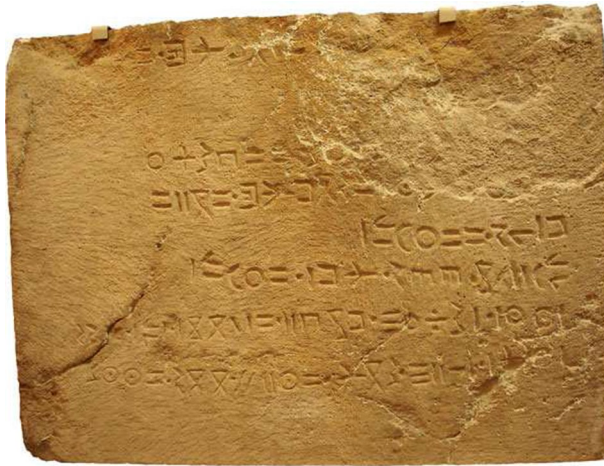


Fig. 2 An ancient Tifinagh script found in the Mausoleum of Atban (Dougga site, Tunisia) [5]

Concerning Morocco which contains the heaviest concentration of Amazigh speakers [1], Amazigh language is officialized since October 17, 2001, [2] after the creation of the Royal Institute of Amazigh Culture (Institut Royale de la Culture Amazigh-IRCAM) [6].

Amazigh language in Morocco has three varieties each of which is spoken in different regions: Tarifite in the north, Tamazight in the Atlas Mountains and Tachelhit in the south regions [2]. The IRCAM institute has adopted the so-called *Tifinagh-IRCAM* alphabet as the official alphabet for Amazigh language [2]. This alphabet is then officially recognized by the International Organization for Standardization (ISO) [7, 8]. Tifinagh-IRCAM contains 33 characters shown with their correspondent's pronunciation in Latin characters in Fig. 3.

The great variability inherent in the nature of handwriting has made this area of research very active. Therefore, with recent advances in computing technologies, several automatic handwriting recognition techniques have been improved and perfected, particularly for the Latin and the Arabic scripts [9–14]. However, despite existing works, establishing an automatic handwritten Tifinagh characters recognition system is an open research challenge and is still at its early beginning.

In this paper, we present a deep convolutional neural network (CNN) for Amazigh handwritten Tifinagh characters recognition system. The obtained results of our proposed system will be discussed and compared with other proposed approaches from the literature.

The structure of the rest of this paper is as follows. Section 2 introduces the related works and their review. Section 3 highlights the convolutional neural networks. The proposed deep CNN architecture is explained in Sect. 4. Experimental results are detailed in Sect. 5. Finally conclusions are made from the experimental results.

Fig. 3 Tifinagh characters adopted by the IRCAM with their pronunciation in Latin characters

ⵢ	ⵣ	ⵤ	ⵥ	ⵦ	ⵧ
Ya	Yab	Yey	Yah	Yu	Yar
ⵢⵕ	ⵣⵓ	ⵤⵓ	ⵥⵓ	ⵦⵓ	ⵧⵓ
Yarr	Yas	Yass	Yach	Yaz	Yazz
ⵢⵏ	ⵣⵏ	ⵤⵏ	ⵥⵏ	ⵦⵏ	ⵧⵏ
Yad	Yadd	Yaf	Yahh	Yae	Yax
ⵢⵓ	ⵣⵓ	ⵤⵓ	ⵥⵓ	ⵦⵓ	ⵧⵓ
Yaq	Yai	Yaj	Yal	Yam	Yan
ⵢⵓ	ⵣⵓ	ⵤⵓ	ⵥⵓ	ⵦⵓ	ⵧⵓ
Yat	Yatt	Yaw	Yay	Yag	Yagg
ⵢⵓ	ⵣⵓ	ⵤⵓ			
Yak	Yakk	Yagh			

2 Related Works

Recognition of Amazigh handwritten characters has become an active research field because of its potentials and various applications. Rachidi et al. [15], presented a state-of-the art and a comparison of scientific research works accomplished and published for automatic recognition of Amazigh characters. In [16], Aharrane et al. gave a comparative study of different supervised algorithms for handwritten Amazigh character recognition. Their goal is to compare performances of some popular classifiers (Bayesian Networks [17], Decision Trees [18] and Multilayer Perceptron [19]), using a set of proposed statistical features. Same authors proposed in [20] a handwritten Amazigh characters recognition system based on a statistical approach with a feature set (densities and shadow features) composed by 79 elements representing each Amazigh characters. In the recognition phase, they used a multilayer perceptron (MLP) as a classifier. The obtained accuracy using only 24,180 characters from the Amazigh Handwritten Character Database (AMHCD) [21] is 96.47%. A little improvement of this system is published later in [22] by combining three classifiers using some majority voting strategies.

Amrouch et al. [23, 24], developed an automatic Amazigh character recognition system based on the Hidden Markov Models (HMMs) [25]. They start by a pre-processing step on the character image, then they use the Hough transformation to represent each character by a string of numbers. The resulting string is injected to an unidirectional HMM (1D-HMM) to achieve the learning task using Baum–Welch algorithm. Finally they use the Viterbi algorithm to recognize the desired character. To evaluate the performance of the proposed system, the authors use only 24,180 among 25,740 characters constituting the entire AMHCD database (as in [22]). The best score obtained is 97.89% with a model having 14 states and 1 or 2 Gaussians.

Es-saady et al. [26, 27], proposed an approach based on the determination of the character's horizontal central line. Based on its position, a set of statistical character's features are calculated using a sliding windows technique. The used classifier is a MLP with: one input layer composed by 90 neurons, one hidden layer and one output layer with 31 neurons. An improvement of this approach was published in [7], by using the character's horizontal baselines instead of central line. They experiment the proposed approach using only 24,180 characters from the AMHCD database as all the above mentioned works and obtain 94.62% as the best recognition rate.

In [28] authors proposed and compare performances of two networks to recognize Tifinagh characters: Convolutional Neural Networks (CNNs) and Deep Belief Network (DBN) [29]. The proposed CNN is composed of 7 convolutional layers with Relu activation function. The DBN is composed of three and four hidden layers by varying the number of neurons in each hidden layer from 500-500-2000-31 to 1000-1000-1000-2000-31. By using AMHCD handwritten character database, authors prove that CNNs outperform the DBN with an accuracy equal to 98.25%.

All mentioned approaches have three major weaknesses: first, they need a preprocessing techniques that require a significant computation time, second, they can cause an important confusion in the recognition of some characters such as 'Yas' and 'Yar', 'Yaz' and 'Yazz', 'Yadd' and 'Yatt', and 'Yay' and 'Yag', third, they use only at most 31 characters instead of 33 Amazigh characters. To overcome these limits, we propose in this paper a robust and fully automatic handwritten recognition system. This system extracts Tifinagh character's features directly without supplementary preprocessing step and it can recognize all characters in the AMCHD database.

3 Convolutional Neural Networks

A convolutional neural networks (CNNs) is a concatenation of an input layer, an output layer, and a multiple hidden layers. Compared with fully connected neural networks, the parameters number of these models is widely reduced by sharing weights and biases [30]. LeNet [31] is the first Convolutional Neural Network (CNN) that were developed for handwritten digit recognition since 1998. At that time CNNs were restricted by high computation and memory resources allocation costs. But after the emergence of GPUs and the use of Relu activation functions, instead of Sigmoid and Tanh, CNNs have demonstrated excellent performance on image recognition and classification tasks, especially for handwritten digits and characters recognition. Several papers have been reported in the literature such as handwritten Chinese [32], handwritten character classification [33] and handwritten digits classification [34].

A typical convolutional architecture using an RGB image as input is shown in Fig. 4. In the first, a convolution operation is applied to the input image using k filters (masks) ($k = 1$ in the figure) for each channels R, G and B. These filters act as feature detector from the original input image. Then, a non-linearity function ψ is then applied to the result of the convolutional operation to obtain the so-called *activation map* (also called *feature map*). Each layer is followed by a pooling layer

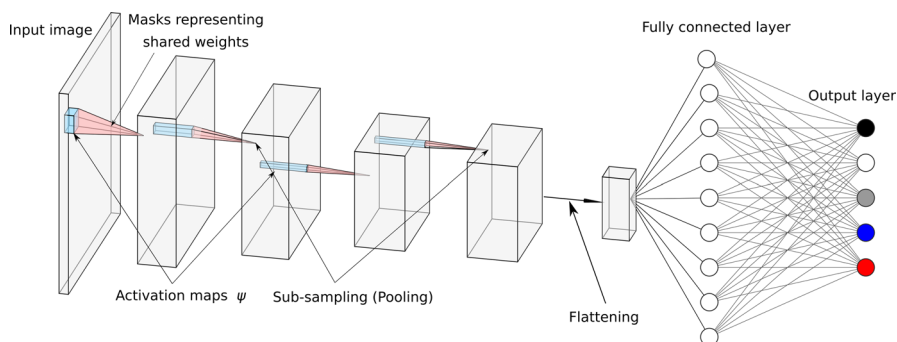


Fig. 4 Typical CNN architecture with a convolutional layer, a maxpooling layer, a flattening layer and a fully connected layer with a fixed number of neurons

to reduce the size of the activation map and to give invariance for small local translations. Finally, a fully connected layer with a softmax activation function is used in the output layer to perform the classification task.

In the literature, many advanced convolutional neural networks have been developed over last years such as AlexNet [35], Inception modules [36], VGG networks [37], and region-based convolutional neural networks [38].

Recently many researchers employ convolutional Neural Networks to design handwriting recognition systems for several languages such as Chinese [39], Arabic [40], Bangla [41], Devanagari [42], Indic [43], Tamil [44]. The obtained results are good and promote the usage of CNNs for handwriting recognition systems for other languages.

4 The Proposed System

The used CNNs architecture is given in Fig. 5. It is composed of five adjacent layers. All training images are labeled and resized with 32×32 pixels. The first three layers are responsible for the feature extraction and the two last ones perform features classification. The first layer (L1) is a convolution layer with 32 activation maps of 32×32 pixels each. Each neuron unit is associated to a convolution by a 3×3 mask with the addition of a trainable bias. In this layer, different activation

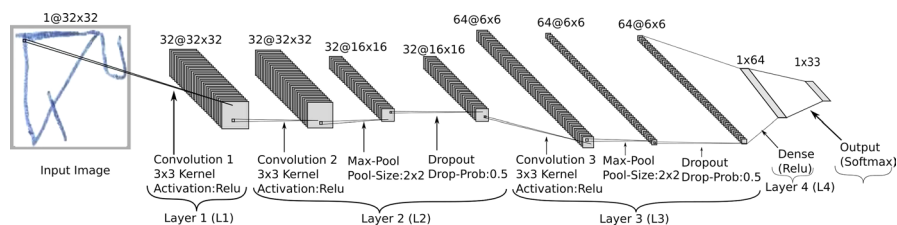


Fig. 5 The proposed system architecture

maps correspond to different trainable masks and biases. Each map has 9 trainable weights plus a trainable bias which leads to 320 (32×10) trainable parameters for this layer (L1). The used activation function is a rectified linear unit (Relu) as shown in Eq. 1.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (1)$$

Layer (L2) is also a convolution layer. It contains 32 activation maps. The implementation of this layer differs from the first one (L1) by adding a maxpolling and dropout layers. The pooling size is 2 by 2 in the x and y directions. We added dropout layer with probability 0.5 for regularization. In total, we have 9248 trainable parameters for this layer.

Next, we add another layer (L3) composed as for (L2) of three layers: a convolutional layer, a maxpooling layer and a dropout layer with 64 output channels. The size of each feature map is 6×6 pixels. In total, we have 18,496 trainable parameters for the (L3) layer. This duplication and combination of the feature extractors lead to the exploration of other high-level features.

The last two layers perform features classification. The first layer is composed by 64 Relu neurons. Each neuron in this layer is fully connected to only one feature map of (L3). The second layer contains 33 neurons indicating the class of the input image. In this layer, we use softmax activation function. We train the system using RMSProp optimizer with an adaptive learning rate. The whole architecture has 177,729 trainable parameters. In Table 1 are given all the parameters of the proposed architecture.

Table 1 The proposed architecture parameters summary

Layer (type)	# kernels	Kernel/pool size	Output shape	#Param
Image (InputLayer)	–	–	(1, 3, 32, 32)	0
Conv1 (Conv2D)	32	3×3	(32, 32, 32)	320
Conv2 (Conv2D)	32	3×3	(32, 32, 32)	9248
Pool1 (MaxPooling2D)	–	2×2	(32, 16, 16)	0
Drop1 (Dropout)	–	–	(32, 16, 16)	0
Conv3 (Conv2D)	64	3×3	(64, 6, 6)	18,496
Pool2 (MaxPooling2D)	–	2×2	(64, 6, 6)	0
Drop2 (Dropout)	–	–	(64, 6, 6)	0
Dense64 (Dense)	–	–	(64)	147,520
Dense33 (Dense)	–	–	(33)	2145
Output (Softmax)	–	–	(33)	0

from 50 to 20% images in the testing set and register the obtained accuracy using these sets. Table 2 illustrates the accuracy of various training set sizes. As we can see, the accuracy reaches the best value 99.1% when 80% of the AMHCD database images are used for training. According to the experimental results in Table 2 the training set size is 80% of images and 20% is for validation set size.

5.2.2 Feature Visualization

In this part of experiments, we show the output of each hidden layer in the proposed CNN system after the training step. Our goal is to see how different filters in different layers are trying to highlight different parts of the image character. The Fig. 7 shows the feature maps of the three convolutional layers using an input image of ‘*Yakw*’ character. As we can see some filters in the first and second layers are acting as edge detectors, others are detecting a particular region of the character. We can note also that the patterns captured by the convolution filters in the third layer are not clear because the feature maps become more sparse and localized.

5.2.3 Accuracy Using the Proposed CNN

To show performance of the proposed system in classifying Tifinagh characters, we give a summary of classification rate of each Tifinagh character in the Fig. 8 and we calculate the confusion matrix as shown in Fig. 9. As we can see, the number of misclassified characters is very small compared to the well-predicted ones (all values are concentrated in the diagonal). Mistakes made by the proposed system are not surprising, they can be explained by the fact that some characters are not well written in the AMHCD database and by the similarities between some Tifinagh characters, such as ‘*Yagw*’ and ‘*Yag*’, ‘*Yak*’ and ‘*Yahh*’, ‘*Yall*’ and ‘*Yan*’, ‘*Yarr*’ and ‘*Yar*’, ‘*Yas*’ and ‘*Yar*’, ‘*Yaz*’ and ‘*Yazz*’ and between ‘*Yi*’ and ‘*Ya*’ as observed in the confusion matrix. According to this confusion matrix the number of misclassified characters doesn’t exceed five characters in the worst case.

5.2.4 Comparison of the Proposed CNN with Other Existing Methods

In first, we compare the proposed system with the well known LeNet-5 network [31]. Our objective by this experiment is to demonstrate that the architecture of our system is more adapted than the LeNet-5 network to the tifinagh characters recognition task. For that, the data set is divided into two sets: training and validation. At

Table 2 Accuracy vs the number of training samples for the AMHCD database

Percentage of training images (%)	Accuracy (%)
50	94
60	98
70	99
80	99.1

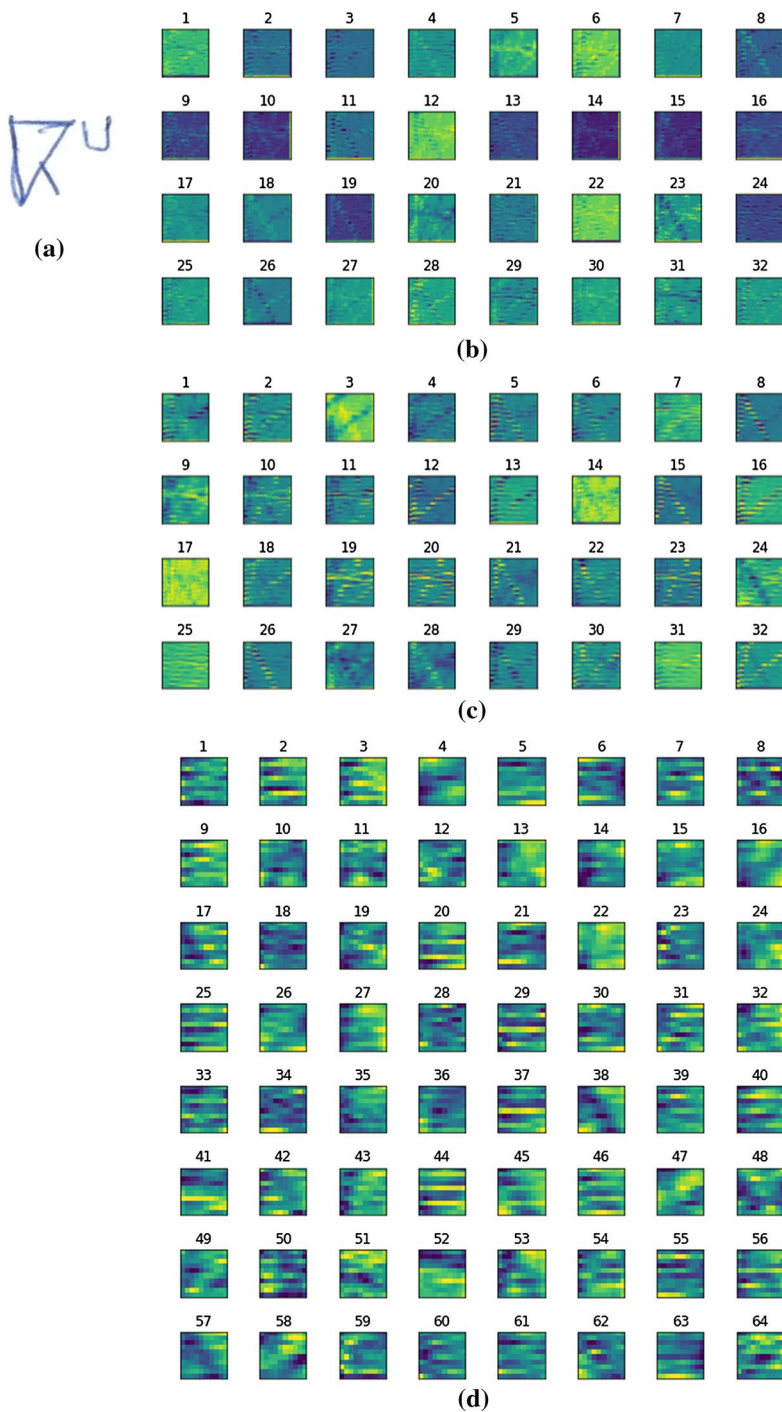


Fig. 7 Illustration of feature visualization using the ‘Yakw’ character for the proposed system: **a** Input image. **b–d** Output feature maps respective to first, second and third convolution layers in the proposed CNN system

Character	Accuracy	Number of classified characters	Character	Accuracy	Number of classified characters
◦	100%	158	l	100%	155
⊖	99.38%	161	ℤ	100%	172
℄	99.34%	153	○	95.15%	159
∧	99.34%	151	Q	99.39%	169
E	100%	144	⊙	98.20%	169
h	98.69%	151	⊘	98.11%	161
ℑ	100%	170	†	100%	156
⌘	98.21%	166	E	100%	143
Ƴ	99.34%	151	⊐	100%	162
ℳ	100%	166	ℵ	100%	155
⊕	98.73%	157	⋈	98.08%	154
∧	96.43%	136	⋈	98.53%	135
I	97.50%	157	⋈	99.33%	151
ℜ	100%	139	⋈	99.31%	143
ℜ	99.44%	178	⋈	98.00%	150
h	100%	153	⋈	100%	157
⊐	100%	166	avg/total	99.10%	5148

Fig. 8 Classification rate of each character in the AMHCD database during the validation phase

each iteration, the proposed system and LeNet-5 are trained with an equal number of images from the training set. By the use of the validation set, we obtain a more realistic estimation of how the networks would perform with unseen data and check the presence of overfitting.

We notice that we kept the original LeNet-5 architecture as in [31] except the activation function to overcome the vanishing gradient problem [45]. So, we used the Relu activation function instead of Tanh and Sigmoid activation functions.

Figures 10 and 11 show the accuracy and loss curves respectively according to the number of epochs obtained by our proposed model and the LeNet-5 architecture with Relu activation function. The Fig. 10, demonstrates that our network is behaving quite well. After only 6 epochs our network reaches 95%. As we can see in Fig. 11, both the training and validation data continue to fall with small minor spikes and no signs of overfitting. Unlike the LeNet-5 architecture that began to learn from data at the 15 first epochs but after that the model overfit as seen in the Figs. 10 and 11. Based on the loss

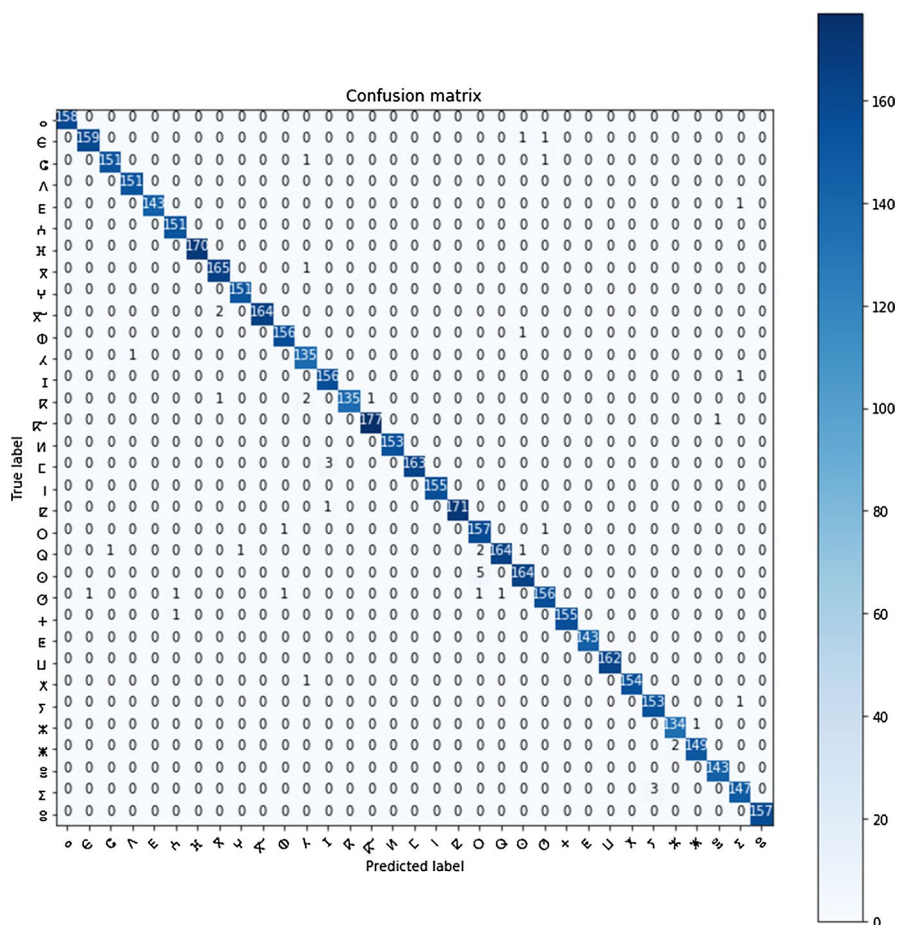


Fig. 9 Confusion matrix for the AMHCD database

and accuracy curves, our proposed CNN has achieved good validation accuracy with high consistency and it has outperformed the LeNet-5 with Relu activation function.

Finally, the proposed system is compared with various existing techniques. Table 3 gives performances obtained by the proposed method and by some existing systems using the AMHCD database. As we can see, the proposed approach gives the best performance without any preprocessing step (such as in [7, 22, 24, 26, 28, 46]), and gives the best accuracy even when we use all images in the AMHCD database, unlike all cited works in Table 3.

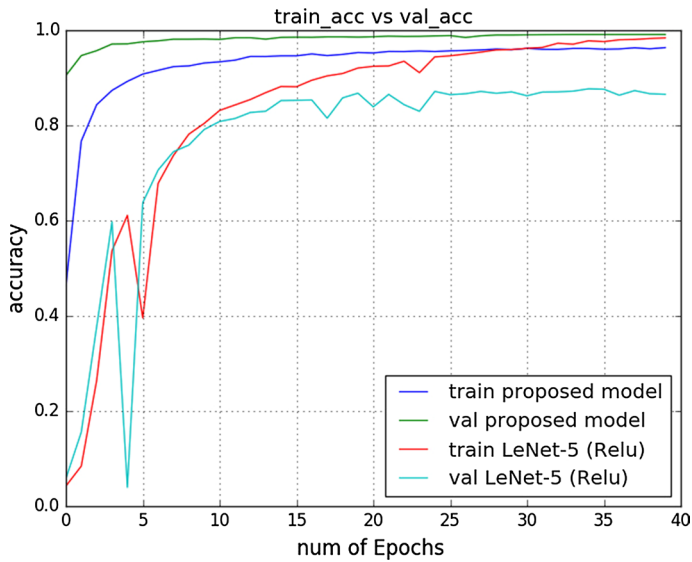


Fig. 10 A plot of the accuracy over the course of 40 epochs for the proposed architecture and the LeNet-5 trained on the AMHCD data set

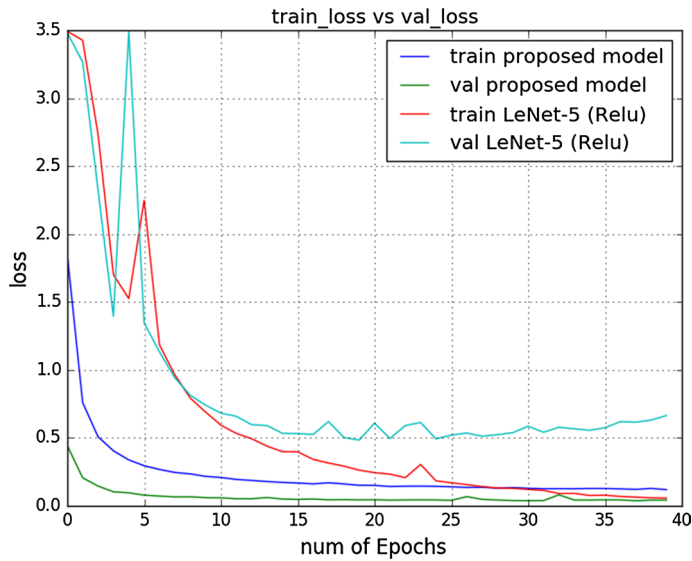


Fig. 11 A plot of the losses over the course of 40 epochs for the proposed architecture and the LeNet-5 trained on the AMHCD data set

Table 3 Comparison with other existing approaches

Methods	Total used images from the AMHCD (%)	Training set size	Test set size	Accuracy (%)
Geometrical method [46]	1700 (06.60%)	1000	700	92.30
Horizontal and vertical centerline of character [26]	20,150 (78.28%)	18,135	2015	96.32
Horizontal and vertical baseline of character [7]	24,180 (93.93%)	21,762	2418	94.96
Continuous HMMs and directional features [24]	20,180 (93.93%)	16,120	8060	97.89
Combination of multiple classifiers with statistical feature [22]	24,180 (93.93%)	16,926	7254	99.03
CNN vs BDN [28]	24,180 (93.93%)	–	–	98.25
Lenet-5 [31] with Relu	25,740 (100%)	20,592	5148	87.68
Proposed system	25,740 (100%)	20,592	5148	99.10

6 Conclusion

In this paper, we have presented a recognition system of the Amazigh handwritten characters, based on the deep convolutional neural networks. The proposed system operates directly with the original character's images where all published works require many preprocessing steps. The proposed system was evaluated using the entire characters in the AMHCD database and gives best performances compared with other existing systems in the literature including those who used the CNN networks. As a future work, we aim to extend our system for sentences recognition and multilingual handwriting recognition.

References

1. Ekkehard Wolff, H. (2018). *Berber languages*. <https://www.britannica.com/topic/amazigh-languages>. Accessed 20 Feb.
2. Ameer, M., Bouhjar, A., Boukhris, F., Boukouss, A., Iazzi, E., Boumalk, A., et al. (2004). Initiation à la langue amazighe. In *Publications de l'Institut royal de la culture Amazighe, manuels*, No. 1, 9.
3. Es-Saady, Y. (2012). *Contribution au développement d'approches de reconnaissance automatique de caractères imprimés et manuscrits, de textes et de documents amazighs*. PhD thesis, Ibn Zohr University, Agadir.
4. Ameer, M., Bouhjar, A., Boukhris, F., Boukouss, A., Boumalk, A., Elmedlaoui, M., et al. (2006). *Graphie et orthographe de lamazighe*. Rabat, Maroc: IRCAM.
5. Casajus, D. (2013). Sur l'origine de l'écriture libyque. quelques propositions. In *Afriques [En ligne], Débats et lectures, mis en ligne le 04 juin 2013, consulté le 28 février 2018*. URL: <http://journals.openedition.org/afriques/1203>. Accessed 20 Feb 2018.
6. <http://www.ircam.ma/>. Accessed 20 Feb 2018.
7. Es-Saady, Y., Amrouch, M., Rachidi, A., El Yassa, M., & Mammass, D. (2014). Handwritten tifi-nagh character recognition using baselines detection features. *International Journal of Scientific and Engineering Research*, 5(4), 1177–1182.
8. Zenkouar, L. (2004). *L'écriture amazighe tifi-naghe et unicode. Etudes et documents berbères. Paris (France)*. (n 22, pp. 175–192).
9. Byun, H., & Lee, S.-W. (2002). Applications of support vector machines for pattern recognition: A survey. In S. W. Lee., & A. Verri (Eds.), *Pattern recognition with support vector machines. SVM 2002. Lecture notes in computer science* (Vol. 2388). Berlin, Heidelberg: Springer.
10. El Abed, H., & Märgner, V. (2011). Icdar 2009-arabic handwriting recognition competition. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1), 3–13.
11. Koerich, A. L., Sabourin, R., & Suen, C. Y. (2003). Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis and Applications*, 6(2), 97–121.
12. Marti, U.-V., & Bunke, H. (2002). The iam-database: An english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1), 39–46.
13. Plötz, T., & Fink, G. A. (2009). Markov models for offline handwriting recognition: A survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(4), 269.
14. Tagougui, N., Kherallah, M., & Alimi, A. M. (2013). Online arabic handwriting recognition: A survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(3), 209–226.
15. Rachidi, A., Eddahibi, M., Essaad, Y., & Amrouch, M. (2014). Amazigh characters automatic recognition: Overview and prospects. *International Journal of Scientific and Engineering Research*, 5(11), 797–803.
16. Aharrane, N., El Moutaouakil, K., & Satori, K. (2015). A comparison of supervised classification methods for a statistical set of features: Application: Amazigh ocr. In *Intelligent systems and computer vision (ISCV)*, (pp. 1–8). IEEE.

17. Pearl, J. (1985). Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th conference of the cognitive science society*, (pp. 329–334).
18. Bhavsar, H., & Ganatra, A. (2012). A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(4), 2231–2307.
19. Haykin, S. (2009). *Neural networks and learning machines* (Vol. 3). Upper Saddle River, NJ: Pearson.
20. Aharrane, N., El Moutaouakil, K., & Satori, K. (2015). Recognition of handwritten amazigh characters based on zoning methods and mlp. *WSEAS Transactions on Computers*, 14, 178–185.
21. Es-Saady, Y., Rachidi, A., El Yassa, M., & Mammass, D. (2011). Amhcd: A database for amazigh handwritten character recognition research. *International Journal of Computer Applications*, 27(4), 44–48.
22. Aharrane, N., Dahmouni, A., El Moutaouakil, K., & Satori, K. (2017). A robust statistical set of features for amazigh handwritten characters. *Pattern Recognition and Image Analysis*, 27(1), 41–52.
23. Amrouch, M. (2012). *Reconnaissance de caractres imprimés et manuscrits, textes et documents basée sur les modles de Markov cachés*. PhD thesis, Faculté des sciences d'Agadir.
24. Amrouch, M., Es-Saady, Y., Rachidi, A., El Yassa, M., & Mammass, D. (2012). Handwritten amazigh character recognition system based on continuous hmms and directional features. *International Journal of Modern Engineering Research*, 2(2), 436–441.
25. Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.
26. Es-Saady, Y., Rachidi, A., El Yassa, M., & Mammass, D. (2011). Amazigh handwritten character recognition based on horizontal and vertical centerline of character. *International Journal of Advanced Science and Technology*, 33(17), 33–50.
27. Es-Saady, Y., Rachidi, A., El Yassa, M., & Mammass, D. (2011). Reconnaissance automatique de l'écriture amazighe à base de ligne centrale de l'écriture. In *4ème Atelier international sur l'amazighe et les TIC*.
28. Sadouk, L., Gadi, T., & Essoufi, E. H. (2017). Handwritten tfinagh character recognition using deep learning architectures. In *Proceedings of the 1st international conference on internet of things and machine learning, IML '17*, (pp. 59:1–59:11). New York, NY: ACM.
29. Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527–1554.
30. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <http://www.deeplearningbook.org>. Accessed 20 Feb 2018.
31. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
32. Zhong, Z., Jin, L., & Xie, Z. (2015). High performance offline handwritten chinese character recognition using googlenet and directional feature maps. In *Document analysis and recognition (ICDAR), 2015 13th international conference on*, (pp. 846–850). IEEE.
33. Claudiu Ciresan, D., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2011). Convolutional neural network committees for handwritten character classification. In *Document analysis and recognition (ICDAR), 2011 international conference on*, (pp. 1135–1139). IEEE.
34. Niu, X.-X., & Suen, C. Y. (2012). A novel hybrid cnn-svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4), 1318–1325.
35. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th international conference on neural information processing systems—Volume 1, NIPS'12*, (pp. 1097–1105). USA: Curran Associates Inc.
36. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–99). Boston, MA. <https://doi.org/10.1109/CVPR.2015.7298594>.
37. Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
38. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1), 142–158.

39. Yang, W., Jin, L., Xie, Z., & Feng, Z. (2015). Improved deep convolutional neural network for online handwritten chinese character recognition using domain-specific knowledge. In *2015 13th international conference on document analysis and recognition (ICDAR)*, (pp. 551–555). IEEE.
40. El-Sawy, A., & Benha, M. L. (2017). *Characters recognition using convolutional neural network*.
41. Purkaystha, B., Datta, T., & Islam, M. S. (2017). Bengali handwritten character recognition using deep convolutional neural network. In *Computer and information technology (ICCIT), 2017 20th international conference of*, (pp. 1–5). IEEE.
42. Jangid, M., & Srivastava, S. (2018). Handwritten devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods. *Journal of Imaging*, 4(2), 41.
43. Sarkhel, R., Das, N., Das, A., Kundu, M., & Nasipuri, M. (2017). A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. *Pattern Recognition*, 71, 78–93.
44. Vijayaraghavan, P., & Sra, M. (2014). Handwritten tamil recognition using a convolutional neural network. MIT Media Lab. https://web.media.mit.edu/~sra/tamil_cnn.pdf.
45. Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02), 107–116.
46. Djematen, A., Taconet, B., & Zahour, A. (1997). A geometrical method for printed and handwritten berber character recognition. In *Proceedings of the fourth international conference on document analysis and recognition, volume 2*, (pp. 564–567).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.