Computer Science • 24(2) 2023 https://doi.org/10.7494/csci.2023.24.2.4783

Kemiche Mokrane
Malika Sadou

DEEP CONVOLUTIONAL NEURAL NETWORK USING NEW DATA SET FOR BERBER LANGUAGE

Abstract *Currently, handwritten character-recognition (HCR) technology has become an interesting and immensely useful technology; it has been explored with impressive performance in many languages. However, few HCR systems have been proposed for the Amazigh (Berber) language. Furthermore, the validation of any Amazigh handwritten character-recognition system remains a major challenge due to the lack of availability of a robust Amazigh database. To address this problem, we first created two new data sets for Tifinagh and Amazigh Latin characters by extending the well-known EMNIST database with the Amazigh alphabet. Then, we proposed a handwritten character-recognition system that is based on a deep convolutional neural network to validate the created data sets. The proposed convolutional neural network (CNN) has been trained and tested on our created data sets; the experimental tests showed that it achieved satisfactory results in terms of its accuracy and recognition efficiency.*

Keywords optical character recognition, handwritten character recognition, CNN, Berber-MNIST data set, EMNIST, Tifinagh, Latin characters

Citation Computer Science 24(2) 2023: 225–241

1. Introduction

Nowadays, there has been an increasing interest in optical character recognition (OCR). This emerging technology is considered to be a popular field of research that has been widely explored in various areas, including handwritten document analyses, advertising content recognition, digitizing document images, etc. OCR studies have also been explored in the automatic processing of many languages. Hence, different techniques of handwriting character recognition have been developed for several languages, such as Latin [10], Chinese [27], Japanese [11], Hindi [25], Arabic, etc. However, few reliable OCR systems are available for the Amazigh language [9]; since it remains a young alphabet, recognition of its handwritten characters is a young area of research. The Amazigh language (also referred to as Berber) is a language that is spoken by Amazigh people all over an area that stretches from the Siwa Oasis in Western Egypt westward to the Canary Islands through Libya, Tunisia, Algeria, and Morocco as well as from the northern coast of the Mediterranean Sea extending southward to Mauritania, Mali, and Niger [2]. The Amazigh language is a very old languages. Originally, it was written with symbols called Tifinagh, but there have since been many attempts to adapt Tifinagh characters into modern usage by introducing new symbols such as Latin and Arabic alphabets [22]. Furthermore, the Tifinagh that were adopted by IRCAM (the Royal Institute of the Amazigh Culture) was officially recognized by the International Organization for Standardization (ISO) [19] as belonging to the basic multilingual planned. In this paper, we are interested in Berber handwritten character recognition by setting up two goals: creating a Berber data set, and validating it with an OCR system. To do this, we needed to first create two new data sets for both Tifinagh and Amazigh Latin character scripts. For such needs, we have extended the standardized EMNIST data set [12] with new Latin characters that belong to the Amazigh language. Accordingly, we created new benchmark Berber-MNIST Amazigh scripts that contain two subsets (Tifinagh, and Amazigh Latin). The first subset was obtained essentially by converting the AMHCD data set [17] to a new one that was much lighter and faster. In the second sub-subset (Latin version), characters that are specific to the Amazigh language were incorporated. The Berber-MNIST data set was created with a new automatized paradigm that is compatible with the one that is used in MNIST (which is described later in this paper). On the other hand, we applied the MNIST paradigm to make our data set compatible with all of the machine-learning algorithms that have already been designed for MNIST data sets. Furthermore, the improvement that was brought about with an image-centering algorithm provides more precision to the classifiers. Our data set has been released at the address in the footnote[1]. In order to validate the created data sets, we established a deep convolutional neural network. The experimental tests showed that it achieved satisfactory results in terms of its accuracy and recognition efficiency.

[1] https://www.kaggle.com/muqran/the-berbermnist-dataset

The rest of this paper is organized in the following manner. Section 2 gives an overview of previous proposed Amazigh character-recognition systems. In Section 3, we first present a brief overview of some databases that are already used in character-recognition work in some languages, and then we discuss the process of creating our data set in detail. The architecture and the functioning of the proposed CNN are explained in Section 4. Section 5 gives the achieved experimental results. Finally, Section 6 summarizes and draws conclusions.

2. Related works

OCR technology has become an active area; it has been applied to several languages like Chinese, Arabic, and English; however, few works have been proposed in the literature for Amazigh character recognition. Saady *et al.* [17] developed a database for handwritten Amazigh characters called AMHCD. This database is used to test most of the research work in Amazigh character recognition. In this setting, the same authors proposed a system for Amazigh handwriting recognition based on the positions of the horizontal and vertical centerlines of the characters [17]. First, this approach estimates horizontal and vertical baseline parameters in order to derive a subset of baseline-dependent features, taking the symmetry of Amazigh characters into account. The extracted features are based on the density of the pixels using the sliding window technique. In the training step, the authors used a multilayer perceptron with one hidden layer step for feature extraction and character recognition. Based on hidden Markov models, Saady *et al.* [16] developed a global approach to the recognition of handwritten Amazigh characters. This method extracts a vector of features from an image of Amazigh characters, which is used in the learning phase. This approach is based mainly on the directional primitives of Amazigh characters during the extraction step and Markov models at the recognition phase. The same authors [7] combined hidden Markov models (HMM) and the Hough transform to propose an automatic system for offline printed Amazigh handwritten character recognition. The Hough transformation method was applied to build the representative chain of each character. The obtained information was translated into a sequence of observations that were used to feed the hidden Markov model. [15] designed an optical character recognition system for a multilingual text that was written in the French and Amazigh languages and transcribed in Latin. In this framework, many pre-treatments and several types of features were studied and compared. On the other hand, the authors developed a corpus that contained different characters that were used in the transcription of the Amazigh language. Another approach that was proposed by [3] was a statistical method that increased the performance of Tifinagh handwriting recognition. This approach was used to develop a new feature set to conceive an optical character-recognition system of Amazigh handwritten scripts. The obtained features were based on specific zoning to represent each Amazigh character. Then, the authors used a multilayer perceptron (MLP) as a classifier in the recognition phase. Abaynarh *et al.* [1] proposed a new hybrid Amazigh character-recognition system based

on an artificial neural network classifier. This contribution had two main steps: preprocessing, and recognition. In the first step, the authors developed a method that extracted optimal character features based on Legendre moments in order to discover similarities between the characters. In the recognition phase, the obtained vector of similarities was used as inputs, and a multilayer neural network with the stochastic backpropagation algorithm was used as a classifier. Sadouk *et al.* [23] proposed two deep-learning approaches for Tifinagh handwritten character recognition, which were convolutional neural networks (CNNs) and deep belief networks (DBNs). These two automatic methods used the AMHCD handwritten character database to test their efficiency. Amrouch*et al.* [5] suggested a new approach that exploited the morphology of Amazigh characters to extract a novel set of structural primitives from the character strokes. In the feature-extraction phase, this method used the character contour points that had the maximum deviations. Furthermore, the recognition phase used a discriminating path that was based on dynamic programming that operated at the global graph of the segments. Benaddy*et al.* [8] designed an offline recognition system based on deep convolutional neural networks. This system operated with the original character images, which makes it flexible – especially in the feature-extraction phase. As with most of the previously mentioned research works, this system also used the AMHCD data set to test its performance. In this framework, the authors of [14] proposed an OCR system that treated Amazigh writing that was transcribed in Latin with the aim to contribute to the preservation of its literary heritage by digitizing it.

3. Berber-MNIST (new data set for Berber characters)

In this paper, we have created a new data set of Berber characters that contains two sub-data sets. To do so, we based it on the widely used EMNIST and AMHCD data sets. Before presenting the process of creating our database, we first give an overview of some databases that have been proposed in the literature. We begin with the databases that we used to build Berber-MNIST, and then we give a brief overview of the most-used databases in automatic language processing.

3.1. Previous proposed data set

The automatic processing of any language (in general) and handwritten character-recognition field (in particular) require large databases of its handwritten characters. In this section, we overview some data sets that have been proposed in the literature to recognize the handwritten characters of some languages.

CASIA-OLHWDB1 is a database of online handwritten Chinese characters [24]; it includes 3866 Chinese scripts and 171 symbols. In this data set, the total samples were partitioned into three main categories. They were also divided into training and test sets. [4] proposed a database for Arabic handwritten characters called AHCR, which contained 28,000 images of handwritten Arabic characters. The purpose of the authors behind the creation of this database was to use it in Arabic character-recognition studies. HACDB (handwritten Arabic character data base) is another

data set for the alphabet of the language (Arabic) that was proposed by [21]. It included 6600 Arabic characters, including one overlapped with the other. Kavitha and Srimathi [20] proposed a recognizing handwritten system for Tamil characters in offline mode. To do so, the authors used an isolated handwritten Tamil character data set; this contained around 82,928 samples of Tamil characters divided into 156 classes. HCD [26] is a relatively new data set for recognizing English handwritten characters and digits. In this study, we are only concerned with the Amazighe language. In this setting, there is only one available database of handwritten Amazigh characters; it is AMHCD (Amazigh handwritten character database) [17]. It consisted of 2540 samples of Tifinagh handwritten scripts divided into 780 classes (a class corresponds to the image of a Tifinagh handwritten character).

3.2. Overview of NIST-based data sets

As mentioned above, the Berber-MNIST data set was derived from the EMNIST database, which is in turn was derived from the NIST and MNIST data sets. Therefore, we first begin by giving a short overview of the two original databases from which our database is derived before starting its presentation.

- NIST data set NIST (the National Institute of Standards and Technology) has developed a reference form-based handprint-recognition system to evaluate optical character recognition (OCR) systems. The NIST database [18] contains digits and upper- and lower-case letters that were collected through the contributions of more than 500 people.
- MNIST data set The Modified National Institute of Standards and Technology (MNIST) [13] provides resources that consist of a collection of handwritten digit images. The MNIST database was derived from a small subset of data from the digit class in the NIST data set. It contains 60,000 training set images and 10,000 testing set images. Its images are in black and white and have a 28×28 pixel standard size. Therefore, the dimensionality of each vector of an image is $28 \times 28 = 784$ pixels.
- EMNIST data set Extended MNIST (EMNIST) [12] is a variant of the full NIST data set that was presented earlier. It follows the same conversion paradigm that was used to create the MNIST data set but extended to lower- and upper-case letters. The resulting image is 28×28 pixels in grayscale.

3.3. Proposed Berber-MNIST data set

In this work, we first developed a new data set called Berber-MNIST, which is the first and most important step in a Berber recognition system (Section 5). This section describes the paradigm of the conception of this new data set in detail. The creation of the Berber-MNIST data set for the Tifinagh and Latin versions is based on the AMHCD data set [17] and the upper-case subcategory of the EMNIST data set, respectively. The Latin data set is extended with specific characters for the Berber language. The letters in the original AMHCD use 64×64 pixel images, which induces

a considerable weight for the pictures. For this reason, we opted to convert the AMHCD data set by using the MIST conversion paradigm in 28×28 with a black background. Indeed, the weight of the obtained pictures is six times lower than those in the original AMHCD. The conversation process makes the processing of classifiers less time-consuming. The design of the Berber-MNIST Latin version data set proceeds in two main steps (automating and simplifying the MNIST conversion paradigm); we then applied the image-centering algorithm 3. Figure 1 illustrates the creation process of our data set and the relationships between the existing ones and the newly created data set.

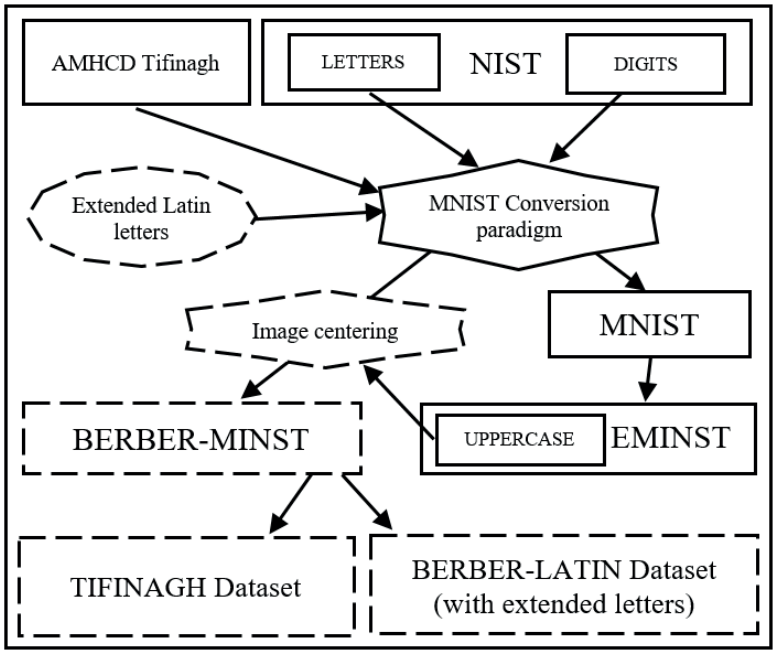

Figure 1. Dashed lines represent our contribution in process of creating Berber-MNIST

3.3.1. Amazigh data set (Latin version)

The whole upper-case subcategory of the EMNIST set was reused in our first data set that was dedicated to the Amazigh script for Latin characters. The difference is that, in the Latin character data set, there are 26 characters, whereas the Amazigh Latin character script contains 33 symbols (+1 additional letter T, which is ignored) – the O, P, and V characters are removed from the data set. Table 1 illustrates the differences between the EMNIST letters and the Amazigh Latin characters as well as the new characters that are specific to the Amazigh language.

The differences in most cases can be noted in dots that are added under a few letters (Ḍ, Ḥ, Ṛ, Ṣ, Ṭ, and Ẓ), inverted circumflexes (Č and Ǧ), an epsilon Ɛ, and

a gamma Γ (ⵖ in lower-case). Excluding Ɛ and Γ, which were obtained by horizontally flipping the images of "3" and "7," the images were taken from the EMNIST digit subset. The other letters that were inserted into our database have only one dot or an inverted circumflex to add to the letters that were already present in the EMNIST data set. We modified the concerning letters according to Algorithm 1. The result is shown in Figure 2.

Table 1
Amazigh Latin characters

Latin	New?	Latin	New?	Latin	New?
A		J		TT	Ignored
Ɛ	Yes	K		W	
B		L		X	
C		M		Y	
Č	Yes	N		Z	
D		Q		**Ẓ**	Yes
Ḍ	Yes	**Γ**	Yes		
E		R			
F		**Ṛ**	Yes	O	Deleted
G		S		P	Deleted
Ǧ	Yes	**Ṣ**	Yes	V	Deleted
H		T			
Ḥ	Yes	**Ṭ**	Yes		
I		U			

Algorithm 1 Extended letter creation

1: **for** each image **do**
2: Shift image up or down depending on whether you add point or inverted circumflex.
3: Let border of 2px.
4: Insert missing point or inverted circumflex in recovered area in image.
5: Position of missing point or inverted circumflex is randomly generated at place where human writer would put this.
6: **end for**

Figure 2. Some extended letters obtained with Algorithm 1

3.3.2. Amazigh data set – Tifinagh script version

In the Amazigh script with the Tifinagh characters, the number of characters is equal to 40 symbols according to Unicode Version 12; however, the IRCAM-Tifinagh utilizes only 33, which are as follows: ⴰ,ⴱ,ⴳ,ⴷ,ⴻ,ⴼ,ⴽ,ⵀ,ⵃ,ⵄ,ⵅ,ⵇ,ⵉ,ⵊ,ⵍ,ⵎ,ⵏ,ⵓ,ⵔ,ⵕ,ⵖ,ⵙ,ⵚ,ⵛ,ⵜ,ⵟ,ⵡ,ⵢ,ⵣ,ⵥ,ⵯ. Like the extended Latin characters, all of these symbols were created by following a new conversion paradigm that was similar to the one that was used to design the MNIST data set. Algorithm 2 describes the adopted approach for creating the Tifinagh data set. The result is shown in Figure 3.

Algorithm 2 Tifinagh conversion (MNIST paradigm)

1: **for** each image **do**
2: Extract ROI (region of interest).
3: Boost pixel brightness.
4: Resize images to 24 x 24 pixels with cubic interpolation.
5: Add 2px border.
6: **end for**

Figure 3. AMHCD Tifinagh conversion

After applying the MNIST paradigm on all of the images of the two versions of the data set (Tifinagh and Latin), we improved the whole set of images by applying the following image-centering Algorithm 3.

Algorithm 3 Image centering

1: **for** each image **do**
2: Remove empty space.
3: Stretch image to border.
4: **end for**

The result of this image-centering algorithm is shown in Figure 4, and its advantage is discussed in Section 5.2.

In this work, we opted to use an efficient technique for image data sets, which was a convolutional neural network to test the accuracy of our data sets. Thus, we present the architecture and the functioning of the proposed CNN in the following section as well as the conducted experiments and obtained results.

Figure 4. Image centering

4. Proposed deep convolutional network

Our data set was composed of images of upper-case letters; therefore, we used one of the best classifiers that are suitable for image data (CNN) to analyze its contents.

4.1. Convolutional neural network

A convolutional neural network can contain two steps (even though the second one is a layer of the same CNN): the first one consists of filtering each input image to extract specific features (for example, S and Ṣ – the dot is the feature that will make the difference between S and Ṣ). The filter is responsible for extracting all of the patterns and features that differentiate any letter from others in the same class (such as S and Ṣ). The data set of images is analyzed by multi-convolution layers with different filter sizes that are defined empirically. Maxpooling is added between each convolution layer to downsize the image further while keeping its features. This is done by choosing the Max value of the window that will move in the image that was obtained from the convolution. Then, we obtain a flatten vector by converting the different images from the last step into a vector. This vector is injected into a fully connected layer as a second step in which the number of output neurons is equal to the number of classes to predict. In order to adjust the weights that are associated with the neuron, a specified number of epochs are required for better prediction. In what follows, we present the overall architecture of the proposed CNN and how the data is processed to be injected into the network.

4.2. Overall architecture of proposed CNN

A global overview of a CCN architecture is given in Figure 5. It is composed of successive convolutional layers in order to extract the characteristics of the images to be provided to the fully connected layer to do the classification.

The proposed CNN model is given in Figure 6. The first two convolution layers that are composed of 32 filters of 3×3 pixels are connected to the maxpooling layer. The resulting layer is connected to the third convolution layer of 64 filters of 3×3 pixels, itself connected to the fourth convolution layer of 128 filters of 3×3 pixels. To improve the efficiency of the treatment, we applied a ReLu activation function between the layers. The images that are generated by the last step are flattened to obtain the flatten vector. This vector is the input of three fully connected layers that are composed of 512, 128, 33 outputs, respectively. The output

layer corresponds to the number of classes to be predicted (33 for both Latin and Tifinagh), with Softmax being used as an activation function. The epoch number is set to 15. To optimize the network weight values, we chose the Adam algorithm.

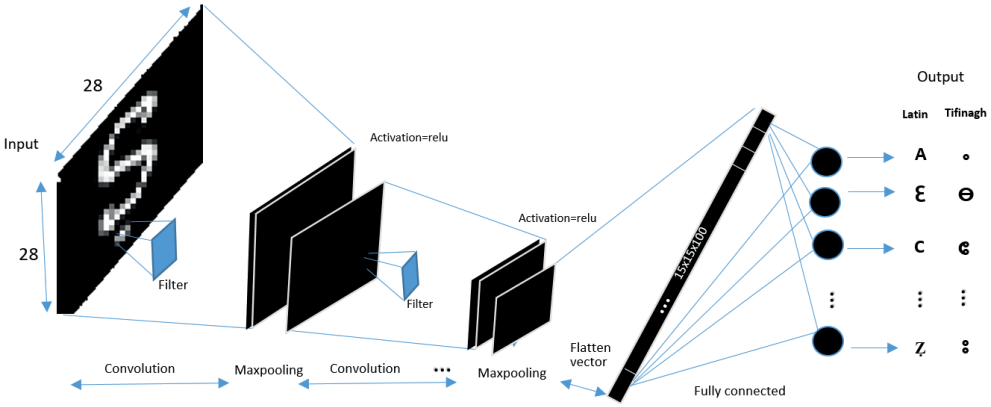

Figure 5. Global overview of CNN architecture

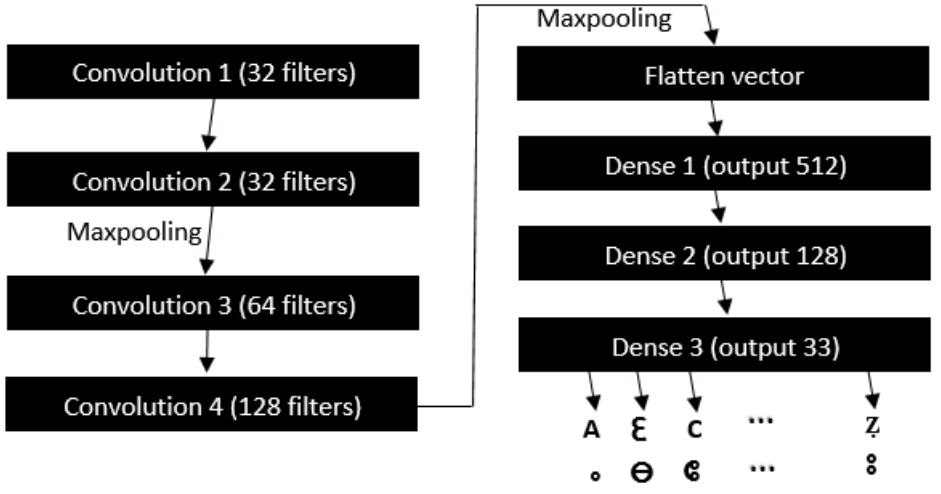

Figure 6. Architecture of proposed CNN

5. Tests and results

To test and evaluate the effectiveness of the proposed CNN, we trained it by using the created Berber-EMNIST data set.

5.1. Data preprocessing

We first split the data set into two subsets: Train (75%), and Test (25%). The Train set was used to train the network, while the Test set was applied to test the accuracy of our classifier in the validation step. Table 2 below illustrates the number of images in each data set.

Table 2
Contents of Latin and Tifinagh data sets

	Train set	Test set
Extended Latin version	185,988	61,996
Tifinagh version	19,305	6435

In order to simplify the computations, the data (image vectors) was preprocessed and normalized with standard scalers and min-max scalers.

5.2. EMNIST upper-case subset testing and evaluating

Before performing the test on Berber-MNIST, we will first see the contribution of Algorithm 3 on the EMNIST upper-case subset [12]. To do this, we tested and compared the original EMNIST by using the previous CNN with an improved version that was obtained by centering the images using the previous Algorithm 3. Table 3 shows the superiority of the improved version.

Table 3
Efficiency of Algorithm 3

	EMNIST upper-case subset [%]	improved EMNIST upper-case subset [%]
CNN error	0.93	**0.86**
Accuracy	99.07	**99.14**
Loss	9.74	**3.5**

5.3. Berber-MNIST evaluation and result discussion

The loss function (which we have to minimize) is the Categorical Crossentropy Loss Function. It is generally used for multi-class classification problems. In the case of our system, there were 33 classes. The loss function was calculated using the following formula:

$$Loss = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} log(\hat{Y_k^l})(1 - Y_k^i)log(1 - \hat{Y_k^l}) \tag{1}$$

where k demonstrates the class, i – the sample number, $\hat{Y_c}$ – the predicted value, Y_c – the ground truth value, m – the sample number in the batch, and K – the total number of classes.

The accuracy of the classification was obtained with the following formula:

$$accuracy = \frac{Correct\_predictions}{Total\_predictions} \times 100 \tag{2}$$

Using the formulas above, the proposed network provides the following results for both the extended Latin version and Tifinagh version data sets with the same parameters.

Table 4
CNN results

	Extended Latin version [%]	Tifinagh version [%]
Train_Loss	3.12	3.13
Val_loss	2.93	2.50
Train_Accuracy	99.08	99.07
Val_accuracy	99.28	99.33
CCN_Error	0.72	0.67

Table 4 summarizes the performance of the proposed CNN for both the Latin and Tifinagh versions in terms of the accuracy and loss functions in 15 epochs. In the accuracy rows, there was a tiny difference in the training and validation sets, which can be clearly seen in the following four curves for the Latin and Tifinagh versions. This signified that the proposed model provided better results.

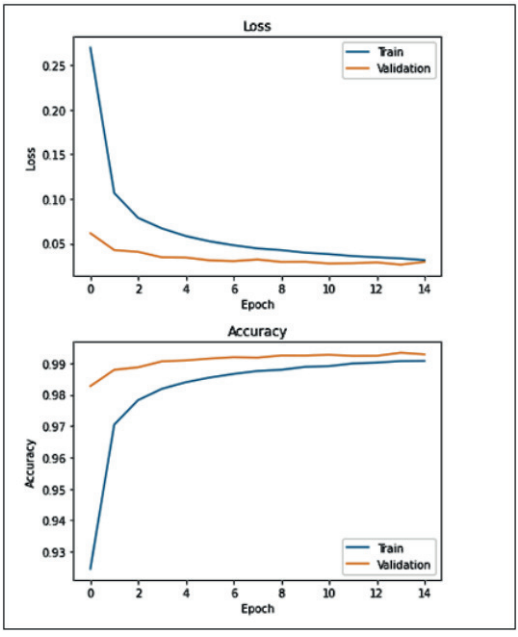

Figure 7. Loss and accuracy for extended Latin version

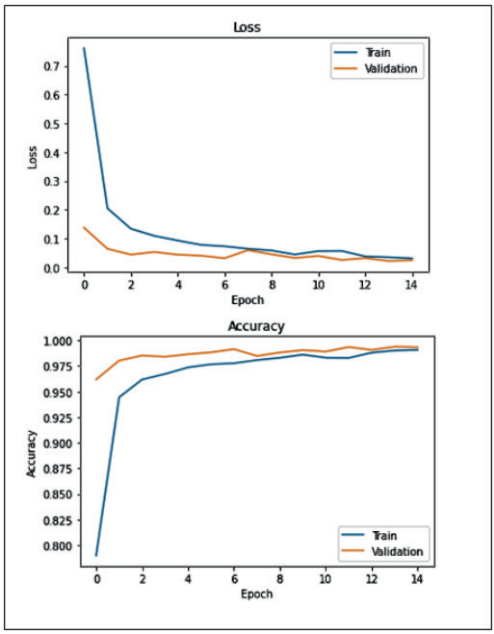

Figure 8. Loss and accuracy for Tifinagh version

These curves (Figures 7 and 8) represent the evolution of the accuracy and loss over 15 epochs for both the Latin and Tifinagh versions. To analyze the false predictions for each character in more detail, we used the following notations: True positive T_p, True negative T_n, False positive F_p, and False negative F_n. The correct predictions are represented by T_p and T_n, and the false predictions are represented by F_p and F_n.

To calculate the success score, we used the following formulas:

$$Precision = \frac{T_p}{T_p + F_p} \tag{3}$$

$$Recall = \frac{T_p}{T_p + T_n} \tag{4}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5}$$

Score F_1 represents a performance measure of the success of prediction when taking precision and recall into account (also known as sensitivity). Since our problem has 33 classes in both versions, we calculated the F_1 score for each class (character). Figure 9 gives the detailed results. Each character is represented by a number (such that 0, 1, 2... correspond to A,ⴹ, B... for the Latin version and ⴰ, ⴱ, ⴲ,..., for the

Tifinagh version). All of the values of these three metrics were close to 1 (as depicted in the figure below), which means that the character was well-predicted.

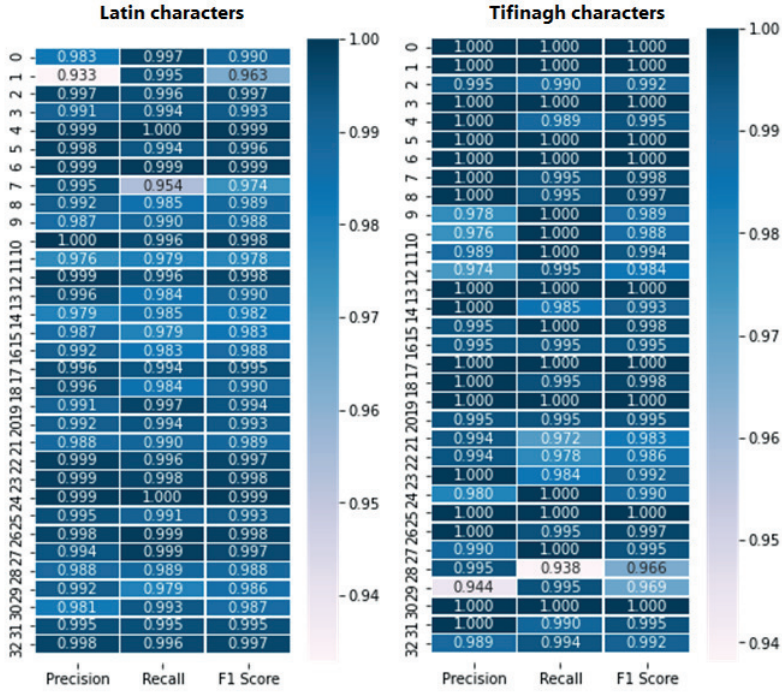

Figure 9. Detail result for each character

All of the functions that were mentioned above are already implemented in Tensorflow, which we used for the realization of this model.

6. Conclusion

Amazigh handwritten character recognition remains a recent young field of research. In this context, we have developed two versions of data sets for both Latin and Tifinagh character scripts. To do so, we extended the upper-case subdirectory of EMNIST with characters that only exist in Latin Amazigh-script (such as Č and Ṭ). Thereafter, we converted the Tifinagh AMHCD data set to a new one based on the MNIST paradigm (which is much lighter and faster). To test the efficiency of these data sets, we proposed a convolutional neural network. Note that the original author of AMHCD used a CNN to evaluate his data set; however, our series of experiments showed that our proposed system provided good performance on the Latin and Tifinagh data sets. So, the obtained results for the Tifinagh version were better than those that were presented in [8], and the required time to train the proposed CNN

was reduced with this new data set. This work opens us interesting prospects in the field of handwritten character recognition. Thus, we intend to develop a complete OCR system in future work that recognizes Berber words and texts with a high recognition rate.

References

[1] Abaynarh M., Elfadili H., Zenkouar K., Zenkouar L.: Neural Network Classifiers for Off-line Optical Handwritten Amazighe Character Recognition, *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 12(6), pp. 28–36, 2012.

[2] Achab K.: *The Tamazight (Berber) Language Profile*, University of Ottawa, 2001.

[3] Aharrane N., El Moutaouakil K., Satori K.: Recognition of handwritten Amazigh characters based on zoning methods and MLP, *WSEAS Transactions on Computers*, vol. 14(19), pp. 178–185, 2015.

[4] AlKhateeb J.H.: A database for Arabic handwritten character recognition, *Procedia Computer Science*, vol. 65, pp. 556–561, 2015.

[5] Amrouch M., Es-Saady Y., Rachidi A., El-Yassa M., Mammass D.: A Novel Feature Set for Recognition of Printed Amazigh Text using Maximum Deviation and HMM, *International Journal of Computer Applications*, vol. 44(12), pp. 23–30 2012.

[6] Amrouch M., Es-Saady Y., Rachidi A., El Yassa M., Mammass D.: Printed amazigh character recognition by a hybrid approach based on Hidden Markov Models and the Hough transform. In: *2009 International Conference on Multimedia Computing and Systems*, pp. 356–360, IEEE, 2009.

[7] Amrouch M., Rachidi A., El Yassa M., Mammass D.: Handwritten amazigh character recognition based on hidden Markov models, *ICGST-GVIP Journal*, vol. 10(5), pp. 11–18, 2010.

[8] Benaddy M., El Meslouhi O., Es-Saady Y., Kardouchi M.: Handwritten Tifinagh Characters Recognition Using Deep Convolutional Neural Networks, *Sensing and Imaging*, vol. 20, pp. 1–17, 2019.

[9] Boufenar C., Kerboua A., Batouche M.: Investigation on deep learning for off-line handwritten Arabic character recognition, *Cognitive Systems Research*, vol. 50, pp. 180–195, 2018.

[10] Chaudhuri A., Mandaviya K., Badelia P., Ghosh S.K.: Optical Character Recognition Systems for Latin Language. In: *Optical Character Recognition Systems for Different Languages with Soft Computing*, pp. 165–191, Springer, 2017.

[11] Clanuwat T., Lamb A., Kitamoto A.: KuroNet: Pre-modern Japanese kuzushiji character recognition with deep learning. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 607–614, IEEE, 2019.

[12] Cohen G., Afshar S., Tapson J., Van Schaik A.: EMNIST: Extending MNIST to handwritten letters. In: *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, IEEE, 2017.

[13] Deng L.: The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web], *IEEE Signal Processing Magazine*, vol. 29(6), pp. 141–142, 2012.

[14] El Gajoui K., Allah F.A., Oumsis M.: Diacritical Language OCR based on neural network: Case of Amazigh language, *Procedia Computer Science*, vol. 73, pp. 298–305, 2015.

[15] El Gajoui K., Ataa Allah F.: Optical character recognition for multilingual documents: Amazigh-French. In: *2014 Second World Conference on Complex Systems (WCCS)*, pp. 84–89, 2014.

[16] Es-Saady Y., Rachidi A., El Yassa M., Mammass D.: Amazigh handwritten character recognition based on horizontal and vertical centerline of character, *International Journal of Advanced Science and Technology*, vol. 33(17), pp. 33–50, 2011.

[17] Es-Saady Y., Rachidi A., El Yassa M., Mammass D.: AMHCD: A database for amazigh handwritten character recognition research, *International Journal of Computer Applications*, vol. 27(4), pp. 44–48, 2011.

[18] Grother P., Hanaoka K.: NIST special database 19. Handprinted forms and characters, 2nd Edition, National Institute of Standards and Technology, Technical Report, vol. 13, 2016.

[19] ISO/IEC JTC N2739R. De Normalisation, Organisation Internationale, 2004.

[20] Kavitha B., Srimathi C.: Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks, *Journal of King Saud University – Computer and Information Sciences*, vol. 34, pp. 1183–1190, 2019.

[21] Lawgali A., Angelova M., Bouridane A.: HACDB: Handwritten Arabic characters database for automatic character recognition. In: *European Workshop on Visual Information Processing (EUVIP)*, pp. 255–259, 2013.

[22] Osborn D.: *African languages in a digital age: Challenges and opportunities for indigenous language computing*, IDRC, 2010.

[23] Sadouk L., Gadi T., Essoufi E.H.: Handwritten tifinagh character recognition using deep learning architectures. In: *IML'17: Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, pp. 1–11, 2017.

[24] Wang D.H., Liu C.L., Yu J.L., Zhou X.D.: CASIA-OLHWDB1: A Database of Online Handwritten Chinese Characters. In: *2009 10th International Conference on Document Analysis and Recognition*, pp. 1206–1210, 2009.

[25] Yadav M., Purwar R.: Hindi handwritten character recognition using multiple classifiers. In: *2017 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence*, pp. 149–154, 2017.

[26] Yousaf A., Khan M.J., Imran M., Khurshid K.: Benchmark dataset for offline handwritten character recognition. In: *2017 13th International Conference on Emerging Technologies (ICET)*, pp. 1–5, 2017.

[27] Zhang X.Y., Bengio Y., Liu C.L.: Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark, *Pattern Recognition*, vol. 61, pp. 348–360, 2017.

Affiliations

Kemiche Mokrane
Research Center for Amazigh Language and Culture, Algeria, k.muqran@gmail.com

Malika Sadou
Research Center for Amazigh Language and Culture, Algeria, malika.sadou142@gmail.com

Received: 04.04.2022
Revised: 07.11.2022
Accepted: 08.11.2022