Recognize Arabic Handwritten using CNN Model

Abdul Muhsin M. **Bashra Kadhim Oleiwi** **Farah F. Alkhalid**

University of Technology, Iraq Control and Systems Engineering Department
60100@uotechnology.edu.iq 60010@uotechnology.edu.iq
10352@uotechnology.edu.iq

ARTICLE INFO
Submission date: 8 / 12 / 2019
Acceptance date: 30/12/2019
Publication date: 31/ 12/ 2019

Abstract

One of the most challenges that face machine learning is handwritten recognition, especially Arabic scripts, because many styles found for Arabic font. In this paper, an investigation model is proposed to make recognition for Arabic handwritten scripts utilizing Convolutional Neural Network (CNN), with multi layers of Normalization and Regularization to reduce training time and increase overall accuracy, with validation accuracy 98% for Kaggle dataset for Arabic handwritten characters and digits using Python.

Keywords: CNN, Deep Learning, Arabic Handwritten, Python.

Introduction

The Arabic content is the composition framework utilized for composing Arabic and a few different dialects of Africa and Asia, for example, Kurdish, Persian, Sindhi, Azerbaijani, Pashto, Urdu, Lurish, Mandink and many other until the sixteenth century, also Arabic was depending to write some texts. in Spanish. Additionally, before, Arabic characters were the official writing system in Turkish. It is the second most broadly used writing system over the world by many countries using it and the third by the count number of speaking user, after Latin language and Chinese language.[1,2]

The handwritten automated recognition of text on scanned images has enabled many applications such as finding for words in large volumes of documents, convenient editing of previously printed documents and automatic sorting of postal mail [3].
Numerous Algorithms intended to perceive manually written letters are less fruitful than printed letters, primarily because of the decent variety in transcribed style and structures of character. Arabic character acknowledgment is a significant issue, since it is a stage that might be required in all testing Arabic word or sentence acknowledgment issue [4]. Character segmentation means to isolate the word into many characters is another difficult issue. The character acknowledgment issue is identified with the easier issue of Arabic numeral acknowledgment which has as of late achieved extraordinary outcomes [5].
Many previous studies intended to present different approaches on Arabic handwritten characters recognition using Deep learning. Initially, in [6] the author presented a novel method based on deep neural networks. The Convolutional neural network model is

© Journal of University of Babylon for Pure and Applied Sciences (JUBPAS) by University of Babylon is licensed under a Creative Commons Attribution 4. 0 International License, 2019.

evaluated for its performance using the database that contains 132,500 datasets of handwritten Amharic characters. Common handwritten recognition systems using machine learning use a combination of both feature extractors and classifiers. Currently, the use of deep learning techniques shows promising improvements for machine learning based classification tasks.

Mr. Khaled in [7] utilized convolutional neural system (CNN) models with regularization parameters to anticipate overfitting and connected the Deep CNN system with the AIA9k dataset and the (AHCD) dataset. The characterization accuracy according to the couple previous mentioned datasets were 94.8 % and 97.6 %, separately. While in 2017 the authors of [8] proposed Alphanumeric Visual Geometry Group (VGG) net for Arabic manually written alphanumeric character acknowledgment. Alphanumeric VGG net is developed by approximately thirteen convolutional complex layers, two layers of max-pooling, and three layers which are completely associated each other. have accomplished promising outcomes, with an approval precision of 99.66. % for the ADBase database and 97.32. % for the HACDB database.

In [9] proposed paper and transcribed an Arabic digit acknowledgment approach that works in two stages. To start with, they utilized the RBM, which is a profound learning strategy that can remove exceptionally helpful highlights from crude information, and which has been used in a few grouping issues as an element extraction system in the element extraction stage. At that point, the removed highlights are sustained to an effective CNN engineering together with a profound directed learning design for the preparation and testing process. In the investigation, we utilized the CMATERDB 3.3.1 Arabic transcribed character dataset for preparing and testing the proposed technique. Test results demonstrate that the suggested technique altogether improves the exactness rate, with precision achieving 98.59 %.

Different strategies proposed and great acknowledgment rates are accounted for handwritten recognition. In this research I shall focus on the recognition part of handwritten Arabic digits and letters recognition that face a few difficulties, incorporating the boundless variety in human penmanship and the enormous open databases.

In this paper CNN is used which performs better with images classification problems as they can successfully boil down a given image into a highly abstracted representation through the layer filters which make the prediction process more accurate.

Model

The proposed system uses subsampling or as known “Pooling... layer” for reducing the dimensionality of every filtered image, at the same time keeps the most interested features in the previous layer. The output image will have the same number of images with fewer pixels for representation each one. This is likewise extremely helpful in overseeing the training load. However, Pooling is discussed that max -pooling is able to be excessive and let convolutional layer to be replaced with increased stride without loss in accuracy with maximized stride without loss in accuracy [10,11].

Dropout. layers are utilized in CNN systems with the objective of diminishing overfitting. This layer "Dropout" an irregular arrangement of neurons inside this layer by configuring their initiation value to zero. It ensures that the system able to do sum up to test information by getting loads that are coldhearted toward preparing tests. Dropout is utilized during preparing with various rates of all out number of neurons in each layer [12].

Implementation

Figure (1), The following workflow highlights the software implementation with main libraries based on Python3.

Figure (1) Software Implementation

Step One: Basically, this model trains and tests using Kaggle dataset for Arabic handwritten letters and digits which available in [13,14].

Step Two: The program starts with importing main important libraries to be used in model like:

- NumPy: It gives a superior multidimensional cluster and fundamental instruments to register with and control these exhibits.
- Pandas: It offers information structures and activities for controlling numerical tables and time arrangement.
- CSV: It makes reading CSV files is possible in pandas. It is highly recommended when have a lot of data to analyze.
- PIL: It adds support for opening, manipulating, and saving many different image file formats.
- Matplotlib: It is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Step Three: After preparing the dataset and needed libraries, the training can be done using predefined CSV file which consists from digits and letters, as shown in Figure (2).

	0	1	2	3	4	5	6	7	8	9	...	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 4096 columns

Specified 60000 (64X64) Arabic Digits for training

Specified 10000 (64X64) Arabic Digits for testing

Specified 13440 (64X64) Arabic Letters for training

Specified 3360 (64X64) Arabic Letters for testing

Figure (2) Trained Letters and digits output

Step Four: Rescale the images that represented in CSV by dividing every pixel value in the image by 255 to make them in range [0, 1] using code in Figure (3)

```
In [28]:
scaled_digits = training_digits_images.values.astype('float32')/255
scaled_digits_labels = training_digits_labels.values.astype('int32')
scaled_test_digits = testing_digits_images.values.astype('float32')/255
scaled_test_digits_labels = testing_digits_labels.values.astype('int32')

scaled_letters = training_letters_images.values.astype('float32')/255
scaled_letters_labels = training_letters_labels.values.astype('int32')
scaled_test_letters = testing_letters_images.values.astype('float32')/255
scaled_test_letters_labels = testing_letters_labels.values.astype('int32')
```

Figure (3) Scaling dataset using dividing by 255

Step Five: Make training the scaled letters (13440X4096) and digits (60000X4096), Figure (4) denoted the output training

```
Out[35]: array([[ 0.,  0.,  0., ...,  0.,  0.,  0.],
                [ 0.,  0.,  0., ...,  0.,  0.,  0.],
                [ 0.,  0.,  0., ...,  0.,  0.,  0.],
                [ 0.,  0.,  0., ...,  0.,  0.,  0.],
                [ 0.,  0.,  0., ...,  0.,  0.,  0.]], dtype=float32)

array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]], dtype=float32)
```

Figure(4) Output training image after scaling

Step Six: This step is very complex and consists from many operations, in short, many layers cooperated to perform high accuracy in less time, Figure (5) shows the model summary, and next section highlights the layers specification in details.

```
In [41]:
model = create_model()
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        160
_________________________________________________________________
batch_normalization_1 (Batch (None, 64, 64, 16)        64
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
batch_normalization_2 (Batch (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 16, 64)        256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_3 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
batch_normalization_4 (Batch (None, 8, 8, 128)         512
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
dropout_4 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 38)                4902
=================================================================
Total params: 103,014
Trainable params: 102,534
Non-trainable params: 480
_________________________________________________________________
```

Figure (5) Model summary

Layers Specification

- A convolutional layer is the first hidden layer. Which it has 16 maps of different features with the 3X3 size and an activation function which is relu. This is called the input layer.
- The second layer is Normalization which solves having distributions of the features vary across the training and test data, which breaks the IID assumption.
- The third layer is the MaxPooling layer. MaxPooling layer is used to down sample the input in order to enable the proposed model to make supposition about the features that leading to reduce overfitting. It also reduces the parameters to learn and reducing the time of training.
- A Regularization layer is the next layer that used Dropout. It is designed to arbitrarily reject 20% of neurons in the layer so as to lessen overfitting.

- Another hidden layer with 32 feature maps with the size of 3×3 and a relu activation function to capture more features from the image.
- Other hidden layers with 64 and 128 feature maps with the size of 3×3 and a relu activation function to capture complex patterns from the image which will decribe the digits and letters later.
- More MaxPooling, Normalization, Regularization and GlobalAveragePooling2D layers.

- The last layer is the output layer with 10 neurons (number of output classes) and it uses softmax activation function. Figure (6) shows the main layers of model.

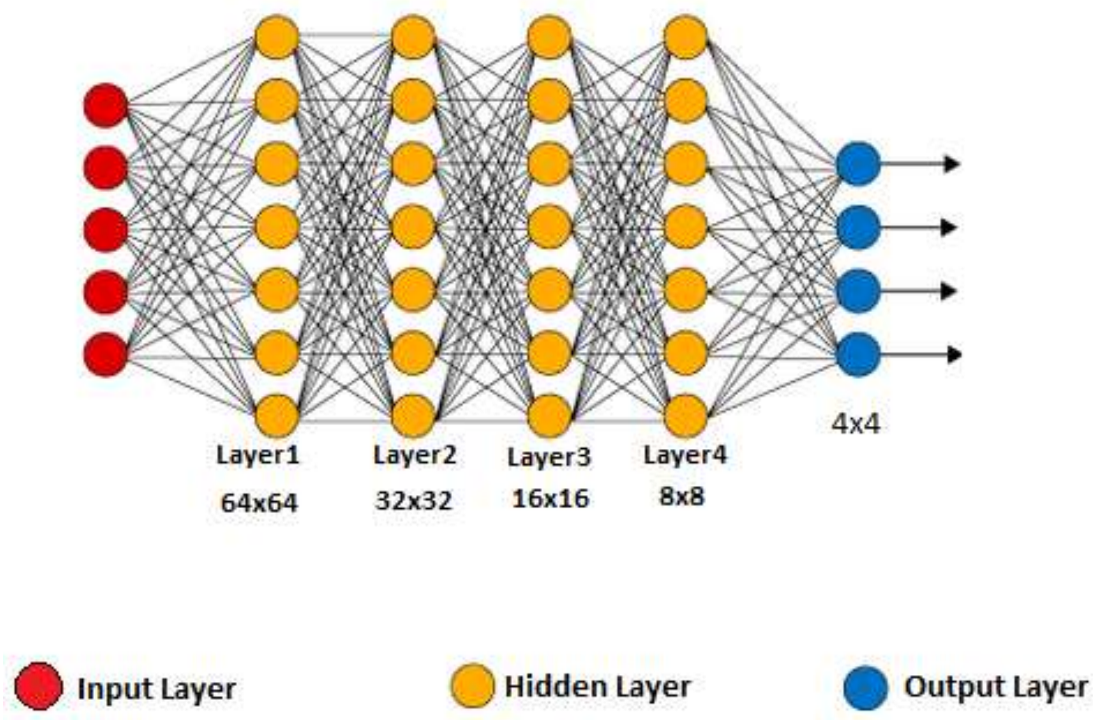

Figure (6) Main layers of suggested model

Dataset

Arabic digits and characters that used to make training and test in this paper is originates from Kaggle kernels, the datasets are CSV files representing the image pixels values and their corresponding label.

Arabic Digits Dataset represents MADBase (modified Arabic handwritten digits database) which contains 60,000 training images, and 10,000 test images. MADBase was written by 700 writers. Each writer wrote each digit (from 0 -9) ten times. To ensure including different writing styles, the database was gathered from different institutions: Colleges of Engineering and Law, School of Medicine, the Open University (whose students span a wide range of ages), a high school, and a governmental institution [13,14], as shown in Figure (7).

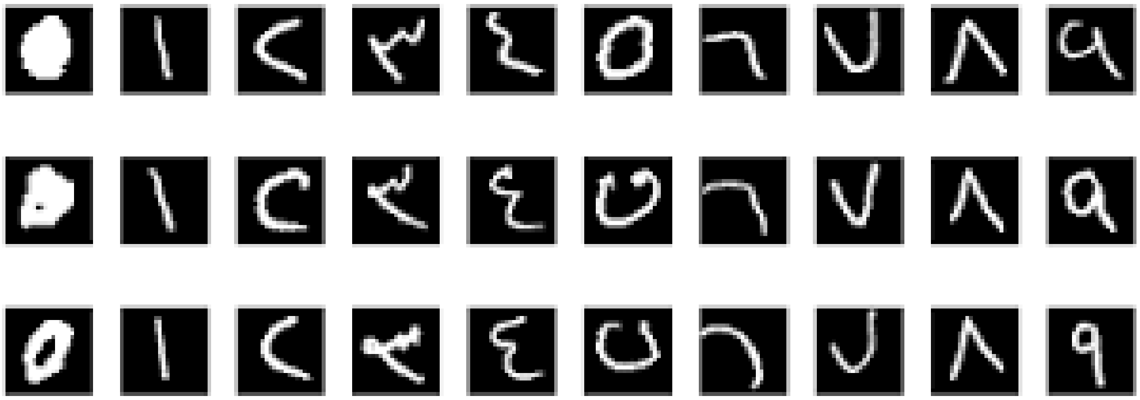

Figure (7) Arabic Handwritten Digits

Arabic Letters Dataset is made out of 16,800. handwritten letters composed by 60 members, the range of age is between nineteen to fourteen years, and 90. % of members are righty hand. Every member composed all letters (from 'alef' to 'yeh') multiple times. These pictures have been examined at the goals of 300dpi. The dataset is parceled into couple sets, a preparation set of 13,440. letters to 480. pictures for each class. and a test set of 3,360 characters to 120 pictures for each class, Figure (8) shows sample of Arabic letters dataset.

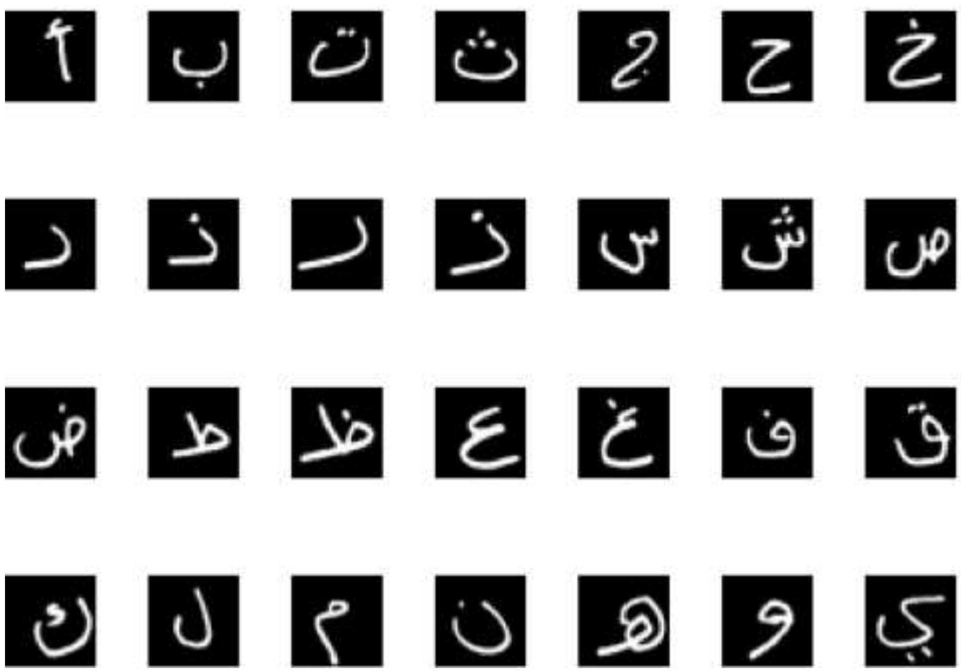

Figure (8) Arabic Handwritten Letters

Convolutional Neural Network (CNN)

The convolutional neural network system, or (CNN) for short, is a specific sort of neural system model intended for handling with two dimensional picture information, in spite of the fact that they can be utilized with one-dimensional and three-dimensional information [15], Models incorporate time-arrangement information, which can be thought of as a 1-D matrix taking examples at customary time interims, and picture information, which can be thought of as a 2-D framework of pixels. Convolutional systems have been massively fruitful in commonsense applications. The name "convolutional neural system" demonstrates that the system utilizes a scientific activity which is known as convolution. [16] A convolution is a duplication activity of all pixels in the picture with each a motivation in the segment, which is thusly another lattice and after that summing the items. The key favorable position of utilizing the convolution task means creating numerous pictures from the first picture that improve various highlights removed from the first picture, which prompts making the order procedure all the more dominant [17].

Conclusion

Deep CNN is used to implement model with Normalization to help in two directions faster learning and higher accuracy. In addition, Maxpooling layer was used to reduce overfitting, number of parameters to learn and training time, which help classify the Arabic handwritten images into digits and letters. The model tested on more than 13000 different characters and digits with all possible classes and got high accuracy of 98.86%.

Conflict of Interests.
There are non-conflicts of interest

References

[1] Mahinnaz Mirdehghan. 2010. Persian, Urdu, and Pashto: A comparative orthographic analysis. Writing Systems Research Vol. 2, No. 1, 9–23.

[2] Das, N.; Mollah, A.F.; Saha, S.; Haque, S.S. Handwritten Arabic Numeral Recognition using a Multi Layer Perceptron. In Proceedings of the National Conference on Recent Trends in Information Systems (ReTIS-06), Kolkata, India, 14–15 July 2006; pp. 200–203.

[3] Babu, U.R.; Venkateswarlu, Y.; Chintha, A.K. Handwritten digit recognition using k-nearest neighbour
classifier. In Proceedings of the 2014 World Congress on Computing and Communication Technologies, (WCCCT 2014), Trichirappalli, India, 27 February–1 March 2014; pp. 60–65.

[4] M. Elleuch, N. Tagougui and M. Kherallah, "Arabic Handwritten Characters Recognition Using Deep Belief Neural Networks," Proc. of the 12th International Multi-Conference on Systems, Signals & Devices (SSD15), pp. 1-5, Mahdia, Tunisia, 16-19 March 2015.

[5] H. Alwzwazy et al., "Handwritten Digit Recognition Using Convolutional Neural Networks," International Journal of Innovative Research in Computer and Communication, vol. 4, no. 2, 2016.

[6] Fetulhak Abdurahman, "Handwritten Amharic Character Recognition System Using Convolutional Neural Networks", DergiPark, Engineering Sciences ,Archive, Volume 14, Issue 2, Pages 71 – 87, 2019,

[7] Khaled S. Younis1, "Arabic Handwritten Character Recognition Based On Deep Convolutional Neural Networks", Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 3, No. 3, December 2017.

[8] MohammedAli Mudhsh, Rolla Almodfer, "Arabic Handwritten Alphanumeric Character Recognition Using Very Deep Neural Network", Information, MDPI, 2017.

[9] Ali A. Alani, "Arabic Handwritten Digit Recognition Based on Restricted Boltzmann Machine and Convolutional Neural Networks", Information, MDPI, 9 November 2017.

[10] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems (NIPS), pp. 1097-1105, 2012.

[11] J. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," arXiv preprint arXiv:1412.6806, 2014.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.

[13] https://www.kaggle.com/mloey1/ahcd1

[14] https://www.kaggle.com/mloey1/ahdd1

[15] Deep learning for computer Vision, A Gentle Introduction to Convolutional Layers for Deep Learning Neural Networks by Jason Brownlee on April 17, 2019 in Deep Learning for Computer Vision.

[16] Deep Learning Book: Chapter 9-Convolutional Networks, Aman Dalmia, Ameya Godbole, Jun 16, 2018.

[17] Y. Le Cun, K. Kavukvuoglu and C. Farabet, "Convolutional Networks and Applications in Vision," Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'10), Paris, France, 2010.

الخلاصة

احد أكثر التحديات التي تواجه التعلم الآلي هو التعرف على الكتابة بخط اليد ، وخاصة النصوص العربية ، لأن هناك العديد من أساليب الكتابة للخط العربي. في هذه الورقة ، يُقترح نموذج تحقيق لتمييز النصوص العربية المكتوبة بخط اليد باستخدام الشبكة العصبية التلافيفية (CNN)، مع طبقات متعددة من التطبيع والتنظيم لتقليل وقت التدريب وزيادة الدقة الإجمالية ، تم الوصول الى دقة تحقق 98 ٪ لمجموعة بيانات Kaggle للغة العربية حيث استخدمت أحرف وأرقام مكتوبة بخط اليد باستخدام Python.

الكلمات الدالة: الشبكة العصبية التلافيفية، التعليم المعمق، الخط العربي، لغة البايثون.