J. Intell. Syst. ? (????), 1–19
DOI 10.1515/jisys-????-??? © de Gruyter ????

The Use of Convolutional Neural Network to Recognize Handwritten Arabic Digits

Ahmed El-Sawy, Mohamed Loey and Hazem EL-Bakry

Dedicated ...

Abstract. Handwritten digits recognition has been an important area due to its applications in several fields. This work is focusing on the recognition as a part of handwritten Arabic digits recognition that face several challenges, including the unlimited variation in human handwriting and the large public databases. Our paper provides a deep learning technique that can be effectively apply to recognizing Arabic handwritten digits. In this paper, we present a Convolutional Neural Networks (CNN) that provide state-of-the-art results for image recognition tasks. The CNN was trained and tested MADBase database (Arabic handwritten digits images) that contain 60000 training and 10000 testing images. Moreover, the CNN is giving an average recognition accuracy of 99.15%. In this work, the optimization methods implemented to increase the performance of CNN. A comparison is held amongst the results, and it is shown by the end that the use of CNN was leaded to significant improvements across different machine learning classification algorithms.

Keywords. Deep Learning, Convolutional Neural Network, Arabic Digits Recognition.

1 Introduction

In recent years, the world are entering the era of **big data**, For example, as of 2016, Every day, our world is creating about 3 quintillion (3×10^{18}) bytes of data per day [35]. Experts estimate that big data volume will reach 35 zetta bytes (35×10^{21}) by 2020 [34]. This flood of data calls for automated methods and algorithms to analysis this big data , which is known as machine learning. Machine learning is a set of methods and algorithms that can automatically detect patterns in data or to perform other kinds of decision making. Today, we want to dive deeper on big data and detect low level feature, this lead to deep learning. Deep learning is a class of machine learning techniques and algorithms that learn multiple levels of representations in hierarchical architecture are exploited for pattern classification [8, 10, 40]. The core of deep learning is to compute hierarchical features of the observational data, where the higher-level features are defined from lower-level features [11]. Various deep learning architectures such as Convolutional Neural Networks (CNN), Deep neural networks, Deep Boltzmann machines and Deep belief networks have been applied to fields like natural language processing, computer vision, automatic speech recognition, audio recognition and bioinformatics [12].

Recognition is an area that covers various fields such as, face recognition, image recognition, finger print recognition, character recognition, numerals recognition, etc. Handwritten Character and Digit Recognition System (HCDR) [39] is an intelligent system able to recognize handwritten characters or digits as human see. The Eastern Arabic numerals consist of ten digits shown in Fig. 1. The rest of the paper is organized as follows: Section 2 gives a review on some of the previous related work done in the area. Section 3 gives a background and preliminaries. Section 4 describes the proposed approach, Section 5 gives an overview of the experiment and results, and we list our conclusions and future work in Section 6.

Western Arabic numerals	0	1	2	3	4	5	6	7	8	9
Eastern Arabic numerals	٠	١	٢	٣	٤	٥	٦	٧	٨	٩

Figure 1. Eastern Arabic numerals

2 Previous Related Work

Various methods have been proposed and high recognition rates are reported for the recognition of Latin handwritten digits [4, 7, 38], Persian [22, 52], Odia [9], South Indian digits [33], Bangla [15]. In recent years many researchers addressed the recognition of text including Arabic. In 2008, Mahmoud [29] proposed a technique for the automatic recognition of Arabic handwritten digits using Gabor-based features and Support Vector Machines (SVMs). They used a medium database have 21120 samples written by 44 writers. The dataset contain 30% for testing and the remaining 70% of the data is used for training. They achieved average recognition rates are 99.85% and 97.94% using 3 scales & 5 orientations and using 4 scales & 6 orientations, respectively.

In 2011, Melhaoui et al. [31] proposed an improved method for recognizing Arabic digits based on Loci characteristic. Their work is based on handwritten and printed numeral recognition. The recognition is carried out with multi-layer perceptron technique and K-nearest neighbour. They trained there algorithm on dataset contain 600 Arabic digits with 200 testing images and 400 training images. They were able to achieve 99% recognition rate on small database. In 2013, Pandi selvi and Meyyappan [41] presented a method to recognize Arabic digits using back propagation neural network. The final result shows that the proposed method provides an recognition accuracy of more than 96% for a small sample handwritten database.

In 2014, Takruri et al. [49] presented three level classifier based on Support Vector Machine, Fuzzy C Means and Unique Pixels for the classification of handwritten Arabic digits. they tested the new algorithm on a public dataset. The dataset contain 3510 images with 40% are used for testing and 60% of images are used for training. The overall testing accuracy reported is 88%. In 2014, Majdi Salameh [37] proposed two methods about enhancing recognition rate for typewritten Arabic digits. First method that calculates number of ends of the given shape and conjunction nodes. The second method is fuzzy logic for pattern recognition that studies each shape from the shape, and then classifies it into the numbers categories. Their proposed techniques was implemented and tested on some fonts. The experimental results made high recognition rate over 95%.

In 2014, AlKhateeb et al. [5] presented a system to classify Arabic handwritten digit recognition using Dynamic Bayesian Network. They used discrete cosine transform coefficients based features for classification. Their system trained and tested on Arabic digits database (ADBase) [2] which contains 70,000 Arabic digits. They reported average recognition accuracy of 85.26% on 10,000 testing samples. In 2015, Kathirvalavakumar and Palaniappan [20] propsed a method for recognizing handwritten Arabic numerals using partitioning approach and K-NN algorithm. They used their dataset that contain 6670 patterns of digits, 3340 of patterns are used for training and 3330 patterns are used for testing. The overall testing accuracy reported is 98.7%.

3 Background and Preliminaries

3.1 Motivation

The motivation of this study is to use cross knowledge learned from multiple works to enhancement the performance of Arabic handwritten digits recognition. In recent years, Arabic digits recognition have different handwriting styles, making it important to find and work on a new and advanced solution for handwritten recognition. A deep learning systems needs a big dataset (images) to be able to make a good decisions. In previous related work they applied the algorithms on a small dataset of Arabic handwritten digits, and have a good classification rate.

3.2 Recognition

Machine learning is usually divided into three main types: supervised learning, unsupervised learning and reinforcement learning. In the supervised learning approach, the goal is to learn a mapping from inputs x to outputs y, given a labeled set of input-output pairs $T_d = \{(x_i, y_i)\}_{i=1}^{N}$ where T_d is the training data and N is the number of training examples. In the simplest setting, each training input x_i is a D-dimensional vector of features or attributes. x_i could be a complex structured object, such as an image or sound. The goal of supervised learning is to learn a mapping from inputs x_i to outputs y_i, where $y_i \in \{1, ..., C\}$, where C is the number classes. If $C = 2$, this is called binary recognition {0,1}; if $C > 2$, this is called multi-class recognition. The performance of leaning system is estimated by measuring the accuracy on a set of samples disjoint from the training data, called testing data.

3.3 Convolutional Neural Network

Currently, Convolutional Neural Network (CNN) has become increasingly important to deep learning, as reviewed by [25]. Convolution Neural Networks (CNN) is supervised learning and a family of multi-layer neural networks particularly designed for use on two-dimensional data, such as images and videos [43]. In more recent work, researchers have applied CNNs to various machine learning problems including fingerprint recognition [32, 48], document analysis [46], face detection [6, 53, 54], speech detection[1] and for more applications reviewed in [28]. In a convolutional neural network data and functions have additional structure. The input data $x_1, x_2, ..., x_n$ are images or sounds. Formally, the input to a convolutional layer is $M \times M \ \times \ C$ image where M is the height and width of the image, $M \times M$ is number of pixels in image and C is number of channels per pixel. For gray scale image have one channel $C = 1$ but RGB image have three channels $C = 3$. A CNN consists of a number of layers (convolutional layers, pooling layers, fully connected layers). The convolutional layer will have K filters (kernels) of $N \ \times \ N \times R$ where N is height and width of filter (kernels) and R is the same number of image channels C or less and may vary for each filter (kernel). The filter (kernel) convolved with the image to produce k feature maps of size $M - N + 1$ shown in Fig. 2. Each feature map is then pooled typically with mean or max pooling over $q \ \times \ q$ where q is range between 2 to 5 for large inputs that illustrated in Fig. 3. After the convolutional layers and pooling layers there may be any number of fully connected layers as in a standard multi-layer neural network.

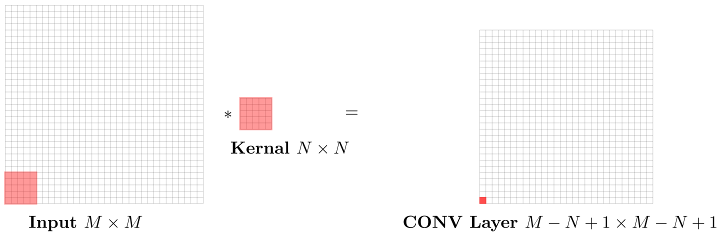

Figure 2. Illustration of convolution process on input image of size $M \times M$ with kernal of size $N \times N$. If the input is 32×32 and the kernal is 5×5 then the output of convolution is 28×28.

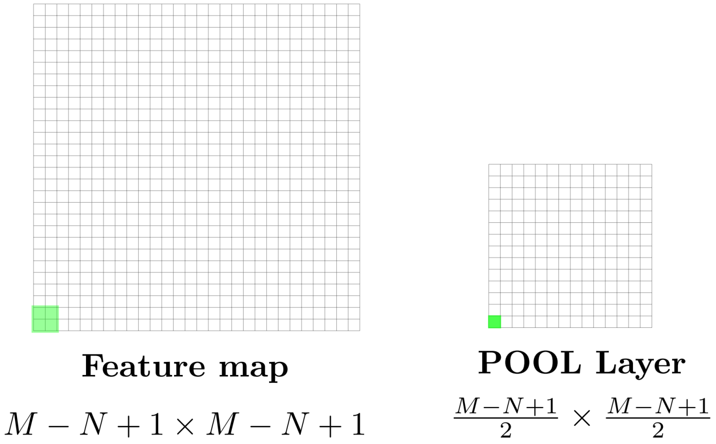

Figure 3. Illustration of pooling process on the feature map that produce a new feature map of size $\frac{M-N+1}{2}$. If the input ia 28×28 then the output of pooling layer is 14×14

3.4 CNN Layers

Convolutional Layers

Let layer l be a convolutional layer. Suppose that we have some $M \times M$ square neuron nodes which is followed by our convolutional layer. If we use an $N \times N$ filter W then convolutional layer output will be of size $(M-N+1) \times (M-N+1)$ which produce k-feature maps that illustrated in Fig. 4. Convolutional layer acts as a feature extractor that extracts features of the inputs such as edges, corners, , endpoints. In order to compute the pre-nonlinearity input to some unit. Then, the input of layer $l-1$ comprises is computed as:

$$Y_i^l = B_i^{(l)} + \sum_{a=1}^{M} \sum_{b=1}^{M} W_i \; X_{(i+a)(j+b)}^{l-1}$$

where $B_i^{(l)}$ is a bias matrix and $W_i^{(l)}$ is the filter of size $N \times N$. Then, the convolutional layer applies its activation function as:

$$Z_i^l = \sigma(Y_i^l)$$

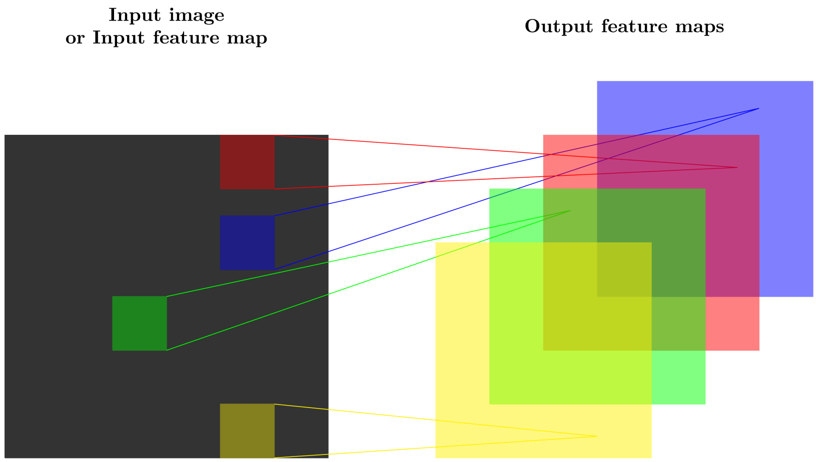

Figure 4. Illustration of a single convolutional layer which produce k-feature maps

Pooling Layers

After each convolutional layer, there may be a pooling layer. Fig. 5 shown the pooling layer takes a small rectangular blocks from the convolutional layer and subsamples it to produce a single output from that block. The pooling layer reduces the resolution of the image that reduce the precision of the translation (shift and distortion) effect. There are several ways to do this pooling, such as taking the average or the maximum, or a learned linear combination of the neurons in the block. Our pooling layers will always be max-pooling layers. It picks out the

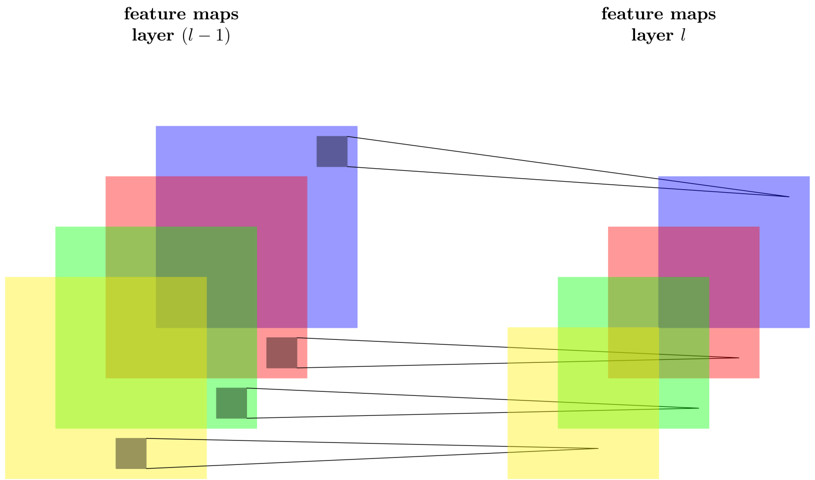

Figure 5. Illustration of pooling layer which produce the same k-feature maps

highest activation in a local region. All the max-pooling is carried out over a 2×2 pixel window. max-pooling layers reduces the size of the activation for the next layer. This lead to smaller number of parameters to be learnt in the later layers. The pooling layer reduces the resolution of the images from convolutional layer thus reduces the precision of the translation (shift and distortion) effect since the feature maps are sensitive to translation in input.

Fully-Connected Layers

Finally, after several convolutional and pooling layers, the high-level reasoning in the neural network is done via fully connected layers. A fully connected layer takes all neurons in the previous layer (be it fully connected, pooling, or convolutional) and connects it to every single neuron it has. Fully connected layers are not spatially located anymore (you can visualize them as one-dimensional), so there can be no convolutional layers after a fully connected layer. The output of fully-connected layer is describe as:

$$Y_i^l = \sum_{j=1}^{D^{l-1}} Y_j^{l-1} * W^l(j,i) + B_i^l$$

where D is the number of neurons in the preceding layer $l-1$, $W^l(j,i)$ is the weight for the connection from neuron j in layer $l-1$ to neuron i in layer l, and

B_i^l is is the bias of neuron i in layer l. Then, the fully-connected layer perform its activation function as:

$$Z_i^l = \sigma(Y_i^l)$$

4 Proposed Approach

4.1 CNN Architecture

The best architecture that we found has two convolutional layers followed by two fully connected hidden layers. Each convolutional layer was followed by a max-pooling layer. All hidden units activation function are Rectified Linear Units (ReLU). In the following, convolutional layers are labeled Cx, pooling layers are labeled Sx, and fully connected layers are labeled FCx, where x is number of layer. In Fig. 6 shown the proposed CNN approach for Arabic handwritten digit recognition that describe as follow : INPUT $\rightarrow$ CONV $\rightarrow$ RELU $\rightarrow$ POOL $\rightarrow$ CONV $\rightarrow$ RELU $\rightarrow$ POOL $\rightarrow$ FC $\rightarrow$ RELU $\rightarrow$ FC $\rightarrow$ Output. For one direction in a channel (feature map) of the convolutional layer in first layer, the output is $((28-5)+1)) = 24$. Layer C1 is a convolutional layer with 80 feature maps. The size of the feature maps is 24 $\times$ 24 pixels. The total number of neurons is 23040 = (24 $\times$ 24 $\times$ 40). There are $((5 \times 5 + 1) \times 40)$=1040 trainable weights. The total number of connection is 599040 = $(24 \times 24 \times (5 \times 5 + 1) \times 40)$. The max pooling layer has non overlapping regions, so it down-samples by 2 in each direction,the output is $24/2 = 12$. Layer S2 is a pooling layer with 40 feature maps. The size of the feature maps is 12 $\times$ 12. The total number of neurons is 5760 (12 $\times$ 12 $\times$ 40). There are (2 $\times$ 40)=80 trainable weights.The total number of connection is 28800 = $(12 \times 12 \times (2 \times 2 + 1) \times 40)$. For one direction in a channel (feature map) of the convolutional layer on third layer, the output is $((12 - 5) + 1) = 8$. Layer C3 is a convolutional layer with 32 feature maps. The size of the feature maps is 8 $\times$ 8 pixels. The total number of neurons is 2048 =(8 $\times$ 8 $\times$ 32). There are $((5 \times 5 + 1) \times 32)$=832 trainable weights. The total number of connection is 53248 = $(8 \times 8 \times (5 \times 5 + 1) \times 32)$. The max pooling layer has non overlapping regions, so it down-samples by 2 in each direction, the output is $8/2 = 4$ on fourth layer. Layer S4 is a pooling layer with 32 feature maps. The size of the feature maps is 4 $\times$ 4. The total number of neurons is 512 = (4 $\times$ 4 $\times$ 32). This is the size of the input to the fully connected layer. There are (2 $\times$ 32)=64 trainable weights.The total number of connection is 2560 = $(4 \times 4 \times (2 \times 2 + 1) \times 32)$. Layer FC5, contain 256 units and is fully connected to S4. Layer FC6 (Output), contain 10 units and is fully connected to FC5. Finally, the FC6 is composed of softmax classifier to produce 10 output class.

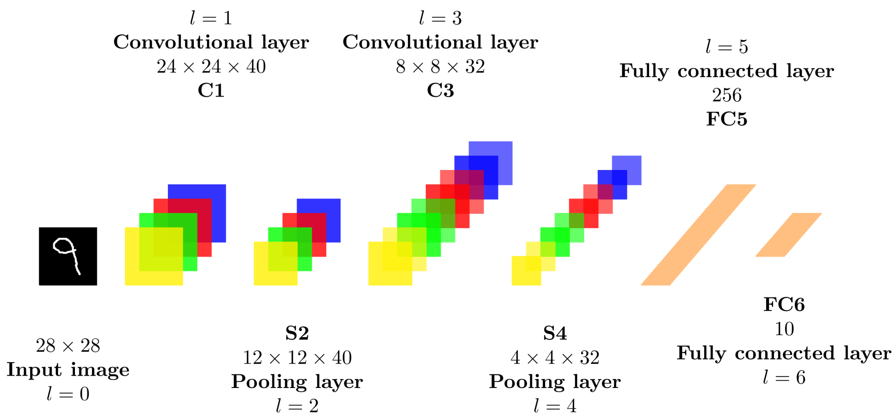

Figure 6. The proposed CNN for Arabic handwritten digit recognition

4.2 CNN Optimization

Learning Rate. Learning rate α is used during the weight update of such architecture. This parameter is crucial in determining the successful of convergence and generalization of such neural network. A too small learning rate leads to slow convergence and oppositely leads to divergence.
Activation Function. The activation function that is used in our proposed approach is Rectified Linear Units (ReLU) that applies the non-saturating $\sigma(Y_i^l) = max(0, Y_i^l)$. We apply the ReLU non-linearity as an activation fuction to the output of every convolutional layer and fuly connected layer. ReLU is more biologically plausible it can be engaged as better activation functions than the widely used logistic sigmoid and hyperbolic tangent functions. The ReLU [17] increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. Th use of ReLUs map more plausibly to biological neurons, makes the training of deep neural network significantly faster and improves its generalization ability.
Stochastic Gradient Descent. The gradient descent algorithm updates the parameters (weights and biases) so as to minimize the error function by taking small steps in the direction of the negative gradient of the loss function:

$$P_{i+1} = P_i - \alpha \nabla E(P_i)$$

where i stands for the iteration number, $\alpha > 0$ is the learning rate, P is the parameter vector, and $E(P_i)$ is the loss function. The gradient of the loss function,

$\nabla E(P_i)$, is evaluated using the entire training set, and the standard gradient descent algorithm uses the entire data set at once [23].

Mini-batch. The stochastic gradient descent algorithm evaluates the gradient, hence updates the parameters, using a subset of the training set. This subset is called a mini-batch. Mini-batch optimization [27] is to divide the dataset into small batches of examples, compute the gradient using a single batch, make an update, then move to the next batch. Each evaluation of the gradient using the mini-batch is an iteration. At each iteration, the algorithm takes one step towards minimizing the loss function. The full pass of the training algorithm over the entire training set using mini-batches is an epoch. We specify the mini-batch size as 256 and the maximum number of epochs is 30.

Momentum. The gradient descent algorithm might oscillate [45] along the steepest descent path to the optimum. Adding a momentum [47] term to the parameter update is one way to prevent this oscillation. The SGD update with momentum is

$$P_{i+1} = P_i - \alpha \nabla E(P_i) + \gamma (P_i - P_{i-1})$$

where γ determines the contribution of the previous gradient step to the current iteration. By default, before training the input data shuffles.

L2 Regularization. One problem with machine learning estimation is that it can result in overfitting. Adding a regularization term [23] for the weights to the loss function $E(P_i)$ is a way to reduce overfitting. The loss function with the regularization term takes the form:

$$E_R(P_i) = E(P_i) + \lambda \Omega(w)$$

where w is the weight vector, λ is the regularization factor (coefficient), and the regularization function, $\Omega(w)$ is:

$$\Omega(w) = \frac{1}{2} w^t w$$

Softmax classifier. The softmax classifier is used in various probabilistic multi-class classification methods. To obtain a probability value for each class, the softmax function is applied to the final output units of the network:

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}}$$

Early stopping. Early stopping monitoring the deep learning process of the network from overfitting. If there is no more improvement, or worse, the performance on the test set degrades, then the learning process is aborted [23].

Dropout. Dropout is a technique of reducing overfitting in CNN. The term âĂIJ-dropoutâĂİ refers to dropping out units (hidden and visible) in a neural network [44]. Dropout is a powerful regularization method introduced by Hinton et.al. [16]. We randomly drop neurons with a probability p during training.

5 Experiment

5.1 DataSet

El-sherif and Abdleazeem [2] released an Arabic handwritten digit database (ADBase) and modified version called (MADBase). The MADBase is a modified version of the ADBase benchmark that has the same format as MNIST benchmark [26]. ADBase and MADBase are composed of 70,000 digits written by 700 writers.

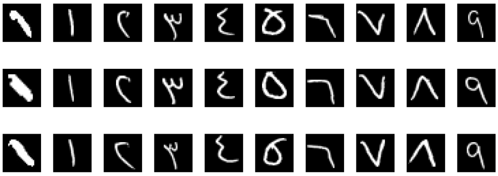

Figure 7. Samples of digits in MADBase benchmark training database

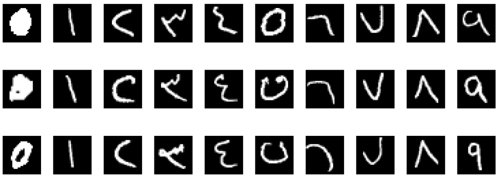

Figure 8. Samples of digits in MADBase benchmark testing database

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. The databases is partitioned into two sets: a training set (60,000 digits to 6,000 images per class) and a test set (10,000 digits to 1,000 images per class).The ADBase and MADBase is available free for researchers from (http://datacenter.aucegypt.edu/shazeem/). Figures 7 & 8 shows samples of training and testing images of MADBase database.

5.2 Results

In this section, the performance of CNN was investigated for training and recognition of Arabic digits. The experiments are conducted in MATLAB 2016a programming environment with CUDA SDK v.7.5. Our CNNs are implemented by using MatConvNet [51] toolbox. MatConvNet is an open source for experimenting with convolutional neural networks that is deeply integrated in MATLAB and allows easy experimentation with novel ideas. All experiments were performed on a 2.6 GHz Core i7 PC with 8G memory and GPU NVIDIA Geforce 840M running on windows.

Figure 9. Confusion matrix of 10000 testing images for proposed Arabic digits recognition

In this section, Our proposed CNN for Arabic digits are evaluating with incremental training approach with mini-batch mode. In figure 9, confusion matrix of

Arabic digits illustrated that the first two diagonal cells show the number and percentage of correct classifications by the trained CNN. For example 884 of class (1) are correctly classified as class (1). This corresponds to 9.8% of all 1000 test images of class (1). The column in the target class show the misclassification of the class. Overall, 99.15% of the predictions are correct and 0.85% are wrong classifications.

Figure 10 shows all 85 misclassified test examples. some of those examples are genuinely ambiguous, but several are perfectly identifiable by humans. this shows that further improvements are to be expected with more training data.

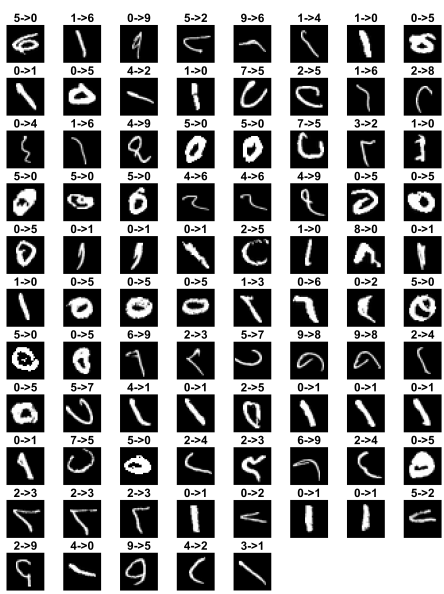

Figure 10. The 85 misclassified handwritten digits on testing images. The higher label is Output (predicted network output) $\rightarrow$ Target (correct answer). These misclassification are mostly caused either by genuinely ambiguous patterns

Finally, in Table 1 shown the obtained results with CNN on MADBase database. It can be seen from Table 1 that the proposed approach have the large dataset and have the best misclassification error. The results are better than the results reported in related work [5, 29, 31, 37, 41, 49], although it is sometimes hard to compare, because previous work has not experimented with large database benchmark.The proposed method obtained 0.85% misclassification error rate on testing data.

Table 1. Comparison between Proposed Approach and Other Approach for Arabic digits

Authors	Database	Images	Error Rate
Takruri et al. [49]	Private	3510	12%
AlKhateeb et al. [5]	ADBase	70,000	14.74%
Majdi Salameh [37]	Fonts	2000	5%
Melhaoui et al. [31]	Private	600	1%
Pandi selvi and Meyyappan [41]	Private	Samples	4%
Mahmoud [29]	Private	21120	0.15% and 2.16%
Kathirvalavakumar and Palaniappan [20]	Private	6670	1.3%
Our Approach	**MADBase**	**70000**	**0.85%**

6 Conclusion

In this paper, we have proposed a CNN architecture to solve handwritten Arabic digits recognition. The purpose to use deep learning was to take advantages of the power of CNN that are able to manage large dimensions input and share their weights. To improve the performance of CNN, we use mini-batch stochastic gradient descent with momentum and ReLU activation functions. Using regularization term (L2 Regularization), early stopping, and dropout, the CNN able to reduce the overfitting problem. The final output layer in our CNN use softmax classifier to obtain a probability value for each class.

In the experiments, we achieved misclassification error rates 0.85% on MADBase benchmark for Arabic handwritten digits. The results on variations of MADBase demonstrate that the new CNN model achieves better classification performance in comparison with several other competitive models. In future work, we plan to work on two directions: one is to improving the performance of handwritten Arabic digits recognition. The other is to work on Arabic handwritten word recognition using deep learning techniques.

Bibliography

[1] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, Convolutional neural networks for speech recognition, *IEEE Transactions on Audio, Speech and Language Processing* **22** (2014), 1533–1545.

[2] Sherif Abdleazeem and Ezzat El-Sherif, Arabic handwritten digit recognition, *International Journal of Document Analysis and Recognition (IJDAR)* **11** (2008), 127–141.

[3] Jafaar Al Abodi and Xue Li, An effective approach to offline Arabic handwriting recognition, *Computers and Electrical Engineering* **40** (2014), 1883 – 1901.

[4] A. Agapitos, M. O'Neill, M. Nicolau, D. Fagan, A. Kattan, A. Brabazon and K. Curran, Deep evolution of image representations for handwritten digit recognition, in: *2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2452–2459, May 2015.

[5] J. H. AlKhateeb and M. Alseid, DBN - Based learning for Arabic handwritten digit recognition using DCT features, in: *Computer Science and Information Technology (CSIT), 2014 6th International Conference on*, pp. 222–226, March 2014.

[6] Grigory Antipov, Sid-Ahmed Berrani and Jean-Luc Dugelay, Minimalistic CNN-based ensemble model for gender prediction from face images, *Pattern Recognition Letters* **70** (2016), 59 – 65.

[7] D. Barbuzzi, G. Pirlo, S. Uchida, V. Frinken and D. Impedovo, Similarity-based regularization for semi-supervised learning for handwritten digit recognition, in: *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 101–105, Aug 2015.

[8] X. W. Chen and X. Lin, Big Data Deep Learning: Challenges and Perspectives, *IEEE Access* **2** (2014), 514–525.

[9] K. S. Dash, N. B. Puhan and G. Panda, Handwritten numeral recognition using non-redundant Stockwell transform and bio-inspired optimal zoning, *IET Image Processing* **9** (2015), 874–882.

[10] Li Deng, A Tutorial Survey of Architectures, Algorithms, and Applications for Deep Learning, *APSIPA Transactions on Signal and Information Processing* (2014).

[11] Li Deng and Dong Yu, *DEEP LEARNING: Methods and Applications*, Report no. MSR-TR-2014-21, May 2014.

[12] Li Deng and Dong Yu, Deep Learning: Methods and Applications, *Foundations and TrendsÂő in Signal Processing* **7** (2014), 197–387.

[13] A. ElAdel, R. Ejbali, M. Zaied and C. B. Amar, Dyadic Multi-resolution Analysis-Based Deep Learning for Arabic Handwritten Character Classification, in: *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, pp. 807–812, Nov 2015.

[14] M. Elleuch, N. Tagougui and M. Kherallah, Arabic handwritten characters recognition using Deep Belief Neural Networks, in: *Systems, Signals Devices (SSD), 2015 12th International Multi-Conference on*, pp. 1–5, March 2015.

[15] T. Hassan and H. A. Khan, Handwritten Bangla numeral recognition using Local Binary Pattern, in: *Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on*, pp. 1–4, May 2015.

[16] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, *CoRR* **abs/1207.0580** (2012).

[17] Farhan Hussain and Jechang Jeong, Efficient Deep Neural Network for Digital Image Compression Employing Rectified Linear Neurons, *Journal of Sensors* **2016** (2016), 7.

[18] R. S. Hussien, A. A. Elkhidir and M. G. Elnourani, Optical Character Recognition of Arabic handwritten characters using Neural Network, in: *Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), 2015 International Conference on*, pp. 456–461, Sept 2015.

[19] M. N. Ibrahim, S. N. Bilmas, A. Babiker and M. Idroas, A Framework of an Online Self-Based Learning for Teaching Arabic as Second Language (TASL), in: *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 255–260, Sept 2013.

[20] T. Kathirvalavakumar and R. Palaniappan, *Recognizing Handwritten Arabic Numerals Using Partitioning Approach and KNN Algorithm*, pp. 226–234, Springer International Publishing, Cham, 2015.

[21] Maâmar Kef, Leila Chergui and Salim Chikhi, A novel fuzzy approach for handwritten Arabic character recognition, *Pattern Analysis and Applications* (2015), 1–16.

[22] K. Kiani and E. M. Korayem, Classification of Persian handwritten digits using spiking neural networks, in: *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 1113–1116, Nov 2015.

[23] Manuel Kroiss, *Introduction to deep neural networks*, pp. 9–29, Springer Fachmedien Wiesbaden, Wiesbaden, 2016.

[24] A. Lawgali, M. Angelova and A. Bouridane, A Framework for Arabic Handwritten Recognition Based on Segmentation, **7** (2014), 413.

[25] Yann LeCun, Yoshua Bengio and Geoffrey Hinton, Deep learning, *Nature* **521** (2015), 436–444.

[26] Yann Lecun, LÃl'on Bottou, Yoshua Bengio and Patrick Haffner, Gradient-based learning applied to document recognition, in: *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[27] Mu Li, Tong Zhang, Yuqiang Chen and Alexander J. Smola, Efficient Mini-batch Training for Stochastic Optimization, in: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 661–670, ACM, New York, NY, USA, 2014.

[28] H. Lu and Q. Zhang, Applications of deep convolutional neural network in computer vision, *Shuju Caiji Yu Chuli/Journal of Data Acquisition and Processing* **31** (2016), 1–17.

[29] S. A. Mahmoud, Arabic (Indian) handwritten digits recognition using Gabor-based features, in: *Innovations in Information Technology, 2008. IIT 2008. International Conference on*, pp. 683–687, Dec 2008.

[30] Sabri Mahmoud, Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models, *Signal Processing* **88** (2008), 844 – 857.

[31] Ouafae EL Melhaoui, Mohamed El Hitmy and Fairouz Lekhal, Article: Arabic Numerals Recognition based on an Improved Version of the Loci Characteristic, *International Journal of Computer Applications* **24** (2011), 36–41, Full text available.

[32] R. F. Nogueira, R. de Alencar Lotufo and R. Campos Machado, Fingerprint Liveness Detection Using Convolutional Neural Networks, *IEEE Transactions on Information Forensics and Security* **11** (2016), 1206–1213.

[33] L. Pauly, R. D. Raj and B. Paul, Hand written digit recognition system for South Indian languages using artificial neural networks, in: *Contemporary Computing (IC3), 2015 Eighth International Conference on*, pp. 122–126, Aug 2015.

[34] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu and Shuo Feng, A survey of machine learning for big data processing, *EURASIP Journal on Advances in Signal Processing* **2016** (2016), 1–16.

[35] Vince Reynolds, *Big Data For Beginners: Understanding SMART Big Data, Data Mining and Data Analytics For improved Business Performance, Life Decisions and More*, 2016.

[36] D. K. Sahu and C. V. Jawahar, Unsupervised feature learning for optical character recognition, in: *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 1041–1045, Aug 2015.

[37] Majdi Salameh, Arabic Digits Recognition Using Statistical Analysis for End/Conjunction Points and Fuzzy Logic for Pattern Recognition Techniques, **4** (2014), 50–56.

[38] L. B. Saldanha and C. Bobda, Sparsely connected neural networks in FPGA for handwritten digit recognition, in: *2016 17th International Symposium on Quality Electronic Design (ISQED)*, pp. 113–117, March 2016.

[39] Ritesh Sarkhel, Nibaran Das, Amit K. Saha and Mita Nasipuri, A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition, *Pattern Recognition* **58** (2016), 172–189 (English).

[40] JÃijrgen Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks* **61** (2015), 85 – 117.

[41] P. P. Selvi and T. Meyyappan, Recognition of Arabic numerals with grouping and ungrouping using back propagation neural network, in: *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*, pp. 322–327, Feb 2013.

[42] Maad Shatnawi and Sherief Abdallah, Improving Handwritten Arabic Character Recognition by Modeling Human Handwriting Distortions, *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **15** (2015), 3:1–3:12.

[43] Suraj Srinivas, Ravi Kiran Sarvadevabhatla, Konda Reddy Mopuri, Nikita Prabhu, Srinivas Kruthiventi and Venkatesh Babu Radhakrishnan, A taxonomy of Deep Convolutional Neural Nets for Computer Vision, *Frontiers in Robotics and AI* **2** (2016).

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research* **15** (2014), 1929–1958.

[45] D. Strigl, K. Kofler and S. Podlipnig, Performance and Scalability of GPU-Based Convolutional Neural Networks, in: *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pp. 317–324, Feb 2010.

[46] X. Sun, F. Gao, C. Li and F. Ren, Chinese microblog sentiment classification based on convolution neural network with content extension method, in: *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pp. 408–414, Sept 2015.

[47] Ilya Sutskever, James Martens, George E. Dahl and Geoffrey E. Hinton, On the importance of initialization and momentum in deep learning, in: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (Sanjoy Dasgupta and David Mcallester, eds.), 28, pp. 1139–1147, JMLR Workshop and Conference Proceedings, May 2013.

[48] A.R. Syafeeza, K.S. Itqan, F.G. Gong, N. Mustafa, Y.C. Wong and M.M. Ibrahim, User identification system based on finger-vein patterns using Convolutional Neural Network, *ARPN Journal of Engineering and Applied Sciences* **11** (2016), 3316–3319.

[49] M. Takruri, R. Al-Hmouz and A. Al-Hmouz, A three-level classifier: Fuzzy C Means, Support Vector Machine and unique pixels for Arabic handwritten digits, in: *Computer Applications Research (WSCAR), 2014 World Symposium on*, pp. 1–5, Jan 2014.

[50] M. Tounsi, I. Moalla, A. M. Alimi and F. Lebouregois, Arabic characters recognition in natural scenes using sparse coding for feature representations, in: *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 1036–1040, Aug 2015.

[51] Andrea Vedaldi and Karel Lenc, MatConvNet: Convolutional Neural Networks for MATLAB, in: *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pp. 689–692, ACM, New York, NY, USA, 2015.

[52] Y. Zamani, Y. Souri, H. Rashidi and S. Kasaei, Persian handwritten digit recognition by random forest and convolutional neural networks, in: *2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 37–40, Nov 2015.

[53] Shu Zhan, Qin-Qin Tao and Xiao-Hong Li, Face detection using representation learning, *Neurocomputing* **187** (2016), 19 – 26, Recent Developments on Deep Big Vision.

[54] Yuanyuan Zhang, Dong Zhao, Jiande Sun, Guofeng Zou and Wentao Li, Adaptive Convolutional Neural Network and Its Application in Face Recognition, *Neural Processing Letters* **43** (2016), 389–399.

Received **???**.

Author information

Ahmed El-Sawy, Faculty of Computer & Informatics, Benha University, Benha, Egypt.
E-mail: `ahmed.el_sawy@fci.bu.edu.eg`

Mohamed Loey, Faculty of Computer & Informatics, Benha University, Benha, Egypt.
E-mail: `mohamed.loey@fci.bu.edu.eg`

Hazem EL-Bakry, Faculty of Computer & Information Sciences, Mansoura University, Mansoura, Egypt.
E-mail: `helbakry5@yahoo.com`