A New Implementation of Deep Neural Networks for Optical Character Recognition and Face Recognition

Khaled S. Younis, Abdullah A. Alkhateeb
The University of Jordan
Computer Engineering Department
Amman, Jordan
E-mail: Younis@ju.edu.jo, akhat47@gmail.com

Abstract

The automatic analysis and recognition of off-line handwritten characters from images is an important area in many applications. Even with the important progress of recent research in optical character recognition, few problems still wait to be solved specially for Arabic characters. The use of Deep Neural Networks may solve these problems. We present a deep neural network for the handwritten OCR problem. The method we used shows favorable performance (accuracy 98.46% on MNIST dataset) with a simple architecture that can be installed on a normal PC. Face recognition of still images is also another important application due to security concerns. Convolutional neural network (CNN) models have demonstrated that they can extract the important features of the face for recognition without extensive preprocessing or feature engineering steps. We present the results of applying a deep CNN on a data collected at the University of Jordan, UJ Face dataset. The method performance was perfect classification after just 80 epochs. The programming environments were TensorFlow and Keras. We showed the importance of regularization on the performance of deep networks.

Keywords: Convolutional Neural Network, Deep Learning, Optical Character Recognition, Face Recognition, Computer Vision

1. INTRODUCTION

The field of optical character recognition is very important especially for offline hand recognition systems as it is more difficult than online recognition systems [1]. The ability to deal with large amount of data in certain context will bring a lot of value. One example of these applications is to automate the text transcription process that intended to be applied on the ancient documents and considering its complexity and irregularity nature due to of the manual aspects of writing [2]. It would be great to access online resources of old documents on online libraries utilizing the revolution of the internet and communication. It is well-known that the work on Arabic optical text recognition is experiencing slowly development process compared to the other languages [3].

On the other hand, the recent terrorist events forced the governments to increase investments in the field of face recognition (FR) to improve current security systems that have weaknesses especially that current authentication systems have many vulnerabilities. It is very useful to apply the latest technologies in the field of face recognition for recognizing people using surveillance camera installed at the malls, airports, universities, and companies.

Deep Learning (DL) is the new application of machine learning for learning representation of data. It is very successful and one Convolutional Neural Network (CNN) DL architecture had won the ImageNet classification challenge in 2012 [4]. Since that time, DL based architectures had won many challenges and competitions in the data science website, Kaggle, that solve real-world problems.

There are several frameworks for deep learning. One of the most popular libraries is TensorFlow that was released by Google in November 2015 [5]. It is an open source for numerical computation. It is written in C++ programming language and capable of using Graphical Processing Units (GPUs) very well. There are Python bindings that doesn't require dealing with C++ directly. TensorFlow can run on different platforms starting from the powerful servers to less powerful devices such as Raspberry Pi and smart phones.

Another simpler framework is Keras [6] (a higher-level API) built on top of TensorFlow or Theano - another deep learning library. Keras uses Python for programming which makes writing programs easier than a native TensorFlow codes.

Deep learning algorithms have been taken the top place in the object recognition field due to the great performance improvement they have provided [7]. Therefore, in this paper we will present the results of utilizing TensorFlow and Keras for building Deep Neural Network (DNN) and CNN for solving the problem of OCR and face recognition. This will open the door to great opportunities expanding the applications of deep learning using these powerful

libraries to problems of ancient handwritten Arabic character recognition. In addition, the we will test the system on other databases for face recognition for security and administrative problems in Jordan.

The contributions of this research are: (i) Taking a closer look at the recent progress in DL, (ii) Working through the different architectures of multi-layer deep neural network. (iii) Applying different techniques to improve the performance of the models such as optimization and generalization methods on standard as well as private databases, and (iv) Utilizing the functionalities offered by TensorFlow and Keras libraries for research purposes at our lab.

The rest of the paper is organized as follows; discussing related work of applying DNN in the fields of OCR and FR. That will be followed by the methodology used for studying and applying the techniques of deep learning in these fields, then we introduce the results obtained, and finally we discuss the conclusions derived from the discussed results and present plans for future work.

2. RELATED WORK

2.1 Optical Character Recognition

Algorithms designed to recognize handwritten characters are still less advanced than that for printed characters, due mainly to the hardships in dealing with the diversity in handwritten characters' shapes and forms. Characters segmentation to automate the recognition process is another problem.

Deep Neural Networks and Convolutional Neural Networks are designed to use many different layers that are capable to learn features automatically. Each neuron in the network takes its input from previous neuron and the activation function decides the information required to be passed to the next neuron. The architecture of the network and it complexity play the major role to determine the ability of the network to predict the desire output from its fed input. Another type of networks called recurrent neural networks is used in different way to discover patterns in unsegmented scripts by working in the sequences of characters. In 2007, the algorithm developed by Graves and Schmidhuber [8] using Hidden Markov Model (HMM) RNN outperformed all entries in the ICDAR 2007 Arabic handwriting recognition competition that correctly classified the IFN/ENIT database of handwritten Arabic word with 91% accuracy.

The interesting fact about Deep Neural Networks is that, by making a large network with many layers it becomes more capable to detect more features automatically. In the work by Jaderberg, et al. [9], the authors used CNN not only to detect words regions in the image, but also to recognizing the words within these regions. They reported good performance on a few benchmark datasets such as 91% accuracy in the character classification of ICDAR 2003. To solve the problem of latency in processing the data, GPUs are used as suggested by Ciresan et al. [10] who trained and tested the CNN network using committee of classifiers and reduced the error rate of MNIST dataset [15] to 2.7%.

Extension of work in Arabic OCR on text embedded in video is presented in [11]. In addition to CNN they had used Auto-encoders with different size filters to extract the features of the characters followed by RNN. The application to images captured from Arabic TV channels is useful and the character recognition rate was 94.6% but the text was not handwritten.

In our work, we apply a simpler DNN network to the problem of MNIST classification and show that with proper choice of the architecture and hyperparameters one can obtain state-of-the-art performance.

2.2 Face Recognition

Face Recognition as a field of study has many challenges and problems in many application of a wide range of areas that makes it attractive to many researchers and companies like Facebook, Google, etc. This had received huge interest in the recent years especially due to the security concerns.

Big success has been reached in the field of the FR using Deep Learning after DL caused a large improvement to the performance that has been found previously on famous datasets. The work in [12] demonstrated the application of two publicly available DL models, namely VGG-Face and Lightened CNN on five benchmark datasets and reported that preprocessing (such as for pose and illumination normalization) can improve the performance significantly even though the classification results varied depending on the database and the amount of alignment. Labeled Faces in the Wild (LFW) database [13] is an example of the dataset that is used a lot to test the progress achieved in term of the accuracy. it also enhances the ability to deal with the variations in images in the dataset due to many factors related to acquisition process and the camera used.

Authors of [14] did a thorough analysis of DL architectures based on Convolutional Neural Networks on LFW database and discussed several important properties of them trying to understand why CNN does good job in FR research. To make this work more beneficial the shared the code and

models for more inspection to the other researchers. Their results showed very close performance for colored versus grey images. Although color images contain more information, they do not deliver a significant improvement.

In general, if we look inside the CNN networks architecture we will find certain types of layers that are used in different orders such as Convolutional Layer, Pooling Layer, and Fully Connected Layer. For the convolutional layer, it uses several filters responsible for finding patterns in the input images, and weights and biases parameters of this layer define the performance of the filtering. then these processed input is sent to the next layer after applying the activation, on the other hand, pooling layer function in a different way where it works to reduce the number of the output data by applying operations such as finding the maximum or average of a window sliding on the input data.

For training purpose, Convolutional Neural Networks use technique of Backpropagation to update its parameters e.g. weights and biases, and using Rectified Linear Unit (ReLU) function that shown its superiority against vanishing gradient problem. Because of the convolution filters being connected to each pixel of the input image, a large number of the previously mentioned parameters is needed, that in turn need a lot of processing which adversely affect performance. This resulting combined layers are repeated with different order and different parameters to recognize a large number of patterns within the input images, and finally, a fully-connected layer is added at the end to produce a number of the output needed as classes. This proposed architecture works in a smart way to extract complex features with an important reduction in computing time and training.

One drawback of CNNs that they require very big data for training to achieve the desired performance. and, they need to be trained using GPUs if we want a speed up of x20 in training compare to a normal computer with CPUs.

In this paper, we will apply one of the best-known models of CNN on privately generated database and use only grayscale images.

3. METHODOLOGY

In this section, we describe the databases used and the architecture parameters for the models used for solving the OCR and FR problems.

3.1 Optical Character Recognition

In this model, we designed a deep multi-layered neural network using TensorFlow to classify the ubiquitous MNIST dataset [15]. MNIST dataset consists of handwritten digits from 0 to 9 and has a training set of 60,000 samples, and a test set of 10,000 samples. The digits are binary 28x28 pixels. The MNIST dataset is suitable because the data is already in the right format and it is built-into the TensorFlow so one could easily work with this dataset and focus on learning TensorFlow syntax and the DNN parameters and not worry about importing the images into the computer, Figure 1 shows part of this dataset.

Figure 1 - MNIST dataset of handwritten digits

Our DNN model consisted initially of 5 layers, the first input layer has 784 neurons that are essentially the input pixels of each MNIST digits image for each successive hidden layer, the input is multiplied by the weights, the products are added to the biases and the results are the input to ReLu activation function. This is cascaded till the output layer. for each hidden layer to extract more detailed features of dataset images,

The first hidden layer consisted of 500 neurons, the second had 1000 neuron, and the third had 250 neurons. To improve the generalization ability of the model for predicting the class of unseen samples, dropout technique at the end of the last layer is used with keep probability parameter of 0.5 where at each epoch training iteration half of the neurons of the last layer get activated while the other half activation will be set to zero. This tends to prevent our network from overfitting by not building a model so is so tightly bonded to training samples. Finally, the output layer had 10 neurons to match the number of classes.

However, before applying the activation function in each layer, we found it useful to apply a technique called Batch Normalization as explained in [22] as a final layer, where the outputs of each neuron are normalized to be zero centered and rescale the logits, this help to fit the data when it fed into the activation function and make the training process faster and effective using backpropagation.

For updating the weights during training, we used

the Categorical Cross-Entropy [17] as a cost function that is the appropriate cost function for multi-class classification problems. We used Adam Optimizer to find the minima of the cost function with a varying learning rate between 0.03 - 0.0001 [19], that recalculate its value after each batch. As we shall see in the Results section, minor modifications were made to this architecture to improve the performance. Figure 2 illustrates the architecture of the OCR model used.

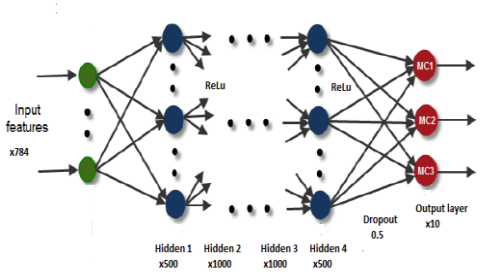

Figure 2 - Character Recognition network architecture.

3.2 Face Recognition CNN model

For the Face recognition task, we wanted to have TensorFlow work on realistic private dataset that was acquired to match the specific project or problem at hand that is taking class attendance at the university by recognizing the faces of students. We built a local database at the University of Jordan and called it UJ Face database by taking fifty clear images for 10 students, male and female, with different pose, zoom, orientation, illumination, and partial occlusion. The images were taken using iPhone 6. The images were resized to become 90 by 160 pixels and then normalized the input image to have zero mean and unity standard deviation, Figure 3 show a sample picture of the dataset used. It is seen that there is difference in pose, scale, and orientation

Figure 3 - Sample images from the UJ Face dataset use in the Face Recognition model

We decided to implement the CNN architecture since it is more powerful than regular DNN. We used the so-called AlexNet by Krizhevski et al. [4] as a starting point for our net and changed it slightly to make it fit our needs. Our CNN has 8 convolutional layers and 4 Max pooling layers followed by two fully connected layers and finally the output layer. Cross validation was used with 20% of the database is randomly selected as a validation set.

Firstly, we stacked two convolutional layers that extract 32 filters of size 5x5 with ReLu as activation function to each layer, then the output of the activation function of the is fed to a max pooling of 2x2 window.

The same architecture is adopted in the following layers with a few changes in the number of filters, size of filters, max pooling layers' specification. The exact architecture is shown in Figure 4. After that, we added two fully connected layers that receive flatten output of the previous layer with 256 neurons and 64 neurons respectively.

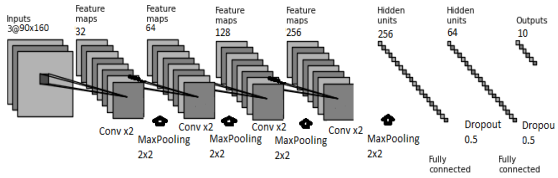

Figure 4 - Face Recognition CNN Architecture

To address the problem of overfitting, we use the Dropout, that was explained in Section 3.1, with 0.5 probability. To reduce overfitting, we also used Early Stopping method [16]. With this method, the model stops learning new weights if the loss function is not reduced for a certain number of epochs. This improves the learner's performance on data outside of the training set. Finally, a 10 neurons layer with a Sigmoid activation function is added to produce the final output of the conventional neural network.

For training purposes, we chose to use Categorical Cross-entropy [17] as loss function which is the appropriate cost function for multi-class classification problems to measure the distance between the output of the model and the true output from the training dataset. The loss value will be minimized using the "RMSprop optimizer" [18] that utilizes the magnitude of recent gradients to normalize the gradients. We used 64 batch size and set the maximum number of epochs to 100.

4. RESULTS

4.1 Optical Character Recognition

The accuracy of the OCR model built as described in Section 3 came to 98.11%. This is a very good accuracy for the first model if we compare it with the performance of other methods on the MNIST

dataset [20]. However, we will see how we can modify our network easily using the tools inherent in TensorFlow to improve the performance. We added dropout to the third hidden layer with a keep probability of 0.5. Next, we decided to go deeper with our network by adding a fourth hidden layer and modifying the number of neurons in the hidden layers to be 500, 1000, 1000, 500, respectively. By having more neurons, we add complexity to the decision solution and improve the modelling capabilities given that we still regularize the solution by dropout. The accuracy improved to become 98.46%. This only required adding 2 lines of code to the TensorFlow model.

The network is trained for 10 epochs with a batch size of 100. This gave a very good result as the accuracy increased from 86.41% after first epoch to 98.46 at the 17th epoch as shown in Figure 5.

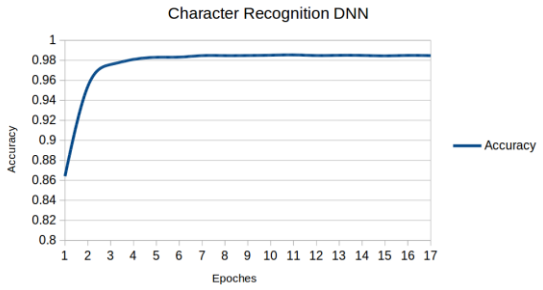

Figure 5 - OCR DNN loss value during training

We could eventually, keep improving our model by adding regularization, tweaking parameters, or applying more advanced CNN but the goal was to prove the concept that TensorFlow can easily be used to program a DNN model to quickly recognize the handwritten digits and convert it to clear digit (class number) with a small effort and high accuracy and without having to extract features. It shows the incredible value of DNN in coming up with model solution and proves the capability of TensorFlow as convenient and fast development framework for deep learning.

It is also important to point out that a convolutional neural network was not built yet for this task as this would be increase the computational complexity. The results are expected to be improved have we used CNN as we shall see in the face recognition case in the next section.

4.2 Face Recognition Results

The results of Face Recognition model validation accuracy are as shown in Figure 6. As can be seen in the chart, the validation accuracy started as 22% then quickly jumped to over 80% after 10 epochs. Then it reached 98.5% after 40 epochs and reached 100% correct classification for the test images after 80 iterations. The Early Stopping criterion was reached and the model stopped at epoch 88 rather than 100 since the loss did not change for 4 consecutive epochs.

CNNs are seen to be very helpful in face recognition problems that give the ability to extract important features of the object in an automated manner. Regardless of the small database that had variable pose, orientation, and zoom factor, the CNN DL network could perform very well. Keras enabled a much simpler implementation.

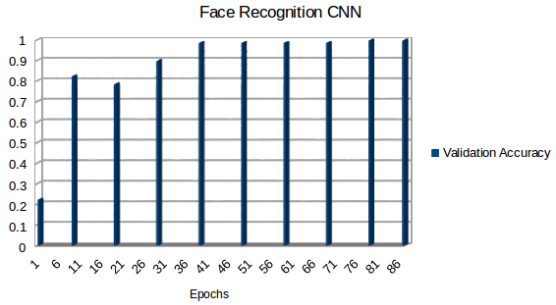

Figure 6 - Faces Recognition CNN Validation accuracy

5. CONCLUSION

In this paper, we present a new implementation for the automatic recognition of handwritten characters that implements deep neural networks with some regularization techniques, namely Dropout and batch normalization and apply it using TensorFlow framework. The additional regularization parameters lead to better outcomes with regard to the accuracy. The accuracy reached by DNN with Dropout in two consecutive layers is almost 1% higher than that when only dropout was used in one layer only. Adding another hidden layer of ReLu activation function also improved the performance by almost 1%. We also applied a CNN DL model on private small data of faces with different poses, and orientation with a perfect 100% correct classification on test data. The implementation was fast using Keras functions and it took only 90s for each epoch on a PC with Intel i5 processor, 4 GB RAM and no GPU. Automated software systems for the recognition of faces could apply such a classifier to record attendance for classes or work. Alternatively, it can be used as a security authentication measure. This utilization will shorten the time taken to record attendance at school with large number of students and improve efficiency, reliability and productivity.

It is important to mention that to get the most out of deep learning; one needs to apply it on huge datasets such as LFW, with truly deep architecture. This of course will require the powerful processing power of GPUs and yet training can take significant time. The results reported in this paper were

obtained using simple hardware specifications with no GPU at all as a starting phase. We are in the process of procurement of GPU workstations for his funded research workstation to speed up learning during experiments and to use different techniques in shorter time. We plan to implement deeper networks with state of the art techniques for regularization and optimization to apply computer vision models in industrial automation application.

6. ACKNOWLEDGMENT

This project is partially supported by a grant from Hamdi Mango Center for Scientific Research (HMCSR) at the University of Jordan, Amman, Jordan.

7. REFERENCES

[1] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

[2] A. Belaïd and N. Ouwayed, Segmentation of ancient Arabic documents, In Guide to OCR for Arabic Scripts, Ed. Volker Märgner and Haikal El Abed, Springer, 2011.

[3] G. Abandah, M. Khedher, and K. Younis, “Handwritten Arabic Character Recognition Using Multiple Classifiers based on Letter Form”. IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2008), Feb 13-15, 2008, pp. 128-133, Innsbruck, Austria.

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[5] TensorFlow™ https://www.tensorflow.org/.

[6] Keras website https://keras.io

[7] Hu, G.; Yang, Y.; Yi, D.; Kittler, J.; Christmas, W.; Li, S.Z.; Hospedales, T. When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 13–16 Dec 2015; pp. 142–150.

[8] A. Graves, J. Schmidhuber. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. Advances in Neural Information Processing Systems 22, NIPS'22, p 545-552, Vancouver, MIT Press, 2009.

[9] Jaderberg, Max, Andrea Vedaldi, and Andrew Zisserman. “Deep features for text spotting.” Computer Vision–ECCV 2014. Springer International Publishing, 2014. 512-528.

[10] Cireşan, Dan C., et al. “Handwritten digit recognition with a committee of deep neural nets on gpus.” arXiv preprint arXiv:1103.4487 (2011).

[11] S. Yousfi, S. Berrani, and C. Garcia, Deep Learning and Recurrent Connectionist-based Approaches for Arabic Text Recognition in Videos, The 13th International Conference on Document Analysis and Recognition (ICDAR 2015), At Nancy, France, Aug 2015.

[12] Mostafa Mehdipour Ghazi, Hazim Kemal Ekenel, A Comprehensive Analysis of Deep Learning Based Representation for Face Recognition; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2016, pp. 34-41

[13] Gary B. Huang, Marwan Mattar, Tamara Berg, Eric Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Workshop on Faces in ’Real-Life’ Images: Detection, Alignment, and Recognition, Oct 2008, Marseille, France. 2008.

[14] Hu, G.; Yang, Y.; Yi, D.; Kittler, J.; Christmas, W.; Li, S.Z.; Hospedales, T. When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 13–16 December 2015; pp. 142–150.

[15] http://yann.lecun.com/exdb/mnist/, MNIST, Yann LeCun,, New York University

[16] Orr, G.B. and Müller, K.-R. (Eds.): LNCS 1524, ISBN 978-3-540-65311-0 (1998).

[17] https://en.wikipedia.org/wiki/Cross_entropy, Cross entropy loss function.

[18] Tieleman, T. and Hinton, G. (2012), Lecture 6.5 - rmsprop, COURSERA: Neural Networks for Machine LearningDiederik P. Kingma, Jimmy Ba, https://arxiv.org/abs/1412.6980

[19] https://en.wikipedia.org/wiki/MNIST_database

[20] Ciresan, Claudiu Dan; Ueli Meier; Luca Maria Gambardella; Juergen Schmidhuber (December 2010). "Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition". Neural Computation. 22 (12). arXiv:1003.0358Freely accessible. doi:10.1162/NECO_a_00052.

[21] https://tinyurl.com/kug6spz. Martin Gorner, “Tensorflow and Deep Learning without PhD”