information

MDPI

Article

Arabic Handwritten Digit Recognition Based on Restricted Boltzmann Machine and Convolutional Neural Networks

Ali A. Alani

Department of Computer Science, College of Science, University of Diyala, Diyala 32001, Iraq;
alialani@sciences.uodiyala.edu.iq; Tel.: +96-077-219-87060

Received: 14 August 2017; Accepted: 8 November 2017; Published: 9 November 2017

Abstract: Handwritten digit recognition is an open problem in computer vision and pattern recognition, and solving this problem has elicited increasing interest. The main challenge of this problem is the design of an efficient method that can recognize the handwritten digits that are submitted by the user via digital devices. Numerous studies have been proposed in the past and in recent years to improve handwritten digit recognition in various languages. Research on handwritten digit recognition in Arabic is limited. At present, deep learning algorithms are extremely popular in computer vision and are used to solve and address important problems, such as image classification, natural language processing, and speech recognition, to provide computers with sensory capabilities that reach the ability of humans. In this study, we propose a new approach for Arabic handwritten digit recognition by use of restricted Boltzmann machine (RBM) and convolutional neural network (CNN) deep learning algorithms. In particular, we propose an Arabic handwritten digit recognition approach that works in two phases. First, we use the RBM, which is a deep learning technique that can extract highly useful features from raw data, and which has been utilized in several classification problems as a feature extraction technique in the feature extraction phase. Then, the extracted features are fed to an efficient CNN architecture with a deep supervised learning architecture for the training and testing process. In the experiment, we used the CMATERDB 3.3.1 Arabic handwritten digit dataset for training and testing the proposed method. Experimental results show that the proposed method significantly improves the accuracy rate, with accuracy reaching 98.59%. Finally, comparison of our results with those of other studies on the CMATERDB 3.3.1 Arabic handwritten digit dataset shows that our approach achieves the highest accuracy rate.

Keywords: handwritten digit recognition; Arabic digit; restricted Boltzmann machine; deep learning; convolutional neural network

1. Introduction

Handwritten digit recognition is a challenging problem in computer vision and pattern recognition; this problem has been studied intensively for many years, and numerous techniques and methods, such as K nearest neighbors (KNNs) [1], support vector machines (SVMs) [2], neural networks (NNs) [3], and convolutional NNs (CNNs) [2,4] have been proposed. Reasonable results have been obtained using datasets with different languages.

Arabic is the main language in the Middle East and the North of Africa, and is spoken widely in many other countries. Statistically, Arabic is one of the top five spoken languages in the present world [5,6]. Arabic numbers are widely important in the regions that write in Arabic. Digit handwritten recognition has received much attention recently because of its wide applications in different fields, such as criminal evidence, office computerization, check verification, and data entry applications. The wide use of those numbers makes the recognition process of these numbers an important field of interest [1]. However, most research has been focused on English digits related to the English language and some other European languages; apparently, English handwriting datasets are widely available, and significant results have been achieved [2,7]. By contrast, little work has been proposed for Arabic handwriting digit recognition due to the complexity of the Arabic language and the lack of public Arabic handwriting digit datasets. Arabic handwritten digit recognition suffers from many challenges, such as writing style, size, shape, and slant variations, as well as image noise, thereby leading to changes in numeral topology [8].To address these challenges, we consider a solution that focuses on the design of an efficient method that can recognize Arabic handwritten digits that are submitted by users via digital devices.

Three main techniques—namely, preprocessing, feature extraction, and classification [7]—are usually used to design an efficient method in pattern recognition. Preprocessing is used to enhance data quality and extract the relevant textual parts and prepare for the recognition process. The main objectives of preprocessing are dimensional reduction, feature extraction, and compression in the amount of information to be retained, among others [9]. The output of the preprocessing produces clean data that can be used directly and efficiently in the feature extraction stage. Meanwhile, feature extraction is the main key factor that affects the success of any recognition method. However, traditional hand-designed feature extraction techniques are tedious and time consuming, and cannot process raw images, in comparison to automatic feature extraction methods by which useful features can be retrieved directly from images. Szarvas, et al. [10] showed that the CNN–SVM combination exhibits good performance in pedestrian detection by use of the automatically optimized features learned by the CNN. Mori et al. [11] used the time domain encoding schemes by modules with different parts of images to train the convolutional spiking NN. In their method, the output of each layer is fed as features to the SVM and 100% face recognition accuracy is obtained on the 600 images of 20 people. Furthermore, the authors in [12] presented an automatic feature extraction method based on CNN. By using the trainable feature extractor plus affine distortions and elastic distortions, the proposed method obtains low error rates of 0.54% and 0.56% for the handwritten digit recognition problem. Therefore, the feature extraction techniques consider the most important steps to increase classification performance; several feature extraction methods are available in [13–18].

The final step in handwritten digit recognition application is image classification, which is a branch of computer vision, and has been extensively applied in many real-world contexts, such as handwriting image classification [1,19], facial recognition [20], remote sensing [21], and hyperspectral image [22]. Image classification aims to classify sets of images into specified classes. Two types of classification methods in computer vision—namely, the appearance-based method and the feature-based method—are used to classify images. The most commonly used method in literature is the feature-based method, which extracts features from the images and then uses these features directly to improve the classification results [23]. In recent years, finding an effective algorithm for feature extraction has become an important issue in object recognition and image classification. Recent developments in graphic processing unit (GPU) technology and artificial intelligence, such as deep learning algorithms, present promising results in image classification and feature extraction. Therefore, in this study, we emphasize the use of deep learning algorithms for the handwritten digit recognition context.

Deep learning algorithms comprise a subset of machine learning techniques that use multiple levels of distributed representations to learn high-level abstractions in data. At present, numerous traditional artificial intelligence problems, such as semantic parsing, transfer learning, and natural language processing [2,5,24], have been solved using deep learning algorithm techniques. The main properties of deep learning methods are that they learn the effect and perform high-level feature extraction by use of the deep architectures in an unsupervised manner without considering the label data [25]. To achieve this goal, layers of network are arranged hierarchically to form a deep architecture. Each layer in the network learns a new representation from its previous layer with the goal of modeling different explanatory factors of variation behind the data [26].

Deep learning algorithms, such as the restricted Boltzmann machine (RBM), involve a powerful feature learning technique using hierarchical deep architectures in an unsupervised manner. RBM is a generative algorithm with a high capability to extract discriminative features from complex datasets in an unsupervised manner, and has been applied in numerous learning domains, including text, speech, and images [27]. CNN is a multilayer NN that can be viewed as the combination of an automatic feature extractor and a trainable classifier. The past few years have borne witness to the increasing popularity of CNN in many different domains such as image classification [28,29] and object and face detection [20,30] over many benchmark datasets.

Numerous digit handwritten recognition methods based on different feature extraction and classifier techniques have been developed. In the last few years, the Latin digit recognition problem has been extensively researched, and a novel CNN–SVM model for handwritten Latin digit recognition was proposed in [2]. The proposed model uses the power of the CNN algorithm to extract the features from the images, and these features are fed to the SVM to generate the predictions. Furthermore, the authors in this work used non-saturating neurons with the efficient GPU implementation of the convolution operation to reduce overfitting in fully connected layers. Ouafae et al. [31] presented a new handwritten digit recognition system using characteristic loci (CL). In their method, each numeral image is divided into four portions, and then the CL is derived from each portion of the image. This work adopted two types of classifiers in the classification stage: multilayer perception and KNN classifiers. Nibaran Das et al. [5] presented a handwritten digit recognition technique using a novel method that utilizes an MLP in which a set of 88 features is used. The feature set is divided into 72 shadow features and 16 octant features. The authors in [4] proposed a CNN deep learning algorithm that uses an appropriate activation function and a regularization layer for Arabic handwritten digit recognition, thereby resulting in significantly improved accuracy compared to that of existing Arabic digit recognition methods. The authors in [32] proposed a handwritten digit recognition method using the perceptual shape decomposition (PSD) algorithm. The proposed approach represents the deformed digits with four salient visual primitives—namely, closure, smooth curve, protrusion, and straight segment—by defining a set of external symmetry axes. The primitives are derived using an efficient set of external symmetry axes based on parallel external chords. The performance of the proposed recognition system was evaluated on five-digit datasets that involve the CMATERDB 3.3.1 Arabic digit dataset. The recognition accuracy on Arabic CMATERDB 3.3.1 was found to be 97.96. Finally, the authors in [33] presented and compared the RBM model along with SVM and sparse RBM-SVM using the MNIST dataset, and the results were 96.9 and 97.5, respectively. The classification results showed the advantage of RBM models compared with other variants and that all RBM methods perform well in terms of classification accuracy.

The main challenges in the handwritten recognition process are the distortions and enormous variability patterns. Therefore, any successful recognition system and image classification require an active and accurate feature extraction technique that can provide distinct features that can be used to distinguish between different numeral handwritten images effectively. Furthermore, an accurate classifier is required to compute the exact distance between the feature vectors of the test and dataset numeral handwritten images. However, most previously proposed methods select only a small number of features as the input, and thus produce insufficient information for correctly predicting the object in the classification process. By contrast, a large number of input features will cause the generalization performance of the model to deteriorate, owing to the problems of dimensionality and increased run time for the training process. Hence, we propose the hybrid RBM–CNN model to address the aforementioned problems, and to introduce a novel method that uses strong feature extraction techniques. In our proposed method, we use the RBM deep learning algorithm, a popular feature extraction technique, to learn and extract features that are optimized and used for classification. Then, the extracted features are fed to the CNN after reshaping for classification. The performance of the proposed method is evaluated using the CMATERDB 3.3.1 Arabic handwritten digit dataset [32,34]. The rest of this article is structured as follows. Section 2 presents the proposed method and provides the basic concept for the used algorithms. Section 3 presents the analysis of the experiment results. Section 4 discusses the results of the proposed method with a relevant literature comparison. Section 5 elaborates the conclusion of the study with a summary.

2. The Proposed Method

In this section, the proposed method is described in detail, in which two deep learning algorithms are used for feature extraction and classification. First, features are extracted using the RBM deep learning algorithm. Then, the extracted futures are fed to the CNN deep learning algorithm for the classification. The two algorithms are described below. Figure 1 presents the block diagram of the proposed method.

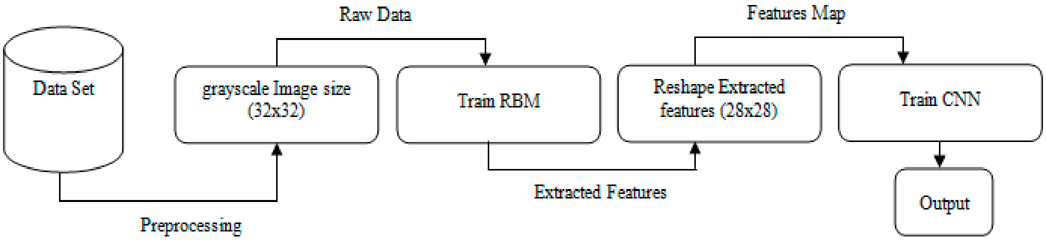


Figure 1. Data flow of the proposed method.

2.1. Restricted Boltzmann Machines

Previous research has used the RBM deep learning algorithm as a feature extraction method, as proposed by Hinton in [27]. RBMs present a high capability for feature extraction and representation; empirical research has proved that using the extracted features from the RBM algorithm instead of the raw data results in significant improvements in different machine learning applications, such as color image classification [35], speech and object recognition [36]. The RBM deep learning algorithm is designed to extract the discriminative features from large and complex datasets by introducing hidden units in an unsupervised manner. RBM is a probabilistic network that learns the probability distribution of its inputs v and a hidden representation h. Figure 2 illustrates the standard RBM algorithm with two layers [37]. The main advantage of the RBM algorithm is that all hidden and visible units are independent, meaning that no connections occur between units in the same layer.

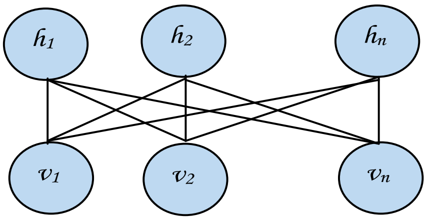


Figure 2. Restricted Boltzmann machine.

RBMs are built by executing a Monte Carlo Markov chain to converge and using the Gibbs sampling method as the transition operator of the chain. Furthermore, RBM can model correlations of the data by use of fast learning algorithms, such as contrastive divergence [38,39]. RBM is controlled by the set of weights and biases across its layers. We suppose that the RBM contains n and m visible and hidden units, the θ parameter contains weight matrix w (m × n), the visible layer bias is $a = a_1, a_2, \ldots, a_m$, and hidden layer bias is $b = b_1, b_2, \ldots, b_n$. Then, the three vectors will determine how the network will represent the input n dimension samples into m dimension features, and these can be defined as a long vector θ = (W, a, b), E(v, h); that is, the energy function. This energy function is defined by θ = (W, a, b); for a set of certain vectors (v, h), the energy function is defined as follows [38]:

$$E(v, h \,|\Theta) = \frac{1}{2}(v^T\,wh + b^T\,v + c^T\,h). \tag{1}$$

The partition function, also called the normalizing factor Z(Θ), is defined as

$$Z(w) = \sum\nolimits_x Exp[-E(x|\Theta)]. \tag{2}$$

The probability function is defined as

$$p(v, h\,|\Theta) = \frac{1}{Z(\Theta)} Exp\,\{-E(v, h\,|\Theta)\}. \tag{3}$$

The visible layer conditional probability is given as follows:

$$p(vi = 1|h) = sigm(bi + wi\,.h). \tag{4}$$

The hidden layer conditional probability is defined as

$$p(hj = 1|v) = sigm(cj + wj\,.v), \tag{5}$$

where sigm function is defined as

$$sigm\ (x) = \frac{1}{1 + exp\ (-x)}. \tag{6}$$

The objective of the RBM algorithm is to rebuild the inputs as accurately as possible. Throughout the forward stage, the input is changed on the basis of the weights and biases, and is then used to activate the hidden layer. In the next stage, the activations from the hidden layer are changed on the basis of the weights and biases and are then sent back to the input layer for activation. At this stage, the input layer seeks the modified activations as an input reconstruction and then uses this input to compare with the original input [40]. Therefore, in our proposed method, we use the advantages of RBM to extract useful features from raw data. The results are presented in Figure 3. The RBM model takes the entire input image with dimensions of 32 × 32 in a single one-dimensional array. The RBM was trained by unsupervised pre-training using contrastive divergence learning. Considering that RBMs only take a one-dimensional array as input, all two-dimensional matrices with pixel values were reshaped to one-dimensional arrays. We trained the RBM with 1024 visible input units (which correspond to all the pixels in a 32 × 32 picture) and 784 hidden output units (which correspond to all the elements in a 28 × 28 feature map). We used mini batches of size 200 with a fixed learning rate of 0.1 for 100 iterations. The corresponding reshaping was also performed on the output of the RBM; the one-dimensional output arrays are reshaped into two-dimensional matrices.

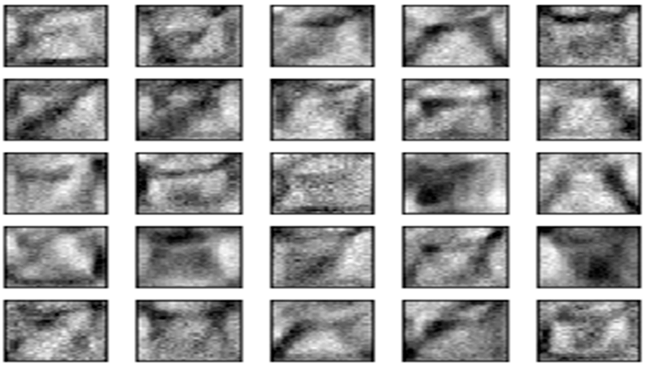


Figure 3. RBM feature map.

2.2. Convolutional Neural Network

As a particular deep learning technique, CNNs have attained success in image classification problems [41–43]. Three main types of layers are used to build CNN architectures; namely, convolutional layer, sub-sampling or pooling layer, and fully connected layer. Normally, a full CNN architecture is obtained by stacking several of the above-mentioned layers. The first layer is a convolutional layer with size [W × H × D], where W represents the width, H represents the height of the input images, and D denotes the dimension of input. In practice, W and H are typically equal (squared image) in image classification applications, and D represents the number of channels of input image (i.e., D = 3 for RGB images or D = 1 to represent black and white images). Each layer contains K filters (kernels) of size [F × F × Q], where F (the receptive field) should be less than W; these filters are of sizes such as 2 × 2 or 5 × 5, and Q in the first convolutional layer represents the number of channels of the input image. In other layers, Q equals the number of filters of the previous layer. In this layer, weights are shared across neurons, thereby leading the filter to learn frequent patterns that occur in any part of the image. Each filter is convolved with the input volume to produce a feature map with a size of W − F + 1; each convolutional layer produces a total of K feature maps of that size [44].

The second layer is a sub-sampling or pooling layer; a common practice in CNN network architecture is inserting a pooling (sub-sampling) layer between two successive convolutional layers. The objective of this layer is to progressively reduce the spatial size of the representation. Thus, such a process will reduce the number of parameters and computations that are required by the network and helps in the overfitting control. The pooling units can perform other functions, such as L2-norm or average pooling. The final layer is a fully connected layer. In this layer, neurons are connected with all activation units in the previous layer, and their activations are computed using a matrix multiplication. This process is followed by a bias offset. This type of layer is standard in a regular NN. This layer holds the net output, such as probability distributions over classes [45,46]. In practice, a parameter-sharing CNN significantly reduces the number of parameters, thereby making the CNN easier to train compared to the traditional fully connected NNs. In summary, a CNN consists of multiple trainable layers stacked on top of each other, followed by a supervised classifier. A set of arrays called feature maps represent the input and output of each stage. In our proposed CNN algorithm, we use the structure as detailed below. The first layer is a convolutional layer that contains 32 feature maps, each with a kernel size of 5 × 5 pixels and a ReLU activation function, which takes images with 32 × 32 pixel values. This layer represents our CNN input layer. Next, we define a MaxPooling layer that uses the maximum value; this layer is configured with a pool size of 2 × 2. The next layer is a regularization layer, which is also called Dropout. It is configured to randomly exclude 0.2 of neurons to reduce overfitting. The following hidden layer is another convolutional layer that also contains 32 feature maps, each with a kernel size of 3 × 3 pixels. Furthermore, this layer uses a ReLU activation function. This layer is followed by another pooling layer that is the same as the previous pooling layer. Afterward, we obtain a layer called Flatten, which converts the two-dimensional matrix data to a vector, thereby allowing the final output to be processed by standard fully connected layers to obtain our next layer. The first fully connected layer contains 128 neurons with the ReLU activation function. Finally, we end the CNN structure with the output layer, which contains 10 neurons for the 10 classes with a Softmax activation function to present the final classification result. Figure 4 represents our proposed CNN method, and Table 1 presents the parameters of the CNN method.

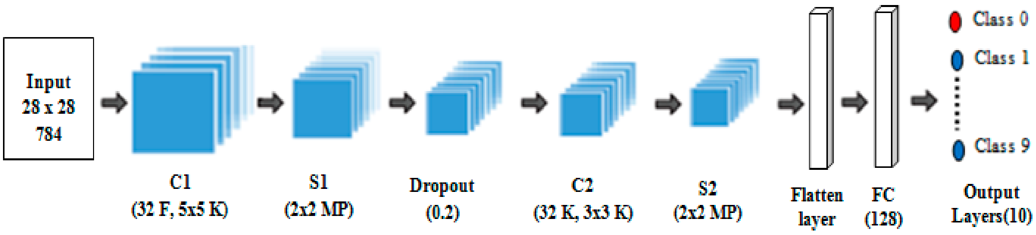


Figure 4. Convolutional neural network (C: convolutional layer, S: sub sampling layer, FC: fully connected layer, F: filters, K: kernels, MP: MaxPooling).

Table 1. CNN Parameters setup.

Layers	Layers Operation	Feature Maps No.	Feature Maps Size	Window SIZE	Parameters No.
C_1	Convolution	32	24 × 24	5 × 5	832
S_1	Max-pooling	32	12 × 12	2 × 2	0
D	Dropout layer	32	12 × 12	2 × 2	0
C_2	Convolution	32	10 × 10	3 × 3	9248
S_2	Max-pooling	32	5 × 5	2 × 2	0
FC	Flatten layer	800	N/A	N/A	0
FC	Fully connected	128	1 × 1	N/A	102,528
FC	Output layer	10	1 × 1	N/A	1290

3. Experimental Results

The proposed RBM–CNN method is trained and tested against the CMATERDB 3.3.1 Arabic handwritten digit dataset. RBM–CNN is also trained for 100 epochs with a batch training size of 70%. The Adam optimizer is used as the optimizing function. Experimental models are implemented in Python programming languages with Theano and Keras Library. Figure 5 shows the structure of our proposed model.

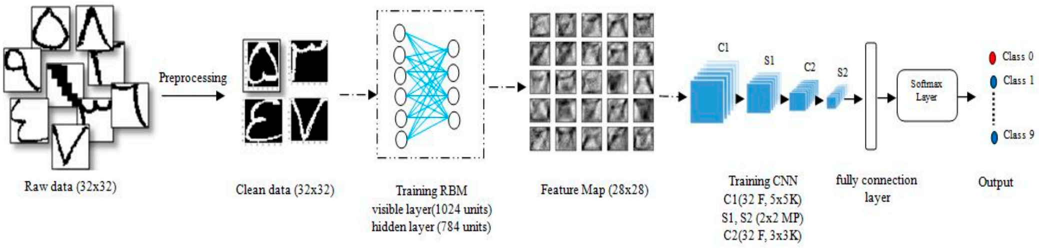


Figure 5. Final structure of the proposed RBM–CNN model (C: convolutional layer, S: sub sampling layer, FC: fully connected layer, F: filters, K: kernels, MP: MaxPooling).

3.1. Dataset Description

Our proposed method is trained and tested on the CMATERDB 3.3.1 Arabic handwritten digit dataset [34], see Table 2. The CMATERDB 3.3.1 dataset was developed by researchers at the Jadavpur University, and is collected from three different sources; namely, class notes of students from different age groups, handwritten manuscripts of popular magazines, and a preformatted data sheet especially designed for collection of handwriting samples [32]. These documents were digitized using HP F380 flatbed scanner at 300 dpi. Each digit contains 300 images of 32 × 32 pixels. A few sample images from the database are shown in Figure 5. No visible noise was found through visual inspection. However, variability in writing style was observed as a result of the high user dependency. We divided the dataset into 70% for the training and 30% for the testing process. The images were prepossessed to convert them into grayscale values. Then, the images were inverted to enhance their features. Furthermore, all the images were normalized to reduce the computation process.

Table 2. Dataset description.

	Dimension				No. of Image		
Dataset	**Classes**	**Width**	**Height**	**Depth**	**Dataset**	**Training**	**Test**
CMATERDB 3.3.1	10	32	32	1	3000	70%	30%

3.2. Evaluation Measures

In evaluating the proposed method against previously proposed methods, the Recall, Precision, and F_1 measure metrics are applied as benchmarks for performance. These metrics are applied to class (i.e., label to be predicted) of the dataset that contains 10 classes. The parameters used to compute the final evaluation are (1) True Positive (TP), which represents the total number of images that can be correctly labeled as belonging to a class x; (2) False Positive (FP), which represents the total number of images that have been incorrectly labeled as belonging to a class x; (3) False Negative (FN), which represents the total number of images that have been incorrectly labeled as not belonging to a class x; and finally, (4) True Negative (TN), which represents the total number of images that have been correctly labeled as not belonging to a class x.

1. Precision (P), also called the positive predictive value, is the fraction of images that are correctly classified over the total number of images classified.

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

2. Recall (R) is the fraction of correctly classified images over the total number of images that belong to class x.

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

3. F_1 combines Recall and Precision; the value of the F_1 measure becomes high if and only if the values of Precision and Recall are high (Table 3). The F_1 formula can be denoted as follows:

$$F_1 = \frac{2\ \text{Precision . Recall}}{\text{Precision} + \text{Recall}}. \tag{9}$$

Table 3. Computed values of Precision and Recall.

	Relevant	**Non-Relevant**
Retrieved	TP	FP
Not-Retrieved	FN	TN

4. Comparison Results and Discussion

RBM is a type of machine learning technique for learning features from training data. We used the training set from the CMATERDB 3.3.1 dataset to learn features from training data and test our model in testing data. In our experiment, we fed the features learned by RBM into the CNN deep learning algorithm, which worked as the feature extraction and classification method. The results of the proposed RBM–CNN method are shown in Table 4.

Table 4. Classification result on the CMATERDB 3.3.1 dataset using RBM-CNN.

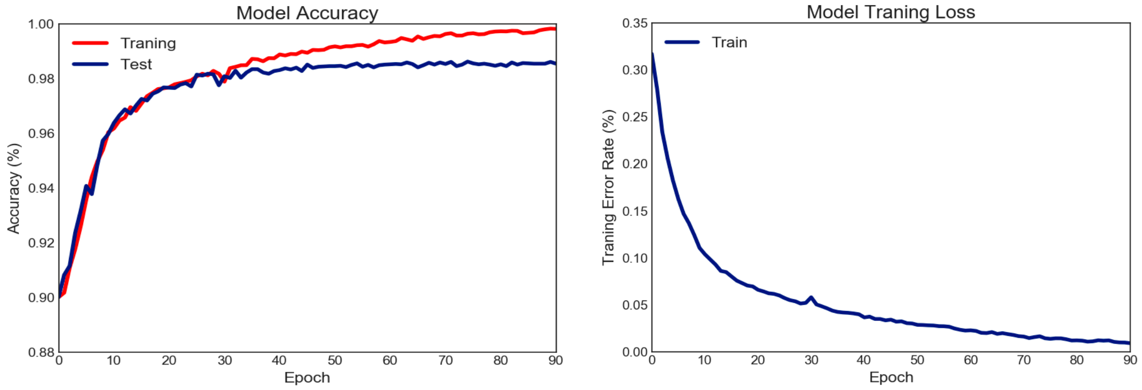
	Evaluation Measures			
Proposed Method	**Precision**	**Recall**	**F_1 Score**	**Accuracy**
RBM-CNN	0.98	0.98	0.98	98.59

Comparison results of the proposed method and state-of-the-art methods on the CMATERDB 3.3.1 dataset are listed in Table 5. We selected the best recognition results generated by related learning algorithms applied on the CMATERDB 3.3.1 training data. From the said table, the highest accuracy rates using the CMATERDB 3.3.1 dataset can be seen to be 97.4% using CNN [4]. However, a significant achievement is made by our proposed method using the RBM deep learning algorithm for feature extraction and the CNN deep learning algorithm for classification. Specifically, the highest accuracy rate of the proposed method reached 98.59%, which is higher than the results for the state-of-the-art methods. These findings demonstrate that feature extraction and dimension reduction via RBM can improve the generalization performance of CNN. As shown in Figure 6a, the proposed approach obtains the best recognition rate on the CMATERDB test set. Figure 6b presents the training error rate of our proposed model.

Table 5. Performance comparison of the proposed and related methods described in [4,33].

Author	Techniques	Accuracy
Ashiquzzaman and Tushar [4]	CNN	97.4
X. Guo et al. [33]	RBM-SVM	96.9
X. Guo et al. [33]	Sparse RBM-SVM	97.5
Our approach	RBM-CNN	98.59

CNN is a sequence of layers, and every layer of a CNN architecture transforms one volume of activations to another through a differential function. In our CNN architecture, we use three main types of layers—Convolutional Layer, Pooling Layer (Max-pooling), and Fully Connected Layer—to control the overfitting process. We added a dropout layer with 20%, and our proposed CNN architecture and parameters are described in Section 2.2. In Table 6, we also compare our proposed CNN architecture with the CNN architecture described in [4].



(**a**) (**b**)

Figure 6. (**a**) Accuracy on the CMATERDB 3.3.1 dataset; (**b**) Training error rates of RBM–CNN on the CMATERDB 3.3.1 dataset.

Table 6. Our CNN proposed architecture compare with CNN architecture proposed in [4].

Our CNN Proposed Architecture			**CNN Architecture as Proposed in [4]**		
Layers Operation	**Feature Maps No.**	**Window Size**	**Layers Operation**	**Feature Maps No.**	**Window Size**
Convolution	32	5 × 5	Convolution	30	5 × 5
Max-pooling	32	2 × 2	Max-pooling	30	2 × 2
Dropout layer	20%		Convolution	15	3 × 3
Convolution	32	3 × 3	Max-pooling	15	2 × 2
Max-pooling	32	2 × 2	Dropout layer	25%	
Flatten layer	800	N/A	Flatten layer	-	N/A
Fully connected	128	N/A	Fully connected Dropout layer	128 50%	N/A
Output layer	10	N/A	Output layer	10	N/A

The computational disadvantage of the RBM-SVM, Sparse RBM-SVM and CNN against the proposed RBM–CNN is that RBM–CNN uses the RBM deep learning algorithm in its first stage to detect the features of images in a highly robust manner. Then, the CNN model is trained over the extracted features, and can easily detect the image class accurately, thereby exhibiting superior performance to RBM-SVM, Sparse RBM-SVM and CNN in classic image recognition problems. In our experiment, we demonstrate that the performance of our proposed RBM–CNN method enhances the accuracy in digit recognition, with an accuracy of 98.59%, which is the highest recorded accuracy for the CMATERDB 3.3.1 Arabic handwritten digit dataset. The confusion matrix of RBM–CNN trained with 100 epochs is shown in Figure 7. The overall classification performance is highly promising.

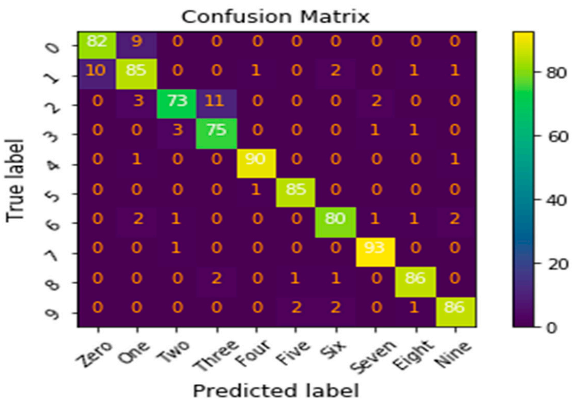


Figure 7. Confusion matrix of RNM–CNN on the CMATERDB dataset.

5. Conclusions

In this study, the RBM–CNN deep learning method is used to handle the problem of Arabic handwritten digit recognition. This method is applied to the CMATERDB 3.3.1 dataset. In our proposed model, the first algorithm is RBM that is used for feature extraction, and then the extracted features are fed into the CNN for the classification process. Experimental results show that the proposed method outperforms existing digit recognition methods with Arabic context in terms of accuracy rate. Our proposed method achieves 98.59% accuracy, which is higher than that of the methods discussed in [4,5,32,33]. This value is the highest recorded accuracy for the dataset used in the experiment. In future work, the performance of combination of different RBMs and CNNs on other benchmarking datasets should be fully explored with more than one RBM depend on the images size.

Acknowledgments: The author would like to thank Georgina Cosma and Taherkhani Aboozar, College of Science and Technology, Nottingham Trent University and Firas D. Ahmed, candidate in Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia for their advice and support.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. 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.
2. Niu, X.X.; Suen, C.Y. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognit.* **2012**, *45*, 1318–1325. [CrossRef]
3. Al-omari, F.A.; Al-jarrah, O. Handwritten Indian numerals recognition system using probabilistic neural networks. *Adv. Eng. Inform.* **2004**, *18*, 9–16. [CrossRef]
4. Ashiquzzaman, A.; Tushar, A.K. Handwritten Arabic Numeral Recognition using Deep Learning Neural Networks. In Proceedings of the 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition, Dhaka, Bangladesh, 13–14 February 2017; pp. 3–6.
5. 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.
6. Abdleazeem, S.; El-Sherif, E. Arabic handwritten digit recognition. *Int. J. Doc. Anal. Recognit.* **2008**, *11*, 127–141. [CrossRef]
7. Impedovo, S.; Mangini, F.M.; Barbuzzi, D. A novel prototype generation technique for handwriting digit recognition. *Pattern Recognit.* **2014**, *47*, 1002–1010. [CrossRef]
8. Mahmoud, S. Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models. *Signal Process.* **2008**, *88*, 844–857. [CrossRef]
9. Suliman, A.; Sulaiman, M.N.; Othman, M.; Wirza, R. Chain Coding and Pre Processing Stages of Handwritten Character Image File. *Electron. J. Comput. Sci. Inf. Technol.* **2010**, *2*, 6–13.
10. Szarvas, M.; Yoshizawa, A.; Yamamoto, M.; Ogata, J. Pedestrian Detection with Convolutional Neural Networks. In Proceedings of the 2005 IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; pp. 224–229.
11. Mori, K.; Suz, T. Face Recognition Using SVM Fed with Intermediate Output of CNN for Face Detection. In Proceedings of the IAPR Conference on Machine VIsion Applications, Tsukuba Science City, Japan, 16–18 May 2005; pp. 1–4.
12. Lauer, F.; Suen, C.Y.; Bloch, G. A trainable feature extractor for handwritten digit recognition. *Pattern Recognit.* **2007**, *40*, 1816–1824. [CrossRef]
13. Cruz, R.M.O.; Cavalcanti, G.D.C.; Ren, T.I. Handwritten Digit Recognition Using Multiple Feature Extraction Techniques and Classifier Ensemble. In Proceedings of the 17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010), Rio de Janeiro, Brazil, 17–19 June 2010; pp. 215–218.
14. Awaidah, S.M.; Mahmoud, S.A. A multiple feature/resolution scheme to Arabic (Indian) numerals recognition using hidden Markov models. *Signal Process. J.* **2009**, *89*, 1176–1184. [CrossRef]
15. Boukharouba, A.; Bennia, A. Novel feature extraction technique for the recognition of handwritten digits. *Appl. Comput. Inform.* **2017**, *13*, 19–26. [CrossRef]
16. Yang, J.; Zhang, D.; Member, S.; Frangi, A.F.; Yang, J. Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 131–137. [CrossRef] [PubMed]
17. Wshah, S.; Shi, Z.; Govindaraju, V. Segmentation of Arabic Handwriting Based on both Contour and Skeleton Segmentation. In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009.
18. Rajashekararadhya, S.V. Isolated Handwritten Kannada and Tamil Numeral Recognition: A Novel Approach. In Proceedings of the First International Conference on Emerging Trends in Engineering and Technology, Nagpur, Maharashtra, India, 16–18 July 2008; pp. 1192–1195.
19. Jackel, L.D.L.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Le Cun, B.; Denker, J.; Henderson, D. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990; pp. 396–404.
20. Tomoshenko, D.; Grishkin, V. Composite face detection method for automatic moderation of user avatars. In Proceedings of the International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, 27–28 March 2013.

21. Cheng, G.; Ma, C.; Zhou, P.; Yao, X.; Han, J. Scene Classification of High Resolution Remote Sensing Images Using Convolutional Neural Networks. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 767–770.
22. Cao, J.; Chen, Z.; Wang, B. Deep Convolutional Networks With Superpixel Segmentation for Hyperspectral Image Classification. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 3310–3313.
23. Chen, S.; Liu, G.; Wu, C.; Jiang, Z.; Chen, J. Image classification with stacked restricted boltzmann machines and evolutionary function array classification voter. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 4599–4606.
24. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [CrossRef]
25. Bengio, Y. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
26. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Member, S. Deep feature extraction and classification of hyperspectral images based on Convolutional Neural Networks. *IEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
27. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural net-works. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1–9.
29. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the 13th European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014.
30. Toshev, D.E.A.; Szegedy, C. Deep Neural Networks for Object Detection. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 1–9.
31. Melhaoui, O.E.L.; El Hitmy, M.; Lekha, F. Arabic Numerals Recognition based on an Improved Version of the Loci Characteristic. *Int. J. Comput. Appl.* **2011**, *24*, 36–41. [CrossRef]
32. Dash, K.S.; Puhan, N.B.; Panda, G. Unconstrained handwritten digit recognition using perceptual shape primitives. In *Pattern Analysis and Applications*; Springer: London, UK, 2016.
33. Guo, X.; Huang, H.; Zhang, J. Comparison of Different Variants of Restricted Boltzmann Machines. In Proceedings of the 2nd International Conference on Information Technology and Electronic Commerce (ICITEC 2014), Dalian, China, 20–21 December 2014; Volume 1, pp. 239–242.
34. Handwritten Arabic Numeral Database. Google Coe Archieve—Long-Term Storage for Google Code Project Hosting. Available online: https://code.google.com/archive/p/cmaterdb/downloads (accessed on 9 November 2017).
35. Larochelle, H.; Bengio, Y. Classification using discriminative restricted Boltzmann machines. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 536–543.
36. Li, M.; Miao, Z.; Ma, C. Feature Extraction with Convolutional Restricted Boltzmann Machine for Audio Classification. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition, Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 791–795.
37. Papa, J.P.; Rosa, G.H.; Marana, A.N.; Scheirer, W.; Cox, D.D. Model selection for Discriminative Restricted Boltzmann Machines through meta-heuristic techniques. *J. Comput. Sci.* **2015**, *9*, 14–18. [CrossRef]
38. Cai, X.; Hu, S.; Lin, X. Feature Extraction Using Restricted Boltzmann Machine for Stock Price Prediction. In Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, 25–27 May 2012; pp. 80–83.
39. Hinton, G.E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **2002**, *14*, 1771–1800. [CrossRef] [PubMed]
40. JXia, Y.; Li, X.; Liu, Y.X. Application of a New Restricted Boltzmann Machine to Radar Target Recognition. In Proceedings of the Progress in Electromagnetic Research Symposimum (PIERS), Shanghai, China, 8–11 August 2016; pp. 2195–2201.
41. Xiao, T.; Xu, Y.; Yang, K.; Zhang, J.; Peng, Y.; Zhang, Z. The Application of Two-level Attention Models in Deep Convolutional Neural Network for Fine-grained Image Classification. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 842–850.

42. Liu, S.; Deng, W. Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition, Kuala Lumpur, Malaysia, 3–6 November 2015.
43. Luong, T.X.; Kim, B.; Lee, S. Color Image Processing based on Nonnegative Matrix Factorization with Convolutional Neural Network. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 2130–2135.
44. Dao-Duc, C.; Xiaohui, H.; Morère, O. Maritime Vessel Images Classification Using Deep Convolutional Neural Networks. In Proceedings of the Sixth International Symposium on Information and Communication Technology—SoICT 2015, Hue City, Vietnam, 3–4 December 2015; pp. 1–6.
45. Scherer, D.; Andreas, M.; Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In Proceedings of the 20th International Conference on Artificial Neural Networks (ICANN), Thessaloniki, Greece, 15–18 September 2010.
46. Spanhol, F.A.; Oliveira, L.S.; Petitjean, C.; Heutte, L. Breast Cancer histopathological Image Classification using Convolutional Neural Networks. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 2560–2567.