Convolutional Feature Learning and CNN Based HMM for Arabic Handwriting Recognition

Mustapha Amrouch[1(✉)], Mouhcine Rabi[2], and Youssef Es-Saady[2]

[1] Laboratory IRF-SIC, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
m.amrouch@uiz.ac.ma
[2] Laboratory IRF-SIC, Ibn Zohr University, Agadir, Morocco
y.essaady@uiz.ac.ma, mouhcineh@gmail.com

Abstract. In this paper, we present a model CNN based HMM for Arabic handwriting word recognition. The HMM have proved a powerful to model the dynamics of handwriting. Meanwhile, the CNN have achieved impressive performance in many computer vision tasks, including handwritten characters recognition. In this model, the trainable classifier of CNN is replacing by the HMM classifier. CNN works as a generic feature extractor and HMM performs as a recognizer. The suggested system outperforms a basic HMM based on handcrafted features. Experiments have been conducted on the well-known IFN/ENIT database. The results obtained show the robustness of the proposed approach.

Keywords: Handwriting recognition · Convolutional Neural Networks · Hidden Markov Models

1 Introduction

The tremendous increase in the power of computers and the availability of large data sources has offered new opportunities of computer vision [1–3]. Indeed, handwritten recognition (HWR) [4–6] has been an active area of research for many decades. It has been successfully applied in many applications such as postal codes, mail sorting, bank check reading, book and handwritten notes transcription, document analysis and retrieval and many other related tasks. etc. Offline handwritten recognition still remains a major challenge and can not yet be considered as solved despite considerable efforts that have led to important progress made recently in some associated applications aforementioned. This difficulty stems from the large pattern variations under which a recognition system must operate. Though high recognition rates are achieved in characters recognition, offline text recognition is not easy according to many factors including the large variability of writing scripts types, variability of handwriting styles between people, confusion between some similar characters, image degradation and noise, the cursive nature of handwriting, size of vocabularies, etc. In this sense, currently several recognition systems exist, which employ different approaches and algorithms for achieving such tasks with promising result and high accuracy. The most powerful recognition systems approaches that have been used include Hidden Markov Models (HMM) [6], Neural Networks (NN) [1, 2, 7] and Support Vector Machines (SVM) [8].

A. Mansouri et al. (Eds.): ICISP 2018, LNCS 10884, pp. 265–274, 2018.
https://doi.org/10.1007/978-3-319-94211-7_29

In view of this, for example, recent results from one impressive research work of Ciresan et al. [9] on the MNIST database, report a further increase in the classification accuracy of handwritten digit to 99.65%, which surpasses the human equivalent recognition rate of 96.1%. The proposed approach developed by using a Deep Neural Network.

Nowadays, the state of art depends on the kind of the studied script. In the case of Latin handwritten recognition, several works have been devoted [10, 11], as a result the Latin HWR approaches seem mature enough to achieve the high accuracies. However, Arabic HWR remains an unresolved problem. Despite the numerous of proposed systems recently in the literature [5, 12–17] including some achieved a significant rate of recognition. The best existing Arabic handwritten recognizers cannot yield satisfactory performance for practical applications. Compared to Latin script, Arabic HWR is a much harder problem because of Arabic script characteristics, especially its cursive nature, for further details a succinctly studies have been presented in the work of Parvez and Mahmoud [18].

This work is the continuity of our previous work developed by Rabi et al. [19], which based on HMM and handcrafted features, and takes into consideration the context of character using a relevant technique of cross learning. Thereby, our goal is to explore the impact of a deep learning in Arabic handwritten recognition, essentially improving the performance of the baseline HMM system. In this context, inspired by the works of Bluche et al. [20, 21], we opted to use a model CNN based HMM in Tandem mode. The obtained features are used thereafter in input for standard HMM. Furthermore, we investigate a powerful CNN for extracting features from the images of Arabic words by comparing two strategies. On the one hand, handcrafted features HMM, on the other hand, CNN-features-HMM. We evaluate the performance of the proposed model on the publicly available IFN/ENIT database. Experimental studies reveal that the suggested model CNN based HMM shows satisfactory classification accuracy and outperformed our previous baseline HMM system and some other exiting methods.

The rest of this paper is organized as follows: The principles of CNN model and a brief overview over our proposed approach are described in Sect. 2. Experimental results are given and analyzed in Sect. 3. Finally, Conclusions and perspectives are drawn in Sect. 4.

2 Method

2.1 The Principle of Model CNN

As illustrated in “Fig. 1”, the architecture represents a CNN adopted for the handwritten recognition inspired from LeNet5 [22]. It incorporates a set of previous layers. At the beginning, the input is processed by a convolutional layer which convolves it with a set of learnable filters or weights, each producing one feature map. Subsequently, the pooling layer (sub-sampling) is used to progressively diminish the dimensionality of the spatial size of the feature map by averaging the features in the neighborhood or pooling for a maximum value. In order, to reduce the amount of parameters and computation in the network. Each convolution layer is pursued by a sub-sampling layer. Successive alternation among convolutional and pooling layers constitutes the feature extractor to retrieves discriminating features from the raw images. Fully connected layers are used at the end of the network for the high-level reasoning after feature extraction and consolidation has been performed by the convolutional and pooling layers. They are used to create final non-linear activation combinations or a softmax of features and for making predictions by the network. Further details on architecture of CNN will be described in Sect. (2.3).

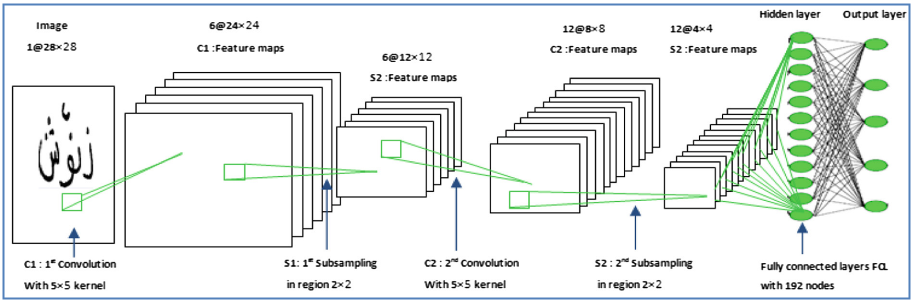

Fig. 1. A typical CNN architecture for proposed CNN for offline Arabic handwriting recognition

2.2 HMM Modeling

The problem of recognizing the Arabic words can be viewed as characters sequence recognition. Let *I* an Arabic word image which contains a set of characters. The modeling of the whole word image is obtained by the concatenation of the sequence of characters arranged horizontally. Each word can be segmented implicitly on units (characters or graphemes). We deal these units as being observed sequentially from a Markov model that pass through states $\mathrm{S} = \mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{k}$. That justifies the use of HMM. A sequence of length T is denoted as $O = o_1,\ o_2\ \ldots\ o_T$, in which o_i corresponds to the ith units. Define $Y = y_1, y_2, \ldots y_L$ as the label of the image. L is the number of units in the image, y_i is the th unit's label. In this study, The used approach is analytical and based on character modelling by HMM. In total, 167 character models HMMs are built [23]. The model architecture $\lambda = \langle \Pi | A | B \rangle$ of a character is right-left topology, where λ represents the HMM. The key parameters of λ are the initial state probability distribution $\pi = \mathrm{p}(q_0 = s_i)$, the transition probabilities $a_{ij} = \mathrm{p}(q_t = s_j | q_{t1} = s_i)$, and a model to estimate the observation probabilities $\mathrm{p}(\mathrm{o}_\mathrm{t} | s_i)$. There is no specific theory to set the number of hidden states in character model, often the solution is empirical. Word model is built by concatenating the appropriate character models.

2.3 The CNN-HMM Model Architecture

The overview of our proposed CNN based-HMM model for offline Arabic handwriting recognition is shown in "Fig. 2", the system was developed to integrate the CNN and the HMM classifiers. We use HMM to model the dynamics of Arabic handwriting and CNN is employed to extract salient features. Our purpose is to improve the performance of our HMM baseline system, though replacing features hand crafted with the CNN features. As illustrated in the diagram “Fig. 2”, the normalized input images are provided to the first convolutional layer and the designed CNN is trained by stochastic gradient descent (SGD) with momentum [24]. Our HMM baseline is trained by a new features vector obtained from the outputs of the hidden layer (FCL). Once the HMM classifier has been well trained, it performs the recognition task and makes new decisions on testing images with such automatically extracted features.

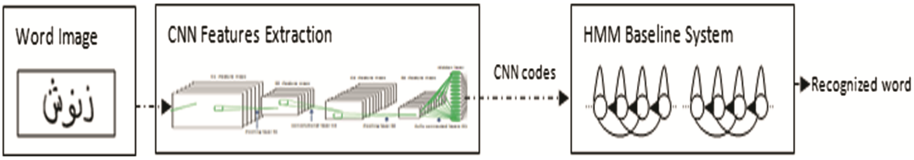

Fig. 2. Structure of the CNN based-HMM model

Instead of using complicated architectures such as AlexNet [2], OverFeat [25], GoogLeNet [26], VGGNet [27], ResNet [28]. Our CNN architecture is similar to LeNet-5 [22] with some modifications (without the second fully connected layer). The adopted structure comprises two convolutional layers with 5 × 5 receptive fields (i.e., kernel) and two sub-sampling layers over non-overlapping regions of size 2 × 2 with fully connected and output layers. In the following, convolutional layers are labeled C_i, sub- sampling layers are labelled S_i, where i is the layer index.

The first convolution layer (C_1) has 6 feature maps with 784 nodes/neurons each(28 × 28 pixels image), each is obtained by applying a distinct kernel of 5 × 5, that contains 25 weights, and a bias. So that it can extract different types of local features. The use of this kernel converts 28 spatial dimension to 24 (i.e., $28 - 5 + 1$) spatial dimension. Therefore, each 1_{st} level feature map size is 24 × 24. Each feature map has different set of weights. All the nodes in a feature map share the same set of weights and so they are activated by the same features at different locations. This weight sharing not only provides invariance to local shift in feature position but also reduces the number of trainable parameters at each layer. This local receptive field can extract the visual features such as oriented edges, end- points, corners of the images. Obtained results using (C_1) are illustrated in “Fig. 3”.

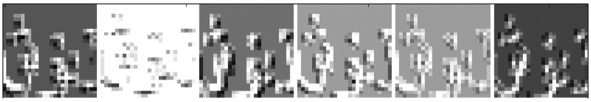

Fig. 3. Visualization of convolutional layer C1 (6 maps produce by using 6 distinct kernels).

In 1st sub-sampling/pooling layer (S_1), the 1st level feature maps are down-sampled from 24 × 24 into 12 × 12 feature maps by applying the max pooling method that checks for the maximum value on its local receptive field, multiplies it by a trainable coefficient, adds a trainable bias and passing through an activation function for generating the output. More formally it can be shown as follows in (1):

$$x_j^l = f\left(\omega_j^l sub\left(x_j^{l-1}\right) + b_j^l\right) \tag{1}$$

where sub(.) represents a sub-sampling function through local region; and are multiplicative coefficient and additive bias, respectively. In this study, we use $s(x) = M(x)$ and a non overlapping scheme (i.e., stride = 2) and a 2 × 2 region, so the output image becomes 2-times smaller of the convolution layer. In addition, this sub-sampling operation reduces both the spatial resolution of the feature map and sensitivity to shift and distortions. “Figure 4”, shows the results obtained by (S_1).

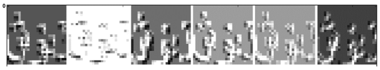

Fig. 4. Visualization of sub-sampling layer S1(6 maps 2-times smaller of the C1).

In the same way, the following layers (C_2 *and* S_2) have the same utility as previous layers (C_1 *and* S_1). When training this architecture, the feature maps generated from (S_2) are merged into a feature vector feeds into the fully connected layers. It means that these 12 feature maps values are considered as 192 (=12 × 4 × 4) distinct nodes those are fully connected to 946 units (the output nodes) represents the size of vocabulary of the IFN/ENIT dataset (946 town/village names). As in classical feed-forward neural networks, in our CNN, we introduce the non linearity by applying the non-linear function ReLU as in (2):

$$f(x) = \max(0; x) \tag{2}$$

The choice of ReLU instead other non-linearities functions is justified by the work of the Nair and Hinton [29]. As mentioned before for training our CNN, we follow recommended training techniques in CNN literature [30, 31]. These aim to minimize the cross-entropy loss between the desired and actual output. Thereafter, we use our pre-trained model on the IFN/ENIT dataset as generic feature extractor. This is done by removing the top output layer and using the activations from the last fully connected layer (CNN codes) as features. These features are used as training input for our previous HMM baseline system. The words models HMMs λ_w training is exactly the arduous task of a recognition system. The CNN features obtained from each image words using the pre-trained CNN, are considered as sequences of observations. We seek to deduce the model that generated them. Once the topologies of the models λ_w were chosen, details of the this procedure are explained in *section B* above, training allows to re-estimate the parameters of each word model HMM λ_w (the probabilities of input, transitions and emissions) which allows to modeling the samples of the dataset. To do this, technically, we determine the parameters of $\lambda w = \langle \Pi_w | A_w | B_w \rangle$ that maximize the likelihood $P(O/\lambda_w)$ of the observations sequence $O = \{o1, o2, \ldots on\}$. The training is performed with Baum-welch algorithm [32] under maximum-likelihood (ML) criterion until the likelihood converges. The best found HMM of each word is saved. Then, all resulting models consisting are the reference models of our system. After the learning phase, Recognition of a word image is performed by maximum a posteriori (MAP) estimation. Given an observation sequence O, we want to find the label sequence S, that satisfies S = argmax$_s$ log P(S/O). We use Viterbi algorithm [33] to get the most probable state sequence. It allows decoding the best state sequence candidates based on a criterion of maximum likelihood. Practically, it takes the word to be recognized as a sequence of observations $O = \{o_1, o_2, \ldots o_n\}$ extracted from the image and determines the sequence of states $S = \{S_1, S_2 \ldots S_n\}$ that has the maximum probability of generating O.

3 Experiments and Results

This section describes the details of our experiments. On the one hand we used the KERAS [34] tool with TensorFlow backend, which is an open source of deep learning written in python, for implementing our CNN. On the other hand, we have used the toolbox HTK [35] to realize our baseline HMM system. All experiments are conducted on a regular PC (2.7 GHz 4-core CPU, 4G RAM and Windows 64-bit OS). To validate the proposed model CNN based HMM, we use the IFN/ENIT database that consists of 946 handwritten Tunisian city names and their corresponding postcodes. Our CNN was trained on this dataset. We split 10% of the training set as validation set. the feed-forward net is trained under cross entropy objective by stochastic gradient descent (SGD) with momentum until convergence (Stability of the error).We use this optimization method with a momentum set to 0.9, a mini-batch size of 50 and the base learning rate was initialized for all trainable parameters at 0.01, and we adjust it manually during the training process, by dividing it by 10 when the validation set performance stops improving. We decrease the learning rates 3 times before stopping the training process, which is terminated at epochs 1000. Several experiments were performed to evaluate the recognition rate of our system according to the test scenarios named “abc-d”, “abcd-e” of IFN/ENIT database. Indeed, the first test that have been done were on scenario “abc-d” (see Table 1 below).

Table 1. Recognition rate on scenario acb-d

System	Models	RR* %
HMM baseline	HMM	87.93
Present work	**CNN-HMM**	**88.95**

*Recognition Rate

The experimental results shown in Table 2, demonstrate that our proposed approach model CNN base HMM outperforms our baseline HMM system. We achieve a rate of 88.95%, involving an increase in the accuracy by 1.02% that confirms the reliability of the suggested improvements. To this error rate, we suggest increasing the size of training data, this hypothesis is based on the idea that a good learning is done on a large data which represents the studied problem. To validate this assumption, we have conducted a second test on scenario ("abcd-e"), four subsets (abcd) are used for training and validation and another one (e) for testing, obtained results given in Table 2, below:

Table 2. Recognition rate on scenario acbd-e

System	Models	RR* %
HMM baseline	HMM	–
Present work	**CNN-HMM**	**89.23**

Finally, the most important gain in recognition rate is the order of 1,3% in Table 2. These tests have shown that CNN based HMM are more effective. The results obtained on the scenario "abcd-e" are more promising than those obtained on the scenario "abc-d", this confirms our hypothesis. A comparative study of the performance of our model was also performed with other results of different approaches published on the same database. Our results are important compared to accuracies achieved by others system on the different scenarios abc-d and abcd-e (See Tables 3 and 4).

Table 3. The comparative results on scenario abc-d

System	Models	RR*%
Ahmad et al. [12]	Contextual sub character	85.12
Alkhateeb et al. [36]	HMM+ re-ranking	83.55
Maqqor [37]	Multiple classifier	76.54
El Moubtahij et al. [38]	HMM	78.95
Jayech et al. [39]	DBN	82.00
Rabi et al. [19]	**HMM**	87.93
Present work	**CNN based HMM**	**88.95**

Table 4. The comparative results on scenario abcd-e

System	Models	RR*%
Giménez et al. [40]	windowed Bernoulli HMMs	84
Hamdani et al. [41]	Multiple HMM	81.93
Kessentini et al. [42]	Multi-stream HMM	79.60
Jayech et al. [39]	DBN	78.5
Present work	**CNN based HMM**	**89.23**

As it can be noted from Tables 3 and 4, most of the previous systems are based on HMM and hand-crafted features-based approach. However, our suggested model CNN based-HMM instead to use hand engineered features, it extract automatically and directly the relevant features from the image of word. In addition, as shown in the Tables (3 and 4) our system outperforms the results obtained with other current methods a significant achievement was made with the recognition rate of 88.95% on the scenario "abc-d" and 89.23% on the scenario "abcd-e". This prove the effectiveness of CNN model, specially its ability to generate a salient features directly from word. In fact, CNN, with automatic feature extractor stage, deduces features that differentiate between words, and then HMM classifier insists on predicting the correct class of word. These learned features, being more robust than computed hand-crafted features, establish an adequate representation for words.

4 Conclusion and Perspectives

In this work, a model CNN based HMM has been presented to solve the Arabic handwritten word recognition problem. This combination took the CNN as an automatic feature extractor and HMM as recognizer. That allows operating directly on the images and extracting relevant characteristics without much emphasis on feature extraction and pre- processing stages. We showed that this model gives promising results on IFN/ENIT which significantly outperforms our previous HMM baseline system based on hand-engineered features. Contrary to our previous work that based on features hand-crafted which is a laborious and time consuming task, the most important advantage of this fusion is the ability to extract automatically salient features directly from raw pixels. As future work, extracted CNN features will be processed by an enhancing HMM using statistical language models that are incorporated as a post-processing into the process of recognition.

References

1. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. In: ICLR (2014)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25 (NIPS 2012)
3. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, S., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Li, F.F.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
4. Hamdani, M., Doetsch, P., Kozielski, M., El-Desoky Mousaand, A., Ney, H.: The RWTH large vocabulary arabic handwriting recognition system. In: 11th IAPR International Workshop on Document Analysis Systems, pp. 111–115 (2014)
5. Graves, A.: Supervised Sequence Labeling with Recurrent Neural Networks. Studies in Computational Intelligence. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2
6. Plotz, T., Fink, G.A.: Markov models for offline handwriting recognition: a survey. Int. J. Doc. Anal. Recogn. **12**(4), 269–298 (2009)
7. Espana-Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., Zamora-Martınez, F.: Improving offline handwritten text recognition with Hybrid HMM/ANN models. IEEE Trans. Pattern Anal. Mach. Intell. **33**(4), 767–779 (2011)
8. Elleuch, M., Maalej, R., Kherallah, M.: A new design based-SVM of the CNN classifier architecture with dropout for offline arabic handwritten recognition. Procedia Comput. Sci. **80**, 1712–1723 (2016). Online publication date 1 Jan 2016
9. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep big simple neural nets excel on handwritten digit recognition CoRR abs/1003.0358 (2010)

10. Kozielski, M., Doetsch, P., Ney, H.: Improvements in RWTH's system for off-line handwriting recognition. In: International Conference on Document Analysis and Recognition (ICDAR2013), pp. 935–939 (2013)
11. Xue, H., Govindaraju, V.: Hidden Markov models combining discrete symbols and continuous attributes in handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 458–462 (2006)
12. Ahmad, I., Fink, G., Mahmoud, S., et al.: Improvements in sub-character hmm model based arabic text recognition. In: 2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 537–542. IEEE (2014)
13. Alabodi, J., Li, X.: An effective approach to offline Arabic handwriting recognition. Int. J. Artif. Intell. Appl. **4**(6), 1 (2013)
14. Azeem, S.A., Ahmed, H.: Effective technique for the recognition of offline arabic handwritten words using hidden markov models. Int. J. Docum. Anal. Recogn. (IJDAR) **16**(4), 399–412 (2013)
15. El Abed, H., Margner, V.: ICDAR 2009-Arabic handwriting recognition competition. Int. J. Docum. Anal. Recogn. (IJDAR) **14**(1), 3–13 (2011)
16. Pechwitz, M., Maddouri, S.S., Märgner, V., Ellouze, N., Amiri, H.: IFN/ENIT-database of handwritten Arabic words. In: The 7th CIFED 2002, Hammamet, Tunis, 21–23 October 2002 (2002)
17. Lawgali, A., Angelova, M., Bouridane, A.: A framework for arabic handwritten recognition based on segmentation. Int. J. Hybrid Inf. Technol. **7**(5), 413–428 (2014)
18. Parvez, M.T., Mahmoud, S.A.: Offline Arabic handwritten text recognition: a survey. ACM Comput. Surv. **45**(2), 23–35 (2013)
19. Rabi, M., Amrouch, M., Mahani, Z., Mammass, D.: Recognition of cursive Arabic handwritten text using embedded training based on HMMs. In: International Conference on Engineering & MIS (ICEMIS), September 2016, INSPEC Accession Number: 16467172. IEEE. https://doi.org/10.1109/icemis.2016.7745330
20. Bluche, T., Ney, H., Kermorvant, C.: Tandem HMM with convolutional neural network for handwritten word recognition. In: 38th International Conference on Acoustics Speech and Signal Processing (ICASSP2013), pp. 2390–2394 (2013)
21. Bluche, T., Ney, H., Kermorvant, C.: Feature extraction with convolutional neural networks for handwritten word recognition. In: 12th International Conference on Document Analysis and Recognition (ICDAR2013) (2013)
22. Le Cun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: International Symposium on Circuits and Systems, pp. 253–256, May 2010
23. Mohamad, R.A.-H., Likforman-Sulem, L., Mokbel, C.: Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. In: IEEE PAMI, vol. 31, no. 7, pp. 1165–1177 (2009)
24. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. JMLR W&CP **28**(3), 1139–1147 (2013)
25. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Le-Cun, Y.: OverFeat: integrated recognition, localization and detection using convolutional network. CoRR (2013)
26. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR abs/1409.4842 (2014)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv technical report (2014)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). https://arxiv.org/abs/1512.03385

29. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp. 807–814 (2010)
30. Le Cun, Y., Bottou, L., Bengio, Y.: Reading checks with multilayer graph transformer networks. In: International Conference on Acoustics, Speech, and Signal Processing (1997)
31. Bengio, Y., Boulanger-Lewandowski, N., Pascanu, R.: Advances in optimizing recurrent networks. CoRR abs/1212.0901 (2012). http://arxiv.org/abs/1212.0901
32. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Stat. **41**(1), 164–171 (1970). https://doi.org/10.1214/aoms/1177697196
33. Forney Jr., G.D.: The Viterbi algorithm. Proc. IEEE **61**(3), 268–278 (1973). https://doi.org/10.1109/proc.1973.9030
34. Keras (2016). https://github.com/fchollet/keras
35. Young, S., et al.: The HTK Book V3.4. Cambridge University Press, Cambridge (2006)
36. Alkhateeb, J.H., Ren, J., Jiang, J., Al-Muhtaseb, H.: Offline handwritten Arabic cursive text recognition using hidden markov models and re-ranking. Pattern Recogn. Lett. **32**, 1081–1088 (2011)
37. Maqqor, A. Halli, A., Satori, K., Tairi, H.: Off-line recognition handwriting combination of mutiple classifiers, In: 3rd International IEEE Colloquium on Information Science and Technology, IEEE CIST 2014, October 2014
38. El Moubtahij, H., Akram, H., Satori, K.: Using features of local densities, statistics and HMM toolkit (HTK) for offline Arabic handwriting text recognition (2016)
39. Jayech, K., Mahjoub, M.A., Amara, N.E.: Arabic handwritten word recognition based on dynamic bayesian network (2016)
40. Giménez, A., Khoury, I., Andrés-Ferrer, J., Juan, A.: Handwriting word recognition using windowed bernoulli HMMs. Pattern Recogn. Lett **35**, 149–156 (2012). Article in Press
41. Hamdani, M., El Abed, H., Kherallah, M., Alimi, A.: Combining multiple HMMs using online and off-line features for off-line Arabic handwriting recognition. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 201–205 (2009)
42. Kessentini, Y., Paquet, T., Ben Hamadou, A.: Off-line handwritten word recognition using multistream hidden Markov models. Pattern Recogn. Lett. **31**, 60–70 (2010)