Keyword spotting with Convolutional Deep Belief Networks and Dynamic Time Warping

Baptiste Wicht[1,2], Andreas Fischer[1,2], and Jean Hennebert[1,2]

[1] University of Applied Science of Western Switerzland
[2] University of Fribourg, Switzerland

Abstract. To spot keywords on handwritten documents, we present a hybrid keyword spotting system, based on features extracted with Convolutional Deep Belief Networks and using Dynamic Time Warping for word scoring. Features are learned from word images, in an unsupervised manner, using a sliding window to extract horizontal patches. For two single writer historical data sets, it is shown that the proposed learned feature extractor outperforms two standard sets of features.

1 Introduction

Although it has been the subject of research for decades, handwriting recognition remains a widely unsolved problem [23]. For large vocabularies, different writing styles and degraded documents, the accuracy of automatic transcription is not perfect. Under these conditions, keyword spotting solutions have been suggested instead of a complete transcription for spotting words in document images [13].

Keyword spotting solutions fall in two categories. *Template-based* methods match a query word image with labeled keyword template images. This approach has the advantage that it is rather easy to gather template images and it is not necessary to know the underlying language or its alphabet. However, for each keyword that is to be spotted, at least one template image is necessary. Furthermore, such systems typically do not generalize well to unknown writing styles. Such systems have been applied to speech [16, 21], poorly printed documents [1, 10] and handwritten text [14]. Many features have been proposed for keyword spotting with Dynamic Time Warping (DTW) and a sliding window [18], such as word profiles [19] and local gradients features [20].

On the other hand, *learning-based* systems are using statistical learning to train a model to score query images. Hidden Markov Model (HMM) were first used for keyword spotting at character level with template images [5]. Similar solutions were developed at word level using local gradient features [2]. Although trained word models are expected to exhibit better generalization than template-based methods, they still need a large amount of training templates. Moreover, such systems are not able to spot out-of-vocabulary keywords. Recently, a lexicon-free approach using character HMMs has been proposed [3], as well as character models based on Recurrent Neural Networks [6].

Both categories are relying on features extracted from the images. Such features are generally handcrafted and optimizing them is often non-trivial. In

Artificial Neural Networks and Machine Learning - ICANN 2016: 25th International Conference on Artificial Neural Networks, pp. 113-120

recent years, the emergence of *Deep Learning* has shown that it was possible to learn features directly from pixels. While Restricted Boltzmann Machines (RBM) have originally been used to initialize the weights of a neural network in an unsupervised manner [8], they also have been extensively used to extract features from a dataset [9]. RBMs can also be stacked into Deep Belief Networks (DBN) to extract multi-layer features [12, 24]. Convolutional RBMs have proved especially successful to extract features from images [12, 25].

In the present paper, we propose a hybrid word spotting system for handwritten text, based on Convolutional Deep Belief Networks and Dynamic Time Warping. While this system is essentially *template-based*, it has the advantage that features are automatically extracted from the images using unsupervised learning, making use of unlabeled handwriting images which are abundantly available. When compared with learning-based approaches, the proposed method has the advantage that no labeled images are needed. However, it requires a segmentation of images into words, which can be prone to errors.

The proposed system has been tested on two well-known benchmark data sets for keyword spotting, namely the George Washington and Parzival data sets. Our features are compared with two benchmark feature sets [15, 20].

2 Keyword Spotting System

Keyword spotting is the task of retrieving keywords from document images. The present research focuses on handwritten documents . The input of the system is a word image and a keyword. For each input, the system must decide whether the image contains the requested keyword or not. The decision for the image X and keyword K is decided by a threshold over a dissimilarity measure: $ds(X, K) < T$. T can be selected based on a trade-off between system precision and recall.

In this work, we focus on perfectly segmented text word images. The images are first binarized and then normalized to remove the skew and slant of the text. The complete normalization process is described in details in [15]. From each input image, patches are extracted using a horizontal sliding window. The patch height is always equal to the height of the image. Each patch is W pixels wide. The window is moved two pixels at a time from left to right.

2.1 Convolutional Restricted Boltzmann Machine

A Restricted Boltzmann Machine (RBM) is a generative stochastic Artificial Neural Network (ANN). It is designed to learn a probability distribution over the inputs. The training of an RBM tries to maximize the Log-Likelihood of the learned input distribution. RBMs only rose to a large audience, after the Contrastive Divergence (CD) algorithm was introduced [7]. CD is a fast learning algorithm to train an RBM, very similar to the gradient descent of a neural network. CD approximates the Log-Likelihood gradients of the input distribution by minimizing the reconstruction error, thus training the RBM into an autoencoder. An RBM has two layers, a visible layer and an hidden layer. There are no connection between units of the same layer (bipartite graph).

Artificial Neural Networks and Machine Learning - ICANN 2016: 25th International Conference on Artificial Neural Networks, pp. 113-120

The RBM model was extended to the Convolutional RBM (CRBM) model [12]. Taking advantage of convolution, a CRBM learns feature detectors shared among all locations in an image. This allows the feature representations to be invariant to local translations in the input and allows learning to scale to realistically sized images. The model is outlined in Figure 1. It is the building block of the proposed feature extraction system. The visible layer is made of $N_V \times N_V$ binary units. The hidden layer is made of K groups of $N_H \times N_H$ binary units. The layers are connected by K convolutional filters of shape $N_W \times N_W$ ($N_W \triangleq N_V - N_H + 1$).

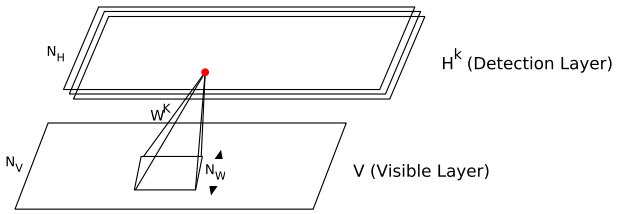

Fig. 1. A Convolutional Restricted Boltzmann Machine.

2.2 Feature Extraction

Features are extracted from one patch using a Convolutional Deep Belief Network (CDBN)[12]. This network is composed of two CRBM. The network is only trained in an unsupervised manner, i.e. labels are not used to train the network. Once the first layer is trained, its weights are frozen and its features are passed to the next layer. The network used for feature extraction is presented in Figure 2.

Generally, higher levels of an ANN encode information about progressively larger input regions. Typical Convolutional Neural Networks use pooling layers to shrink the representation by a small factor. Probabilistic Max Pooling was introduced for generative models to support both top-down and bottom-up inference [12]. This operator shrinks the representation by a factor C. Each layer of the proposed CDBN model uses this operator in order to improve translation-invariance, reduce the computational cost and reduce the number of features.

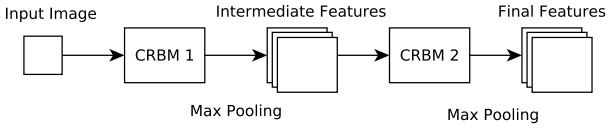

Fig. 2. Convolutional Deep Belief Network used for feature extraction

One patch is passed to the first layer. Then, the activation probabilities of the pooling layer are computed. These probabilities are passed to the second layer, which computes the final features for the patch from its pooling layer. From the network, we define $F(X)$ as a sequence of feature vectors (one for each patch):

$$F(X) = [CDBN(x_1), CDBN(x_2), ..., CDBN(x_N)] \quad (1)$$

The features are normalized so that each feature vector has zero-mean and unit variance.

2.3 Dynamic Time Warping

Dynamic Time Warping (DTW) is a technique used to find an optimal alignment between two sequences of different length. Sequences are warped non-linearly so that they match each other. It is well established in the field of keyword spotting [19]. The cost of an alignment is the sum of the $d(x, y)$ distances of each aligned pair. This system uses the squared Euclidean distance.

The DTW distance $D(F(X), F(Y))$ of two feature vector sequences $F(X)$ and $F(Y)$ is given by the minimum alignment cost. For speeding up the process and improving the results, a Sakoe-Chiba band [22] is used. When several occurrences of the keyword are available in the training set, the example that minimizes the distance for the currently tested image is selected. The DTW distance over the features is used as the final dissimilarity measure $ds(X, K)$.

3 Experimental Evaluation

We compare the features extracted by the proposed system with two other feature sets known to work well with DTW. Marti2001 [15] is a well-established heuristic set of features and has been used repeatedly for keyword spotting. It is made of nine geometrical features per column of the image. Rodriguez2008 [20] uses local gradient histogram features with overlapping windows.

The proposed system was evaluated using two benchmark data sets. The George Washington data set (GW) [11] is composed of 20 pages of letters written by George Washington and his associates. Due to the small amount of samples, a four-fold cross validation is used for experimental evaluation. It is made of 4894 word images. The Parzival data set (PAR) [4] contains 45 pages of a medieval manuscript, written in the 13th century. The set contains 23485 word images. Although the data sets have several writers, the styles being very similar, they are considered as *single-writer*. The system uses the normalized word images, ground truth, keywords, training sets, validation sets and test sets made available by [3].

For evaluation, a set of keywords is spotted on the test set of both data sets. The performance is measured for two different scenarios. The *global* scenario measures the Average Precision (AP) of the system, using a single *global threshold*. The *local* scenario measures the Mean Average Precision (MAP), using a *local threshold* for each keyword. These values are considered to assess the system performance. The `trec_eval`[3] software is used to compute these values [3].

[3] `http://trec.nist.gov/trec_eval`

Since the DTW algorithm requires an example in order to compute a distance, the keywords considered for performance evaluation are constrained to those that appear at least once in the training set and once in the test set.

3.1 System setup

The parameters for training the model and the architecture parameters were optimized for the task. For each data set, these parameters have been optimized individually with respect to the MAP and AP performance on the validation set. The performance of the system is measured on the independent test set.

Both networks have 2 layers of CRBM with Probabilistic Max Pooling. Each patch is 20 pixels wide (W). The GW network first layer is made of 8 9×9 filters followed by 8 3×3 filters. The PAR network has 12 9×9 filters followed by 10 3×3 filters. The pooling ratio (C) for each layer has been set to 2. The networks have been trained for 50 epochs of Contrastive Divergence, using mini-batch training. To improve generalization, L2 weight decay has been applied to all weights. The filters have been initialized using a zero-mean normal distribution with a variance of 0.01, the hidden biases to -0.1 and the visible biases to 0.

4 Results and discussion

Table 1. Mean Average Precision (MAP) and Average Precision (AP) for the different features. The relative improvement over the best baseline is also mentioned. For the GW data set, the results have been averaged over the four cross validation runs.

	GW		PAR	
System	AP	MAP	AP	MAP
Marti2001	33.24	45.26	50.67	46.78
Rodriguez2008	41.20	63.39	55.82	47.52
Proposed	**55.65**	**67.43**	**58.82**	**62.42**
Improvement	35.07%	6.37%	5.37%	31.35%

The experimental results are presented in Table 1. In both scenarios and for both data sets, the proposed system outperformed both reference feature sets. In the following discussion, the relative improvements are reported with respect to the `Rodriguez2008` system which always outperforms `Marti2001`.

For the GW data set, in the global scenario, the proposed system clearly outperformed both reference systems, by 35.07%. In the local scenario, our system is also able to outperform the local gradient features by 6.37%. For the PAR data set, the proposed system performs much better than the benchmark in the local scenario, outperforming it by 31.35%. In the global scenario, our system also outperforms the baseline by 5.37%.

Artificial Neural Networks and Machine Learning - ICANN 2016: 25th International Conference on Artificial Neural Networks, pp. 113-120

Overall, the proposed system exhibits more stable performance than the two baselines. While both datasets are quite different, the performance are quite similar, showing the utility of the unsupervised feature learning system over handcrafted features that are harder to generalize over different datasets. This can be observed with the local histogram features that are clearly outperforming the local geometrical features on GW, but are almost on par on the PAR dataset.

In spite of the significant improvements, optimization of our model proved quite challenging. The model has many parameters and their parametrization is very important. Moreover, the model needs to be tuned in order to provide features that can be used with DTW. The number of outputs revealed very important to tune with respect to the system performance on the independent validation set. Due to the simple Euclidean Distance used in the DTW distance, having too many output features can decrease the performance. Therefore, we focused on networks yielding reasonable number of features. Models with only one layer proved to learn only low-level features and produced too many features. On the contrary, the inputs were not complex enough for a three layer network, which failed to generalize. For these reasons, a two-layer model was selected. The number of filters (K) has different effects. Increasing it improves the learning capacity of the model. Thus, it is typically large in convolutional networks, ranging from 50 to 400 per layer. However, increasing the number of filters of the final layer also increases the number of features used by the DTW. Experiments have shown that large number of filters strongly decreased the performance.

The patch width proved an important factor. This parameter was limited by the size of the convolutional filters (the patch must be at least as wide as the filter), so they had to be optimized together. Experimentally, for both data sets, the optimal patch width was found to be 20 pixels. Interestingly, this is slightly larger than the average width of a character in the data sets. Narrower patches proved rather unsuccessful and wider patches only increased the computational burden of the system without increasing its performance.

While binary hidden units proved to work well for both data sets, Rectified Linear Units (ReLU) [17] proved more effective on the PAR data set. They improved the AP by 20% and the MAP by 24%. While producing good results on the GW data set, they did not prove as effective as binary hidden units, being around 5% to 8% less effective. It seems that they were not able to learn generic features with the small number of available samples, while the large number of images in the PAR data set helped them generalize more effectively. This may indicate that there were too many ReLUs for the small number of samples.

For the network with binary units, enforcing sparsity of the hidden units improved the performance by 21% in the global scenario and 13% in the local one, on the validation set. This helped learning generic features, better for discrimination. While the network was able to learn reconstruction without sparsity, the features were not generic enough. We followed Lee et al. regularization method [9] where updates are made to the visible biases to reach a certain sparsity with some learning rate. The sparsity parameters have been chosen so that the sparsity was reached while still allowing the network to learn.

Artificial Neural Networks and Machine Learning - ICANN 2016: 25th International Conference on Artificial Neural Networks, pp. 113-120

5 Conclusion and Future Work

A keyword spotting system extracting features using Convolutional Deep Belief Networks and scoring word with Dynamic Time Warping was presented for handwritten keyword spotting. The proposed system was experimentally compared with two other sets of features on two different benchmark data sets. On both data sets, the proposed system outperformed the two baselines. The best improvements were observed in the scenario where a single threshold is used for the whole data set when deciding whether or not a word is spotted and very few templates per keyword were available. Moreover, the proposed system proved similarly effective on two very different data sets.

Future work could go in several directions. The discriminative power of the learned features could be improved by training the network for classification, using the word labels after pretraining. This could lead to more discriminative features. Augmenting the data set with geometrical distortions may also lead to a more generic feature extractor. Better normalization of the extracted features is also likely to improve the results. Testing the system on a multiple writer data set would prove useful in evaluating the genericity of the extracted features.

The C++ implementations of the proposed system[4] and our CDBN library[5] are freely available on-line.

References

1. Chen, F.R., Wilcox, L.U., Bloomberg, D.S.: Word spotting in scanned images using Hidden Markov Models. In: Proceedings of the IEEE Int. Conf. on Acoustics Speech and Signal Processing. vol. 5, pp. 1–4. IEEE (1993)
2. Choisy, C.: Dynamic handwritten keyword spotting based on the NSHP-HMM. In: Proceedings of the IEEE Int. Conf. on Document Analysis and Recognition. vol. 1, pp. 242–246. IEEE (2007)
3. Fischer, A., Keller, A., Frinken, V., Bunke, H.: Lexicon-free handwritten word spotting using character HMMs. Pattern Recognition Letters 33, 934–942 (2012)
4. Fischer, A., Wüthrich, M., Liwicki, M., Frinken, V., Bunke, H., Viehhauser, G., Stolz, M.: Automatic transcription of handwritten medieval documents. In: Proceedings of the Int. Conf. on Virtual Systems and Multimedia. pp. 137–142. IEEE (2009)
5. Forsyth, D., Jaety, E., Teh, Y.W., Maire, M., Bock, R.B., Vesom, G.: Making latin manuscripts searchable using gHMMs. In: Proceedings of the Advances in Neural Information Processing Systems. vol. 17, p. 385. MIT Press (2005)
6. Frinken, V., Fischer, A., Manmatha, R., Bunke, H.: A novel word spotting method based on recurrent neural networks. IEEE Transactions oni Pattern Analysis and Machine Intelligence 34, 211–224 (2012)
7. Hinton, G.E.: Training Products of Experts by minimizing Contrastive Divergence. Neural Computation 14, 1771–1800 (2002)
8. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)

[4] `https://github.com/wichtounet/word_spotting/tree/paper_v2`
[5] `https://github.com/wichtounet/dll`

9. Honglak, L., Chaitanya, E., Ng, A.Y.: Sparse Deep Belief Net Model for Visual Area V2. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 873–880 (2008)
10. Kuo, S.s., Agazzi, O.E.: Keyword spotting in poorly printed documents using pseudo 2-D Hidden Markov Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 16, 842–848 (1994)
11. Lavrenko, V., Rath, T.M., Manmatha, R.: Holistic word recognition for handwritten historical documents. In: Proceedings of the Int. Workshop on Document Image Analysis for Libraries. pp. 278–287. IEEE (2004)
12. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional Deep Belief Networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the Int. Conf. on Machine Learning. pp. 609–616. ACM (2009)
13. Manmatha, R., Croft, W.: Word spotting: Indexing handwritten archives. Intelligent Multimedia Information Retrieval Collection pp. 43–64 (1997)
14. Manmatha, R., Han, C., Riseman, E.M.: Word spotting: A new approach to indexing handwriting. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 631–637. IEEE (1996)
15. Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. Int. journal of Pattern Recognition and Artificial intelligence 15, 65–90 (2001)
16. Myers, C., Rabiner, L., Rosenberg, A.: An investigation of the use of Dynamic Time Warping for word spotting and connected speech recognition. In: Proceedings of the IEEE Int. Conf. on Acoustics Speech and Signal Processing. vol. 5, pp. 173–177. IEEE (1980)
17. Nair, V., Hinton, G.E.: Rectified Linear Units improve Restricted Boltzmann Machines. In: Proceedings of the Int. Conf. on Machine Learning. pp. 807–814 (2010)
18. Rath, T.M., Manmatha, R.: Word image matching using Dynamic Time Warping. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition. vol. 2, pp. 521–527. IEEE (2003)
19. Rath, T.M., Manmatha, R.: Word spotting for historical documents. Int. Journal of Document Analysis and Recognition (IJDAR) 9, 139–152 (2007)
20. Rodrıguez, J.A., Perronnin, F.: Local gradient histogram features for word spotting in unconstrained handwritten documents. In: Proceedings of the Int. Conf. on Frontiers in Handwriting Recognition. pp. 7–12 (2008)
21. Rose, R.C., Paul, D.B.: A Hidden Markov Model based keyword recognition system. In: Proceedings of the Int. Conf. on Acoustics Speech, and Signal Processing. pp. 129–132. IEEE (1990)
22. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics Speech and Signal Processing 26, 43–49 (1978)
23. Vinciarelli, A.: A survey on off-line cursive word recognition. Pattern recognition 35, 1433–1446 (2002)
24. Wicht, B., Hennebert, J.: Camera-based Sudoku recognition with Deep Belief Network. In: Proceedings the of IEEE Int. Conf. of Soft Computing and Pattern Recognition. pp. 83–88. IEEE (2014)
25. Wicht, B., Hennebert, J.: Mixed handwritten and printed digit recognition in Sudoku with Convolutional Deep Belief Network. In: Proceedings of the IEEE Int. Conf. on Document Analysis and Recognition. IEEE (2015)

Artificial Neural Networks and Machine Learning - ICANN 2016: 25th International Conference on Artificial Neural Networks, pp. 113-120