Script Identification using Bag-of-Words with Entropy-weighted Patches

Jan Zdenek Hideki Nakayama

Graduate School of Information Science and Technology,
The University of Tokyo

The increasing interest in scene text reading in multilingual environments raises the need to recognize and distinguish between different writing systems. In this paper, we propose a novel method for script identification using convolutional features for the traditional bag-of-words model in a combination with weighting by means of intra-cluster information entropy. This approach exploits the expressive representation of convolutional neural networks, which have displayed outstanding performance in many text analysis and recognition tasks in recent years, discriminative power of script-characteristic features, and generalization abilities of bag-of-words model. The proposed method is evaluated on two public benchmark datasets for script identification. The experiments demonstrate that our method outperforms the baseline and yields competitive results.

1. Introduction

Script identification is a task of recognizing the writing system, a script such as Latin alphabet and Chinese characters, used in a piece of text. Due to increasing globalization of the world and mixing demographics, we can encounter text in various writing systems in our everyday lives. In such multilingual environments, text script identification is a very important task and a pre-requisite for successful automatic scene text reading.

In this paper, the problem of script identification at the text line level in natural scene images is addressed. Most end-to-end text reading systems for scene text are either designed to recognize text only in Latin alphabet. Some can be used for recognition of multiple writing systems, but they do not support the actual recognition of what writing system is used in the detected text, which makes them unable to utilize different text recognition models for different scripts. Therefore, the script identification is a crucial task whose employment in an end-to-end text reading system can open the path to fully-automatic scene text reading supporting a multitude of writing systems.

The early works tackling the problem of script identification, which precede the surge of interest in scene text reading, focused solely on printed or handwritten documents [1, 2].

The script identification in scene text is a challenging problem mainly due to

- high variance in font, color, and flow of the text, which is not constrained by a regular layout as in case of documents,
- possible presence of complex background and other noise caused by the surrounding environment,
- subsets of characters appearing in more than one script, making it more difficult to recognize the script on the word level and impossible on character level.

The method presented in this paper employs convolutional features, which have been shown to exhibit high expressiveness in representation, in a combination with the traditional bag-of-words approach which provides useful generalization. In addition, weighting by means of the information entropy in partitioned feature space is introduced.

2. Related Work

Gllavata et al. [3] were the first one to perform script identification on images with more complex backgrounds than printed or handwritten documents. They use several handcrafted features to train a k-nearest neighbor classifier to recognize two categories, Latin alphabet and other ideographic scripts such as Chinese characters, in video captions.

Shi et al. [4] use a convolutional neural network (CNN) with a spatially-sensitive pooling layer, which accepts inputs of arbitrary widths and makes the network invariant to horizontal positions of responses while keeping information about vertical positions. In their more recent work [5], they extend their spatially-sensitive pooling layer with discriminative encoding learned by a discriminative clustering method proposed by Singh et al. [6]

Gomez et al. [7] have explored clustering convolutional features from random text patches to find more discriminative patches and weigh them by their distance to the closest patch template of a different class. More recently, Gomez et al. have trained an end-to-end model which consists of an ensemble of identical networks conjoined at the last fully-connected layer [8] to make the network learn more discriminative features, taking inspiration from Siamese networks [9].

Our method, similarly to [5, 7], tries to explicitly discover discriminative features in the text by clustering and weighing individual text patches used for global classification of whole text lines. However, we try to calculate the relevance and discriminative power of respective clusters by the amount of information in the clusters.

Contact: Jan Zdenek, The University of Tokyo, Graduate School of Information Science and Technology, 1-1 Yayoi, Bunkyo, Tokyo, jan@nlab.ci.i.u-tokyo.ac.jp

3. Methodology

Our method performs script identification on text line level, which means that every text line is assigned one class label in the classification process. Images of pre-segmented text lines are converted into gray-scale color space and used to create a training set of patches by extracting square patches from the text line images in a sliding window fashion at two scales, full text line height and two third of the text line height. The extracted patches are resized to a fixed height of 32 × 32 pixels and they are used to train a convolutional neural network (CNN).

Each image can be classified by taking the average of class probabilities of all patches extracted from the image and fed to CNN to form a global classification rule. However, in ordinary patch-based classification, all patches are treated equally and have the same importance in the global classification. As pointed out in [7], certain characters or parts of characters have low information value for recognition of the script used in the text as they appear in multiple scripts, whereas other characters or their parts are more discriminative by being unique to only one or a small number of scripts. Therefore, it is desirable to give priority to the patches containing more discriminative features and ignore those with low discriminative power in the global classification.

3.1 Entropy-weighted Patches

In order to prioritize the more discriminative patches in the global classification of a text line, we introduce weights into the global classification rule. To learn the weights, we take the outputs of the penultimate fully-connected layer from a trained CNN as feature vectors and subsequently use them to partition the feature space by the k-means algorithm.

The patches with highly discriminative features are expected to be further away from patches of other classes in the feature space while patches with low discriminative power are likely to be mixed up with patches of other classes. Therefore, clusters containing text patches with high discriminative power are expected to be purer and contain patches of one or only a few classes while clusters with patches of low discriminative power will contain patches of many classes and have low purity. These attributes are employed to compute the weights of each cluster by means of intra-cluster information entropy. The weight ω_{cl} of a cluster cl is defined as

$$\omega_{cl} = \ln C - \sum_{i=1}^{C} \frac{n_i}{N_{cl}} \ln \frac{n_i}{N_{cl}}, \tag{1}$$

where C refers to number of classes, N_{cl} to the total number of samples in a cluster, and n_i to the number of samples of a specific class i in a cluster cl. The clusters of high purity thus have big weights while impure clusters have low weights. The weight values are scaled to fit within the $\langle 0, 1\rangle$ range.

At testing time, the weights of all patches are determined by what cluster the respective patch belongs to, which is performed by finding the nearest cluster center point. The global classification rule is then defined as the average of weighted softmax outputs $f(\cdot)$ over all N patches x_i extracted from a text line:

$$y = \frac{1}{N} \sum_{i=1}^{N} \omega_i f(x_i). \tag{2}$$

Input	1x32x32 image patch
Conv1	96 channels, 5x5 filter, padding 2x2, ReLU
MaxPooling1	pool size 3x3, stride 2
Conv2	256 channels, 3x3 filter, padding 2x2, ReLU
MaxPooling2	pool size 3x3, stride 2
Conv3	384 channels, 3x3 filter, padding 1x1, ReLU
MaxPooling3	pool size 3x3, stride 2
Conv4	512 channels, 3x3 filter, padding 1x1, ReLU
MaxPooling4	pool size 3x3
FC5	512 neurons, ReLU, dropout 0.5
FC6	96 neurons, ReLU, dropout 0.5
FC7	*#classes* neurons, softmax

Table 1: The details of the network architecture used in the experiments. *#classes* refers to the number of classes in a dataset.

3.2 Bag of CNN Words

The patch-based approach of our method allows us to adopt the traditional bag-of-words method which can be employed to train a global classifier for the whole text lines.

The codeword dictionary for bag-of-words is created by applying the k-means clustering algorithm in the same way as described in 3.1 and selecting the found cluster centers as the codewords. A batch of patches extracted from the input image is fed into the trained CNN and the outputs from the penultimate layer are used as feature vectors which are transformed into codewords by finding their respective nearest cluster centers in the feature space. The generated batch of codewords is then represented as a sparse histogram of codewords. Finally, the bag-of-words representation of all images in the training set is used to train a multi-layer perceptron (MLP) classifier for the global classification of unlabeled text line images.

The bag-of-words approach and the approach described in 3.1 are combined by using the entropy-based weight values to weight the individual codewords of the bag-of-words dictionary, which is performed by applying the weights directly to individual values in the codeword histograms by element-wise multiplication defined as

$$\mathbf{h}_{\omega} = \boldsymbol{\omega} \odot \mathbf{h}, \tag{3}$$

where $\mathbf{h}$ is a vector of codeword histogram values, $\boldsymbol{\omega}$ the cluster weights, and $\mathbf{h}_{\omega}$ the weighted histogram. The weighted histograms are then used to train the MLP for final classification in the same fashion as non-weighted bag-of-words histograms.

4. Experiments

The experiments were conducted on two datasets, SIW-13 [5] and MLe2e [8].

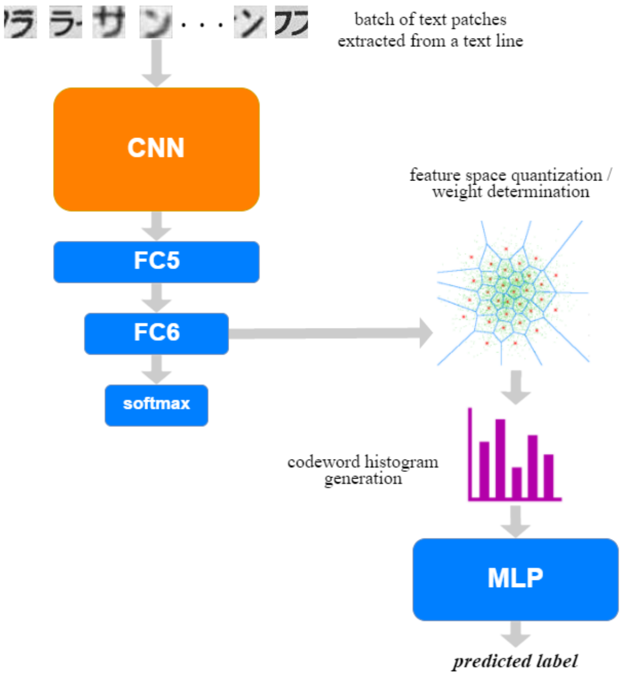
batch of text patches extracted from a text line
CNN
FC5
FC6
softmax
feature space quantization / weight determination
codeword histogram generation
MLP
predicted label

Figure 1: Overall structure of the whole pipeline of the proposed method using weighted bag-of-words model. Outputs from the penultimate layer of CNN are clustered by k-means and codeword dictionary is created for bag-of-words. The histograms of codewords are weighted by calculated weights based on information entropy and they are used to train the final MLP classifier.

The SIW-13 dataset consists of images of pre-segmented text lines in thirteen different scripts: Arabic, Cambodian, Chinese, Latin alphabet, Greek alphabet, Hebrew, Japanese (all three script types used in Japanese considered as one), Kannada, Hangul, Mongolian, Cyrillic alphabet, Thai, and Tibetan.

The MLe2e dataset comprises text lines in four scripts: Latin alphabet, Chinese, Kannada, and Hangul.

4.1 Implementation Details

We follow previous works and use a network architecture [10] which shows outstanding results on tasks related to text analysis and we further optimize the network for our task by repeated experiments. The structure of the network is described in detail in Table 1. Categorical cross-entropy is used as the loss function and the network is trained using stochastic gradient descent with learning rate set to 0.01 and momentum to 0.9. Dropout [11] is used between the fully-connected layers to reduce early overfitting on training data.

The structure of the whole pipeline of the proposed method is shown in Figure 1.

4.2 Baseline Method

The baseline method uses the same CNN architecture as the proposed method and the same text patches for training. The classification rule is defined as the average of softmax outputs over all patches extracted from a text line, which means the average of class probabilities over all patches. The main drawback of the baseline method is that it considers each patch in the text line equally important for classification of the whole text line.

	SIW-13	MLe2e
Shi et al. [5]	89.4	-
HUST [12]	88.0	-
Gomez et al. [8]	94.8	94.4
Nicolaou et al. [13]	83.7	-
CNN baseline	91.14	94.70
Entropy-weighted CNN	91.43	94.86
Bag of CNN words	91.86	95.47
Bag of entropy-weighted CNN words	91.94	95.60

Table 2: Comparison of classification performance of our proposed method (bottom) with the baseline method (middle) and related work (top) on SIW-13 and MLe2e datasets.

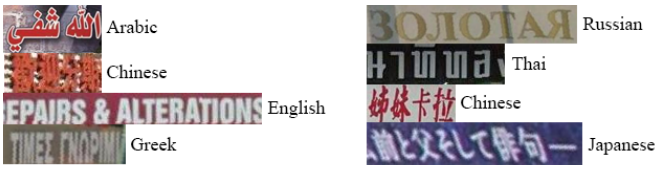

Figure 2: Examples of correct classifications of difficult samples in the SIW-13 dataset.

4.3 Results

As can be seen in Table 2, the proposed method outperforms the baseline method on both benchmark datasets, yielding competitive results. The results indicate that both entropy-based weighing and bag-of-words contribute to an enhancement of the performance as they both improve the classification accuracy when used separately. The Figure 5 shows that misclassifications most frequently happen between scripts which share a subset of characters or exhibit clearly noticeable similarities.

Figure 4 displays examples of text patches in a cluster of low entropy and high entropy, respectively. It is apparent that text patches gathered in a cluster of low entropy contain distinctive features which make it easy to recognize the script while text patches in a high-entropy cluster consist of a lot of unclear strokes and patterns appearing in multiple scripts.

5. Conclusion

A novel method for script identification in real-world scene text has been proposed. The method combines prioritizing of more discriminative features in the text using information entropy with the traditional bag-of-words ap-

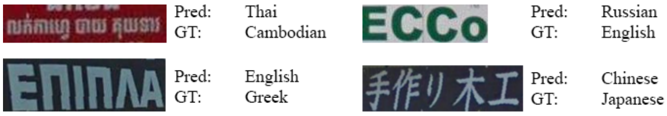

Figure 3: Examples of misclassifications on the SIW-13 dataset.

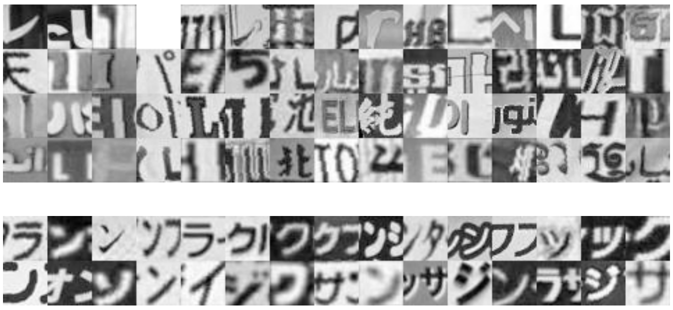

Figure 4: Examples of samples in a cluster of high entropy (top) and samples in a cluster of low entropy (bottom). It shows that the clusters with low entropy contain text patches with features that are unique to specific scripts, thus are more discriminative and important for global classification than patches belonging to clusters of high entropy.

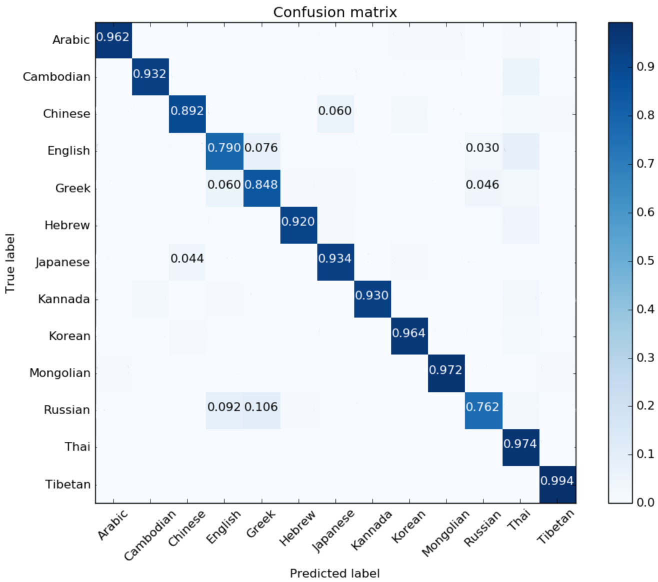

Figure 5: Classification confusion matrix on SIW-13 dataset using bag of entropy-weighted CNN words. The results show that the Latin, Greek, and Cyrillic alphabets, which are all derived from the same origin and share many same characters, are the most difficult to correctly recognize because of their high mutual similarities. Similarly, Chinese and Japanese are difficult to distinguish due to mutual sharing of a subset of characters.

proach to enhance the already powerful representation of convolutional features.

Experiments conducted on public benchmark datasets for scene text script identification show that the proposed method produces competitive results.

References

[1] R. Unnikrishnan and R. Smith, "Combined script and page orientation estimation using the tesseract ocr engine," in *Proc. of the Int. Workshop on Multilingual OCR*, 2009.

[2] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic script identification from document images using cluster-based templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 176–181, 1997.

[3] J. Gllavata and B. Freisleben, "Script recognition in images with complex backgrounds," in *Signal Processing and Information Technology. Proc. of IEEE Int. Symposium on*, 2005.

[4] B. Shi, C. Yao, C. Zhang, X. Guo, F. Huang, and X. Bai, "Automatic script identification in the wild," in *Document Analysis and Recognition (ICDAR), Int. Conference on*, 2015.

[5] B. Shi, X. Bai, and C. Yao, "Script identification in the wild via discriminative convolutional neural network," *Pattern Recognition*, vol. 52, pp. 448–458, 2016.

[6] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *European Conference on Computer Vision (ECCV)*, 2012.

[7] L. Gomez and D. Karatzas, "A fine-grained approach to scene text script identification," in *Document Analysis Systems (DAS), IAPR Workshop on*, 2016.

[8] L. Gomez, A. Nicolaou, and D. Karatzas, "Improving patch-based scene text script identification with ensembles of conjoined networks," *Pattern Recognition*, vol. 67, pp. 85–96, 2017, to be published.

[9] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Sackinger, and R. Shah, "Signature verification using a siamese time delay neural network," *Int. Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, vol. 7, no. 4, pp. 669–688, 1993.

[10] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *Document Analysis and Recognition (ICDAR), Int. Conference on*, 2011.

[11] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] N. Sharma, R. Mandal, R. Sharma, U. Pal, and M. Blumenstein, "ICDAR2015 competition on video script identification (CVSI 2015)," in *Document Analysis and Recognition (ICDAR), Int. Conference on*, 2015.

[13] A. Nicolaou, A. D. Bagdanov, L. Gómez, and D. Karatzas, "Visual script and language identification," in *Document Analysis Systems (DAS), IAPR Workshop on*, 2016.