Connected Component level Multiscript Identification from Ancient Document Images

Sheikh Faisal Rashid
Technical University of Kaiserslautern, Germany
s_rashid09@informatik.uni-kl.de

Faisal Shafait
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
faisal.shafait@dfki.de

Thomas M. Breuel
Technical University of Kaiserslautern, Kaiserslautern, Germany
tmb@informatik.uni-kl.de

ABSTRACT

In a multilingual optical character recognition (MOCR) environment, a MOCR system usually requires script identification of each word or character before passing it to the OCR engine. Many existing script identification techniques mainly depend on various features extracted from document images at character, word, text line or block level. This feature extraction process is not robust and extracted features for one script identification problem are not fully applicable to other script identification problems. In this paper, we present a novel and efficient technique for multi-script identification at connected component level using convolutional neural network. The convolutional neural network acts as a discriminative learning model, where suitable script identification features are automatically extracted and learned as convolution kernels from the raw input. We test our approach on a dataset of ancient Greek-Latin mix script document images. We achieve 96.37% accuracy on a test dataset at connected component level and this accuracy is further improved to 98.40% by using class majority in the left-right neighboring area. The main advantage of our approach is that it can be easily adapted for the identification of other scripts and it can give 100% accuracy at block level.

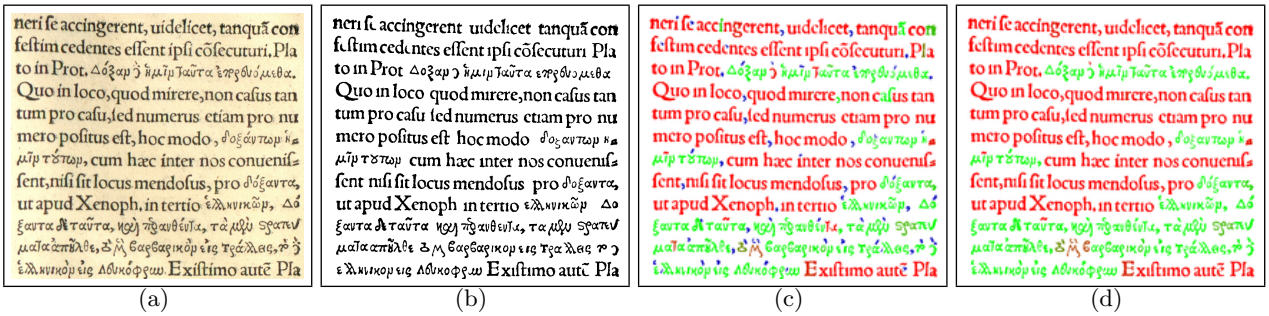
1. INTRODUCTION

Recently, there has been a growing interest in the area of multilingual OCR due to variety of documents available in bilingual or even in trilingual form, for example ancient mix script documents, multilingual dictionaries, books with line by line or column wise translation, and official documents from some countries like passport forms, examination questions and money order forms etc. Usually, a MOCR system combines character level or word level classifiers for different languages or scripts and recognition of a particular character or word is done by its respective classifier. Therefore prior knowledge of a script for each character or word is essential for the selection of an appropriate classifier in MOCR environment. An OCR classifier can be trained on more than one languages by adding individual characters from all languages into training process but this can lead to more classification errors due to the increase in number of classes.

DAS '10, June 9-11, 2010, Boston, MA, USA

Previous methods for script identification can be broadly grouped into two approaches, global approaches and local approaches. This categorization is based on feature extraction process employed at global level (a region of text lines) or local level (character, word or single text line) for each individual script. The survey paper of Abirami [1] presented a precise overview of some of these methods. Most of the existing methods for script identification work at document level or these methods assume that the document images have different scripts only at a particular place (in a particular column, paragraph or text line) and only few of them consider word level multi-script identification. Hochberg et al. [4] used cluster based templates for the script identification at document level. Later, Spitz [14] presented language identification in Han-based and Latin-based scripts by using vertical position distribution of upward concavities, optical density distribution and most frequently occurring word shapes characteristics. Pal and Chaudhuri [11, 10] worked on separation of text lines from different scripts using projection profiles, water reservoir concepts and existence of head-line(a feature specific to Bangla and Devanagari scripts). Zhou et al. [7] presented the Bangla-English script identification by analyzing connected component profiles and head-line features. Busch et al. [2] described the use of texture as a tool for determining the script of a document image. Joshi et al. [5] also employed the texture based Gabor filter at multiple scales and then they further used some script dependent feature like head line information, statistical features, local features and horizontal profile information. Ma and Doermann [8] performed word level script identification for scanned document images by using Gabor filter and compared the performance of different classifiers. Ramakrishnan and Pati [12] reported word level multi script identification by using Gabor and discrete cosine transform (DCT) features.

In this paper we present a discriminative learning approach for multi-script identification using convolutional neural network (CNN) at connected component level. Convolutional neural networks with properties of local receptive fields, weight sharing, and spatial sub-sampling layers have ability to learn discriminative feature from the raw image data by gradient based learning technique [6]. In the current method we use this property of convolutional neural network for learning the complex features for multi-script identification at con-



(a) (b) (c) (d)

Figure 1: Pipeline for multi-script identification system (a) Original image (b) Preprocessed image (c) CNN output (d) Final output

nected component level. We demonstrate the effectiveness of our method on a dataset of ancient Greek-Latin mix script document images. We extract connected components from binarized document images and train the CNN for each connected component with its respective script label using back propagation algorithm [3]. Our approach works at connected component level and therefore it does not require layout analysis for image segmentation into text lines or blocks. Another advantage is that the discriminative features for script identification are extracted and learned automatically by CNN and there is no need to define specific features for a particular script.

2. METHOD DESCRIPTION

This section describes the whole method in a pipeline from document image preprocessing to script identification. The complete pipeline of the system is shown in Figure 1 and is described in the following subsections.

2.1 Preprocessing

The document images are binarized before extracting the connected components. In this method we use local adaptive thresholding technique [13] for binarization of the complete dataset. Binarized document images may contain small noise (salt and pepper) or big noise (merging components or borders). The noise is removed by using a heuristic rule, based on size of the connected components. A connected component is considered as a noise if its height and width is less than or equal to 0.3 times or greater than 5 times of the median height and width of all components. This noise removal process may vary from dataset to dataset. Diacritics and punctuations are the small components that occur in both scripts. We also removed these small components from document images using the similar heuristic rule i.e. a component is considered as a diacritic or a punctuation if its width and height is equal to or less than 0.7 times the median height and width of all components. Figures 1(a) and 1(b) show the original color image and preprocessed image.

2.2 Feature Vector Generation

Instead of extracting complex geometrical, morphological or statistical features, a convolutional neural network is used to extract relevant features from raw input data. Connected components are extracted and rescaled to 40x40 dimensions and the pixel intensities are normalized between -1 and +1. These rescaled connected components describe the raw input to the CNN for training and evaluation. In this rescaling process we downscale or upscale the connected components depending on the size of their width or height to the size of 40x40 dimensions while keeping their aspect ratio intact. This step is very important because change in aspect ratio destroys shape of the connected component and this shape degradation effects the CNN classification accuracy.

2.3 Convolutional Neural Network Training

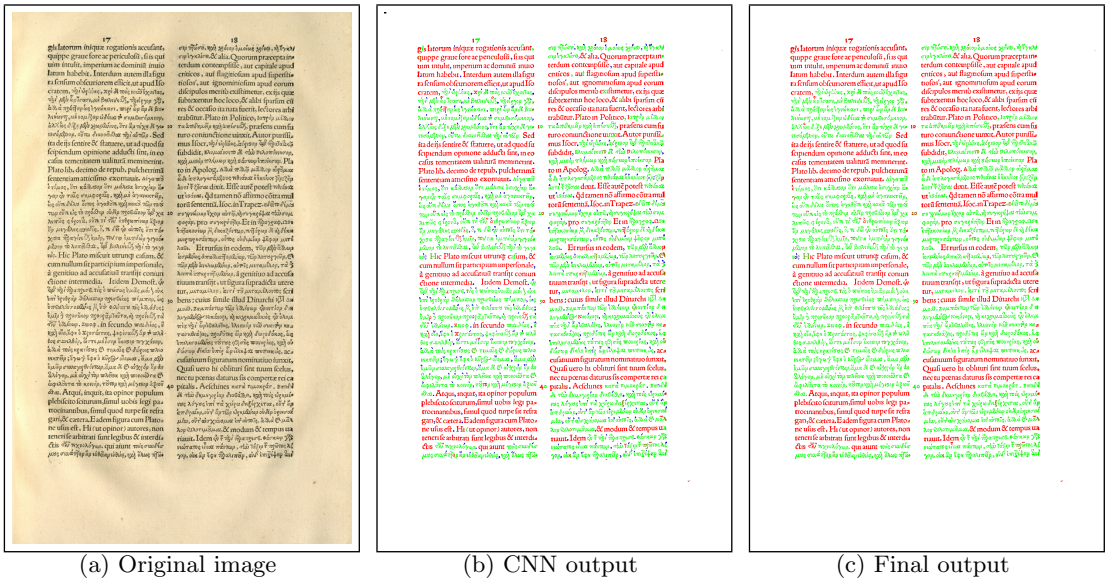
Convolutional neural networks have been applied in various image classification problems when sophisticated feature extraction is to be avoided and classification is done based on raw image data [9]. A typical CNN consists of convolutional layers, sub-sampling layers, one hidden layer and one output layer. Each unit in a layer receives inputs from small neighborhood (local receptive fields) in the previous layer and with these local receptive fields, neuron can extract the initial features like edges, corners, and end-points. The subsequent layers then combine these initial features into more higher-order features. Units in a layer are organized in planes within which all the units share the same set of weights. The set of outputs of the units in such a plane constitutes a feature map. The CNN used in this experiment consists of two convolutional layers with four and eight feature maps followed by two sub-sampling layers. The input layer of CNN receives the feature vector as described in section 2.2. The values of the feature map are obtained by convolving the input map with the respective kernel and applying an activation function to the result. We train the CNN over 19600 training samples and 2000 validation samples of Greek and Latin scripts for 200 epochs with 0.1 learning rate. An online error backpropagation algorithm [3] is used for CNN training.

2.4 Script Classification

For script classification the test document image is first preprocessed to extract the feature vectors and these feature vectors are passed to the CNN. The CNN gives two values (corresponding to two script labels) at output layer. A particular script is classified based on the highest output value for each feature vector. The classification output is represented by a corresponding color coded output image for each document image. We represent Greek script with green color, Latin script with red color and small components with blue color as shown in Figure 1(c). As small

Table 1: Script Identification Accuracy at Connected Component Level for Greek and Latin Scripts

	Training set		Validation set		Test set		
	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Accuracy after post-processing (%)
Greek	9800	99.41	1000	96.40	11302	95.16	97.65
Latin	9800	98.92	1000	97.80	10828	97.58	99.15
Overall	19600	99.16	2000	97.10	22130	96.37	**98.40**



(a) Original image (b) CNN output (c) Final output

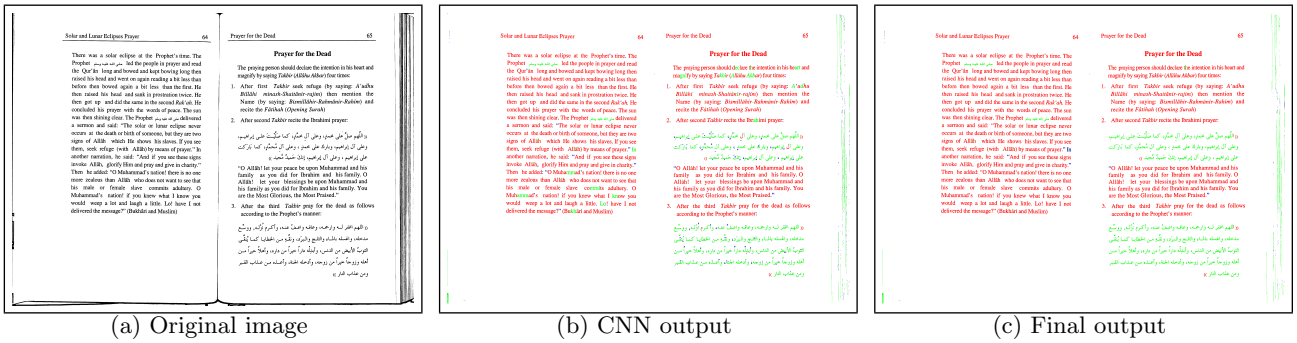
Figure 2: Greek-Latin multi-script identification results in color coding format. Red represents Latin script, green represent Greek script, and blue represents small components

connected components like diacritics and punctuations are filtered out during pre-processing, therefore script identification of these small connected components is not done by CNN. We post-process the CNN output to classify these small connected components and to further improve the script recognition accuracy at connected component level. In this post-processing the small connected components are classified using closest neighboring connected component script information. To improve the script recognition accuracy we extend the bounding box of each connected component to its left and right by a factor of its height or width (whichever is greater) and then use the class majority within the neighboring area to relabel the script of that component. Final output after applying the post-processing is shown in Figure 1(d). Figure 2 shows original document image, CNN output and final output for one of the sample image from test dataset document images.

We also tested our approach for Arbic-Latin mix script document images and we get the similar results. Figure 3 shows one of the sample output for Arabic-Latin mix script document image.

3. EXPERIMENTAL RESULTS

We evaluate the performance of the described multi-script identification system on a dataset of ancient Greek-Latin mix script document images consisting of 19 documents. Twelve documents are used for training and validation of the approach and remaining seven documents are used as a test dataset. The complete dataset is manually processed to generate the ground truth for testing and evaluation of the algorithm. For evaluation, we remove the Greek script text and keep the Latin script text from each document image. This provides us the ground truth for the Latin script. Similarly we generate the ground truth for Greek script text by removing the Latin script text from the original document images. We obtain 97.58% accuracy for Latin, 96.4% accuracy for Greek, and 96.37% overall accuracy at connected component level for the test dataset. The overall script recognition accuracy is further improved to 98.40% after post-processing. Table 1 gives the recognition accuracy percentages for the whole dataset used in this paper.



(a) Original image (b) CNN output (c) Final output

Figure 3: Arabic-Latin multi-script identification results in color coding format. Red represents Latin script, green represent Arabic script, and blue represents small components

4. DISCUSSION

In this paper we presented a multi-script identification method for ancient multi-script document images. We use a dataset of ancient Greek-Latin mix script document images for the evaluation of our method. We use a convolutional neural network as discriminative learning model to extract and learn the suitable features for Greek-Latin script identification. We obtain overall 96.37% accuracy from test dataset of seven ancient Greek-Latin mix document images at connected component level. The accuracy is further improved to 98.40% by using class majority in the left-right neighboring area. The dataset used in this experiment has a lot of noise in terms of touching or broken characters and smudge (e.g. ink spots or spread). It is observed that most of the recognition errors are due this noise and we may obtain better results on a clean dataset. It is also observed that CNN is sensitive to the object shapes in terms of slight variations due to the noise and this problem can be overcome by adding more training examples. The approach is also tested on Arabic-Latin mix script document images and we obtain similar results as we get for Greek-Latin mix script document images.

5. REFERENCES

[1] S. Abirami and D. Manjula. A survey of script identification techniques for multi-script document images. *International Journal of Recent Trends in Engineering*, 1(2):255–257, May 2009.

[2] A. Busch, W. W. Boles, and S. Sridharan. Texture for script identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1720–1732, November 2005.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2000.

[4] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns. Automatic script identification from document images using cluster-based templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2).

[5] G. D. Joshi, S. Garg, and J. Sivaswamy. *Script Identification from Indian Documents*, pages 255–267. Springer Berlin/Heidelberg, 2006.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11).

[7] Y. L. Lijun Zhou and C. L. Tan. *Bangla/English Script Identification Based on Analysis of Connected Component Profiles*, pages 243–254. Springer Berlin/Heidelberg, 2006.

[8] H. Ma and D. Doermann. Word level script identification for scanned document images. In *Proc. SPIE Document Recognition and Retrieval XI*, pages 124–135, San Jose, CA, USA, December 2003.

[9] C. Nebauer. Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, 9(4):685–696, Jul 1998.

[10] U. Pal and B. Chaudhuri. Script line separation from indian multi-script documents. In *International Conference on Document Analysis and Recognition*, pages 406–409, 1999.

[11] U. Pal and B. Chaudhuri. Automatic identification of english, chinese, arabic, devnagari and bangla script line. In *International Conference on Document Analysis and Recognition*, pages 790–794, 2001.

[12] P. B. Pati and A. Ramakrishnan. Word level multi-script identification. *Pattern Recognition Letters*, 29(9):1218 – 1229, 2008.

[13] F. Shafait, D. Keysers, and T. M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proc. SPIE Document Recognition and Retrieval XV*, pages 101–106, San Jose, CA, USA, January 2008.

[14] A. L. Spitz. Determination of the script and language content of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3).