Handwritten and Machine-Printed Text Discrimination Using a Template Matching Approach

Mehryar Emambakhsh, Yulan He and Ian Nabney
School of Engineering and Applied Science
Aston University, UK
Email: {m.emambakhsh, y.he9, i.t.nabney}@aston.ac.uk

***Abstract*—We propose a novel template matching approach for the discrimination of handwritten and machine-printed text. We first pre-process the scanned document images by performing denoising, circles/lines exclusion and word-block level segmentation. We then align and match characters in a flexible sized gallery with the segmented regions, using parallelised normalised cross-correlation. The experimental results over the Pattern Recognition & Image Analysis Research Lab-Natural History Museum (PRImA-NHM) dataset show remarkably high robustness of the algorithm in classifying cluttered, occluded and noisy samples, in addition to those with significant high missing data. The algorithm, which gives 84.0% classification rate with false positive rate 0.16 over the dataset, does not require training samples and generates compelling results as opposed to the training-based approaches, which have used the same benchmark.**

1. Introduction

Handwritten/machine-printed classification (HMC) is the process of labelling an image containing text segments, in order to discriminate handwritten from machine-printed text. It has numerous applications, particularly in (improving) Optical/Intelligent Character Recognition (OCR/ICR), automatic document analysis and anonymisation [1]. Since the outputs are one of the two classes, binary classification techniques have widely been used to resolve this problem. The variations of machine-printed samples are significantly lower than those of the handwritten class. As a result, the feature space extracted from the machine-printed samples are more concentrated, while the same features of the handwritten text samples are mapped to a significantly wider range [2]. This fact has been employed by many of the previous approaches to assign classification boundaries between the two classes, which provided acceptable results over particular types of documents. However, these approaches have a major disadvantage. Although features are intended to have high class separability for the two classes, they can easily overlap for unseen documents, resulting in high deterioration of the classification performance. The main reason is that the feature extraction methods can not be generalised to the significantly higher variance of the handwritten samples than the machine-printed ones, resulting in the algorithm being overfit to the training samples. In order to address this problem, the classifiers should be re-trained and updated using new samples or features, which might be infeasible for the deployed systems. In order to address this issue, the proposed algorithm views HMC as a machine-printed detection problem and resolves it by a template matching approach. A gallery containing characters of different fonts are created. Then, after performing preprocessing and word-block level segmentation a parallelised iterative algorithm is utilise to detect and then exclude machine-printed text. The algorithm is very robust against noise, missing data due to binarisation and overlapping text, capable of discriminating very similar handwritten text blocks to machine-printed. Quantitative evaluations have been conducted over the publicly available Pattern Recognition & Image Analysis Research Lab-Natural History Museum (PRImA-NHM) dataset [3], giving compelling results as compared with the state-of-the-art. The main contributions of the proposed HMC algorithm are as follows: 1) As opposed to [4], [5], [6], [7], [8], [9], [10], the proposed algorithm is training-free as does not rely on a trainable classifier. This provides the capability to update the gallery samples, without re-designing or modifying the decision making step. The application of the proposed gallery generation can be extended to perform HMC for other languages. 2) Unlike the geometry-based feature extraction algorithms used by [6], [7], [10], [11], [12], [13], which are sensitive to missing data, occlusion or cluttered text, the proposed approach is robust against occlusions and noise over the scanned text. 3) The algorithm is based on a well-known template matching technique with numerous publicly available CPU and GPU implementations.

The paper proceeds to give a literature review in section 2. Our proposed template matching approach is presented in section 3. The experimental results are discussed in section 4. Finally, section 5 concludes the paper.

2. Literature Review

An Eigenface-based approach has been proposed for the HMC task in [2]. As in [15], first, principal compo-

• *This work is funded by Innovate UK under the grant number 101779.*

nent analysis (PCA) is used to create a set of normalised characters in different font styles. Then using the fact that the printed characters are reasonably clustered more tightly than the handwritten ones, a threshold is chosen for decision making. The algorithm to classify handwritten and printed texts is character-based and an adaptive method is used to split the characters. In [14], K-nearest neighbour classifiers are trained for HMC using features derived from statistical information of word segments. Based on the fact that the handwritten text generates more Radon components than the machine-printed, Zemouri *et al.* [16] use the Radon transform for feature extraction and train support vector machines (SVMs) to detect handwritten or machine-printed classes. HMC is then performed by incorporating the Rough set theory. Kumar *et al.* [17] also used SVM to detect handwritten text. After a Voronoi segmentation step, Canny edge detector is applied and Triple-Adjacent-Segment (TAS) is used to extract features from the edges. In [10], the input text is assumed to contain three different regions: 1) the within document text; 2) the highlighted or underlined parts; 3) the marginal notes. First, Density-Based Spatial Clustering of Application with Noise (DBSCAN) algorithm is used for clustering an input image. Then the rectangularity of the segments are measured and classified using a decision tree classification algorithm. A boosted version of Bonsai trees is defined for multi-class classification in [7], which based on the given results, outperforms a multi-class SVM for text classification. The features consist of geometrical, statistical moments, and contours histograms. In [6], eleven features mainly based on the ratio of the statistical and geometrical features of the segmented words to the geometrical size of their bounding boxes (width or height) are extracted from the regions within the bounding boxes.

There are a number of issues associated with the algorithms explained above, which limit their capability for a robust HMC. First, they lack "generalisability", i.e. a significant number of the previous algorithms are designed for specific types of documents. This is because of the algorithms' sensitivity to the type of training data. The second issue relates to their training-based frameworks. The trained machine learning methods require to be re-trained for any new data, resulting in the deterioration of the algorithms flexibility. In addition to this, the algorithms can overfit the training data and fail to correctly classify unseen images. Furthermore, the sensitivity to the shape deformations resulted from noise, binarisation, clutter and occlusions by overlapping text can significantly reduce the classification performance of, particularly, geometrical features-based HMC approaches.

A novel unconventional, yet very successful approach is proposed in [3], [4]. After performing preprocessing and document segmentation, scale-invariant feature transform (SIFT) [18] is applied to each text block. Post-processing is then performed to reduce the number of feature points, for which a visual word is selected from the Bag of Words (BoW). Two SVMs (with radial basis function kernels) are trained and used to classify each segment into three classes: 1) handwritten; 2) printed; 3) noise. In terms of classification, the algorithm outperforms multi-class SVM and Random Forest approaches and in terms of features separability for each class, the approach is more robust for handling text lines and noise, and less vulnerable to the segmentation failures than the baseline Gabor features. Nevertheless, as will be shown in the experiments section, our approach outperforms the method proposed in [3] by over 13% in classification accuracy and gives similar results as in [4] despite using no labelled training data.

3. Our Proposed HMC Approach

Our proposed template matching approach is illustrated in Figure 1 which basically consists of three steps: pre-processing, gallery creation and parallelised template matching. Each of the steps is described in more details below.

3.1. Preprocessing: Denoising, Text Block Segmentation and Alignment

The scanned document images are first pre-processed by using median filtering with mask size 5×5 to reduce the spike noise effects caused by scanners. Then to enhance the performance of the word-block segmentation and reduce the over-segmentation, the possibly existing lines and circles caused, for example, by tables and paper punch marks are detected and removed. Line detection is performed using the Hough transform. In order to detect and exclude the circles from the images, first, all the connected components are detected. Then for each connected component $\mathbf{CC}_i$, its area A_i and perimeter R_i are computed. If $\mathbf{CC}_i$ corresponds to a circle, the ratio of $A_i/{R_i}^2$ should be $\approx 1/4\pi$. Therefore, if $D_i = \left| \frac{A_i}{{R_i}^2} - \frac{1}{4\pi} \right|$ is lower than a given threshold, $\mathbf{CC}_i$ is labelled as circle and discarded from the image. Then the line segmentation algorithm in Google's Tesseract is used to segment each word-block [19]. The segmentation algorithm is capable of isolating rotated and skewed regions by generating the coordinates of a rectangle around the text segments. The segmentation is provided in various sizes, such as single word-length, sentence-length (combination of adjacent words) or paragraph-length (combination of text lines). In order to accelerate the parallelisation performance by reducing the computation times over each text segment, the smallest segment types, which is the single word-length is used in our work. Since the segments can have different poses, due to particular rotated text blocks, such as the stamp marks, or slight rotations when the scanning is performed, an alignment algorithm is applied to each segment. For different angular rotation θ_i selected from the range of $[0°, 180°]$, the lengths of the horizontal (P_h^i) and vertical (P_v^i) projections of the i^{th} text block $\mathbf{S}_i$ is computed. The ratio P_v^i/P_h^i peaks at the optimal alignment angle θ^{opt}, in which the segment is horizontally aligned, is computed as follows,

$$\theta^{opt} = \underset{\theta_i}{\text{argmax}} \left(P_v^i / P_h^i \right) \tag{1}$$

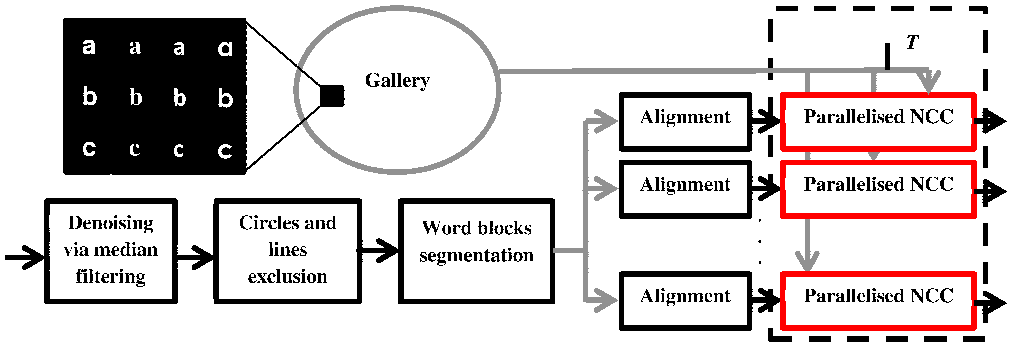


Figure 1: Our proposed template matching-based HMC approach.

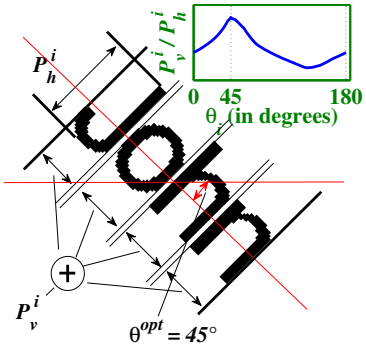


Figure 2: An example of the alignment step over a segmented text block.

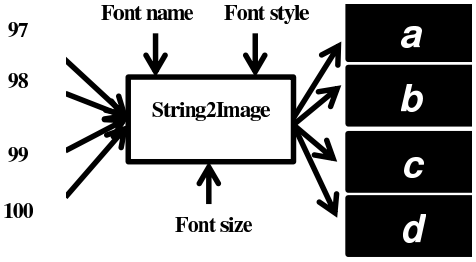


Figure 3: An example of an application of the `String2Image` function. Inputs: ASCII codes 97, 98, 99 and 100, font name: Arial, font size: 25 (pixels), font style: italic; Outputs: images of 'a', 'b', 'c' and 'd' in the given font specifications.

An example of this procedure is shown in Figure 2. The input image is rotated for 45°, where P_v^i/P_h^i reaches a peak at $\theta^{opt} = 45°$, as illustrated at the top right corner.

3.2. Gallery Creation

In order to create a gallery of machine-printed text, a function called `String2Image` is defined, which takes as input the American Standard Code for Information Interchange (ASCII) code of every character, a font name, style and size. It then estimates the size of an image in pixels, which fits the character with the given size and creates an image containing the given character according to the given font style and name. In other words, this procedure maps its input, which is an ASCII code, to the pixel space of a binary image, whose foreground shows the input character. An example is illustrated in Figure 3. By repeating this procedure for different characters and font attributes, a large gallery containing various characters written in different machine fonts can be generated.

3.3. Parallelised Template Matching

The gallery samples are matched with the segmented regions using the normalised cross-correlation (NCC) algorithm, which is a well-known extensively used technique for template matching [20]. Although NCC has the advantage of being occlusion-robust, particularly for binary images, being translation invariant and capable of generating normalised matching scores, it has two main disadvantages. It is not a scale- and rotation-invariant matching algorithm. Also, since it requires 2D convolution calculation of the template over the whole image domain, it has high computational complexity. In order to address the sensitivity of NCC to the pose variations, the alignment algorithm explained in section 3.1 is applied to each word-block, prior to the NCC computation. To decrease the sensitivity of NCC to the scale variations, three methods are used. The first, utilises the `String2Image` function explained in section 3.2 to create a gallery of characters in various sizes (**PNCC3**). Once the correct character size is matched, the output of NCC is maximised. Alternatively, instead of adding characters with various sizes to the gallery, the correct font size is estimated by resizing the gallery samples, which initially, all are in a same size. This approach is implemented using two different procedures. The first, resizes the gallery images to have the same number of size as the current word-block segments (**PNCC1**). For the other approach, the gallery templates are incrementally resized using the nearest neighbour interpolation from a given initial size (which in this work, is $0.5\times$ the number of the current word-block segment rows) to eventually have the same number of rows as the segmented text block (**PNCC2**).

Finally, the parallelised architecture illustrated in Fig-

ure 1 is used to decrease the computation cost of NCC. The parallelisation is performed in two macroscopic and microscopic phases. In the macroscopic scale (shown in dashed gray in Figure 1), since the processing over each segmented text block is independent from the other blocks, the NCC matching over the text blocks is performed in parallel. On the other hand, in the microscopic scale shown in red in Figure 1, the mask sliding procedure of NCC is computed in parallel for every region over the segmented text block. Once the output of NCC (the matching score) at every instance of the parallelised algorithm, is higher than a threshold T, the parallel calculation of NCC stops and the region underlying the mask, in which highest matching score is achieved is labelled as machine-printed and excluded from the image. Otherwise, if for all of the gallery samples the matching scores are lower than T, the text block is labelled as handwritten. As a result of this parallelisation procedure, in comparison with a serialised direct NCC calculation over the text blocks, the average computational speed is increased ≈ 9 times and ≈ 15 times for a CPU and GPU implementation, respectively. The NCC equation used in this work is as follows,

$$\begin{aligned} &\mathbf{C}_{i,j} = \mathbf{G}_j \star \mathbf{S}_i \\ &\mathbf{C}_{i,j}(x,y) = \frac{\sum_{x',y'} (\mathbf{S}_i(x',y') - \mu_i)(\mathbf{G}_j(x'-x,y'-y) - \lambda_j)}{\sqrt{\sum_{x',y'} (\mathbf{S}_i(x',y') - \mu_i)^2 \sum_{x',y'} (\mathbf{G}_j(x'-x,y'-y) - \lambda_j)^2}} \end{aligned} \tag{2}$$

in which $\mathbf{G}_j$ is the j^{th} gallery image, whose average is λ_j, μ_i is the mean of those parts of the i^{th} text segment $\mathbf{S}_i$ underlying $\mathbf{G}_j(x+x', y+y')$ and is a function of x and y, $\star$ is the NCC operator and $\mathbf{C}_{i,j}$ is the matching image, whose elements $\mathbf{C}_{i,j}(x,y)$ are the matching scores in the range of $[0,1]$. The location and value of the maximum of $\mathbf{C}_{i,j}$ is computed as follows,

$$\begin{cases} [x_m, y_m] = \underset{x,y}{\text{argmax}} \mathbf{C}_{i,j} \\ C_{i,j}^{max} = \underset{x,y}{\max} \mathbf{C}_{i,j} \end{cases}, \tag{3}$$

in which $[x_m, y_m]$ gives the location where the maximal matching with value $C_{i,j}^{max}$ occurs. As illustrated in Figure 4, if for all gallery samples $C_{i,j}^{max} \leq T$, $\mathbf{S}_i$ completely contains handwritten text. On the other hand, if for a gallery sample $\mathbf{G}_j$, $C_{i,j}^{max} > T$, $[x_m, y_m]$ and $\mathbf{G}_j$ are utilised to crop the maximal matched region from $\mathbf{S}_i$ and the same process is repeated over the updated $\mathbf{S}_i$.

4. Experiments

4.1. Datasets

The Pattern Recognition & Image Analysis Research Lab-Natural History Museum (PRImA-NHM) dataset [3], containing 100 images of both handwritten and machine-printed text, is used to evaluate the proposed algorithm. The overall number of handwritten and machine-printed segments are 100 and 415, respectively. In order to binarise the coloured images in the dataset, they are first mapped to the YCbCr colour space, then Otsu's automatic thresholding is utilised to create the binary images. The locations of the surrounding pixels for each handwritten or machine-printed region has been provided in the dataset, using the Page Analysis and Ground-truth Elements (PAGE) format framework [21]. In order to create the ground truth for our quantitative evaluation, first the locations are read from the provided XML files. The convex hull of the points are then computed and the (polygonal) region of interests are converted to regional masks. This process creates the binary segmentation maps. Computing the convex hull assures that each binary mask contains the whole region of the corresponding handwritten or machine-printed segment.

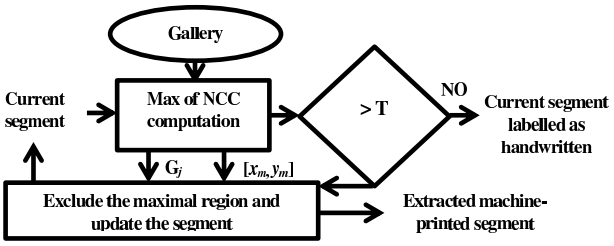


Figure 4: The parallelised NCC algorithm to perform handwritten detection via machine-printed text exclusion.

Figure 5: Gallery characters for Arial, Times New Roman and Calibri fonts, in normal, italic and bold at the first, second and third three lines for each font, respectively.

4.2. Gallery Creation and the HMC Results

The threshold used for circle detection is 0.05. The AddTextToImage Matlab code[1] is modified to create the gallery images, while the publicly available Tesseract OCR[2] is embedded for the word-block segmentation. The characters in the gallery are created using the `String2Image` function explained in section 3.2. The gallery used for the following experiments contains Arial, Times New Roman and Calibri fonts, all with size 50 and normal, italic and bold styles, as illustrated in Figure 5. Because of the possible high similarity between the vertical lines, which may exist in the handwritten segments and '1', 'I', 'l', 'j' and 'i', these characters are excluded from the gallery samples to reduce false positives for the handwritten class.

The decision making threshold T, which is applied to the maximum output of NCC, determines whether a segment is machine-printed or handwritten. If the NCC maximum is

1. mathworks.com/matlabcentral/fileexchange/40959-add-text-to-image
2. https://code.google.com/p/tesseract-ocr/

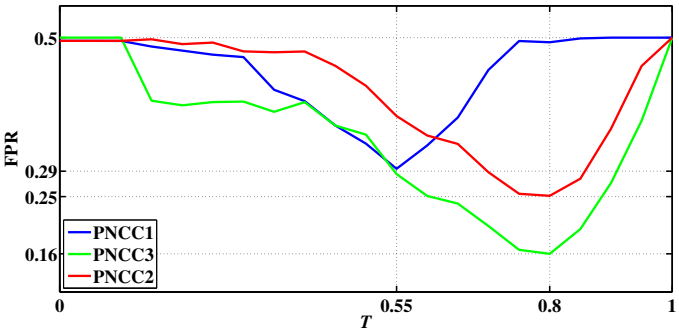
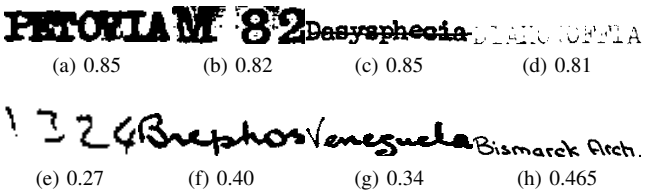


Figure 6: FPR vs. T: the plot shows the effects of varying T over the classification performance.

lower than T for all the gallery characters, the segment is labelled as handwritten and otherwise, as machine-printed. The value for T is highly effective on the false positive rate (FPR). A too small value for T can result in a high FPR for the machine-printed class, while bigger values tighten the decision boundary for the machine-printed class, resulting in machine-printed segments being labelled as handwritten, which increase FPR for the handwritten class. Therefore, there should be an optimal point for T when plotted against FPR, as verified in Figure 6.

The performance of the proposed HMC approach is evaluated for different values of T for all three methods PNCC1, 2 and 3. Since the number of samples for the two classes are unequal in the used dataset, in order to have a fair comparison for the classification performance of both classes, instead of dividing the number of incorrect classified samples to the overall existing samples in the dataset, the average of FPR for each class is reported, i.e. FPR $= 0.5 \times (\text{FPR}_\text{H} + \text{FPR}_\text{P})$. When $T = 0$ and $T = 1$, all the samples are classified as handwritten or all as machine-printed, respectively, resulting in FPR $= 0.5$. When $0 < T < 0.55$, PNCC1 outperforms PNCC2. The reason is that directly resizing the templates to the size of the current segment may increase false detections for the handwritten class. However, for $T > 0.55$, PNCC2 provides a higher performance in maintaining FPR for both classes. The resizing procedure of PNCC1 and PNCC2 can match a template to a region of a handwritten segment, which is similar to a machine-printed text, resulting in misclassification of the segment and deterioration of FPR. PNCC3 avoids this error by incorporating different sizes of characters in the gallery. The best performances for PNCC1, PNCC2 and PNCC3 occur at $T = 0.55$ and $T = 0.8$, with FPR 0.29, 0.25 and 0.16, respectively. In order to check the algorithm does not overfit to the PRImA-NHM samples, similar thresholds are verified over another dataset containing scanned clinical documents (Due to the sensitivity of the data, the results on the clinical dataset is unpublishable).

Some examples of the correctly classified (binarised) samples from the PRImA-NHM dataset are shown in Figure 7. Figure 7-a and -b show cluttered segments, whose characters have been merged due to the scanning error.



(a) 0.85 (b) 0.82 (c) 0.85 (d) 0.81

(e) 0.27 (f) 0.40 (g) 0.34 (h) 0.465

Figure 7: Challenging examples in the PRImA-NHM dataset and their matching scores classified by PNCC2 with $T = 0.8$: (a-d) Machine-printed; (e-h) Handwritten.

Algorithm	Performance	Samples
PNCC1 PNCC2 PNCC3	70.6% 74.9% **84.0%**	No training samples required
Zagoris *et al.* [4] Gabor features [3]	**84.2%** 70.6%	15% train 85% test

TABLE 1: Comparison of the classification performances over the PRImA-NHM dataset.

The noise in the data resulted in binarisation failure and missing data in Figure 7-d, while the overlapping line created an occluded segment in Figure 7-c. On the other hand, although the handwritten segments shown in Figure 7-e to -h have close regularities with machine-printed text, they are correctly classified as handwritten since none of the gallery characters can produce a matching score higher than the threshold T. Because of the high similarity with the machine-printed sample, the geometric features, such as area and rectangularity used in [6], [7], [10], [12], fail to create separable classes and the samples will wrongly be classified as machine-printed.

Table 1 shows the comparison of our approach with the results provided by Zagoris *et al.* [3], [4], in which 15% of the samples in the PRImA-NHM dataset are utilised for train and the remaining 85% for the test phase. The bounding boxes-based PRImA Layout Evaluation Framework [4], [22] is used to compute overlapping regions of the classified segments with the ground truth. Although the proposed PNCC algorithms do not use any training samples, their performance is only slightly lower than the one reported in [4] and $\approx 14\%$ higher than Gabor features [3].

5. Conclusions

This paper proposes an algorithm to discriminate the handwritten text from machine-printed ones. Because of significantly higher variations in the handwritten text samples, the HMC problem is resolved using a machine-printed detection approach, which incorporates a gallery consisting of machine-printed samples. After an initial preprocessing over the scanned document images, including denoising and circles/lines exclusion, word-block level segmentation is performed. Then a parallelised algorithm is utilised to align and match gallery characters with the the segmented regions

using a parallelised NCC. The results show high robustness of our proposed approach in classifying cluttered, occluded and noisy samples in addition to those with significant high missing data. The proposed approach is training-free, in the sense that no training sample is required for a classifier. The gallery creation procedure makes the algorithm flexible to new font specifications, new characters, or even other languages. Although the algorithm proposed in this paper is generic, for the applications that the font specifications (name, style or size) is initially known, the size of the gallery can be reduced by limiting its samples to only the available font types. Also, for the cases in which the rotations of the test segments are already known, for example for the border texts, which are usually $\pm 90^{\circ}$ rotated, the alignment can be omitted. Instead, given the rotational angle of the text segments, the gallery can be extended by adding the manually rotated versions of the characters, which result in higher computational speed. HMC can be viewed as a pattern recognition problem with imbalance class distributions, in the sense that one class has limited known number of samples, while the other is sampled from an unlimited space. This is seen in other applications such as spam detection, one-class classification or irregularity detection problems. The similar methodology proposed in this paper, which is based on generating and matching of the sample of the known class, can be applied to resolve such problems. In addition to this, the explained method to create gallery of characters using different font specifications can be used to generate labelled data for those pattern recognition algorithms which rely on huge amount of labelled samples, such as the deep neural networks.

The proposed algorithm in the paper can potentially be improved in several respects, particularly by adding a font detection step and embedding a more robust alignment algorithm to overlapping text. A successful font detection step can reduce the number of false positives for the machine-printed class and reduce the search time for the correct template from the gallery. The robustness of the algorithm in discriminating the overlapping samples can be further evaluated using more challenging datasets.

References

[1] S. Brunessaux and P. Giroux and B. Grilheres and M. Manta and M. Bodin and K. Choukri and O. Galibert and J. Kahn, "The Maurdor Project: Improving Automatic Processing of Digital Documents," in 11^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2014, pp. 349-354.

[2] S. Pinson and W. Barrett, "Connected component level discrimination of handwritten and machine-printed text using Eigenfaces," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1394–1398.

[3] K. Zagoris, I. Pratikakis, A. Antonacopoulos, B. Gatos, and N. Papamarkos, "Handwritten and machine printed text separation in document images using the bag of visual words paradigm," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 103–108.

[4] K. Zagoris, I. Pratikakis, A. Antonacopoulos, B. Gatos, and N. Papamarkos, "Distinction between handwritten and machine-printed text based on the bag of visual words model," *Pattern Recognition*, vol. 47, no. 3, pp. 1051–1062, 2014.

[5] E. Zemouri and Y. Chibani, "Machine printed handwritten text discrimination using Radon transform and SVM classifier," in *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2011, pp. 1306–1310.

[6] L. da Silva, A. Conci, and A. Sanchez, "Automatic discrimination between printed and handwritten text in documents," in *XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, 2009, pp. 261–267.

[7] Y. Ricquebourg, C. Raymond, B. Poirriez, A. Lemaitre, and B. Coasnon, "Boosting bonsai trees for handwritten/printed text discrimination," in *Proc. SPIE*, 2013, pp. 5–12.

[8] J. Eduardo Bastos Dos Santos, B. Dubuisson, and F. Bortolozzi, "Characterizing and distinguishing text in bank cheque images," in *XV Brazilian Symposium on Computer Graphics and Image Processing*, 2002, pp. 203–209.

[9] J. Guo and M. Ma, "Separating handwritten material from machine printed text using hidden Markov models," in *Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 439–443.

[10] A. Mazzei, F. Kaplan, and P. Dillenbourg, "Cognitive and social effects of handwritten annotations," in *Red-conference, rethinking education in the knowledge society*, 2011.

[11] U. Patil and M. Begum, "Word level handwritten and printed text separation based on shape features," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 590–594, 2012.

[12] Y. Zheng and H. LI and D. Doermann, "Machine printed text and handwriting identification in noisy document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 337–353, 2004.

[13] P. Barlas and S. Adam and C. Chatelain and T. Paquet, "A typed and handwritten text block segmentation system for heterogeneous and complex documents," in 11^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2014, pp. 46–50.

[14] M. Hangarge, K. C. Santosh, S. Doddamani, and R. Pardeshi, "Statistical texture features based handwritten and printed text classification in south Indian documents," *Computer Vision and Pattern Recognition*, vol. 1, no. 32, 2013.

[15] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[16] S. Narayan and S. Gowda, "Discrimination of handwritten and machine printed text in scanned document images based on rough set theory," in *World Congress on Information and Communication Technologies (WICT)*, 2012, pp. 590–594.

[17] J. Kumar, R. Prasad, H. Cao, W. Abd-almageed, D. Doermann, and P. N. A, "Shape codebook based handwritten and machine printed text zone extraction," in *Document Recognition and Retrieval XVIII*, 2011, pp. 1–8.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2002.

[19] R. Smith, "An overview of the Tesseract OCR engine," in *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, 2007, pp. 629–633.

[20] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," *Proc. SPIE*, vol. 4387, pp. 95–102, 2001.

[21] S. Pletschacher and A. Antonacopoulos, "The PAGE (Page Analysis and Ground-truth Elements) format framework," in *International Conference on Pattern Recognition (ICPR)*, 2012, pp. 257–260.

[22] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Scenario driven in-depth performance evaluation of document layout analysis methods," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2012, pp. 1404–1408.