2014 11th IAPR International Workshop on Document Analysis Systems

Forgery Detection Based on Intrinsic Document Contents

Amr Ahmed
Faculty of Computer Science and Engineering
German University in Cairo, Egypt
amora.1136@gmail.com

Faisal Shafait
School of Computer Science and Software Engineering
The University of Western Australia, Crawley WA, Australia
faisal.shafait@uwa.edu.au

***Abstract*—Nowadays, Document forgery detection is becoming increasingly important as forgery techniques are becoming available even to untrained users. Hence, documents that do not contain any extrinsic security features (e.g. invoices) have become easier to forge. We previously presented a method to detect manipulated documents based on distortions introduced during the forgery creation process. In this paper, several approaches are explored to improve accuracy and time taken to detect forgeries based on document distortions. The main idea behind the presented approaches is to automatically identify which parts of a document belong to the template (and hence would remain static across different documents originating from the same source) and then detect distortions in those parts only. An improvement up to 29% in accuracy of forgery detection is observed compared to our previous work. Furthermore, we also present an approximation of the original method that results in a reduction in run time of the method by several orders of magnitude, while having only a marginal reduction in its accuracy.**

I. Introduction

Document forgery is a common problem affecting many areas of our daily lives. For example, customers may present fake documents to banks in order to obtain a loan or even present tampered documents to insurance companies in order to obtain the insurance amount. Several approaches may be used to falsify a document. For example, documents may be copied using copying machines, remade using word processing software or scanned, digitally modified and finally printed [1]. During the scanning and printing processes involved in creating scan/print type forgeries, some distortions are introduced in the document. The most important distortion, which is the focus of this paper, is non-uniform vertical scaling. When a printed document is scanned and printed again, the contents of the re-printed document have slightly different vertical distances as compared to the original one [2]. This effect can be observed in a direct comparison with the original document. Similarly, remaking a document in a word processing program is also likely to introduce alignment imperfections.

Creating every day documents in a more secure way (e.g. by introducing holographic images [3] or specialized printing techniques [4]) incurs extra costs. Therefore, most of such documents are created using off-the-shelf paper and printers. On the other hand, many of these documents are repeatedly produced (e.g. medical invoices from a specific hospital, or repair costs from a specific workshop). In scenarios where one has access to a number of documents from the same source, one can leverage that to use a model-based document authentication algorithm. van Beusekom et al. [5] used a simplified approach to align documents coming from the same source. The major challenge in aligning different documents (e.g. invoices) from the same source (e.g. a doctor) is to distinguish between the static parts of the document (i.e. header and footer) and the non-static part (the actual content of the invoice). Different invoices from the same source might have nothing in common, apart from the issuing party details. Note that although it is likely that the actual forgery would be done in the non-static part of the document (e.g. modifying the date or amount of an invoice), we aim at catching the distortions in the static part of the documents as a by-product of the forgery creation process. The preliminary approach developed in this direction [5] required manually marking the static and non-static parts of the document. This restriction was removed in [2] to be able to effectively detect document distortions without distinguishing its static and non-static parts.

In this paper, we develop an *automatic* approach for identifying static parts of a document and show that it significantly improves the results of [2]. First, we briefly introduce the DocAlign algorithm [2] in Section II. Then, Section III presents different approaches for improving accuracy of DocAlign using automatic identification of static parts of documents. An approximation of the DocAlign algorithm is presented in Section IV. Experimental results are given in Section V, followed by a conclusion in Section VI.

II. Document Alignment (DocAlign) Algorithm

We begin by presenting the original work [2] briefly for completeness. Documents in the training set are matched to each other using the RAST algorithm (see Section II-A) producing a matrix of pairwise matching results. Afterwards, the test document is matched to all documents in the training set in a pairwise fashion using the RAST algorithm. A summed score (of matching every document in the training set to the test document) is calculated and a variant of Grubbs test (see Section II-B) was run to detect the outliers. The main aim of the algorithm is to mark a forged document as an outlier.

978-1-4799-3243-6/14 $31.00 © 2014 IEEE
DOI 10.1109/DAS.2014.26

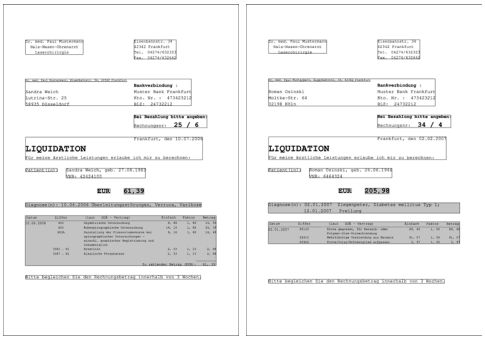

Fig. 1. Blocks varying in size are highlighted in grey. This work aims at automatic removal of such variable parts of documents to align static parts only.

A. RAST algorithm

The RAST term stands for Recognition by Adaptive Subdivision of Transformation space. In this algorithm, two images are matched together and a matching score is computed. This matching score can be seen as the number of identical characters that have the same position in both document images. The algorithm aligns them in such a way that the matching score is maximized. The alignment consists of defining the transformation space with parameters (t_x , t_y, s, a) where t_x and t_y are translations in the x and y directions respectively, s is the scale factor and a is the rotation angle required for optimal alignment of the two documents. The transformation space is initialized given by $[t_{x_{\min}}$, $t_{x_{\max}}] \times [t_{y_{\min}}$, $t_{y_{\max}}] \times [s_{\min}$, $s_{\max}$] $\times$ $[a_{\min}, a_{\max}]$, where these ranges are the initial ranges of the parameters. An optimal branch-and-bound search algorithm is used to find the optimal parameters set (see [5], [6] for details).

B. Variant of Grubbs Test

In the original Grubbs test, the test searches for an outlier in the data set, removes it and then continues searching for other outliers and removes them. The test stops if no more outliers are found or if the data set size becomes six [7]. In our variant of the Grubbs test, the test is done only on the incoming test image to determine whether it is an outlier or not (since all other documents are genuine and thus there is no need to do the test on them.) As an evaluation measure, the true positive rate (r_{t_p}) and the true negative rate (r_{t_n}) are determined where

$$r_{t_p} = t_p/(t_p + f_n)$$

and

$$r_{t_n} = t_n/(t_n + f_p)$$

t_p, t_n, f_p, f_n are the number of true positives, true negatives, false positives and false negatives respectively.

	Forged	Genuine
Outlier detected	True positive	False positive
No outlier detected	False negative	True negative

III. Accuracy Improvements in DocAlign

A. Layout Filtering

This approach aims at filtering out layout varying blocks and leaving the layout static ones by means of a mask before matching the documents using RAST algorithm. Layout static blocks are defined to be blocks that do not vary greatly in size or in position across different documents. In this approach, the documents in the training set are segmented into blocks using the XY-cut algorithm [8] (thus extracting the documents layouts where a document layout is the set of blocks forming a document.) The layouts are then matched together in a pairwise fashion. In this comparison the Hungarian algorithm is used to match blocks of one layout to blocks of another layout (Figure 2) and this matching of layouts is assigned a score which represents how close the layouts are to each other (the Hungarian algorithm aims at maximizing this score). The cost matrix of the Hungarian algorithm is first constructed where cell (x,y) in this matrix represents how similar block x in layout one and block y in layout two are. If there is an overlap between the two blocks, similarity score is the number of overlapping pixels. If there is no overlap, the Manhattan distance d is calculated between the centers of the blocks and similarity score is

$$cost = \alpha(1 - \frac{d}{\beta})$$

We empirically set $\alpha = 500$ and $\beta = 5000$ in our experiments. The obtained value has two interesting properties. First, as the distance between the block centers decreases the similarity score increases. Second, the value obtained is relatively low in comparison to the number of overlapping pixels between the blocks and thus it is guaranteed that if there is an overlap between the two blocks currently considered, it will have a higher contribution and only when the block does not overlap with any other block, the distance will come into play (searching for the nearest block to it). Finally if the number of blocks was not equal, dummy blocks are added to the document with the lower number of blocks and the similarity measure between any block and a dummy block is zero. The similarity score of two layouts is the sum of similarity scores of blocks that matched together. A summed score is then calculated for each layout in the training set (matching a layout in the training set to all other layouts in the same set) and the layout with the highest summed score is determined. This is called the data set representative layout. Each block from this layout is then compared to its corresponding blocks (to which it was matched by the Hungarian algorithm) in the other layouts in the training set. If the difference in size of the two blocks (that in the representative layout and its corresponding block in another layout) is within a fixed threshold (at most 10% of the size of the smaller block), a block score for the block in the representative layout is incremented by one. Blocks of the representative layout scoring above a certain threshold (threshold chosen manually to be 80% of the size of the training set) are called layout static blocks. Those

blocks form a mask that, when applied to another document, leaves everything in the position of the layout static blocks in the representative layout and removes other blocks and thus extracts the layout static blocks from all the documents (training and test set.) It is assumed that the corresponding blocks in other documents will have roughly the same position as in the representative layout. After applying the mask to all documents, documents are matched using the RAST algorithm as in DocAlign [2] .

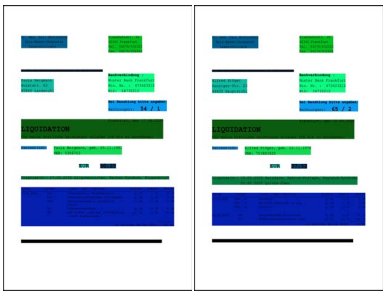

Fig. 2. After establishing correspondence between blocks of two pages, matching blocks are shown in the same color.

B. Layout + OCR Filtering

In this approach, the layout mask extracted in the previous approach is refined. This is done by removing blocks that are not OCR static. OCR static means that the text contents of the blocks are relatively fixed and not varying across the documents. Thus in the representative layout, layout-static blocks previously determined are considered one by one, their text contents are compared to the text contents of their corresponding blocks in the rest of the training set using the Levenshtein distance algorithm. The similarity score of matching the texts of two blocks using the Levenshtein distance algorithm is the minimal number of edits required to transform one text to the other. The cost of insertion, deletion and substitution of a character is set to one. It should be noted that such a score should be normalized first by dividing by the length of the longer (cost now ranges from zero to one after normalization.) A summed score of matching the text of one of the layout static blocks in the representative layout to all other texts of corresponding blocks in the training set is determined. The summed score of each block should range between zero (if the considered block in the representative layout had the same text as all the corresponding blocks in all the other documents) and size of training set minus one (the case when each block in the training set documents has a completely different text such that the normalized Levenshtein distances are always one, except when block compared to itself the score will be always zero.) Afterwards a threshold is set, below which the block is considered to be an OCR static block. The threshold chosen in this context is 4% of the size of the training set. This can be thought of as allowing text variations between corresponding blocks up to 4%, above that the block is considered as an OCR non static block. After determining the OCR static blocks, the mask extracted in using the layout filtering approach is refined so as to include only OCR static blocks. This mask is applied to all the document images (training and test set) and anything outside this mask is removed.

C. Manual (Semantic) Filtering

In this approach, a mask was manually created to extract only blocks corresponding to the headers and footers of the documents. In bills and vouchers, it is expected to find such headers and footers that repeat across different documents from the same source. For example, the block containing the name of the bill issuer, the block containing his/her address and also the block of the bank account number were found to repeat across the doctor bills data set used in this work. The manually created mask was applied to the whole data set to extract these three blocks.

IV. Computation Time Reduction in DocAlign

In the previous approaches, incoming documents were matched to all documents in the training set. The drawback of this method is that it takes a lot of time in addition to being dependent on the size of the training set. To reduce the computation time at the testing stage, a representative of the training set is first determined (as in the Layout filtering approach). Then, the test image is matched only to the single representative document chosen from the training set. This match score is given to the Grubbs outlier detection method along with the pre-computed scores of matching the training set representative to all of the training set images. This approximation can be thought of in the following way: if the scores of matching the training set images to the training set representative are present, does the score of matching the test image to the training set representative fit into these scores or is it an outlier?

V. Results and Analysis

A. Experimental Setup

Experiments were done on a doctor bills data set used in DocAlign [2]. It consists of 40 genuine documents, 40 copied documents, and 12 forged documents. As a preprocessing step, documents were binarized using Otsu algorithm [9], deskewed as in [10], and finally their page frames were aligned together. Page frame detection involves detecting the frame surrounding the documents contents [11]. Detecting such frame helps removing the noise (that may occur at the margins of the document images) as well as aligning the documents together by shifting the page frames of different documents so that their top left corners coincide. This is an important step to eliminate document tilting and shifting that may occur while scanning the documents.

In this paper, a variant of the N fold cross validation was used throughout the experiments. The genuine images were divided into n folds. One fold was taken as the training set while the remaining n-1 folds in addition to the copies and forgeries were taken as the testing set. Experiments were then repeated n times, each time a different fold of the genuine

images was taken as the training set and the rest of the documents were taken as the testing set. The true positive, true negative rates and the average testing time were calculated. Results of the different repetitions were averaged. Finally, experiments were repeated with different values of N (N = 2; 3; 4; 5) to test the effect of different training set sizes on the results.

B. Reproducing Results of DocAlign

First, the previous work (DocAlign [2]) was reproduced to set a base line for the results. It can be observed in Figure 3 that reproduced results are generally better than results in DocAlign [2] (with the exception of the true positive rates at n=2 and n=3.) These differences are due to two factors. The first one is that in DocAlign, the library Cuneiform v1.0 was used to extract the OCR information instead of the Tesseract library used in this paper. The second one is that another variant of the Grubbs test was used (see Section II-B).

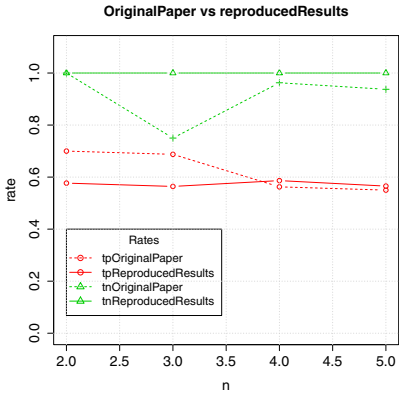

Fig. 3. True positive and true negative rates in the original work [2] and in the reproduced results against number of folds n

C. Analysis of Accuracy Improvements

In this section, the true positive and true negative rates of the different approaches are compared. In the three approaches presented in Figure 4 and 5, the true positive rates have increased compared to the base line of the reproduced results. The highest true positive rate increase was obtained using the layout filtered data set (up to 30% at n = 2.) This is because in that approach layout non static blocks were filtered out. Those blocks vary in size and thus in content as well. Filtering them out leaves blocks that are relatively close in size. The blocks left have in most of the cases the same content or even close content. The non static blocks vary greatly in size (and thus in content) across the documents and thus do not help identifying whether the document is a genuine or a forged one. Incorporating OCR information lead to a slighter improvement (up to 25% at n = 2.) This is possibly due to the effect of the excessive block removal. Documents have lost some of their features (blocks) which make it harder to differentiate between genuine documents and forgeries. Using the manually filtered data set has lead to the slightest improvement in the true positive rate (up to 19% at n = 4.) In this approach, the effect of excessive block removal plays a greater role (only three blocks from an average of 14 blocks per document are present). Regarding the true negative rate (Figure 5), it remains at 1.0 in the reproduced work and the layout-OCR filtered blocks method. In the layout based filtering approach, the true negative rate dropped only to 0.975 at n = 2, otherwise it is 1.0. However, using the manual filtering caused a significant drop in the true negative rate (lies in the interval [0.90, 0.95]) due to an excessive block removal.

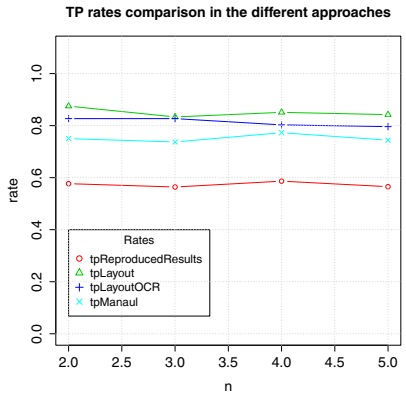

Fig. 4. True positive rate in reproduction of the original work, layout, layout-OCR and manually filtered data set against number of folds n

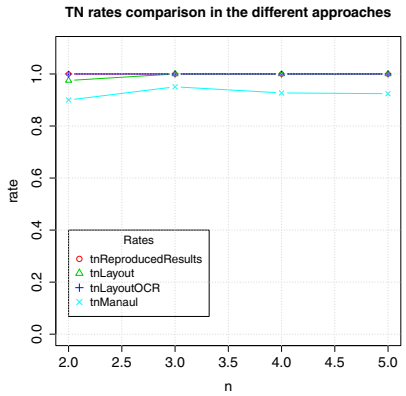

Fig. 5. True negative rate in reproduction of the original work, layout, layout-OCR and manually filtered data set against number of folds n

D. Analysis of Computation Time Reduction

In the previous approaches, incoming documents were matched each to all other documents in the training set. In this approach, the incoming document is matched only to the document with the representative layout in the training set. This approach is aimed mainly at improving the average testing time of incoming documents. In Figure 6, there is a slight drop in the true positive rate when using the single document matching technique at n = 2, 4 and 5. A slight increase in the true positive rate occurred at n = 3. The true negative rate experiences a slight decrease. It lies in the

interval [0.98, 0.99] instead of [1,1]. In the single document matching technique, the new document is matched using RAST algorithm only to the document with the representative layout of the whole training set. This leads to a major reduction in testing time (as shown in section V-D) with a slight degradation in performance in comparison to the results of the DocAlign [2].

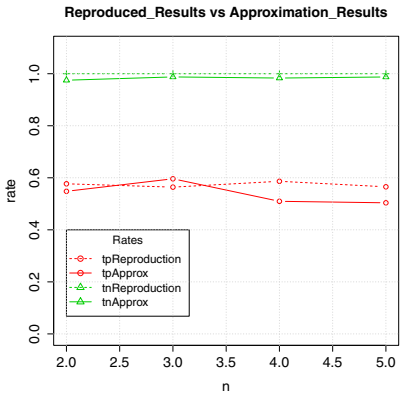

Fig. 6. True positive and true negative rates in reproduction of the original work and in the single document matching approach against number of folds n

To benchmark computation time, all experiments were repeated on a 2.4 GHz AMD Opteron, eight core Unix server. For each of the layout, layout-OCR and manual approaches, average time of classifying an incoming document at number of folds ($n = 2$) was recorded. This is the time taken to compare an incoming document to 20 other documents of the training set partition (total training set size is 40, number of folds is 2). In the single document matching approach, average time of matching incoming documents to the document with representative layout was recorded. In DocAlign, it takes six minutes to classify an incoming document into a genuine or a forgery (see Figure 7). Computation time decreased to four minutes for layout-based filtering method, and to two minutes for the layout+OCR and manual filtering approaches. Using the single document matching approach caused the greatest decrease in the classification time (from 6 minutes to 16 seconds). This is because in this approach, incoming documents are matched only to one document in the training set.

VI. Conclusion

In this paper, several approaches were explored to improve accuracy of model-based document forgery detection and average time taken to classify an incoming document. Documents were filtered based on their contents (layout, layout-OCR and logical contents) removing varying parts. The layout filtering approach caused the greatest improvement in the true positive rate due to removal of non-static blocks. Further filtering in the other two approaches (layout-OCR and manual filtering) caused slighter improvements due to the excessive block filtering effect. The true negative rates remained largely unaffected.

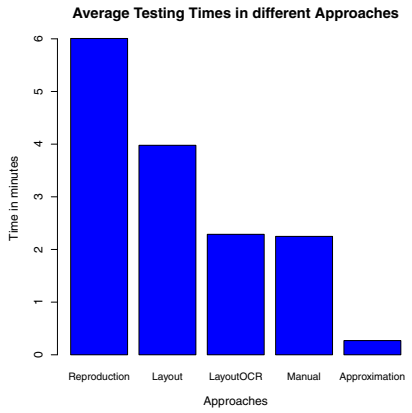

Fig. 7. Average testing time of the different approaches

Thus, layout-based filtering approach can be effectively used to identify static parts of a set of documents. Furthermore, an approximation of the algorithm to match a test document to only one representative training set document caused the average testing time to decrease from 6 minutes to 16 seconds with no major degradation in accuracy of forgery detection.

Acknowledgments

This research work was partially funded by The University of Western Australia's FECM research grant.

References

[1] J. van Beusekom, F. Shafait, and T. M. Breuel, "Text-line examination for document forgery detection," *Int Jour on Document Analysis and Recognition*, vol. 16, no. 2, pp. 189–207, 2013.

[2] J. van Beusekom and F. Shafait, "Distortion measurement for automatic document verification," in *11th Int. Conf. on Document Analysis and Recognition, ICDAR11*, Beijing, China, Sep. 2011.

[3] P. Smith, P. O'Doherty, C. Luna, and S. McCarthy, "Commercial anticounterfeit products using machine vision," *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, vol. 5310, pp. 237–243, June 2004.

[4] I. Amidror, "A new print-based security strategy for the protection of valuable documents and products using moire intensity proles," *Optical Security and Counterfeit Deterrence Techniques V*, vol. 4677, p. 89100, June 2002.

[5] J. van Beusekom, F. Shafait, and T. M. Breuel, "Document signatures using intrinsic features for counterfeit detection," in *2nd Int. Workshop on Computational Forensics*, Washington DC, USA., Aug. 2008.

[6] T. M. Breuel, "A practical, globally optimal algorithm for geometric matching under uncertainty," *Electronic Notes in Theoretical Computer Science*, vol. 46, 2001.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Ground truth data for document image analysis," pp. 199–205, 2007.

[8] F. Shafait and T. M. Breuel, "The effect of border noise on the performance of projection-based page segmentation methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 846–851, Apr 2011.

[9] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys., Man, and Cyber.*, vol. 9, no. 1, pp. 62–66, 1979.

[10] J. van Beusekom, F. Shafait, and T. M. Breuel, "Combined orientation and skew detection using geometric text-line modeling," *Int. Jour. on Document Analysis and Recognition*, vol. 13, no. 2, pp. 79–82, 2010.

[11] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel, "Document cleanup using page frame detection," *International Journal on Document Analysis and Recognition*, vol. 11, no. 2, pp. 81–96, nov. 2008.