Document Authentication using Printing Technique Features and Unsupervised Anomaly Detection

Johann Gebhardt*, Markus Goldstein*, Faisal Shafait† and Andreas Dengel*
*German Research Center for Artificial Intelligence (DFKI GmbH), D-67663 Kaiserslautern, Germany
†School of Computer Science and Software Engineering, The University of Western Australia, 6009 Perth, Australia
Email: first.last@dfki.de; faisal.shafait@uwa.edu.au

***Abstract*—Automatically identifying that a certain page in a set of documents is printed with a different printer than the rest of the documents can give an important clue for a possible forgery attempt. Different printers vary in their produced printing quality, which is especially noticeable at the edges of printed characters. In this paper, a system using the difference in edge roughness to distinguish laser printed pages from inkjet printed pages is presented. Several feature extraction methods have been developed and evaluated for that purpose. In contrast to previous work, this system uses unsupervised anomaly detection to detect documents printed by a different printing technique than the majority of the documents among a set. This approach has the advantage that no prior training using genuine documents has to be done. Furthermore, we created a dataset featuring 1200 document images from different domains (invoices, contracts, scientific papers) printed by 7 different inkjet and 13 laser printers. Results show that the presented feature extraction method achieves the best outlier rank score in comparison to state-of-the-art features.**

I. Introduction

A. Motivation

With today's availability of low-cost scanning devices, high-quality printers and better color copy machines, it gets much easier to manipulate printed documents. For this reason, authentication of printed documents is a task of increasing importance [1]. Today, still a lot of governmental and business transactions depend on paper based documents, for example invoices, contracts, certifications and other official documents, which are all valuable targets for forgery. Another reason for an increasing demand of automated document authentication is that today systems automatically digitize and process documents without any human inspecting the original paper prints [1].

There are several approaches to secure printed documents by adding additional security features, so called extrinsic features. A very popular approach, often used in practice, are watermarks [2]. These extrinsic features usually provide a good protection against forgery, but they are costly and often not included in real-life documents. Therefore, an interesting field of study is the authentication of documents without added security features. In this case, features which describe characteristics of the normal document creation process are used. These intrinsic features do not require to change anything of the document creation process itself [3], [4], which is very useful if the document is created by a third party.

One possible course of action is to identify the device used to create a document, in this case the printer. If somebody is trying to forge or modify a document, he usually has to print the document again on a different printer. A first step towards the identification of the printer is the recognition of the printing technique which was used to print the document. In some cases, the detection of a different printing technique can already result in a successfully detected fraud attempt.

In this paper, a system for detecting documents printed with a different printing technique is presented. The whole process can be split up into two basic steps: (1) Feature extraction and (2) anomaly detection. In the first step, suitable intrinsic features are extracted from the document image to describe the used printing technique. In the second step, features extracted from a set of documents are compared with the goal of identifying documents which are not printed with the same printing technique as the majority of the documents.

In addition to the system, a dataset to evaluate the presented algorithms has been created. In short, the following contributions have been made:

- Features for documents scanned at reasonably low resolution (400 dpi) are proposed,
- The application of unsupervised anomaly detection on the data without the need for prior training,
- A dataset containing a large variety of inkjet and laser printed documents has been created,
- Various experiments have been performed in order to find the best combination of feature extraction and anomaly detection

B. Related Work

Document authentication has been studied by several researchers using different features and approaches. Van Beusekom et al. [4] have presented a system that uses tracking patterns ("yellow dots"), integrated into the printing process by many printer manufacturers, to expose the source of a document. Another approach by the same authors is using text-line rotation and alignment to detect documents that have been changed with a malicious intent [5]. The process of printer or printing technique recognition has also been studied: Mikkilineni et al. [3] have presented an intrinsic and extrinsic approach using graylevel co-occurrence texture features for printer recognition. While the results are promising, the documents have to be scanned with a very high resolution (2400 dpi). Furthermore, a classifier system was used which only works on printers that have been used for training beforehand. Lampert et al. [1] have presented a system that uses local features, such as line edge roughness, area difference and correlation coefficients, focusing on single characters of a document. Again, the documents were scanned with a very high resolution (3200 dpi) and a classifier system which requires training has been used. Printing technique recognition has also been studied by Schreyer et al. [6]–[8] using discrete cosine transform (DCT) features. The evaluation used both, low and high resolution scans. Unfortunately, only one document was used in the evaluation which was printed by different printers. To overcome these shortcomings, our paper is based on a dataset providing a large number of unique documents for every used printer. Furthermore, we eliminate the training process by using unsupervised anomaly detection, which is in our view much more applicable in practical systems.

Anomaly detection is the process of finding data points which do not follow the expected behavior of the majority within a dataset. A good overview of the area is given in the survey paper of Chandola et al. [9]. Basically, anomaly detection algorithms can be divided into three main categories with respect to the availability of labels in the data: Classification based anomaly detection algorithms need to be trained with normal and anomalous data and are similar to traditional machine learning. Semi-supervised algorithms only require normal samples to train a prediction model and finally, unsupervised anomaly detection does not require any training at all. The detection of outliers is performed based on intrinsic characteristics of the data only. The basic idea is that anomalies are rare and different from normal instances with respect to their feature values. More recent algorithms usually score the data instances according to their probability of being an outlier instead of assigning a binary label only.

There are several techniques for unsupervised anomaly detection: Clustering-based anomaly detection clusters the data and uses the distance of each instance to a centroid as an anomaly score. Nearest-neighbor based algorithms use the distance or density with respect to their nearest neighbors as an anomaly score and finally, statistical methods use parametric and non-parametric models to compute the degree of being an outlier. In this paper, the statistical Grubbs' test [10] and the global k-NN, a nearest neighbor based approach, have been evaluated. Additionally, the RaipdMiner[1] anomaly detection extension[2] by Amer et al. [11] has been used for parts of the evaluation.

II. Feature Extraction

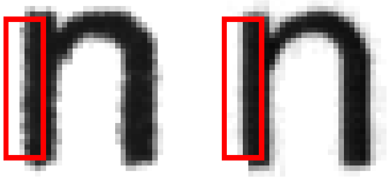


Fig. 1. Comparison of edge roughness between an inkjet (left) and a laser printer (right). The inkjet printer produces a higher degree of edge roughness.

The purpose of the feature extraction is to extract a value (or a set of values) which can be used to classify the used printing technique. Figure 1 illustrates the difference between a character printed by an inkjet and by a laser printer. The inkjet printer produces a higher degree of edge roughness/ degeneration, which holds for many printers [6]. For that reason, this work focuses on extracting features describing the edge roughness of characters.

A. Standard Deviation on Fixed-size Windows

The degree of edge roughness is calculated using the standard deviation of pixel gray values along vertical edges of characters in the document. Rough edges with a lot of change in gray levels along a vertical edge will result in a high standard deviation. The standard deviations are calculated for all connected components with a minimum length vertical edge independently. It is expected that inkjet printed documents have higher deviations than laser printed documents.

1) Preprocessing: As a preprocessing step the connected components of the document image are extracted. This is achieved with the help of image binarization using the Otsu method [12]. A connected component is a cluster of directly connected black pixels in the binarized image.

2) Edge detection and value extraction: This step needs to be repeated for every selected connected component (CC). The goal is to find the longest vertical edge in every CC and, if it is long enough, to extract a $n \times n$ window of pixel values and calculate the standard deviation of the columns. The procedure returns n values (one corresponding to each column of the window) for every CC. To detect the longest edge, the binarized picture is compared to a theoretical perfect edge. In this context, a perfect edge is defined as $\frac{n}{2}$ white pixels next to $\frac{n}{2}$ black pixels. Starting at the bottom left corner of the bounding box of the connected component, a $n \times 1$ window is extracted from the binarized image and compared to the perfect edge. This window is moved over the component until the longest vertical edge is found. Due to degeneration, edges are usually not perfect in the binarized image. Therefore, there is an additional rule that allows irregularity: A window that match a pattern where one black or white pixel is shifted horizontally is allowed as long as it is followed by a perfect edge again (cf. Figure 2).

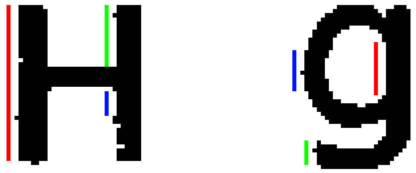


Fig. 2. Example of vertical edge detection. In both example characters the red edges are the longest and used for feature extraction.

Figure 2 illustrates the detected edges for two example characters. The colors represent separated edges as detected by the algorithm. In this case, the red edge is the longest in both characters and will be used to extract the feature values. If the longest edge is more than T ($T > n$) pixel long, a $n \times n$ window of pixel values is extracted from the middle of the edge. As mentioned above, the window is centered on the transition from white to black, to ensure that all edge information is included. Edges smaller than n pixels are ignored since it is likely that these are part of a curved character edge. Then, the standard deviation of every column in the window is calculated and stored for further investigation in a vector of n values for every selected connected component.

B. Standard Deviation on Dynamic Height Windows

In addition to the basic version of the algorithm, this alternative uses a window with the full height of the longest

[1] http://www.rapidminer.com/

[2] http://madm.dfki.de/rapidminer/anomalydetection

edge leading to a $T \times n$ sized window for every CC instead of centering it. The reason for enlarging the window is that one might expect more robust standard deviation values.

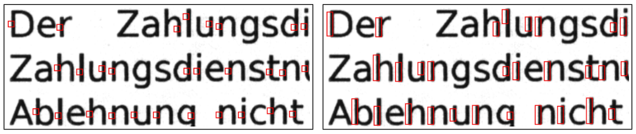


Fig. 3. A comparison of the fixed size windows (left) and dynamic height windows (right) approach. The red boxes mark the detected edges which serve as the basis for the feature extraction.

C. Intensity Variations utilizing OCR

Another approach for local feature extraction utilizes optical character recognition (OCR). OCR is the process of extracting and recognizing characters out of scanned documents. In comparison with using all connected components, this opens the chance to limit the used components to characters which definitely have a straight edge. In the above methods, it is possible to declare something as an edge, despite it only is appearing to be a straight edge in the binarized image. This effect is already reduced by only using long edges, but with the help of OCR it is possible to reduce it further. When using OCR, the examined characters can be reduced to characters like B,D,E or F, while characters like A,G or O can be ignored. To extract the characters, the tesseract-OCR [13] engine is used as a preprocessing step. Tesseract returns a list of all characters together with their coordinates in the document. This list can be reduced to contain only characters of interest. The rest of this method uses the same edge detection and feature value extraction as explained in the two previous sections, based on the extraction of connected components with the coordinates generated by tesseract-OCR.

III. Anomaly Detection

For detecting documents being printed using a different printing technique than the majority of documents, unsupervised anomaly detection is used. In contrast to previous work, our approach does not require any training data or labels. This yields to the advantage that even unknown printing techniques could be detected, for example dye-sublimation among inkjet printers. Also a poor-quality laser printer could be separated from other laser printers.

To this end, the Grubbs' test [10] and the global k-NN anomaly detection algorithm [11] is used. The Grubbs' test is a statistical anomaly detection algorithm that assigns a score depending on the mean and standard deviation of the examined data such that the score is higher for data points lying far away from the mean [10]. The score is calculated as

$$z\_score = \frac{value - mean}{SD}$$

where $mean$ is the arithmetic mean and SD is the standard deviation of the dataset. Grubbs' test is designed for data with an underlying normal distribution and only works for one dimensional data. K-NN is a nearest neighbor based anomaly detection algorithm that assigns a score depending on the distance to the closest k neighbors to an instance. The score is calculated as the arithmetic mean of the euclidean distances to the next k neighbors. It does not require a specific underlying distribution of the data and also works when having more than one dimension.

IV. Experiments and Evaluation

The experiments are conducted on a newly created dataset. All the presented methods are tested and evaluated on this dataset. In the following, the dataset and test setup is explained as well as the results of the experiments are presented and interpreted.

A. DFKI Printing Technique Dataset

The created dataset contains unique documents for every used printer. There are three different document types having different page layouts. This diversity features unique challenges for the feature extraction and anomaly detection process. For every printer, a unique dataset has been created in order to ensure a content independent feature extraction system. The following document types have been used:

1) Contracts: The first document type contains plain text contracts. The contract only contains text, but in different font types and sizes. In this dataset, a contract never contains pictures, lines and diagrams. The contracts were created automatically using a Python script.

2) Invoices: The second used document type represents invoices. In addition to different font types and sizes, the invoices also feature vertical and horizontal ruling lines as well as logos, composed of a small picture and colored text. Like the contracts, these documents are also created using a Python script simulating a bunch of different invoice parties.

3) Scientific Literature: The last type contains real-world examples, pages taken from existing scientific papers and books. For this reason, they feature a large variety of content, e.g different font types and sizes but also all kind of pictures, diagrams and formulas.

Twenty unique pages from every document type are put together to form a package for one printer. This means that there are 60 unique pages printed by every printer. Due to errors during the printing and scanning process, a few pages had to be removed from the dataset to ensure a valid evaluation. The printed pages have been scanned at 400 dpi. The images are pre-processed to remove any major skew [14] or border noise [15]. To the best of the authors' knowledge, this dataset is the first of its kind. It features realistic document types of varying difficulty, a huge number of unique pages and a good selection of different printers, as well as the original PDFs allowing reprinting or the possibility to increase the number of used printers. The dataset is publicly available[3]. All chosen literature has originally been released under a license, which allows reusing.

B. Experiments

The anomaly detection algorithm assigns a score to each document such that normal documents get small scores. To evaluate the algorithms, the examined documents are ranked by their score and the rank of the outlier document is examined.

[3] http://madm.dfki.de/downloads-ds-printing-technique

A successfully detected anomaly would have a rank of one. To evaluate the presented methods, all possible combination of documents (20 normal and 1 outlier document of the same document type but different printing techniques) are tested and the average rank of the outlier document is calculated. An average rank of 1 would be perfect while random guessing would result in an average rank of ≈ 11.

1) Basic Experiments: For the following experiments a window size of $n = 6$ is used while edges have to be at least $T = 10$ pixels long. For feature extraction with OCR, characters featuring a long, straight edge have been extracted[4]. These values have been selected based on experiments. Using the middle columns of the window (the columns lying directly on the transition from white to black) lead to the best outlier detection results. To combine the values of all connected components, taking the median yields the overall best results.

2) Comparison of the presented Features: Table I shows the average rank of the outlier document for the different feature extraction methods shown in Section II combined with Grubbs' test for anomaly detection.

TABLE I. COMPARISON OF THE AVERAGE RANK WHEN USING FIXED-SIZE OR DYNAMIC WINDOW HEIGHT AS WELL AS USING ALL CONNECTED COMPONENTS AND USING OCR.

method	contract	invoice	paper
stanDev (fixed-size)	**1.16** ±0.62	**1.74** ±2.57	2.13 ±3.36
stanDev (fixed-size + OCR)	1.44 ±1.76	1.80 ±2.65	**2.13** ±3.22
stanDev (dynamic)	2.21 ±3.61	2.24 ±3.39	3.31 ±5.11
stanDev (dynamic + OCR)	2.52 ±4.05	2.05 ±2.92	3.07 ±4.75

Comparing the results of using all connected components to using only specific characters shows that using only specific characters yields surprisingly no significant improvement. A possible explanation is that by limiting the observed connected components to specific characters, the sample size for one page is too small. When using all connected components on average 1432 windows are extracted from one contract, when using only certain characters on average only 227 windows are extracted. The results also show that using dynamic window height does not improve the results.

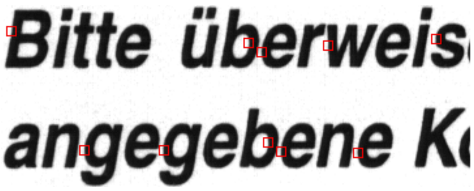
TABLE II. COMPARISON OF THE AVERAGE OUTLIER RANK FOR DIFFERENT FEATURE EXTRACTION AND ANOMALY DETECTION METHODS.

method	contract	invoice	paper	overall
stanDev Grubbs	**1.16** ±0.62	1.74 ±2.57	2.13 ±3.36	1.67 ±2.49
stanDev k-NN	1.21 ±0.85	1.78 ±2.80	**1.57** ±1.43	**1.52** ±1.90
global DCT [7]	1.99 ±2.29	**1.47** ±1.68	3.05 ±3.63	2.17 ±2.74
areaDiff [6] Grubbs	4.34 ±5.59	8.10 ±6.84	6.30 ±5.82	6.23 ±6.30
areaDiff [6] k-NN	4.49 ±5.79	7.48 ±7.00	5.04 ±4.35	5.66 ±5.98
cCoeff [6] Grubbs	5.65 ±6.82	6.55 ±6.96	11.04 ±6.58	7.71 ±7.20
cCoeff [6] k-NN	4.29 ±5.21	4.79 ±4.88	7.03 ±4.70	5.35 ±5.12

3) Comparison with previous work: Table II shows the resulting outlier rank of our presented features compared to previous work by Schreyer [6], [7]. It shows the results for the different document types separately as well as the results when using all documents. When using the overall best working feature extraction and anomaly detection algorithms, an improvement compared to previous work has been achieved.

C. Discussion of the Results

1) The Influence of Font Types: Italic font types can have a bad influence on the results as illustrated in Figure 4. Including pages like this results in a significantly worse average rank of the outlier document. Due to the askew edges of the characters the standard deviation along an edge is increased (compared to non italic characters). Consequentially, documents with a lot of italic characters can be falsely detected as outliers.

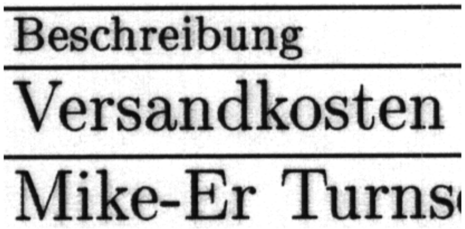


Bitte überweis

angegebene K

Fig. 4. Incorrect chosen windows due to an italic font. The windows are often selected from round or askew "edges".

2) Text rendered as a Picture: Sometimes during the printing process, text can be rendered as picture, reducing the quality and adding a lot of noise around the edges. As this can effect both inkjet and laser printers it distorts the results. Documents, where the whole page is effected by this problem, have been excluded from the dataset, because that effect was deemed to be out of the scope.



Beschreibung

Versandkosten

Mike-Er Turns

Fig. 5. An example of bad printing quality due to rendering text as a picture. Due to the noise around most characters, selecting windows from the edges in this picture would result in a higher standard deviation than expected.

3) Performance of Invoices and Scientific Literature: In many test setups, invoices and scientific literature produce worse results than contracts. This is most likely due to the effects described in the two previous sections. Concerning the invoices, the logo is sometimes printed or scanned with a low quality, in the scientific literature this effect can be found (often in formulas), too. Also, both document types contain italic fonts. If those effects are only found in parts of the documents, the pages containing them have not been removed from the dataset, as they are valid use cases. Those effects create a higher variance in the extracted features, which in turn results in a worse average rank.

Another reason may be due to the fact that on average less windows are extracted from those documents than from the contracts. For example, when using the feature standard deviation on static windows, an average amount of 451 windows is extracted from the invoices, from the scientific literature an average amount of 840 windows are used, both cases are significantly lower than the average of 1432 windows that are extract from contracts.

The different presented feature extraction methods have a

[4] used characters: B,D,E,F,H,I,J,K,L,M,N,P,R,T,U,b,h,k,l,p,r,u

varying influence on both, reducing or increasing the described effects as well as the number of used windows, which explains the varying performance of those two document types.

4) Comparison of Grubbs and k-NN: One topic that has only been briefly addressed so far is the performance of the anomaly detection algorithms, namely Grubbs' test and k-NN. Table III shows the results when using different anomaly detection algorithms for the standard feature using fixed window size and all connected components. It is interesting that both Grubbs and k-NN perform equally well on the contract and invoices type documents while k-NN performs better on the scientific literature documents. When using only specific characters instead of all connected components the difference between Grubbs and k-NN is greater, but the general trend stays the same.

TABLE III. COMPARISON OF THE AVERAGE OUTLIER RANK WHEN USING FIXED-SIZE WINDOW HEIGHT COMBINED WITH GRUBBS' TEST OR k-NN ANOMALY DETECTION.

anomaly detection	**contract**	**invoice**	**paper**
stanDev Grubbs	1.16 ±0.62	1.74 ±2.57	2.13 ±3.36
stanDev k-NN	1.21 ±0.85	1.78 ±2.80	1.57 ±1.43
stanDev Grubbs OCR	1.44 ±1.76	1.80 ±2.65	2.13 ±3.22
stanDev k-NN OCR	1.56 ±2.18	1.67 ±2.25	1.62 ±1.15

A possible explanation is that Grubbs can only detect data points lying far away from the mean. Grubbs' test assumes an underlying normal distribution in the data. K-NN on the other hand can also detect outliers lying in between the rest of the data. For example, when having two clusters, one having a high mean and one having a low mean, k-NN would be able to detect a outlying point in between those two clusters as an outlier. Grubbs on the other hand would not be able to detect that point as an outlier, because the estimated mean would be somewhere between the clusters (and therefore the "outlier" would actually be relatively close to that mean). For example, documents containing formulas might make two clusters – one containing those documents in which formulas are printed as text resulting in high quality edges, and the other containing formulas rendered as images resulting in poor quality edges. This can lead to outlier documents that can not be detected by Grubbs' test.

V. CONCLUSION AND FUTURE WORK

A new document authentication approach using printing technique features with unsupervised anomaly detection is presented in this paper. The advantage of using unsupervised anomaly detection is that no prior training and manual labeling of the data has to be performed. We developed simple and efficient features to distinguish printing techniques. Comparison to the state-of-the-art features showed that overall the presented features achieved the best performance. It is important to mention that the developed features work reasonably well at 400 dpi scanned documents. Hence, it is possible to integrate them in existing document management workflows where identifying potential forgeries is of interest.

The presented work should be understood as a basis for intrinsic document authentication in the context of low resolution scans. Possible improvements of the presented features could include a more robust edge detection to increase the quality of the features. Another potential improvement could be the preprocessing of the documents in order to identify images or italic text, which often cause incorrect scores (cf. Section IV-C1 and IV-C2). Furthermore, other printing technique features could be combined with the presented features to further improve the results of the multivariate unsupervised anomaly detection. We have made the dataset prepared in this work publicly available and hope that this will trigger more research in the topic of printer / printing technique identification.

ACKNOWLEDGMENT

This project was partially funded by the Rheinland-Palatinate Foundation for Innovation, project AnDruDok (961-38 6261 / 1039).

REFERENCES

[1] C. Lampert, L. Mei, and T. M. Breuel, "Printing technique classification for document counterfeit detection," in *Computational Intelligence and Security (CIS)*, 2006.

[2] M. Barni, C. Podilchuk, F. Bartolini, and E. Delp, "Watermark embedding: hiding a signal within a cover image," *Communications Magazine, IEEE*, vol. 39, no. 8, pp. 102–108, 08 2001.

[3] A. Mikkilineni and P. C. et al, "Printer identification based on graylevel co-occurrence features for security and forensic applications," in *Proc. of the SPIE Int. Conf. on Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, 2005, pp. 430–440.

[4] J. van Beusekom, F. Shafait, and T. M. Breuel, "Automatic authentication of color laser print-outs using machine identification codes," *Pattern Analysis & Applications*, pp. 1–16, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1007/s10044-012-0287-5

[5] J. van Beusekom, F. Shafait, and T. Breuel, "Text-line examination for document forgery detection," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 2, pp. 189–207, 2013.

[6] M. Schreyer, "Intelligent printing technique recognition and photocopy detection using digital analysis for forensic document examination," Master's thesis, Kaiserslautern University of Technology, 2008.

[7] C. Schulze, M. Schreyer, A. Stahl, and T. M. Breuel, "Using DCT features for printing technique and copy detection," in *IFIP Int. Conf. Digital Forensics*, 2009, pp. 95–106.

[8] C. Schulze, M. Schreyer, A. Stahl, and T. Breuel, "Evaluation of graylevel-features for printing technique classification in high-throughput document management systems," in *Proc. of the 2nd Int. Workshop on Computational Forensics*, 2008, pp. 35–46.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[10] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, February 1969.

[11] M. Amer and M. Goldstein, "Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer," in *Proc. of the 3rd RapidMiner Community Meeting and Conferernce*, 2012, pp. 1–12.

[12] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[13] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633.

[14] J. van Beusekom, F. Shafait, and T. M. Breuel, "Combined orientation and skew detection using geometric text-line modeling," *International Journal on Document Analysis and Recognition*, vol. 13, no. 2, pp. 79–92, 2010.

[15] 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, 2008.