Department of Computer Science
University of Kaiserslautern, Germany

Multimedia Analysis and Data Mining Competence Center
German Research Center for Artificial Intelligence (DFKI GmbH)
Kaiserslautern, Germany

Document Authentication using Printing Technique Features

Bachelor Thesis

Author:	Johann Gebhardt
Supervisor:	Markus Goldstein
	Dr. Faisal Shafait
Reviewer:	Prof. Dr. Andreas Dengel
	Markus Goldstein
Submission Date:	10 October, 2012

Abstract

Inkjet and laser printers differ in their produced printing quality. This is especially notable at the edges of printed characters. In general, inkjet printers produce a higher degree of edge roughness than laser printers. In this thesis, a system using the difference in edge roughness to distinguish laser printed documents from inkjet printed documents is presented. Several methods for feature extraction have been developed and implemented. They use different ways of edge detection and varying ways of calculating the edge roughness. The edge detection is done using binarized images. A window of pixel values is extracted from this edge. For comparison another window positioning algorithm, which minimizes the difference between a perfect edge and the actual values in the non-binarized image was developed and implemented. The extracted pixel values are used in different ways to calculate the edge roughness. One way is using the standard deviation or mean change in grey level along the edge. Another approach is calculating DCT coefficients along the edge.
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 of documents. Two different anomaly detection algorithms, Grubbs and k-NN, have been implemented and evaluated. In addition to the system, a newly created dataset featuring unique pages for different printers is presented. The dataset has been printed on 13 laser and 7 inkjet printers. For evaluation, 20 documents from one printing technique are paired with 1 page from a different printing technique (the outlier). The possible combinations of features and anomaly detection are compared by calculating the average rank that outlier document is assigned to when testing all possible combinations of printers. A perfect anomaly detection would result in an average rank of 1, randomly ranking the documents would results in average rank of 11. Compared to previous work from Schreyer et al [11], an overall improvement from an average rank of 2.19 to 1.44 has been achieved.

Contents

1 Introduction

1.1 Motivation

Today, document authentication is a task of increasing importance [8]. Invoices, contracts, certifications etc... are valuable targets for forgery. A possible way of forging a document would be scanning, subsequently modifying and then printing it or by simply creating it from scratch to look like the original. Therefore, verifying that documents are what they claim to be, is an important task.
Another reason which increases the need for automated document authentication is because today only few printed documents are examined by humans. Instead they are processed by automated systems [8].
There are several approaches to secure printed documents by adding additional security features, so called extrinsic features, such as watermarks [3].
However, an interesting field of study is the authentication of documents without added security features. In this case, only intrinsic features, features that are produced during the normal document creation, are used. This is useful because adding extrinsic features costs time and money, while using only intrinsic features does not change the creation process of the documents [5, 9].
One possible course of action is to identify the device used to create a document, in this case the printer. If someone was trying to forge or modify an invoice, the forger would have to print the document again. It is possible that the original printer is not available for the forgery.
A first step towards the identification of the printer is the recognition of the printing technique 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 thesis a system to distinguish different printing techniques has been developed and implemented. The whole process can be split up into two basic steps: feature extraction and anomaly detection. During the feature extraction, the document is examined and a feature classifying the printing technique is extracted. In the second step, a set of documents is analyzed 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:

- Feature extraction which works with documents scanned at reasonably low resolution (400 dpi)
- Combining of the feature extraction with unsupervised anomaly detection such that no training for the anomaly detection needed

- A dataset reflecting a realistic fraud attempts has been created for evaluating the feature extraction and anomaly detection
- A large number of experiments have been performed in order to find the best combination of feature extraction and anomaly detection

In this thesis, the feature extraction process as well as the anomaly detection algorithm are explained. Finally, the experiments are presented and interpreted.

1.2 Related Work

Document authentication has been studied by several groups using different approaches. For example, Beusekom et al [5] have presented a system that uses tracking patterns, 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 [4].
The process of printer or printing technique recognition has also been studied by several groups. Mikkilineni et al [9] have presented two methods, one intrinsic and one extrinsic, using graylevel co-occurence texture features for printer recognition. While the results are promising, they scanned the documents with very high resolution (2400 dpi). Further they used a classifier system, which only works on printers that have been used for training beforehand.
Lampert et al [8] 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 needs to be trained has been used.
Printing technique recognition has also been studied by Schreyer et al [11–13]. They focus on discrete cosine transform (DCT) features. In fact, this thesis builds on Marco Schreyer's diploma thesis [11], in which a wide range of different features have been presented and evaluated. The evaluation used both low and high resolution scans. However, there was only one document, printed by different printers, examined during the evaluation. One advancement of this thesis is a dataset providing a large number of unique documents for every used printer. Another advantage is the use of unsupervised anomaly detection.
Anomaly detection, the process of finding data points which do not follow the expected behavior among a dataset, has been studied by Chandola et al [6]. According to them, anomaly detection algorithms can be distinguished into several categories. There are classification based anomaly detection algorithms which need to be trained with training data. These algorithms are not used in this thesis, as one of the goals is to use unsupervised (untrained) anomaly detection.
Also there are statistical anomaly detection algorithms. Statistical anomaly detection assumes that the data set is created by a statistical model and points are labeled as anomalies, if the probability that the assumed model produced the data point is low.
Other anomaly detection algorithms are based on nearest neighbor assumptions. Generally speaking, they assume that a point is an anomaly if it lies far away from its neighbors.

Another group of anomaly detection algorithms are based on clustering. They group similar data points into clusters. Points not belonging to one of the clusters are assumed to be anomalous.
In this thesis the Grubbs test [7] - a statistical anomaly detection algorithm - and *k*-NN - a nearest neighbor based anomaly detection - have been implemented. Additionally the RaipdMiner[1] anomaly detection[2] extension by Amer et al [2] has been used for parts of the evaluation.

[1]http://rapid-i.com/content/view/181/190/
[2]http://madm.dfki.de/rapidminer/anomalydetection

2 Feature Extraction

Feature extraction is the process of transforming a large amount of input data into a smaller and descriptive representation, the so called features, of that data. In this case the input data is the examined document and the feature should extract the relevant information needed to describe the used printer type.
In the examination of documents there are two different basic types of features. There are **global features** obtained by global document examination, which examine the whole document at once. There also are **local features** obtained by local document examination, which analyze the connected components (CCs) or characters of the document separately from each other [11]. This thesis focuses on **local features**.

2.1 Intensity Variation along Vertical Edges

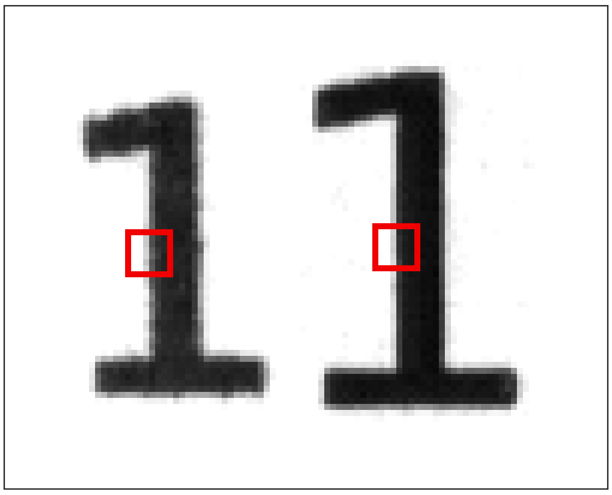


Figure 2.1: Comparison of edge roughness between inkjet(left) and laser printer(right). The inkjet printer produces a higher degree of edge roughness.

Figure 2.1 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. According to Schreyer [11], it can be generally assumed that inkjet printers generate a higher amount of edge roughness than laser printers. All presented features calculate a value or a set of values that represent the edge roughness of the document. They differ in their ways to obtain the edges as well as in their way to calculate a representation of the degeneration.

2.1.1 Standard Deviation on Static Windows

The first feature represents the degree of edge roughness by calculating the standard deviation of pixel values along vertical edges of the document. A high standard deviation means a lot of change in pixel values (grey level) along an edge and therefore represents a higher edge roughness. The standard deviation is calculated for all connected components with a minimum length vertical edge. It is expected that inkjet printed documents have higher deviation than laser printed documents.

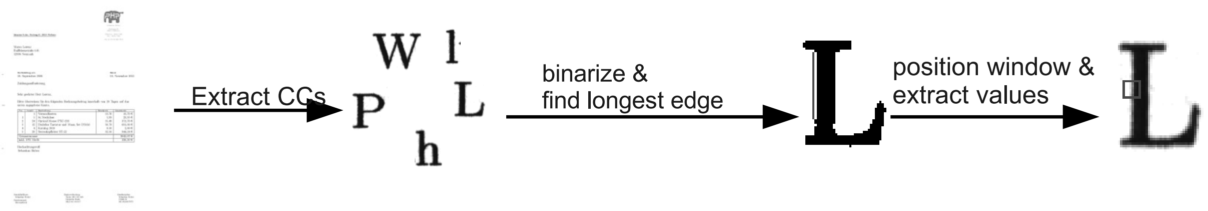


Figure 2.2: General overview of the feature extraction process. First the connected components (CCs) are extracted, then the binarized image is searched for the longest vertical edge. Finally the grey level values are extracted from the middle of the longest edge.

Figure 2.2 gives an general overview over the feature extraction process used in this method. The first step is to extract all connected components. After that, the longest edge of every component is found with the help of image binarization. Finally, the grey level values of a 6x6 window are extracted from the middle of that edge. The window is centered on the transition from white to black, so that there are three columns of white (or nearly white) pixels next to three columns of black (or nearly black) pixels. Because documents usually have a lot of CCs, there is no need to examine all of them. Therefore, connected components which do not have a vertical edge are not used. Also, a lot of printing or scanning errors (e.g. due to paper positioning distortion [14]) are found at the margin of a document. Therefore all CCs near the margin are neglected during the rest of the process.

Preprocessing

The first step - the extraction of the CCs - is achieved with a method implemented by Schreyer [11]. The method uses an image as input, extracts the CCs with the help of image binarization, and returns the coordinates of bounding boxes of every CC. These boxes are slightly larger than needed to ensure that no edge information is lost. For example,

for documents scanned with 400 dpi, there are four padding pixels added around the box [11].
Image binarization separates the pixel values into two groups. All pixels below a certain threshold belong to one group (white or 255) and all other pixels to the other group (black or 0). In this case, the threshold is found using the Otsu method [10]. The Otsu method divides the image into two classes, objects and background, and then finds the threshold that minimizes the intra-class variance. A connected component is a set of pixels which is directly connected in the binarized imaged.

Edge finding and Value Extraction

Algorithm 1 Find longest edge

```
1:  for all $x$ in $CC$ do                                          ▷ starting left
2:      for all $y$ in $CC$ do                              ▷ starting at the bottom
3:          extract 6x1 window
4:          compare to perfect edge
5:          if identical then
6:              $count$++
7:              $last = true$
8:          else
9:              compare to nearly perfect edge
10:             compare to next window to perfect edge  ▷ next window: window for y+1
11:             if identical and $last == true$ then
12:                 $count$++
13:                 $last = false$
14:             else
15:                 $count = 0$
16:                 $last = true$
17:             end if
18:         end if
19:     end for
20:     save x for longest edge                                     ▷ highest count
21: end for
```

The next step needs to be repeated for every selected connected component. The goal is to find the longest vertical edge in every CC and, if it is long enough, to extract a 6x6 window of pixel values and calculate the standard deviation of the columns. The procedure returns six values (6 columns in the window) for every CC. To find the longest edge, the binarized picture is compared step by step to a theoretical perfect edge. Here, a perfect edge has the values [255,255,255,0,0,0].
Starting at the bottom left corner of the bounding box, a 6x1 window is extracted from the binarized image and compared to the perfect edge. If the compared windows are identical, an edge has been found and a counter is incremented. The window moves now one pixel in vertical direction and the comparison is repeated. This continues until the

pattern does not match. In that case the counter is reset, and if it is the longest edge found so far, saved.
Due to the degeneration the edges are usually not perfect in the binarized image. Therefore, there is an additional rule that allows irregularity. More precisely, windows that match the pattern [255,255,255,255,0,0] or [255,255,0,0,0,0] are allowed as long as they are followed directly by a perfect edge.

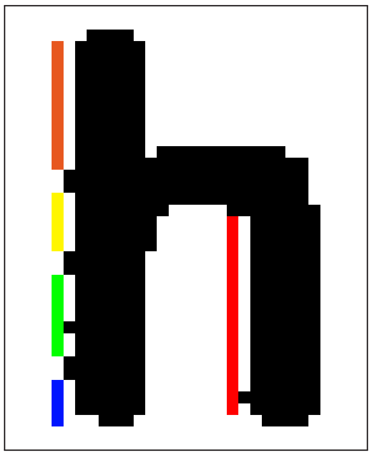


Figure 2.3: Example of detected edges. Different colors mark different edges. In this case the red one is the longest edge that will be used.

Figure 2.3 illustrates which edges are detected in an example character. All colors represent one isolated edge, as detected by the algorithm. In this case the red edge is the longest and will be the one used to extract the values.
If the longest edge is more than 10 pixel long, a 6x6 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 10 pixel are ignored, because they often only appear as edges in the binarized image, while they are actually a flat curve in the normal image. Also, when using a small edge there is a high chance of positioning the window next to the transition from an edge into a curve or corner. In both cases the results are distorted.

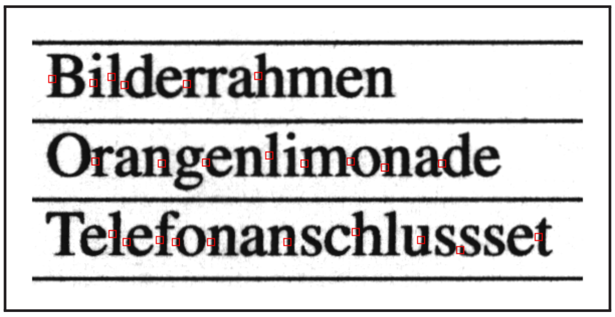


Figure 2.4: An example of extracted windows. The red boxes mark the extracted windows.

Figure 2.4 shows an example of the extracted windows. Every red box represents one detected edge with its extracted values. In each window the standard deviation of every column is calculated and stored for further investigation. This leads to having a vector of six values for every selected connected component.

2.1.2 Standard Deviation on Dynamic Windows

In addition to the basic version of the algorithm, there are two different variants building on the same principle. One alternative is to make the window height variable. The idea behind this is to minimize the impact of the upright position of the window in hope of stabilizing the results. In the basic version the window is placed in the middle of the edge. It is possible that the middle randomly shows a lower or higher degree of edge degeneration than other parts of the edge. Stretching the window to the whole edge may result in a better and more robust feature.

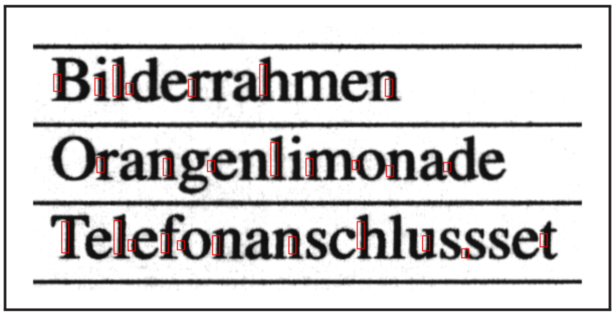


Figure 2.5: An example of extracted windows with variable height. The red boxes mark the extracted windows.

2.1.3 Mean Change in Grey Level

Another variant utilizes a slightly different way of calculating the edge roughness. Instead of using the standard deviation, it calculates the mean change in grey level when moving along the edge. The rest of the feature extraction is done the same way as explained in Section 2.1. It is expected that this will not result in a huge change of the final feature.

2.2 Intensity Variations utilizing OCR

A different approach to the local feature extraction utilizes optical character recognition (OCR). OCR is the process of extracting and recognizing characters out of scanned documents. In comparison to using all connected components, this opens the chance to limit the used components to characters which definitely have a usable edge. In the first method described above, it is possible to declare something as an edge, despite it only appearing to be a straight edge in the binarized image. This effect is already reduced by only using long enough edges, but with the help of OCR it is possible to reduce it further.

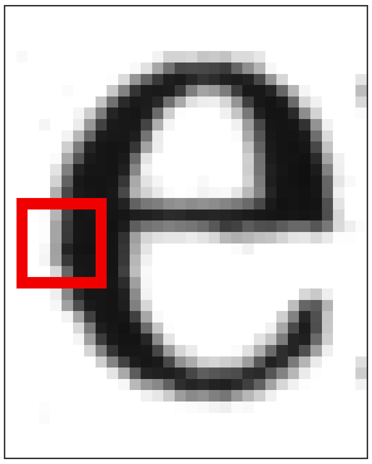


Figure 2.6: An example of a badly chosen edge when using all characters.

Figure 2.6 shows an example of a questionably chosen edge. In this case the two middle columns indicate that, in this case, the standard deviation is influenced by the fact that the window is on a curve. The window is used because in the binarized version this curve would be detected as an edge. However, using OCR the examined characters can be reduced to characters like B,D,E,F. . . , while characters like A,G,O. . . are ignored. This eliminates the described effect.

2.2.1 Description

To extract the characters **Tesseract-OCR** [1] is used. The OCR engine is used separately from the rest of the program as a preprocessing step. **Tesseract-OCR** returns a list of all characters together with their coordinates in the document. This list can be reduced to contain only characters of interest. One possibility which was tested is to use the same edge finding and value extraction methods as explained in section 2.1.1, only replacing the extraction of connected components with the coordinates generated by **tesseract-OCR**.
Additionally, a new edge finding algorithm only working in connection with OCR, has been implemented. This algorithm, called Enhanced Window Positioning, is explained in the following.

2.2.2 Enhanced Window Positioning

This procedure finds the longest edge by searching for the part in the character that minimizes the difference between the actual values and a theoretical perfect edge. In contrast to the method used in the previous feature, this methods works on the original image instead of the binarized image. Here, a perfect edge is defined as two columns of white pixels next to two columns of black pixels. The edge finding is done for every

[1] http://code.google.com/p/tesseract-ocr/

Algorithm 2 OCR Enhanced Window Positioning

```
for all $x$ in $character$ do
    for all $y$ in $character$ do
        $mean\_white$ += $255 - [x, y] + 255 - [x-1, y]$        ▷ difference between
        $mean\_black$ += $0 + [x+1, y] + 0 + [x+2, y]$        ▷ perfect edge and edge
    end for
    $mean\_white = mean\_white / y$
    $mean\_black = mean\_black / y$
    if $mean\_white + mean\_black < min\_difference$ then $save\ x$
    end if
end for
```

extracted character. Starting from the left side of the character four neighboring columns of pixel values are extracted from the input picture. Then the difference between the extracted values and a theoretical perfect edge is calculated. For the two left rows (white in the perfect edge) that is $255 - extracted\ pixel\ value$ and for the two right rows (black in the perfect edge) the difference is $0 + extracted\ value$. This is repeated for all columns in the characters. The column showing the least difference is the one closest to an edge. When comparing two edges, the smaller edge definitely has one row more than the longer edge, where at least three of the examined columns are black or white. Therefore the shorter edge differs more from the perfect edge than the longer one. If the selection of characters was not limited, this method would find an edge in every connected component, even if the edge is only one pixel long. Limiting the used characters to a selection of letters which have a suitable edge ensures that only suitable edges are found.
After finding the edge the window positioning continues in a similar way by moving the window along the edge and finding the position where the the difference between actual values inside the window and a perfect edge is minimal.
After positioning the window the value extraction continues as described in Section 2.1.1.

2.3 Discrete Cosine Transform (DCT)

The DCT transforms the grey level values of the input image from the spatial domain into the frequency domain. The image information is transformed into terms of cosine functions. Schreyer [11] shows that there is a relation between the intensity changes in the original document and the frequency values generated by DCT. Therefore it is possible to use the DCT values as a feature to differentiate between different printing techniques. The DCT is used as a local feature. The extraction of connected components or characters, as well as the edge finding and window placement, can be done in the same way as described in the previous features. However, instead of calculating the standard deviation or mean change of pixel values level along the edge, a one dimensional DCT is performed on both columns being closest to directly on the transition from white to black. In this case the DCT coefficients reflect the change of intensity along the edge. Therefore the coefficients can be used as a representation of edge roughness.

In the previous mentioned features there is only one value for every column of the extracted window, as the columns are examined separately from each other. In this case one dimensional anomaly detection is sufficient. In the case of the DCT, all extracted DCT coefficients are needed. This complicates the anomaly detection process, because a multi dimensional algorithm is needed. For more information about the anomaly detection see Chapter 3.

2.4 Global DCT

The Global DCT feature was implemented by Schreyer [11]. It performs a DCT on the whole picture and then extracts a sub band of boxes from the top and left edge. For every box the mean and standard deviation of DCT coefficient are extracted and used as the feature values. See [11] for a more in depth explanation. This feature is used as a comparison to the former presented features.

3 Anomaly Detection

Anomalies, often referred to as outliers, are data points among a dataset which do not follow the expected behavior. The process of finding these anomalies is called anomaly or outlier detection [6]. In this case that means finding a document among a set of documents which is not printed by the same printer type used for the majority of the documents. For one dimensional features two different anomaly detection algorithms have been implemented. For multi dimensional features the RapidMiner[1] anomaly detection[2] extension has been used.

3.1 Grubbs Test

The first anomaly detection algorithm which has been implemented is the Grubbs test. The Grubbs test was originally designed for data with an underlying normal distribution. It detects the data point being the furthest away from the rest of the data and checks if it is an outlier. If the data point is an outlier, it is removed and the test is repeated for the rest of the data points. [7]
The Grubbs test assigns a z_score to every data point [7].

Algorithm 3 Grubbs Test

```
1: for all data points x do
2:     calculate mean
3:     calculate standard deviation sd
4:     calculate z_score $z = \frac{x-mean}{sd}$
5: end for
6: find maximum z
7: if $z > threshold$ then $x = outlier$
8: else no outlier
9: end if
```

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

where *mean* is the arithmetic mean of the dataset and *SD* is the assumed standard deviation of the dataset. The z_score is a measure of how far away a data point lies

[1]http://rapid-i.com/content/view/181/190/
[2]http://madm.dfki.de/rapidminer/anomalydetection

from the rest of data. If the z_score of a data sample is high, the sample is far away from the majority of the data points and is a potential outlier. A high z_score can be achieved by points having a lower value than the mean as well as by points having a higher value than the mean. To check if a point is an outlier, the z_score is compared to a threshold. The threshold is influenced by the examined significance level. The significance level α expresses the probability that a point above the threshold is not generated by the underlying normal distribution. Also, using a one or two sided test influences the threshold. Two sided means that points lying above as well as points lying beneath the mean are checked. For a two sided test the threshold T is calculated as follows:

$$T = \frac{N-1}{\sqrt{N}} \sqrt{\frac{(t_{\alpha/(2N),N-2})^2}{N-2+t_{\alpha/(2N),N-2})^2}}$$

where N is number of data points and $t_{a/(2N),N-2)}$ is the critical value of the t-distribution with a significance level of $\alpha/(2N)$ and $N-2$ degrees of freedom [1]. Alternatively the threshold can be looked up in tables showing the thresholds for different number of data points and different significance levels. This is done in this implementation. The tables for different significance levels can be found in [7]. Usually, Grubbs is used to test whether there is exactly one outlier or no outlier at all in the dataset. However, there are variants designed for multiple outliers.
To evaluate the features from the previous chapter, a slightly modified version of Grubbs is used. In this version all points are ranked - in descending order - according to their z_score. The value with the highest z_score is assigned rank one. This is used to evaluate the performance of the different features presented in Chapter 2. A potential problem using the Grubbs test is that Grubbs assumes an underlying normal distribution of the data. Therefore, an additional anomaly detection algorithm is used to verify the experiments.

3.2 K-Nearest Neighbor

The second anomaly detection algorithm which was implemented is the k-nearest neighbor (k-NN) algorithm. K-NN ranks all data points by their distance to the k nearest

Algorithm 4 k-NN

```
1: for all data points x do
2:     find k nearest neighbors
3:     calculate average distance to them
4: end for
5: rank data points according to distance
```

neighbors. The distance is the euclidean distance between two points. During the experiments, k is set to 5, because tests showed that in this case 5 is a good compromise between performance and accuracy. That means that the arithmetic mean of the distance to the nearest five points is calculated and used as a score for the current point. The points are then ranked by their score in descending order, the highest distance is assigned rank one. As in Grubbs, this ranking is used to evaluate the performance of the different features presented in Chapter 2.

3.3 Multivariate Anomaly Detection

For the multi-dimensional features, the RapidMiner[3] anomaly detection extension[4] is used. For more information about the algorithms implemented see Amer et al [2]. All algorithms in the extension assign an outlier score to every data point. This score can be used to rank the data points in the same way as in Grubbs and k-NN.

[3]http://rapid-i.com/content/view/181/190/

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

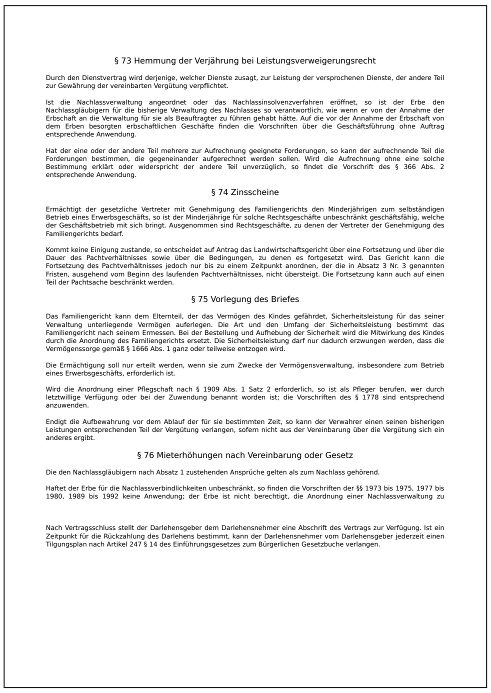
4 Evaluation

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

4.1 Dataset

The created dataset contains unique documents for every used printer. There are three different page layouts, featuring different difficulties 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:

4.1.1 Contracts



§ 73 Hemmung der Verjährung bei Leistungsverweigerungsrecht

Durch den Dienstvertrag wird derjenige, welcher Dienste zusagt, zur Leistung der versprochenen Dienste, der andere Teil zur Gewährung der vereinbarten Vergütung verpflichtet.

Ist die Nachlassverwaltung angeordnet oder das Nachlassinsolvenzverfahren eröffnet, so ist der Erbe den Nachlassgläubigern für die bisherige Verwaltung des Nachlasses so verantwortlich, wie wenn er von der Annahme der Erbschaft an die Verwaltung für sie als Beauftragter zu führen gehabt hätte. Auf die vor der Annahme der Erbschaft von dem Erben besorgten erbschaftlichen Geschäfte finden die Vorschriften über die Geschäftsführung ohne Auftrag entsprechende Anwendung.

Hat der eine oder der andere Teil mehrere zur Aufrechnung geeignete Forderungen, so kann der aufrechnende Teil die Forderungen bestimmen, die gegeneinander aufgerechnet werden sollen. Wird die Aufrechnung ohne eine solche Bestimmung erklärt oder widerspricht der andere Teil unverzüglich, so findet die Vorschrift des § 366 Abs. 2 entsprechende Anwendung.

§ 74 Zinsscheine

Ermächtigt der gesetzliche Vertreter mit Genehmigung des Familiengerichts den Minderjährigen zum selbständigen Betrieb eines Erwerbsgeschäfts, so ist der Minderjährige für solche Rechtsgeschäfte unbeschränkt geschäftsfähig, welche der Geschäftsbetrieb mit sich bringt. Ausgenommen sind Rechtsgeschäfte, zu denen der Vertreter der Genehmigung des Familiengerichts bedarf.

Kommt keine Einigung zustande, so entscheidet auf Antrag das Landwirtschaftsgericht über eine Fortsetzung und über die Dauer des Pachtverhältnisses sowie über die Bedingungen, zu denen es fortgesetzt wird. Das Gericht kann die Fortsetzung des Pachtverhältnisses jedoch nur bis zu einem Zeitpunkt anordnen, der die in Absatz 3 Nr. 3 genannten Fristen, ausgehend vom Beginn des laufenden Pachtverhältnisses, nicht übersteigt. Die Fortsetzung kann auch auf einen Teil der Pachtsache beschränkt werden.

§ 75 Vorlegung des Briefes

Das Familiengericht kann dem Elternteil, der das Vermögen des Kindes gefährdet, Sicherheitsleistung für das seiner Verwaltung unterliegende Vermögen auferlegen. Die Art und den Umfang der Sicherheitsleistung bestimmt das Familiengericht nach seinem Ermessen. Bei der Bestellung und Aufhebung der Sicherheit wird die Mitwirkung des Kindes durch die Anordnung des Familiengerichts ersetzt. Die Sicherheitsleistung darf nur dadurch erzwungen werden, dass die Vermögenssorge gemäß § 1666 Abs. 1 ganz oder teilweise entzogen wird.

Die Ermächtigung soll nur erteilt werden, wenn sie zum Zwecke der Vermögensverwaltung, insbesondere zum Betrieb eines Erwerbsgeschäfts, erforderlich ist.

Wird die Anordnung einer Pflegschaft nach § 1909 Abs. 1 Satz 2 erforderlich, so ist als Pfleger berufen, wer durch letztwillige Verfügung oder bei der Zuwendung benannt worden ist; die Vorschriften des § 1778 sind entsprechend anzuwenden.

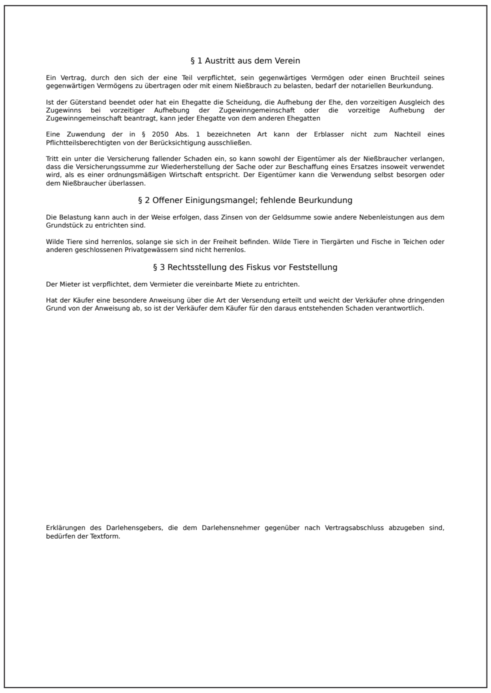
Endigt die Aufbewahrung vor dem Ablauf der für sie bestimmten Zeit, so kann der Verwahrer einen seinen bisherigen Leistungen entsprechenden Teil der Vergütung verlangen, sofern nicht aus der Vereinbarung über die Vergütung sich ein anderes ergibt.

§ 76 Mieterhöhungen nach Vereinbarung oder Gesetz

Die den Nachlassgläubigern nach Absatz 1 zustehenden Ansprüche gelten als zum Nachlass gehörend.

Haftet der Erbe für die Nachlassverbindlichkeiten unbeschränkt, so finden die Vorschriften der §§ 1973 bis 1975, 1977 bis 1980, 1989 bis 1992 keine Anwendung; der Erbe ist nicht berechtigt, die Anordnung einer Nachlassverwaltung zu

Nach Vertragsschluss stellt der Darlehensgeber dem Darlehensnehmer eine Abschrift des Vertrags zur Verfügung. Ist ein Zeitpunkt für die Rückzahlung des Darlehens bestimmt, kann der Darlehensnehmer vom Darlehensgeber jederzeit einen Tilgungsplan nach Artikel 247 § 14 des Einführungsgesetzes zum Bürgerlichen Gesetzbuche verlangen.



§ 1 Austritt aus dem Verein

Ein Vertrag, durch den sich der eine Teil verpflichtet, sein gegenwärtiges Vermögen oder einen Bruchteil seines gegenwärtigen Vermögens zu übertragen oder mit einem Nießbrauch zu belasten, bedarf der notariellen Beurkundung.

Ist der Güterstand beendet oder hat ein Ehegatte die Scheidung, die Aufhebung der Ehe, den vorzeitigen Ausgleich des Zugewinns bei vorzeitiger Aufhebung der Zugewinngemeinschaft oder die vorzeitige Aufhebung der Zugewinngemeinschaft beantragt, kann jeder Ehegatte von dem anderen Ehegatten

Eine Zuwendung der in § 2050 Abs. 1 bezeichneten Art kann der Erblasser nicht zum Nachteil eines Pflichtteilsberechtigten von der Berücksichtigung ausschließen.

Tritt ein unter die Versicherung fallender Schaden ein, so kann sowohl der Eigentümer als der Nießbraucher verlangen, dass die Versicherungssumme zur Wiederherstellung der Sache oder zur Beschaffung eines Ersatzes insoweit verwendet wird, als es einer ordnungsmäßigen Wirtschaft entspricht. Der Eigentümer kann die Verwendung selbst besorgen oder dem Nießbraucher überlassen.

§ 2 Offener Einigungsmangel; fehlende Beurkundung

Die Belastung kann auch in der Weise erfolgen, dass Zinsen von der Geldsumme sowie andere Nebenleistungen aus dem Grundstück zu entrichten sind.

Wilde Tiere sind herrenlos, solange sie sich in der Freiheit befinden. Wilde Tiere in Tiergärten und Fische in Teichen oder anderen geschlossenen Privatgewässern sind nicht herrenlos.

§ 3 Rechtsstellung des Fiskus vor Feststellung

Der Mieter ist verpflichtet, dem Vermieter die vereinbarte Miete zu entrichten.

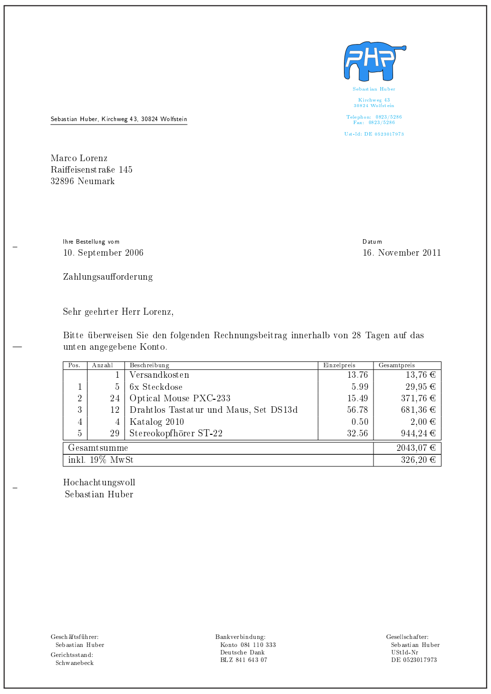
Hat der Käufer eine besondere Anweisung über die Art der Versendung erteilt und weicht der Verkäufer ohne dringenden Grund von der Anweisung ab, so ist der Verkäufer dem Käufer für den daraus entstehenden Schaden verantwortlich.

Erklärungen des Darlehensgebers, die dem Darlehensnehmer gegenüber nach Vertragsabschluss abzugeben sind, bedürfen der Textform.

Figure 4.1: Two example contracts taken from the dataset

The first document type contains plain text contracts (see Figure 4.1 for an example). 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.

4.1.2 Invoices



Sebastian Huber

Kirchweg 43
30824 Wolfstein

Telephon: 0823/5286
Fax: 0823/5286

Ust-Id: DE 0523017973

Sebastian Huber, Kirchweg 43, 30824 Wolfstein

Marco Lorenz
Raiffeisenstraße 145
32896 Neumark

Ihre Bestellung vom
10. September 2006

Datum
16. November 2011

Zahlungsaufforderung

Sehr geehrter Herr Lorenz,

Bitte überweisen Sie den folgenden Rechnungsbeitrag innerhalb von 28 Tagen auf das unten angegebene Konto.

Pos.	Anzahl	Beschreibung	Einzelpreis	Gesamtpreis
	1	Versandkosten	13.76	13,76 €
1	5	6x Steckdose	5.99	29,95 €
2	24	Optical Mouse PXC-233	15.49	371,76 €
3	12	Drahtlos Tastatur und Maus, Set DS13d	56.78	681,36 €
4	4	Katalog 2010	0.50	2,00 €
5	29	Stereokopfhörer ST-22	32.56	944,24 €

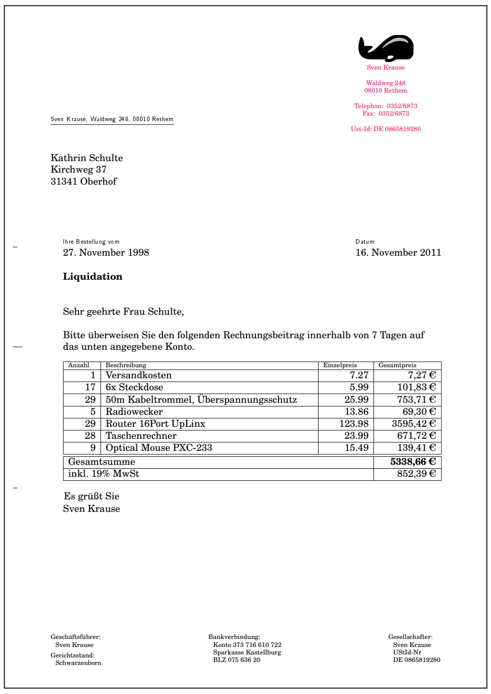
Gesamtsumme	2043,07 €
inkl. 19% MwSt	326,20 €

Hochachtungsvoll
Sebastian Huber

Geschäftsführer:
Sebastian Huber
Gerichtsstand:
Schwanebeck

Bankverbindung:
Konto 084 110 333
Deutsche Dank
BLZ 841 643 07

Gesellschafter:
Sebastian Huber
UStId-Nr
DE 0523017973



Sven Krause

Waldweg 248
08010 Rethem

Telephon: 0352/6873
Fax: 0352/6873

Ust-Id: DE 0865819280

Sven Krause, Waldweg 248, 08010 Rethem

Kathrin Schulte
Kirchweg 37
31341 Oberhof

Ihre Bestellung vom
27. November 1998

Datum
16. November 2011

Liquidation

Sehr geehrte Frau Schulte,

Bitte überweisen Sie den folgenden Rechnungsbeitrag innerhalb von 7 Tagen auf das unten angegebene Konto.

Anzahl	Beschreibung	Einzelpreis	Gesamtpreis
1	Versandkosten	7.27	7,27 €
17	6x Steckdose	5.99	101,83 €
29	50m Kabeltrommel, Überspannungsschutz	25.99	753,71 €
5	Radiowecker	13.86	69,30 €
29	Router 16Port UpLinx	123.98	3595,42 €
28	Taschenrechner	23.99	671,72 €
9	Optical Mouse PXC-233	15.49	139,41 €

Gesamtsumme	**5338,66 €**
inkl. 19% MwSt	852,39 €

Es grüßt Sie
Sven Krause

Geschäftsführer:
Sven Krause
Gerichtsstand:
Schwarzenborn

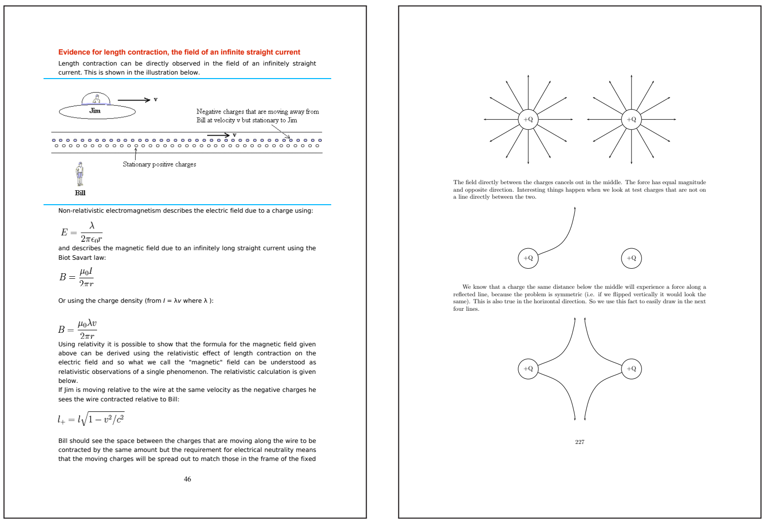
Bankverbindung:
Konto 373 716 610 722
Sparkasse Kastellburg
BLZ 075 636 20

Gesellschafter:
Sven Krause
UStId-Nr
DE 0865819280

Figure 4.2: Two example invoices taken from the dataset

The second used document type are invoices (see Figure 4.2 for an example). In addition to different font types and sizes, the invoices also feature vertical and horizontal 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.

4.1.3 Scientific Literature



Evidence for length contraction, the field of an infinite straight current

Length contraction can be directly observed in the field of an infinitely straight current. This is shown in the illustration below.

v
Jim
Negative charges that are moving away from Bill at velocity v but stationary to Jim
v
Stationary positive charges
Bill

Non-relativistic electromagnetism describes the electric field due to a charge using:

$$E = \frac{\lambda}{2\pi\epsilon_0 r}$$

and describes the magnetic field due to an infinitely long straight current using the Biot Savart law:

$$B = \frac{\mu_0 I}{2\pi r}$$

Or using the charge density (from $I = \lambda v$ where λ):

$$B = \frac{\mu_0 \lambda v}{2\pi r}$$

Using relativity it is possible to show that the formula for the magnetic field given above can be derived using the relativistic effect of length contraction on the electric field and so what we call the "magnetic" field can be understood as relativistic observations of a single phenomenon. The relativistic calculation is given below.
If Jim is moving relative to the wire at the same velocity as the negative charges he sees the wire contracted relative to Bill:

$$l_+ = l\sqrt{1 - v^2/c^2}$$

Bill should see the space between the charges that are moving along the wire to be contracted by the same amount but the requirement for electrical neutrality means that the moving charges will be spread out to match those in the frame of the fixed

46

The field directly between the charges cancels out in the middle. The force has equal magnitude and opposite direction. Interesting things happen when we look at test charges that are not on a line directly between the two.

We know that a charge the same distance below the middle will experience a force along a reflected line, because the problem is symmetric (i.e. if we flipped vertically it would look the same). This is also true in the horizontal direction. So we use this fact to easily draw in the next four lines.

227

Figure 4.3: Two example papers taken from the dataset

The last type contains real-world examples, pages taken from existing scientific papers and books (see Figure 4.3 for an example[1]). Therefore they feature a large variety of content, e.g different font types and sizes but also all kind of pictures, diagrams and formulas.

4.1.4 DFKI Printing Technique Dataset

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. In total, 50 packages have been produced. 13 packages have been printed by different laser printers. Seven packages have been printed by inkjet printers. Two packages have been printed by dot matrix printers. However the dot matrix printers have not been used in the experiments. They will still be included in the dataset for future work. The rest of the packages have not been printed, but are provided in their original form for future use.
Due to errors during the printing and scanning process several pages had to be removed from the dataset to ensure a valid evaluation. One error occurring multiple times was caused by missing font types. In this case the text got rendered as a picture, which reduces the quality to a level which distorted the results. Other issues include nearly empty ink carriages, different paper types or paper jamming. Table 4.1 shows a list of all used printers, their printing technique and which dataset (DS) they printed. The printed pages have been scanned at 400 dpi.

[1] Taken from http://en.wikibooks.org/wiki/Special_relativity (left example) and http://en.wikibooks.org/wiki/FHSST_Physics (right example)

To the best of my knowledge this data set 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 can be downloaded at http://www.madm.eu/downloads-ds-printing-technique. Both, the original PDFs, as well as the printed and scanned documents are available. All chosen literature has originally been released under a license, which allows reusing.

DS	**Type**	**Model**
01	Ink	Officejet 5610
02	Laser	Samsung CLP 500
03	Laser	Ricoh Aficio MPC2550
04	Laser	HP LaserJet 4050
05	Laser	OKI C5600
06	Laser	HP LaserJet 2200dtn
08	Laser	Ricoh Afico Mp6001
11	Ink	Epson Stylus Dx 7400
13	Ink	unknown
17	Dotmatrix	unknown
19	Laser	HP Color LaserJet 4650dn
20	Laser	Nashuatec DSC 38 Aficio
21	Laser	Canon LBP7750 cdb
22	Ink	Canon MX850
23	Ink	Canon MP630
24	Laser	Canon iR C2620
25	Dotmatrix	Epson LX300+
26	Ink	Canon MP64D
31	Laser	Hp Laserjet 4350 o.4250
32	Ink	unknown
49	Laser	Hp Laserjet 5
50	Laser	Epson Aculaser C1100

Table 4.1: A list of all used printers and which dataset(DS) they printed. Missing datasets were either not printed or not used during the evaluation.

4.2 Test Setup

The different document types are evaluated separately from each other. The standard test setup contains twenty pages from one document type printed on the same printer. As an anomaly to be detected, the data set also contains one single document of the same document type printed on a different printer belonging to a different printing technique. To evaluate the methods all possible combinations of laser and inkjet printers are used. All pages of every inkjet printer are paired with all laser printers and vice versa. For the contract document type (where no pages had to be removed from the dataset) that leads to 3640 different test cases. For invoice and scientific literature (where several pages had been removed) more than 3000 test cases remain.

4.3 Experiments

As explained in Chapter 3 the anomaly detection algorithms assign a score to all tested documents. Lower scores mean more normal documents. A document is considered to be normal when the extracted features are similar to those of the majority documents of the current test sample. The outlier (in this case the document from a different printer) should always have the highest score. To evaluate the feature extraction and anomaly detection, the documents are ranked by their score (highest score is rank 1) and the rank of the outlier is examined. If the rank is 1, the outlier was detected correctly. The higher the rank, the worse the feature performed. This is done for all test compositions of all possible printer combinations. The average rank of the outlier document is calculated and used for evaluation. A perfect result would be an average rank of 1, which would mean that the anomaly document has been detected in every test composition. Randomly ranking the documents would result in an average rank of $\frac{1+21}{2} = 11$ for the outlier document, as the standard test setup contains 21 documents (20 normal documents and 1 outlier document). Due to the need of removing several pages from the dataset, the average rank of a randomly ranked outlier is acutally slightly lower, because some test cases have less than 21 documents. As a baseline, an average random rank of 10.5 is assumed, which is actually slightly better than the actual random rank, as most test combinations have all 21 documents and only very few have less than 19 test samples.

4.3.1 Comparison of using different Columns of the Extracted Window

The first feature (see Section 2.1) extracts 6x6 windows, centered on the white to black edges of the document, and calculates the standard deviation of pixel values for every column. This leads to 6 different values (one for every column) which can be used for the anomaly detection. To test which of the columns leads to the best results, this experiment compares the average rank of the outlier document when using different columns. Both, Grubbs (see section 3.1) and k-NN (see section 3.2) have been tested.

Results

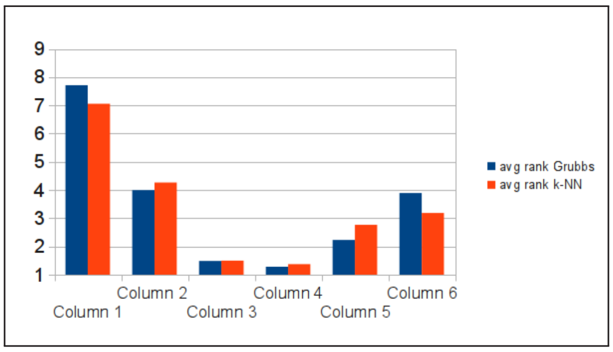


Figure 4.4: Comparison of the results of using different columns of the extracted windows. The two middle columns (3 and 4) show the best results. Interestingly, column 5 and 6, which are in the black part of an edge, show better results than column 1 and 2 (which are in the white part of the edge).

Overall, the forth column shows the best result with an average rank of 1.281 for the Grubbs test (and 1.373 for k-NN). The third column is slightly worse with an average rank of 1.486 (1.499). The other columns, while still being better than a random guess, yield significantly worse results. A possible explanation is that the difference between inkjet and laser printers is most significant directly at the transmission from white to black. The black columns (5 and 6) show better results than the white (1 and 2) columns. This was unexpected, but looking at the pictures reveals that inkjet printers indeed often show a higher variance then laser printers in those columns.

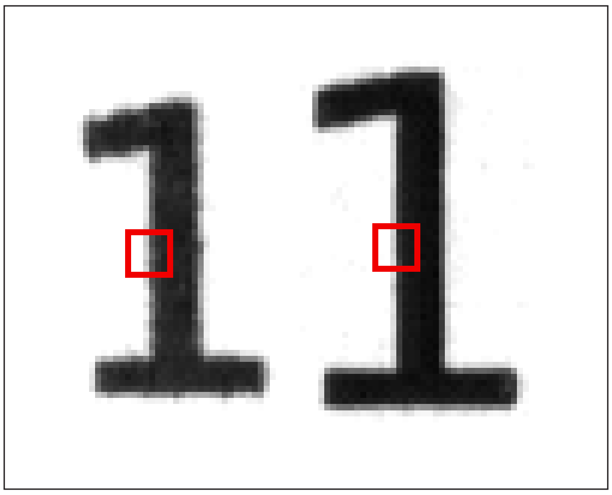


Figure 4.5: Comparison of the black columns in the red window between inkjet (left) and laser printer (right). The Figure reveals that the variance in the black columns of the inkjet printer is visibly higher than in the columns of the laser printer.

As the forth column shows the most promising results, it is used for the rest of the experiments unless otherwise noted.

4.3.2 Comparison of Mean, Median and Standard Deviation

The local features presented in Chapter 2 extract values for every connected component (CC). These values need to be reduced to a single value for every page. Taking the mean, median or standard deviation of all connected components are three different ways of doing this, a comparison of which is shown below.

Results

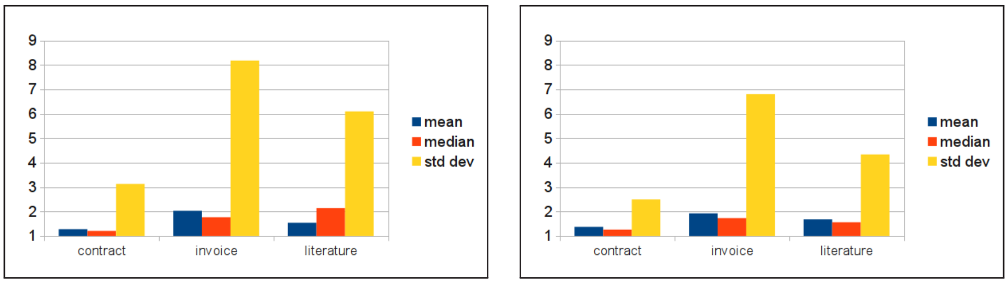


Figure 4.6: Comparison of the average rank using different statistical methods to combine column 4 of all extracted windows. In this case, the median shows the overall best results for both Grubbs (left) and k-NN (right). The only notable exception is the scientific literature combined with the Grubbs test, in which the mean shows the best results.

The results (see Figure 4.6) show that when using the fourth window column, for most of the document types the median of all connected components yields the best results. An exception is the literature document type combined with the Grubbs test in which the mean shows the best result. A difference between mean and median is that the mean is influenced by every value, including extreme values. In contrast, the median is more robust against extreme values. The results indicate that these few extreme values are mostly anomalies and should not be taken into account. The data shows that using the standard deviation to combine the connected components is generally a lot worse than using the mean or median. Interestingly, this looks different when using the third instead of the fourth window column. In the following the same experiment is done using the third window column.

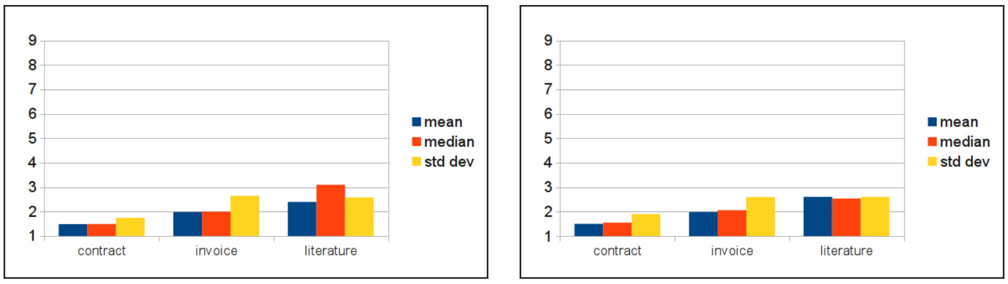


Figure 4.7: Comparison of the average rank using different statistical methods to combine all extracted windows using colum 3. In this case the mean shows the overall best results for both Grubbs (left) and k-NN (right). The only notable exception is the scientific literature combined with the Grubbs test in which the median shows the best results.

As mentioned above, the results shown in Figure 4.7, look different in the third window column. Here the arithmetic mean shows slightly better results than the median. Also using the standard deviation results in a better average rank than in the forth column. In the following experiments the median is used, as it shows the best results in the forth column, and those results are better than the arithmetic mean of the third column.

4.3.3 Comparison of Static and Dynamic Window Size

In Section 2.1.2 a variant of the feature standard deviation on static windows has been presented. The variant uses a dynamic window height instead of the static 6x6 window. The window height is stretched to the length of the detected edge in hope of creating a more robust feature. This experiment analyses the effect of the window height on the outlier detection.

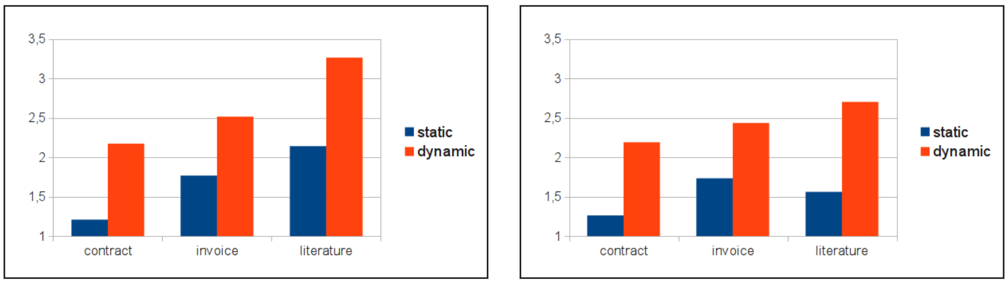


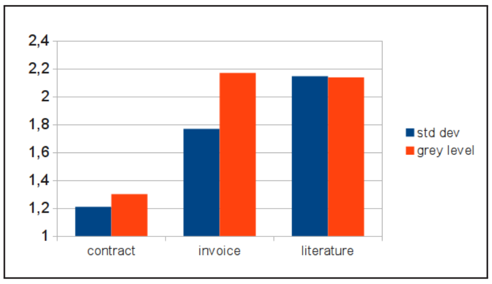
Figure 4.8: Comparison of the resulting average rank when using static or dynamic window high. In this feature the fixed window size shows better results for both Grubbs (left) and k-NN (right)

Results

The results show that using a variable window size does not improve the algorithm. In Chapter 2 it has been explained that the algorithm from the feature standard deviation on static windows (see Section 2.2) sometimes detects a part of a connected component as an edge, despite it only appearing to be an edge due to the binarization of the image. It is possible, and due to the results most likely, that this effect is enhanced when using variable window height. Both anomaly detection algorithms are effected in a comparable manner by the dynamic window size. This will be further inspected later, when comparing the results of using all connected components to using only certain characters.

4.3.4 Comparison of Standard Deviation and Mean Change in Grey Level

Another variation of the standard deviation on a static window feature, is to use the mean change in grey level alongside a vertical edge instead of the standard deviation (see Section 2.1.3). This experiment is different than the experiment from Section 4.3.2, because here the standard deviation alongside a vertical edge is compared to the mean change in grey level alongside a vertical edge. In the experiment from Section 4.3.2 the standard deviation was used as a way to combine all connected components from one document into a single value.



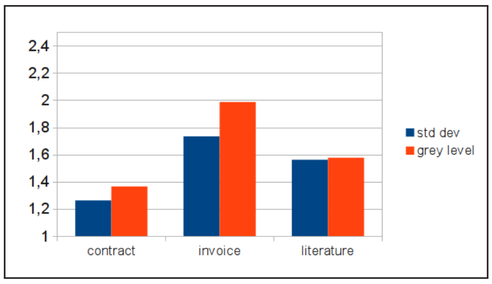


Figure 4.9: Comparison of the average rank calculated by both Grubbs (left) and k-NN when using the standard deviation with using the mean change in grey value to calculate the edge roughness. The results are worse when using the mean change in gray level instead of the standard deviation.

Results

Overall, using the standard deviation along a vertical edge produces better results than using the mean change in grey level. The only exception is the document type scientific literature, but the difference between the two methods is small.

4.3.5 Comparison of using all Connected Components and using OCR

This experiment compares the results of using tesseract-OCR[2] to select specific characters instead of using all connected components. The rest of the feature extraction is done in the same way as described in the feature standard deviation on static windows (see Section 2.1.1). In the following, the average rank calculated for all connected components is compared to the resulting average rank when using only specific characters. The following characters have been extracted in this experiment: BDEFHIJKLMNPRTUbhklpru.

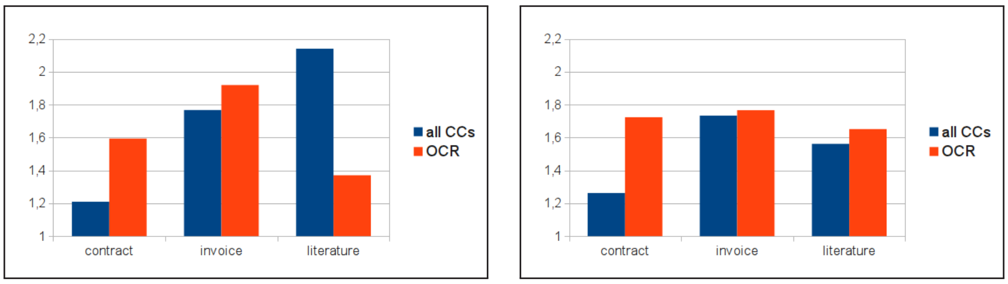


Figure 4.10: Comparison of the resulting average rank calculated by Grubbs (left) and k-NN (right) when using all CCs with using only the listed characters extracted with OCR. A very good improvement is seen for paper document type combined with Grubbs test, for the rest of the document types and all of k-NN the results are worse.

Results

The results, especially the ones for the contract-like documents, are unexpected. For this document type the average rank calculated by both anomaly detection algorithms is a lot worse when limiting the extracted characters compared to using all connected components. A possible explanation is that by limiting the observed connected components to specific characters, the sample size for one page is to small. When using all connected components on average 1448.01 windows are extracted from one contract, when using only certain characters on average only 203.28 windows are extracted. For the invoice type, the results are slightly worse when using only the listed characters. Again this could be influenced by the lower sample size. A huge improvement from an average rank of 2.144 to an average rank 1.3699 is seen for the scientific literature combined with the Grubbs test. It seems that the original idea of limiting the characters does work in this case. The results indicate that, when using the literature documents together with all connected components, a lot of “bad” windows are extracted, possible from pictures or formulas. These influence the results of the Grubbs test. By limiting the extracted connected components, less “bad” windows are chosen, and the results improve. When using k-NN this is not reproduced, a discussion about the different results of the anomaly

[2]http://code.google.com/p/tesseract-ocr/

detection algorithms can be read in Section 4.4.4, in which the general performance of Grubbs and k-NN is analyzed.

4.3.6 Comparison of using Static and Dynamic Window Size when using OCR

This experiment compares the results of using a static window height with the results of using a dynamic window height in combination with limiting the extracted characters with the help of OCR.

Result

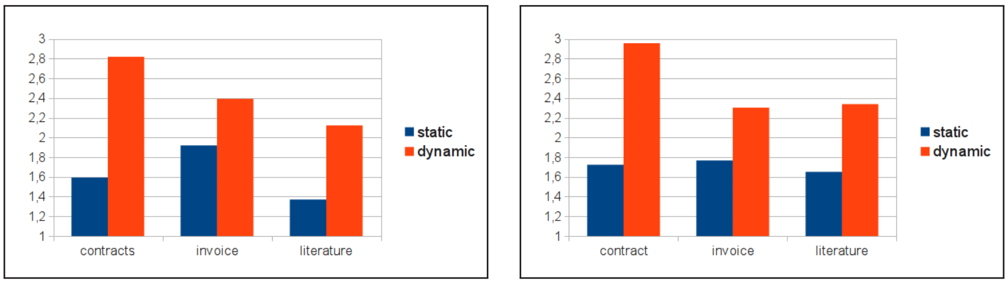


Figure 4.11: Comparision of the average rank when using static and dynamic window size together with OCR. The dynamic window approach produces worse results in all cases.

.

The results are similar to the results of the comparison of static and dynamic window sizes for all connected components (see Section 4.3.3). The dynamic window size produces worse results in all cases. It was expected that the results from dynamic window heights would be better when using OCR. Limiting the characters to characters which have a long, straight edge (often one edge from the top to the bottom of the character) should eliminate the effect described in Section 2.2, in which sometimes non-vertical edges are chosen. Those non-ideal edges seem to be a possible explanation of the poor results when using a dynamic window height together with all connected components. This explanation is obviously wrong because using OCR resulted in no improvment, therefore another explanation has to be found. Possible ideas include that longer windows increase the effect of italic fonts or enhance problems created by paper position distortion [14].

4.3.7 Comparison of using OCR to OCR with Enhanced Window Positioning

In Section 2.2.2 an alternate window positioning algorithm has been presented. This experiment compares the results of using the new window position with the old edge detection described in Section 2.1 and used in all previous experiments.

Results

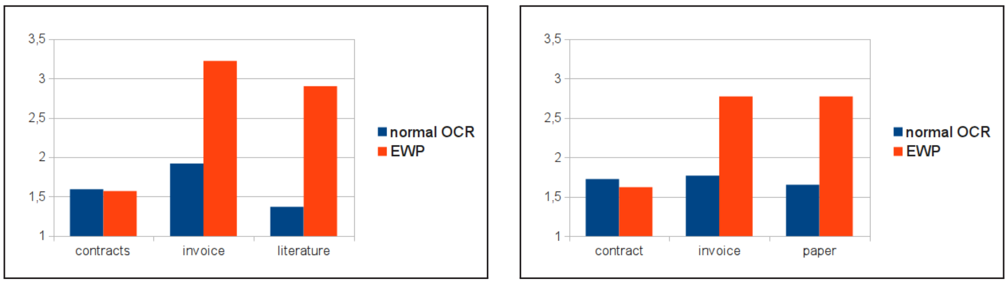


Figure 4.12: Comparision of the average rank when using the standard window position with using the Enhanced Window Positioning. The Enhanced Window Positioning produces slightly better results for the contract documents. For the other document types the results are worse.

.

The results in Figure 4.12 show a slight improvement for the contract documents. In this case the average rank improves from 1.59 to 1.57 when using Grubbs and from 1.73 to 1.62 when using k-NN. This indicates that the enhanced window position theoretical works. However, the results for the invoices and scientific literature are significantly worse with enhanced window positioning. The reasons are, that in the invoices and scientific literature sometimes text get rendered as a picture in parts of the document or is in an italic font type (see 4.4.1 and following). **Tesseract-OCR** still detects characters in this parts. When using "normal" window positioning as described in Section 2.1.1 some of this characters are rejected during the edge detection, for example because they do not have long enough edges (due to the reduced printing quality). When using enhanced window positioning all of those characters are used, and in this case the results are distorted. Possible improvements could be made by either filtering these characters out during the OCR process, or maybe by using thresholding during the window positioning.

4.3.8 DCT

Local DCT

In Section 2.3 a feature using DCT coefficients to represent the edge degeneration was presented. In this experiment, 2x8 windows were extracted directly from the edge of all available connected components. The connected components extraction as well as the edge finding was done in the same way as explained in the feature Standard Deviation on Static Windows (see 2.1.1). A one dimensional DCT was performed on both extracted columns. In this case all extracted DCT coefficients have been used in the anomaly detection. Therefore multivariate anomaly detection needs to be used. For this task the RapidMiner[3] anomaly detection[4] extension was used. In this variant the following anomaly detection algorithms have been used: k-NN, LOF, LOCI. For more information see [2].

Results

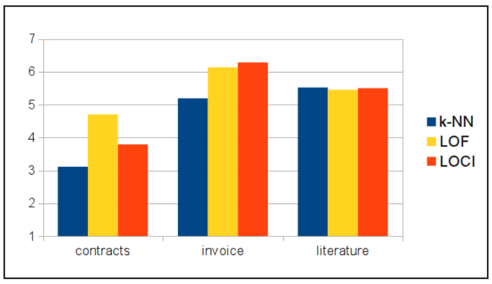


Figure 4.13: Comparison of the average rank calculated by different anomaly detection algorithms. k-NN shows the overall best results.

Overall, the results when using k-NN are best. For the contracts (and invoice) documents k-NN delivers the best results with an average rank of 3.11 (5.91) compared to the average rank of 4.70 (6.13) by LOF or 3.79 (6.28) by LOCI. For the scientific literature however, k-nn produces the worst results with an average rank of .,52 compared to the average ranks of 5.45 (LOF) and 5.50 (LOCI). In the case of the literature the difference between the anomaly detection algorithms is relatively small. That is why k-NN is overall rated as the best.
However, the results are a lot worse than the results produced beforehand by the other features and their variants. It has been shown by Schreyer et al [11–13] that using DCT coefficients as features is possible for printing type recognition. Therefore the problem is most likely created by the way in which the DCT coefficients were used. Possible improvements could be:

- Extraction of horizontal window
- Multi-dimensional DCT instead of multiple one-dimensional DCTs

The next experiment shows that DCT coefficients, extracted in a different way, can produce better results on the used dataset.

Global DCT

Schreyer [11] reported very good results when using a global DCT feature. The used feature has been outlined in section 2.4. This is a multivariate feature, therefore the

[3]http://rapid-i.com/content/view/181/190/
[4]http://madm.dfki.de/rapidminer/anomalydetection

RapidMiner[5] anomaly detection[6] plugin has been used. The results of using the global dct in combination with different anomaly detection algorithms are shown below.

Results

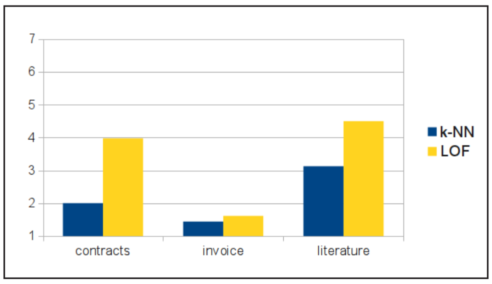


Figure 4.14: Comparison of the average rank using different anomaly detection algorithms in conjunction with the global DCT feature extraction. k-NN shows the overall best results.

The results of using the global DCT feature by Schreyer [11] are significantly better than those using the presented local DCT feature. This is true for all document types and all anomaly detection algorithms. The best results are produced by k-NN, which calculates an average rank of 2.003 for the contracts, an average rank of 1.44 for the invoices and average rank of 3.124 for the papers.
Most interesting are the results for the invoices, which are better than all methods presented by us. Contrary, some of the presented methods work better for the other two document types. An explanation is probably that global DCT is a global feature and it is possible that these are slightly influenced by the content of the documents. On a local level, the contracts are the most similar to each other as they only contain text. On a global level, however, they show a high variance in the amount of text they contain. The invoices on the other hand, show higher variance on a local level than the contracts. For example due to the logo. However, they are relatively similar to each other on a global level. They have comparable amounts of text and all of them have a logo at roughly the same position. It is possible that the global DCT feature is effected by the similarity of the documents. The even worse results for the scientific literature can be explained that way too. That could be an explanation why it produces better results for the invoice than for the other document types.

4.4 Discussion

This section presents several observations that have been made during the experimentation.

[5]http://rapid-i.com/content/view/181/190/
[6]http://madm.dfki.de/rapidminer/anomalydetection

4.4.1 The Influence of Font Types

Italic font types have a bad influence on the result. This is, for example, shown in dataset 6, page 19. When including page 19 (during the other tests it was always included) the standard deviation on static windows feature calculates an average rank of 1.977 when comparing dataset 6 with all inkjet printers. Removing page 19 improves the rank to 1.111. Due to italic font type page 19 gets a relatively high score. The score is higher than the score of most inkjet printed documents and therefore page 19 becomes rank 1 most of the time. When removing page 19, the inkjet printed documents get the highest score most of time which improves the result.

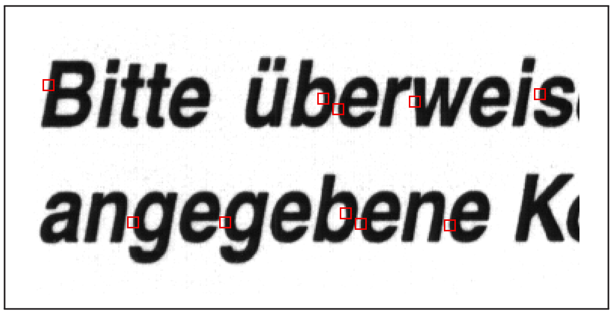


Figure 4.15: Incorrect chosen windows due to an italic font. The windows are often selected from round or askew "edges".

Figure 4.15 shows an example of the window selection not working properly with an italic font type. The windows are often selected from either askew edges or round "edges".

4.4.2 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. This happened for example in the dataset 3 page 13.

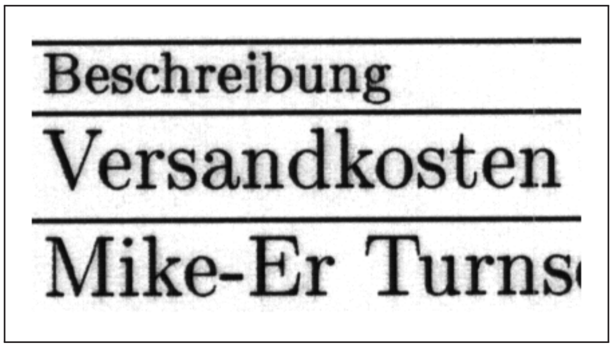


Figure 4.16: An example of bad printing quality due to rendering text as a picture. Due to the noise around most characters, selecting windows from the edges in this picture would result in a higher standard deviation than normal.

Documents, where the whole page is effected by this problem, have been excluded during the experimentation, because that effect was deemed to be ouf of the scope. Including these pages would have an effect similar to the italic fonts, because it increases the outlier score in a similar manner.

4.4.3 Performance of Invoices and Scientific Literature

In most tested methods invoices and scientific literature produce worse results than contracts. Possible reasons for this are that in both document types the effects described in the previous Sections 4.4.2 and 4.4.1 are partly present in some of the documents. Concerning the invoices, the logo is sometimes printed or scanned with 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 a valid use case. Those effects create a higher variance in the extracted features, which in turn results in a worse average rank.
Another disadvantage 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 461.92 windows is extracted from the invoices, from the scientific literature an average amount of 799.81 windows are used, both cases are significantly lower than the average of 1448.01 windows that are extract from contracts.
The different presented feature extraction methods have an 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.

4.4.4 Comparison of Grubbs and k-NN

One topic that has only been briefly addressed during the experiments is the performance of the anomaly detection algorithms, namely Grubbs and k-NN. It is interesting that Grubbs usually performs better on the contract type documents while k-NN performs better on the invoice and scientific literature documents. A possible explanation is that Grubbs can only detect data points lying far away from the mean. As already mentioned in section 3.1,the 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 point lying between those clusters as an outlier. Grubbs on the other hand would not 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 the mean). This is relevant in some test subjects. Previously it was explained how several fonts (missing, italic ...) can influence the outlier detection. On the invoice and literature documents some pages have either italic fonts (often in formulas in the literature) or text rendered as pictures in parts of the document. If those

effects are only present in parts of the document, the document is still tested (as this is a relevant use case). When, for example, comparing different literature documents from the same printer it is possible that some of the pages are influenced from those effects and others are not. In this case the normal documents would form two clusters after the feature extraction. When comparing those two normal clusters to an outlier it is possible that the outlier lies in between those clusters and in that case only k-NN can detect it correctly.
An example of this effect can be found in being the case are the invoices of dataset 21. When using the standard deviaton on static windows (see section 2.1), 4 pages of the invoice types produce a significantly higher standard deviation than the rest of the documents, due to effects explained above. When comparing dataset 21 with all inkjet printed invoices using Grubbs an average rank of 5.81 is achieved. When using k-NN on the same test an average rank of 3.84 is achieved. While the result of k-NN is far from perfect it is a significant improvement compared to Grubbs (in this specific test sample).

4.4.5 General Reasons that lower the average Rank

Another reason that lowers the average rank regardless of the used anomaly detection is that the feature extraction results into similar feature values for "bad" laser printers compared to modern and expensive inkjet printers. In this case the extracted values of the inkjet printed documents are only slightly higher than the values extracted from laser printed documents. This sometimes results in a wrongly detected outlier document.

4.5 Summary of the Experiments

In the following a comparison of all different methods is given. For a better overview the best performing algorithms have been depicted in the following comparison. Schreyers global DCT feature [11], with k-NN for anomaly detection, is used as a baseline to show the improvements of this thesis. From the presented methods the features standard deviation on static windows using all connected components and the same feature combined with OCR are shown. The results for both Grubbs and k-NN are presented.

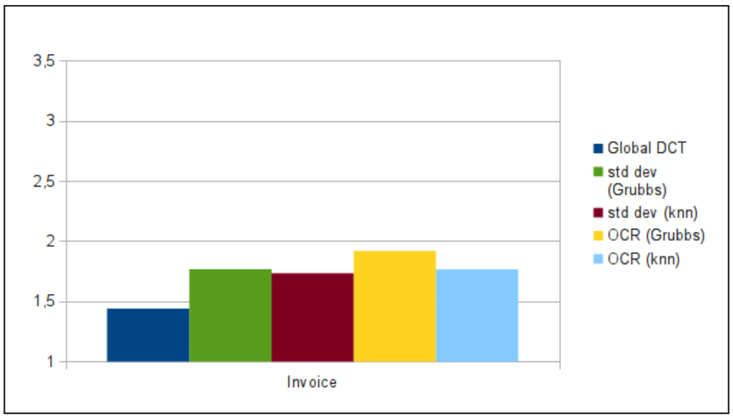


Figure 4.17: Comparison of the achieved results for invoice documents of the presented methods with the global DCT feature. Using this document type Schreyer's global DCT achieves the best results. Here, the presented methods generally work better together with k-NN.

Figure 4.17 shows the comparison of the results achieved for the invoice documents. In this case Schreyer's global DCT feature achieves the best results. A possible explanation why the global DCT works so good for invoice documents has been presented in Section 2.4. A possible explanation why the presented features do not perform as good for the invoice documents can be read in Section 4.4.3. Comparing k-NN with Grubbs shows that k-NN outperforms Grubbs in this case, a possible explanation for this is given in section 4.4.4.

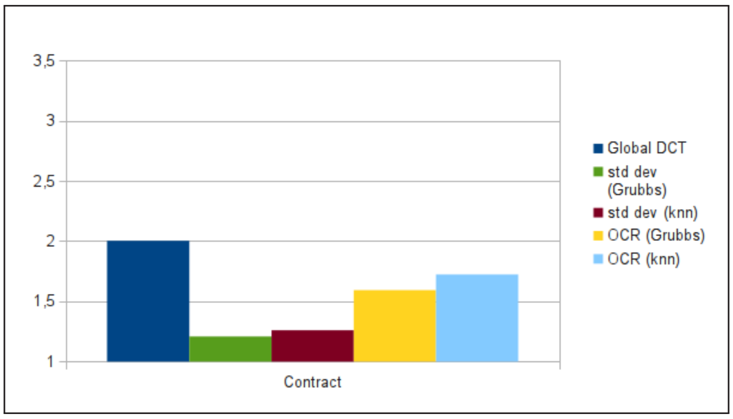


Figure 4.18: Comparison of the achieved results for contract documents of the presented methods with the global DCT feature. For this document type the standard feature using all connected components shows the best results. Grubbs is generally better than k-NN for contracts.

Figure 4.18 shows the comparison of the results achieved for contract-like documents. Contrary to the invoices, the presented methods outperform global DCT. OCR has a negative influence on the results for this document type. Possible reasons for this have been discussed in Section 4.3.5.

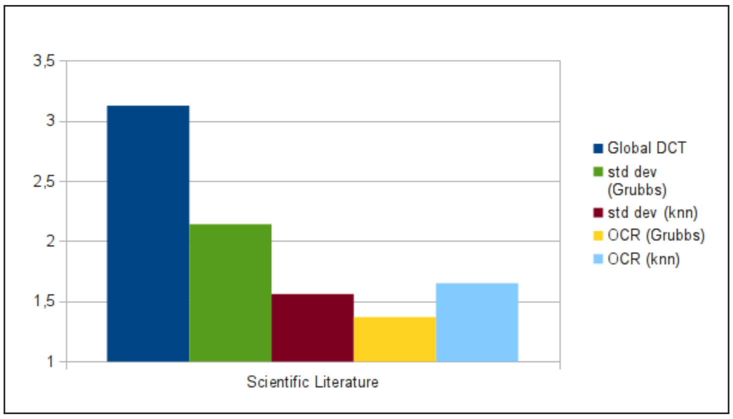


Figure 4.19: Comparison of the achieved results for scientific literature documents of the presented methods with the global DCT feature. For this document type a significant improvement has been achieved. Grubbs in conjunction with OCR yields the best results. However k-NN performs better than Grubbs without OCR.

Figure 4.19 shows the achieved results for the scientific literature. This document type shows the best improvements compared to the global DCT features. The best result is achieved when combining OCR with Grubbs test, even though k-NN outperforms Grubbs without OCR, because of reasons explained in Section 4.4.4 . Combining OCR with Grubbs outperforms k-NN in this case. Possible reason for this have been discussed in Section 4.3.5.

So far, the document types have only been examined separately from each other. This makes sense because of the varying performance of the presented methods for the different document types. For a last comparison the overall results, when combining all document types, are presented.

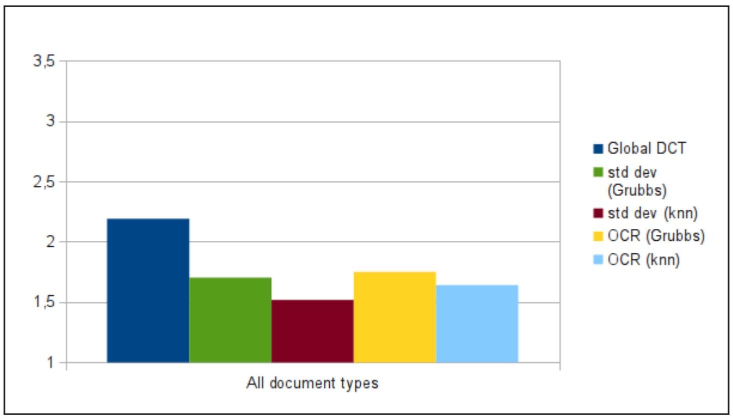


Figure 4.20: Comparison of the achieved results for all documents of the presented methods and the global DCT feature. Overall, when using all document types at once, k-NN works better than Grubbs and using all connected components works better than OCR.

Figure 4.20 shows the results of the different methods when using all documents at once. The best results are achieved when using k-NN together with the standard deviation on static windows. Compared to globald DCT an improvement from an average rank of 2.19 to an average rank of 1.52 is achieved.
The above comparison shows the results when running one of the presented method over all document types. Another possibility is to calculate the overall result when using the best method for every document type. This means for invoices the standard deviation on static windows with k-NN, for contracts the standard deviation on static windows with Grubbs and for scientific literature the standard deviation on static windows combined with OCR has been chosen.

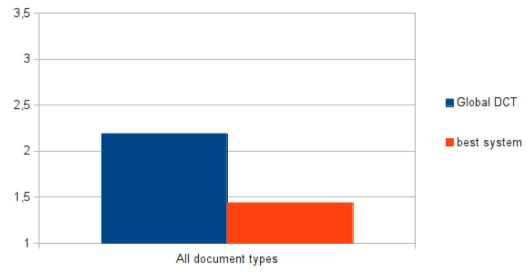


Figure 4.21: Comparison of the overall achieved results when combining the best methods for every document type and global DCT.

Figure 4.21 shows a comparison of the overall results of global DCT and the results achieved when combining the best presented methods for all document types. It shows an overall improvement from an average rank of 2.19 (global DCT) to an average rank of 1.44 when using the best presented methods.

5 Conclusion

As outlined in Section 1.1, distinguishing between different printing techniques is a useful process in the context of document authentication. Therefore, the goal of this thesis was to create a system being able to discriminate between different printer types. It should work with unsupervised anomaly detection and documents scanned at a moderately low resoultion (400 dpi). The whole process was divided into two main steps: feature extraction and anomaly detection.
Several features with different variants, have been implemented and compared to the global DCT feature by Schreyer [11]. Admittedly, not all ideas to improve the feature extraction worked as expected. The experiments have shown that some improvement ideas of the features did not achieve better results. Possible reasons and ideas how to further improve them have been discussed in the evaluation Chapter (4). None the less, the final results of the best working features and anomaly detection algorithms look very promising. Combining the best possible features resulted in a significant improvement compared to global DCT. Examining the different documents types separately from each other resulted in a good improvement for two out of three document types. Reasons for the varying results of the different document types have been discussed as well within the evaluation chapter.
In addition to the feature extraction, two different unsupervised anomaly detection algorithms - Grubbs and k-NN - have been implemented and tested. Both show promising results in different test cases. Possible reasons for the varied results in the different test cases have been examined and presented.
Furthermore a dataset featuring unique test cases for different printers has been created and published. The dataset offers a huge varian of unique pages of three different basic types. Due to releasing both the scanned documents as well as the original PDFs the dataset can either be reused for research as it is or expanded, by adding more printers, for future work.

5.1 Future Work

Some possible improvements for variants of the presented feature have already been discussed in the evaluation in Chapter 4. This thesis focuses on edge roughness and degeneration. In the diploma thesis from Schreyer [11], several other differences between printing techniques have been presented. For example, all printing techniques produce a varying amount of noise. According to Schreyer [11] it can be generally assumed that laser printers generate more noise than inkjet printers. This and other printing technique features, could be combined with the presented features, which focus on edge roughness

and degeneration, for a possible improvement of the results of the unsupervised anomaly detection approach.
Also the printing technique recognition could be used in conjunction with other approaches for document authentication. Several possible approaches have been outlined in Section 1.2.
Increasing the sample size of available printers in the dataset could also be a future goal. The dataset can be used for all kinds of experiments, therefore continually increasing the sample size would be desirable.

6 Bibliography

[1] NIST/SEMATECH e-Handbook of Statistical Methods. URL `http://www.itl.nist.gov/div898/handbook/eda/section3/eda35h1.htm`.

[2] Mennatallah Amer. Comparison of Unsupervised Anomaly Detection Techniques, 2011. URL `madm.dfki.de/_media/theses/bachelorthesis-amer_2011.pdf`.

[3] M Barni, CI Podilchuk, F Bartolini and EJ Delp. Watermark embedding: hiding a signal within a cover image. In *Communications Magazine, IEEE*, 39(8):102–108, 08 2001.

[4] Joost van Beusekom, Faisal Shafait and Thomas Breuel. Text-line examination for document forgery detection. In *International Journal on Document Analysis and Recognition (IJDAR)*, Online First:1–19, 2012. URL `http://www.springerlink.com/content/d2x6827517755293/`.

[5] Joost Beusekom, Faisal Shafait and Thomas M. Breuel. Automatic authentication of color laser print-outs using machine identification codes. In *Pattern Analysis & Applications*, pages 1–16, August 2012. ISSN 1433-7541. doi:10.1007/s10044-012-0287-5. URL `http://dx.doi.org/10.1007/s10044-012-0287-5`.

[6] Varun Chandola, Arindam Banerjee and Vipin Kumar. Anomaly detection: A survey. In *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009. ISSN 0360-0300. URL `http://doi.acm.org/10.1145/1541880.1541882`.

[7] Frank E. Grubbs. Procedures for Detecting Outlying Observations in Samples. In *Technometrics*, 11(1):1–21, February 1969.

[8] Christoph H. Lampert, Lin Mei and Thomas M. Breuel. Printing Technique Classification for Document Counterfeit Detection. In *Computational Intelligence and Security (CIS)*. 2006.

[9] Aravind K. Mikkilineni, Pei-Ju Chiang, Gazi N. Ali, George T. C. Chiu, Jan P. Allebach and Edward J. Delp. Printer identification based on graylevel co-occurrence features for security and forensic applications. In *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 430–440. 03 2005.

[10] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. In *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979. URL `http://dx.doi.org/10.1109/TSMC.1979.4310076`.

[11] Marco Schreyer. Intelligent Printing Technique Recognition and Photocopy Detection Using Digital Analysis for Forensic Document Examination, 2008.

[12] Marco Schreyer, Christian Schulze, Armin Stahl and Wolfgang Effelsberg. Intelligent Printing Technique Recognition and Photocopy Detection for Forensic Document Examination. In Gesellschaft für Informatik, editor, *GI-Informatiktage 2009*. Gesellschaft für Informatik e.V. (Gesellschaft für Informatik e.V.), 2 2009.

[13] Christian Schulze, Marco Schreyer, Armin Stahl and Thomas M. Breuel. Evaluation of Graylevel-Features for Printing Technique Classification in High-Throughput Document Management Systems. In *Proceedings of the 2nd international workshop on Computational Forensics*, IWCF '08, pages 35–46 (Springer-Verlag, Berlin, Heidelberg), 2008. ISBN 978-3-540-85302-2. URL `http://dx.doi.org/10.1007/978-3-540-85303-9_4`.

[14] Faisal Shafait, Joost van Beusekom, Daniel Keysers and Thomas Breuel. Document cleanup using page frame detection. In *International Journal on Document Analysis and Recognition*, 11(2):81–96, 11 2008. URL `http://www.dfki.de/web/forschung/km/publikationen/renameFileForDownload?filename=2009-IUPR-21Aug_0932.pdf&file_id=uploads_356`.