Printing Technique Classification for Document Counterfeit Detection

Christoph H. Lampert Lin Mei Thomas M. Breuel
German Research Center for Artificial Intelligence (DFKI) GmbH,
Image Understanding and Pattern Recognition Lab,
Kaiserslautern, Germany.
{chl,lin,tmb}@iupr.dfki.de

Abstract

The detection of counterfeit in printed documents is currently based mainly on built-in security features or on human expertise. We propose a classification system that supports non-expert users to distinguish original documents from PC-made forgeries by analyzing the printing technique used. Each letter in a document is classified using a support vector machine that has been trained to distinguish laser from inkjet printouts. A color coded visualization helps the user to interpret the per-letter classification results.

1 Introduction

In the last years, the number of observed forgeries of official documents has increased strongly. This is less the case for bank notes, where an ever ongoing race between better security features and better forgery techniques takes place. But a strong growth of forged documents can be observed in the area of identification, e. g. passports, visa and drivers licenses, for printed documents of potentially high financial value, like cheques, invoices or certificates of authenticity.

Even for documents of little individual value, e. g. bus tickets, forgeries frequently occur, because their high number and their simple creation process make them an interesting target as well. Digital imaging techniques have evolved to a level where forgeries can be created within seconds, e. g. using color photocopy machines, that are indistinguishable from the original for the untrained human eye – at least at a quick glance. In addition, an increasing number of documents are not processed by humans anymore at all, but by automatic document management systems, which are built to extract information, but not to check its authenticity. Thus, there is a large demand for automatic systems that can decide if a document is genuine or not, and the interested parties range from governmental organizations over large companies like banks and insurances down to small companies like pharmacies and even end users.

2 Image Based Counterfeit Detection

In principle there are two ways that the problem of counterfeit detection for documents can be accessed: model-based or generically. The model-based approach requires pre-knowledge on characteristic features of a document to be checked and then searches specifically for them. Often, the document are already created with the possibility for such checks in mind by including *security features* that are easy to check for later, either by the human eye or by using special devices. Typical examples are banknotes and credit cards, which contain watermarks, holograms or special ink, which is only visible in ultraviolet light. Correctly applied, these methods provide the highest level of security and many approaches to model-based forgery detection exist, e. g. Smith et al., who use machine vision to detect forgeries of CD holograms[1]. The field is developed well enough that even full textbooks have been written on it, e. g. [2]. However, model-based systems have the drawback that only those documents can be checked, for which a model of the security features is available, e. g. from a database. Many classes of documents, e. g. stamps, come in such a great variety of characteristics that a database of all such models is impractical. Other important documents can be generated by anybody on-the-fly, e. g. invoices, making a database of all originals impossible.

Some of these drawbacks are avoided in the alternative, generic way. Here, a general selection of features is extracted from a document, and the decision if a document is genuine is based only on a class membership of a document and statistical information of the expected features. Because of this limited knowledge, generic counterfeit detection systems show a larger rate of error than model-based, but have the advantage of being applicable for a wider class of documents. Typically, the generic approach does not really test if a document is genuine, but rather if a certain method of forgery has been applied.

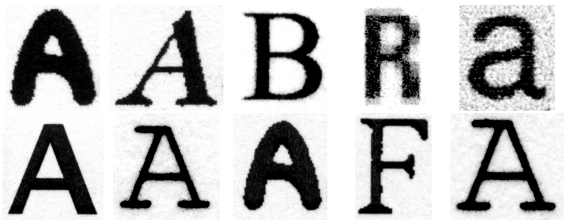


Figure 1. Sample letter from different inkjet printouts (top) and laser printouts (bottom). Laser printed letter typically show sharper contours than inkjet printed ones.

(a) (b)

Figure 2. Enlarged contour region of a inkjet printed (a) and a laser printed character (b). The laser image shows a sharper black-to-white transition and fewer ink/toner droplets outside of the black contour.

2.1 Per-Letter Classification

A canonical choice for getting a hint on the genuineness of a printed document is to study its creation process, in particular the type of printer that has been used. However, little research has been done in this area so far.

One exception are studies on the dot patterns of inkjet printers by Yamashita et al.[3]. Tchan analyzed the sharpness of image features to detect manipulations of existing printouts[4]. A more specialized solutions is proposed e. g. by Akao et al., who search for spur marks, which inkjet printers commonly leave on the paper during the printing process[5].

The approach proposed in this paper is to generically detect forgeries by using a per-letter classification of the printing technique. It relies on the fact that for different printers have different visual characteristics, especially in the edge area of the letters. From the grayscale distribution in this image area, it can be decided for each letter in the document what kind of printer created it. In parts, this is similar to Tchan[6], but there the target was to distinguish different printer models, which required details knowledge on the creation of the original document, access to the machines to be detected, and also specialized hardware for the image acquisition. The method we propose is more generic, it is possible to target a wide range of different documents types. Another advantage is, that the method does not require any specific hardware but can work with a standard consumer imaging device like a scanner or a digital camera.

The suitability of the proposed technique to counterfeit detection is backed up by forensic results on how document forgeries are usually done these days: only a very small fraction of forgers are professionally equipped with an expensive laboratory of advanced printing machinery. Instead, the vast majority of forgeries are done on an ordinary home PC with a scanner and a printer. This includes the replacement of passports photographs by inkjet printouts, and scanning bus-tickets and printing copies on a color laser printer. A surprisingly large number of forgeries is even simply done by hand, e. g. adding or manipulating digits and an invoice. It is characteristic for these kinds of forgeries, that the forged information relies on a different printing technique than the original. This can be the case that the whole document was created using a wrong technique, like for the bus-ticket, or that some parts have different, like on the invoice.

Both scenarios can be detected by the proposed method: if the original printing technique of a document is known, it can as a whole be validated against the detection technique in the document to be checked. Additionally, documents which show only few positions of different printing technique than the main body show strong evidence of a manipulation.

3 Printing Technique Classification

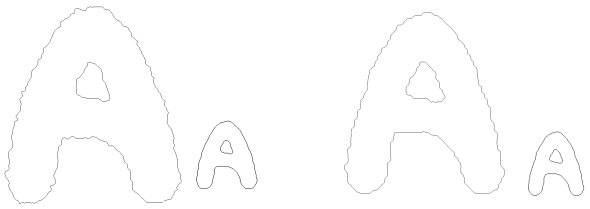
The proposed method consists of four steps: *preprocessing, feature extraction, classification* and *visualization*, which we describe in the following in more detail.

3.1 Image Preprocessing

For a more compact description, we leave out the more parts of image acquisition and assume that the system is provided with high resolution image material. Since we are dealing with document images, we usually cannot rely on color information, and we therefore assume the material to be in 8 bit grayscale format. We identify individual objects on the page by a connected component analysis on a binarized version of the image. Of the connected components we keep all which have roughly the right size and aspect ratio to be characters and extract the classification features from them.

3.2 Feature Extraction

When studying high resolution scans of different printer types, sharp edges are the most interesting regions to distinguish between them visually. Here, the techniques differ,



(a) Letter Contours for Inkjet (b) Letter Contours for Laser

Figure 3. Line Edge Roughness for Inkjet (a) and Laser (b) print. In low resolution, laser and inkjet letter contours are roughly equally smooth (small letters). In full resolution, the inkjet print has rougher contours than the laser one (large letters). Therefore, the ratio between high res and low res contour lengths can be used as a distinguishing feature.

e. g. in sharpness and existence of droplets. This is illustrated in Figure 2. From these areas, we extract the following qualities to form a feature vector: *Line Edge Roughness, Area Difference, Correlation Coefficient, Texture*.

Line Edge Roughness It is a basic observation that in very high resolution images, the contour line of an inkjet print typically is rougher than that of a laser print, and it therefore has a longer perimeter. However, if the image is scaled down, the contours of inkjet and laser printouts both become smoother and it is generally not possible to distinguish them anymore. Figure 3 shows this effect.

For classification, we calculate the ratio of the lengths of the high res and the low res contours, and we include the *scale* as a factor as well, because the roughness measure then becomes invariant to the actual resolutions used:

$$\text{roughness} = \frac{\text{perimeter of downsized object contour} * \text{scale}}{\text{perimeter of object contour}} \tag{1}$$

For our experiments, we use a scale factor of 10.

Area Difference Another characteristic property of inkjet prints is that the grayscale distribution of a black-to-white edge transition contains more intermediate grayscale values for inkjet than for laser prints. To measure this in a size-invariant matter, we use the area difference between two different image binarizations. One is created using Otsu's automatic threshold [7], the other by using a threshold δ units larger. From the difference between the area A_{otsu} of the first binary shape and the area $A_{otsu+\delta}$ of the second binary shape we obtain the number of intermediate grayscale

Figure 4. Area Difference: The size of a binarized letter (left) is increased slightly by raising the binarization threshold (center). The area of the additional pixels (right) typically is larger for inkjet printers than for laser and can be used for classification.

pixels, which we normalize by the total area:

$$\text{area difference} = \frac{|A_{otsu} - A_{otsu+\delta}|}{A_{otsu}} \tag{2}$$

This is illustrated in Figure 4 using $\delta = 20$, which is the same value that we used in our experiments.

Correlation Coefficient It is also characteristic for a printer how close its output of a letter contour comes to an idea step edge. We measure this by calculating the correlation coefficient between the original image and the segmented binary image. Since only the contour region is important to us, we use an edge image that had been dilated with a circular mask of radius 7 as a region-of-interest (ROI) mask. Figure 5 illustrates this.

When A is the original gray value image and B the Otsu-binarized version, the correlation coefficient is calculated as

$$\text{correlation} = \frac{\sum\limits_{\substack{[i,j]\in \\ ROI}} (A[i,j]-\bar{A})(B[i,j]-\bar{B})}{\sqrt{\sum\limits_{\substack{[i,j]\in \\ ROI}} (A[i,j]-\bar{A})^2} \sqrt{\sum\limits_{\substack{[i,j]\in \\ ROI}} (B[i,j]-\bar{B})^2}} \tag{3}$$

where $\bar{A}$ and $\bar{B}$ are the mean of A and B respectively over the ROI.

Texture To obtain a texture descriptor, we use a technique that is similar to the gray value co-occurrence matrix, but measures the co-occurrence of values in two different images or *channels*. One is the original image, another is a transformed version of the original. We do this for three different transforms:

Gaussian Filter We obtain the the second channel from blurring the original using a *Gaussian filter*.

Wavelet Filter We obtain the second channel from computing the first low pass approximation in a wavelets representation of the original image.

(a) Contour image and dilation (b) ROI in original and binarization

Figure 5. A region is interest (ROI) mask is constructed by dilating the contour image (a). The cross-correlation between the ROI-pixels in the original image and in the binarized image (b) measures how close the contour region is to an ideal step edge.

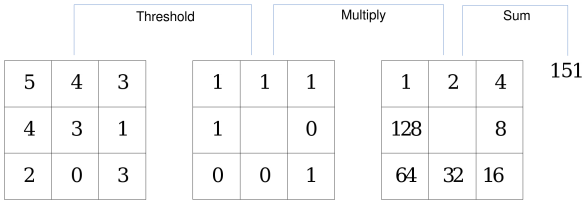


Figure 6. Local binary pattern: A 3×3 windows around each position is extracted (left). Its center value is used as threshold for binarization (center). The 8 binarized neighbors are multiplied with a mask of powers of two and summed up (right). This arranges them in clockwise order forming a byte value, in this example $1 + 2 + 4 + 16 + 128 = 151$.

Local Binary Maps We calculate the *rotation invariant local binary map*[8]. It can be illustrated in the following way: Assume a 3×3 window in the image. We calculate a binary version of this window using the center point as threshold. The resulting 8 single-bit values are combined into a bitstring in clockwise order, resulting in an 8-bit integer value. This process is illustrated in Figure 3.2. The resulting value depends on which bit the combination started with. To make the feature rotationally invariant, we using each bit as a starting point once and keep the minimum resulting value as output. The same process can also be done for larger window sizes. In our experiments, we use a 5×5 window. The values at its boundary are binarized and combined into a 16 bit output value, which is again made rotationally invariant by minimizing over all 16 possible rotations.

For each of the transformed images, we create a $2D$ value co-occurrence histogram $p[i, j]$, i.e. the histogram of how often an original grayscale value i occurs in the region of interest of image I together with a value j at the same position in the transformed image J.

$$p[i,j] = \frac{\#\{(x,y) \in ROI \ : \ I[x,y] = i \wedge J[x,y] = j\}}{\#\{(x,y) \in ROI\}}$$

From each such histogram, we extract the four features *contrast*, *correlation*, *energy* and *homogeneity* for classification:

$$\begin{aligned}
\text{contrast} &= \sum_{i,j} |i-j|^2 p[i,j] \\
\text{correlation} &= \frac{\sum_{i,j}(i-\mu_x)(j-\mu_y)p[i,j]}{\sigma_x \sigma_y} \\
\text{energy} &= \sum_{i,j} p^2[i,j] \\
\text{homogeneity} &= \sum_{i,j} \frac{p[i,j]}{1+|i-j|}
\end{aligned}$$

where

$$\mu_x = \sum_{i,j} i\, p[i,j] \quad \mu_y = \sum_{i,j} j\, p[i,j]$$

$$\sigma_x = \sqrt{\sum_{i,j}(i-\mu_x)^2 p[i,j]} \quad \sigma_y = \sqrt{\sum_{i,j}(j-\mu_y)^2 p[i,j]}$$

This results in a $4D$ feature vector per transformation, i.e. the total vector for texture features is 12-dimensional.

3.3 Classification

For classification, the 15-dimensional feature vectors are reduce to 12-dimensions using the PCA transform and normalized to the range $[0, 1]$. Afterwards, they are used as input to a support vector machine with Gaussian kernel (RBF-SVM). Support vector machines have proven to be a powerful tool for classification tasks where high accuracy is required. By use of a Gaussian kernel function, we avoid the frequent problem of parameter section, because a RBF-SVM has only two free parameters, the penalty C and the kernel width σ, and there are straightforward testing method for choosing them[9]. In our case, parameters were set to $C = 20$ and $\sigma = 1$.

Another reason for choosing a RBF-SVM was, that they utilize the advantages of a high (even infinite) dimensional classification space, and can be thought of as covering the more fundamental linear kernel SVMs as well[10].

3.4 Visualization

A big problem of fully automatic classification problems is that users don't fully trust their decisions. This is even more the case in sensitive areas like document authenticity,

where matters of national security or financial transactions are involved. We have therefore designed our system to not independently make decisions, but to guide the user into making his or her own decision. This is done by presenting the user with the classification results in form of a color annotation of the original document image. Green represents laser printed characters and red represents inkjet.

This representation allows the user to see at a single glance if a document is uniquely colored in the way he or she expects, indicating that it is an original, or if the color is uniformly wrong, indicating a complete forgery, or if it consists of different colors, which can be an indicator for a partial forgery. In the last case, the color coding shows its special strength, because the user can immediately see which parts of the documents have been detected as potentially forged. Based on the semantic meaning of these part, he or she might can decide to either consult the original document for an in-depth check, or to attribute the detection as a false positive and proceed. Typically, names and numbers are positions where a missing a forgery is potentially dangerous, whereas in the company logo or the running text of a letter, a detection error is more likely. Encoding this information into the system would require a lot of background knowledge about the documents to be checked, something that a generic counterfeit detection system typically does not have. By instead leaving the choice to the user in an interactive fashion, the system stays overall generic. We also believe that the acceptance rate with a user will be higher, because of the positive feedback to decide for oneself instead of giving the control to "a machine".

4 Experiments

4.1 Setup

We implemented a prototype of the described system in MatLab, and tested it on a dataset of 26 printouts of 8 laser and 5 inkjet printouts. All documents show only text and were scanned at a resolution of 3200 dpi. Since our algorithm works on the level of individual letters, the document images were split into connected components, resulting in 9217 image regions in total.

For evaluation, we performed two kinds of experiments. To measure the numerical classification accuracy, we performed leave-one-out testing, each time selecting all regions of one document for testing and all regions of all other documents for training. This resulted in a classification accuracy of 94.8%. However, the classification accuracy varied rather strongly, for some documents reaching 100% but in one case also dropping as low as 78%. We believe that this is caused by a lack diversity in the training material, since we only had access to a limited number of printer and paper combinations.

To demonstrate our approach of visualization and counterfeit detection, we created an example document from laser printed stationary with inkjet printed text body. It was classified and the result is presented in Figure 7. As one can see, the laser part is recognized perfectly, whereas in the inkjet section, some letters are falsely marked as laser print. However, the visualization makes it possible to see that the errors happen mainly for very small bounding boxes in particular punctuation, and not for critical parts like names or bank data. A human observer should therefore be able to classify the letter as non-forged.

4.2 Discussion

For the prototypical stage of the system, we believe that the results presented are good. It also looks promising that they can be improved by more careful parameter selection and in particular more training material. However, the current error rate of 5.2% is not low enough to establish a fully automatic system. For that the task of counterfeit detection itself is much too sensitive to errors: a single forged letter makes a whole document a forgery. Therefore, a single false decision can cause a genuine document to be detected as forgery, and a single false negative can cause a forged document to be overlooked.

A system working with such a strict decision rule would instead require an error rate in the range of 0.01% (one error per 10 documents) which appears unreachable with current methods. We have avoided this problem by not letting the system decide for itself, but by presenting the user with a visualization of the classification results in form of an assistance system.

5 Conclusion

We have presented a system that uses machine learning to detect different types of printing techniques on the level of individual letters, or even parts of letters. Furthermore, we have described a setup to detect counterfeit or manipulation of printed documents. Here, the aim was not to build a completely autonomous system in form of a black box classifier, but a system that assists a possible user in his own decision. The reported results indicate that automatic classification of the technique used to create a printed document is indeed possible with ordinary consumer hardware. So far we used two classes, inkjet and laser, but by switching to a multi-class classifier and with additional research on suitable features, it should be possible to distinguishing even more types of print and also other writing devices like ball pens. Our goal is to create a system that can indeed fulfill a useful purpose outside of the lab environment, e. g. by integration into a content management system handling invoices or receipts, as they are frequently used today at

(a) (b) (c)

Figure 7. Example of classifier visualization: A letter, that is composed of both laser and inkjet print, is classified using the proposed system. The left image shows the original scan in reduced resolution; the center and right images show the red and green color channel of the classification output. The header and footer have been identified correctly as *laser* (b). The text body shows few errors, but is mainly correctly identified as *inkjet* (c). In the software GUI, the output is color coded instead of showing the color channels separately.

companies, who have to handle a large amount of incoming paper documents, e. g. insurance companies.

6 Acknowledgements

This project was supported in part by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03) and by the Stiftung Rheinland-Pfalz für Innovation, project BIVaD (15202-386261/737). We would like to thank Jane and Thomas Bensch and Oleg Nagaitsev for their help in creating and scanning the ground truth document printouts.

References

[1] Smith, P.J., O'Doherty, P., Luna, C.: Commercial anticounterfeit products using machine vision. In: Proceedings of SPIE – Optical Security and Couterfeit Deterrence Techniques V. (2004) 237–243

[2] van Renesse, R.L.: Optical Document Security. Artech House (1997)

[3] Yamashita, J., Sekine, H., Nakaguchi, T., Tsumura, N., Miyake, Y.: Spectral based analysis and modeling of dot gain in ink-jet printing. In: International Conference on Digital Printing Technologies IS&T's NIP19. (2003) 769–772

[4] Tchan, J.: The development of an image analysis system that can detect fraudulent alterations made to printed images. In: Proceedings of SPIE – Optical Security and Couterfeit Deterrence Techniques V. (2004) 151–159

[5] Akao, Y., Kobayashi, K., Sugawara, S., Seki, Y.: Discrimination of inkjet-printed couterfeits by spur marks and feature extraction by spatial frequency analysis. In: Proceedings of SPIE – Optical Security and Couterfeit Deterrence Techniques IV. (2002) 129–137

[6] Tchan, J.: Classifying digital prints according to their production process using image analysis and artificial neural networks. In: Proceedings of SPIE – Optical Security and Couterfeit Deterrence Techniques III. (2000) 105–116

[7] Otsu, N.: A threshold selection method from gray level histograms. IEEE Trans. Systems, Man and Cybernetics **9** (1979) 62–66

[8] Mäenpää, T., Pietikäinen, M.: Texture analysis with local binary patterns. In C. H. Chen, L.F.P., Wang, P.S.P., eds.: Handbook of Pattern Recognition and Computer Vision. 3rd edn. World Scientific Publishing Company, River Edge, NJ, USA (2005) 197–216

[9] Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University (2003)

[10] Keerthi, S.S., Lin, C.J.: Asymptotic behaviors of support vector machines with gaussian kernel. Neural Computing **15**(7) (2003) 1667–1689