https://doi.org/10.2352/EI.2022.34.4.MWSF-325

Smartphone-supported integrity verification of printed documents

Waldemar Berchtold, Dani El-Soufi, Martin Steinebach; Fraunhofer Institute SIT — ATHENE; Darmstadt, Hesse/Germany

Abstract

This work discusses document security, use of OCR, and integrity verification related to printed documents. Since the underlying applications are usually documents containing sensitive personal data, a solution that does not require the entire data to be stored in a database is the most compatible. In order to allow verification to be performed by anyone, it is necessary that all the data required for this is contained on the document itself. The approach must be able to cope with different layouts so that the layout does not have to be adapted for each document. In the following, we present a concept and its implementation that allows every smartphone user to verify the authenticity and integrity of a document.

Introduction

Every day, people deal with documents in different forms - paper and digital. Although some documents are now issued digitally, the majority of documents issued by government agencies are still printed due to unresolved privacy and online security issues. Advances in scanning, printing and imaging tools and technologies have made it easy to manipulate and forge the original document. These advances have tempted fraudsters to create fake and counterfeit documents that are indistinguishable from original documents. Therefore, verifying the authenticity and integrity of paper documents has become an important research topic.

The growing awareness and need to secure paper-based documents has occupied research for more than two decades. At the beginning of this research topic, the solution was to create a traditional seal through visual cryptography. In various studies over the last two decades, digital signatures have been visualized by visual cryptography and complex image frequency modulation to resemble a traditional seal [1, 2, 3]. Verification of the integrity and authenticity of presented documents has traditionally relied on recognizable visual features of the document [2]. These solutions usually require expensive and specially developed printing techniques, special ink, fluorescent particles in the paper, watermarks, and holograms [3].

Winter et al. [4] proposes the use of cryptographic digital signatures to secure important features of a document. The document is digitally sealed by enclosing all required digital certificates and important document features in a high capacity matrix code. Integrity verification is based purely on checking the information in the matrix code. Integrity verification of printed documents is the most difficult part in practice. Newer approaches require the user to manually verify the integrity of the document by comparing the information visible in the application with that printed on the document. In addition, these approaches propose the use of online services to meet the security requirements of authenticity and integrity [5].

Digitally issued documents can be copied and redistributed. Therefore, traditional methods of sealing the document can no longer be used. Digital signatures provide inherent security features required to prove authenticity, information integrity, and non-repudiation of a document. However, documents must be presented in paper form to various entities under different circumstances. Printing a digitally signed document negates all security features because verification is performed at the bit level of the digital document. In addition, the person responsible for processing a submitted document does not know the actual characteristics and content of the authentic document. Consequently, this person is not able to verify the authenticity and integrity of the document. Manual verification of these documents is a tedious and time-consuming task that requires human interaction at multiple organizational levels.

Document integrity verification is an important aspect, and it can be optimally performed with smartphone support. Smartphone cameras enable the capture of high-quality images and the scanning and decoding of high-capacity barcodes. High capacity barcodes, such as the JAB code, enable the inclusion of all data necessary for verification. The high capacity barcode, in combination with various compression algorithms, can store enough data so that the authenticity and integrity of the document can be verified without online services. Therefore, privacy-sensitive information does not need to leave the boundaries of the document. In addition, advances in machine learning (ML) and natural language processing (NLP) have improved OCR recognition rates. Together, they serve as the foundation of the work.

The goal is to present a concept that can be adapted to Winter et al. [4] and to evaluate it for paper-based documents using available smartphone technology. It uses the current API on iOS as well as the MLKit on Android, which are based on different technologies, such as ML and NLP, for text recognition. The available techniques apply OCR to a given image to convert analog text into machine readable text after a document is recognized. To achieve the goal, requirements were defined and an analysis was performed in comparison to the state of the art. These requirements ensure robustness of OCR text and document recognition, usability, privacy, and performance of the overall system. It also analyzes the state of the art of OCR implementation using different font types and sizes under different ambient lighting conditions. The speed of the system was measured and finally the results are discussed.

Related work

Liu et al [6] proposes a concept in which the image of a seal certificate is integrated into a digital certificate. The purpose of this solution is to bridge the gap between traditional seals and digital signatures. The solution defines new private extensions to the X.509 v3 certificate structure that include an Object Identifier (OID). This allows a seal image to be embedded in a digital certificate. The seal image represents the traditional seal type, such as a signature or name, which is a distinctive and recognizable constant sign for the signer. The seal image and associated information are submitted to the issuing certification authority, which issues a new public key certificate. The certificate is embedded in the document and a digital fingerprint is calculated. Then the digital fingerprint is signed with the private key to create a digital signature of the document. The recipient is presented with the signer's digital certificate along with the received sealed and signed document for verification. Sun et al [3] proposes a solution for authentication of printed documents based on optical watermarking. The security of the system is provided by a content-based key sharing scheme derived from visual cryptography. The applied visual cryptography technique is used to generate the watermark. The watermark is embedded in the printed document by frequency modulation. To improve image quality during authentication, the watermark is embedded in smooth areas of the document. The watermark image, such as the logo and other relevant information of the third trusted party (TTP), the user, and the characteristics of the document, are fed into predefined visual key generators to produce a set of secure authentication keys. Only one of the generated keys is used as a watermark seed that is embedded in the document. The other keys are used to verify the authenticity of the document by overlaying them. When overlaid, the watermark image becomes visible if the document is genuine.

Requirements

We define three categories of requirements for an integrity verification system, namely robustness, usability, and privacy.

Robustness We first address the requirements for a robust integrity verification system, such as typography selection, good document recognition, skew correction, image thresholding to improve OCR performance, and recognition error correction.

Typical OCR systems are trained with large data sets to handle multiple languages, characters of any font, and handwriting. The accuracy of such systems depends on the size of the training dataset and the similarity of the different fonts [9]. Therefore, the recognition rate varies depending on the font type and size in a document. Depending on the type of document, a recommended list of font types and sizes must be defined to achieve high and consistent recognition rates. Public agencies typically use self-defined corporate design guidelines that describe the consistent look and feel. The guidelines include a logo, color palette, document styles, and preferred fonts and sizes. Here we use five of the most commonly recommended fonts and sizes at German universities, Charter, Arial, Helvetica, Times New Roman, Verdana. Fonts have different body, weight and body sizes. Therefore, words may appear smaller with the same font size, but with different fonts. Also, the smartphone camera is usually positioned so high that the edges of the document are visible. Therefore, smaller fonts lose sharpness at a greater distance, which affects recognition rates. In this work, three different font sizes, i.e., 9, 10.5, and 12 points, are considered, which allows understanding which fonts and sizes achieve more consistent recognition rates.

Rectangle detection approaches are applied to detect documents with different aspect ratios in the camera's field of view. However, recognizing a document in an image does not automatically mean that the recognized document contains text. For us, it is important that only documents containing text are recognized and passed to OCR. Another aspect of robustness is the recognition of documents with different contrast between paper and support. Depending on the rectangle recognition algorithm, the recognition performance varies. In addition, it should be possible to recognize documents from different camera positions that lead to skewing. Skew occurs whenever the position of the camera and the physical document are not perfectly aligned. The camera plane should be perpendicular to the physical document, and the horizontal lines of the document should be aligned parallel to the camera plane for optimal alignment. Such optimal conditions are usually difficult to achieve without special equipment. Document images with skew may change the proportions, size, shape, and angle of the physical document. These perspective distortions typically have a negative impact on OCR text recognition rates. A perspective transformation should mitigate distortion effects and correct for possible skew in the document image.

Physical documents can have artifacts such as spots and curvatures. The smartphone camera smartphone camera can exacerbate these artifacts due to uneven illumination. It can create faint characters on the document image and affect the text recognition rate of OCR. The image thresholding process is an essential step of image pre-processing in OCR. The process segments the color document image into text and background by removing existing artifacts and highlighting black text on a white background. However, depending on the uniformity of the light source and luminance, the thresholding process can produce soft and faint images. Therefore, a thresholding process that adapts to different lighting conditions should be used.

Integrity verification can fail due to text recognition and text alignment errors generated by OCR. Physical text alignment errors are difficult to detect because they change the physical layout of the document, making detection difficult without prior knowledge of their likely occurrence. In comparison, logical text alignment errors are easier to detect. Depending on whether the error is an over-segmentation error or an under-segmentation error, different approaches to segmentation correction must be applied. Text recognition errors are even more difficult to correct because an attacker can corrupt the words. Therefore, a list of the best recognized candidates from OCR should be used to test for possible word matches.

Usability Usability requirements are defined below, such as the efficiency of document capture, the importance of recognition error correction from a usability perspective, and the visualization of erroneous text positions.

The speed and accuracy of document scanning depends mainly on the performance of the smartphone and its camera. The simplicity of the methods used at each stage, i.e. from document recognition to image capture and thresholding, leads to higher process efficiency and lower latency. Slow acquisition processes require a longer waiting time for a consistent focus to capture a better quality image. Therefore, an efficient document recognition and capture process is required.

The detection and correction of OCR errors, are essential for the robustness requirements of the integrity checking process and to improve the user experience. In text recognition, there are always error rates and over-segmentation and under-segmentation errors that cause false positives and user confusion. Therefore, detection and correction of recognition errors are necessary to improve the user experience.

OCR errors or document falsification must be displayed to the reviewer so that reviewers can make a decision. The errors should preferably be highlighted on the image of the scanned document in the application. In addition, the authentic characteristics of the document must be displayed above each erroneous item so that reviewers can classify the document as genuine or a forgery after manual review.

Privacy Privacy is one of the main concerns when it comes to documents with personal references. Systems proposed so far provide online services to verify authenticity and integrity. On the one hand, this means that personal data is stored in a database and must be transmitted over the network. This creates security and privacy risks. Online services can be compromised, and sensitive information can leak out. Therefore, an offline authenticity and integrity verification system is essential to achieve this goal.

Concept

The approach as shown in Figure 1 is used to verify the integrity of digitally sealed printed documents using a smartphone application. The problem is addressed by first applying a document recognition algorithm to each smartphone camera image. A cropping algorithm is applied to extract the document portion of the image, followed by skew and perspective correction and thresholding for the recognized document. A modern OCR technique embedded in the smartphone operating system is used to digitize the printed text. Documents exist in various complex layout formats. Therefore, a layout-independent text alignment technique is implemented to correctly match the machine-readable text with the expected ground truth. The proposed text alignment technique allows the definition of fixed-length skip sequences. The skip sequence allows to skip optional parts of the document that are irrelevant to integrity. Integrity irrelevant parts of the document may unintentionally increase the size of the payload printed on the document. Due to the size limitations of a physical document, the size of the payload data must be optimized. The payload data is encoded into a matrix code that is read by the smartphone application. The proposed text alignment method generates a control sequence of the expected ground truth that is matched with the OCR output. This sequence is used to verify the integrity of the presented document. After the verification procedure, the application presents the validation result to the user by highlighting all mismatched positions in the document image.

Implementation

The application design follows the commonly used Model-View-Controller (MVC) software design pattern. The model is a dynamic data structure that represents and manages the application's information. The view is responsible for representing data, and it is directly visible to the user. The controller accepts input and converts it to commands for the model or the view. Figure 2 shows the application view components and the navigation flow between them.

The CameraViewController is the starting point of the application. The controller configures the smartphone's rear main camera and initializes a new camera session for high-quality photo captures. The controller also instantiates a document detector and an image processor for document recognition and image preprocessing. The camera continuously transmits image data to the document detector. The detector delegates a high-quality image capture when a document is detected. The captured image is transferred to the image processor, which extracts the detected document from the image frame, performs skew correction, and applies various image preprocessing methods to the image. Then, the CameraViewController instantiates the DocumentViewController and passes it the processed image. The DocumentViewController is the main view in the application's navigation workflow. It allows inspection of the image by navigating and passing the image to the DocumentDetailViewController. The Document Detail View implements various gestures, such as panning and pinching, and saves the image to the user's photo gallery. The DocumentViewController allows validation of the scanned document by pressing the validate button in the view. The control sequence received from the JAB code is inserted into the input field of the displayed modal ValidateSequence view. The validation process is triggered by pressing the validate button in the modal view. The integrity validation process is divided into several steps. First, the document image is passed as input to a text detector, which triggers a text recognition request to perform OCR. Three candidates with the highest confidence rate are selected for integrity checking. The coordinate system of the detected text positions in the vision framework starts from the lower left corner of the image. Therefore, the y-coordinate of each recognized text position is inverted to match the natural coordinate system of the UIKit framework, which starts in the upper left corner of the screen. Also, the x and y coordinates of each text position are scaled to the size of the image and sorted by their x and y coordinates. Then, the identified text positions are passed to the control sequence validator. Finally, the unmatched text positions are returned and highlighted on the document image, and the validation result is displayed in a separate modal view.

The document detection process consists of a detection of a rectangle. We configured a maximum size of 0.5, maximal observation of 1, minimum confidence of 0.99, quadrature tolerance

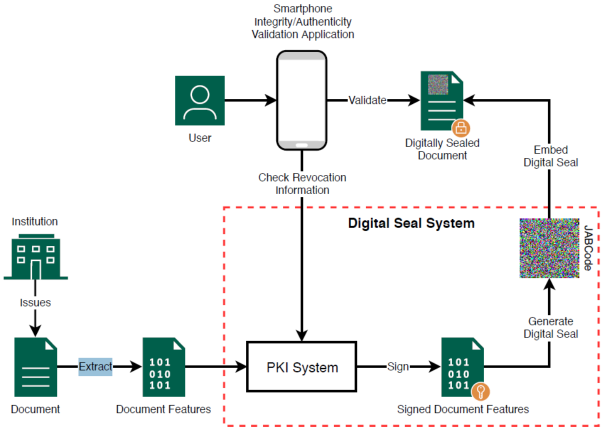

Figure 1. *Digital seal generation, integrity and authenticity validation system concept.*

of 20 and aspect ratio in the range of $[0.65, 0.75]$. Each image frame of the camera is passed onto the detection request for rectangular detection. A document is considered to be in the frame, only if one or more text positions are detected. The used Kits provides image perspective correction filter that takes four vectors for all four corners of the image as input to calculate the perspective transformation. Consequently, the scaled vertices are passed to the perspective correction filter to transform the perspective corrected image.

Mainly two influencing factors affect the quality of the image threshold, namely the image quality output from the camera and the improvement level of the image thresholding technique. Smartphone cameras use various image pre-processing techniques to enhance the image and, in many cases, help reduce blur and overexposure and improve low-light images in difficult scenes. The first step, therefore, is to configure the smartphone's main camera to achieve better image quality. The configuration is set as follows: Automatic High Dynamic Range (HDR), monitoring of subject changes, automatic improvement in low-light conditions, continuous autofocus at close range, continuous automatic adjustment of exposure level, continuous automatic adjustment of white balance level. In the next step, a focus adjustment observer is registered to receive notifications about whether the camera is adjusting its focus to delay image capture. Once the focus of the camera Once the focus of the camera is locked, the document detector processes the next image to detect the position of the document in the image. A point of interest (POI) is then calculated in the center of the detected document. Then the camera is adjusted to refocus on the detected POI. A high quality image is captured to which a predefined set of automatic image enhancement filters are applied to the image, listed at the Apple API under CIImageAutoAdjustmentOption.

Error Correction and visualization There are two types of logical errors that can affect the integrity checking process, namely character recognition errors or over- and under-segmentation. These errors depend heavily on the recognition performance of the OCR. Character recognition errors are characterized by one or more transposed characters in a single word and unrecognized characters or words in a text segment. Validation checks against the control sequence that expects words to be at a specific position in the text. The integrity check of the recognized sequence fails if the word is misspelled or another word is expected at that position. OCR returns multiple recognized text observations of a recognized word, sorted by recognition confidence rate. Therefore, three text observations that have the highest confidence rate are selected. Recognition error correction is performed in the following steps. First, the control sequence is validated using the candidate with the highest confidence rate. If the validation fails, the control sequence is validated using the second and third candidates. Validation may also fail if a different word is expected at the current position, which is the case if a word is not recognized by OCR. An unrecognized word creates an offset in the validation position of the control sequence, resulting in severe validation errors. Therefore, a look-ahead to the control sequence is implemented to detect and correct the position of the control sequence. The look-ahead on the control sequence is initiated when the validation of the sequence fails based on the first, second and third text candidates. The look-ahead checks whether the current sequence matches the position for these three candidate observations and the next control sequence position. Accordingly, the current segment is marked as invalid and the control sequence position is advanced.

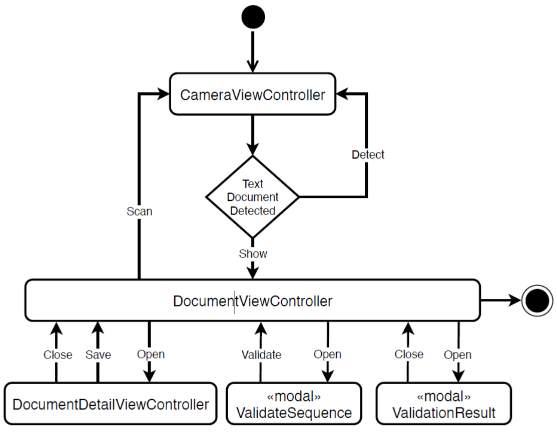

Figure 2. Application workflow chart for different views and actions.

Horizontal segmentation errors, such as merging and splitting, are detected and corrected by advancing either the control sequence or the detected sequences. Horizontal merging occurs when OCR unexpectedly merges two segments into one. This means that the position of the expected control sequence has changed for validation of two expected sequences into one. Therefore, a look-ahead technique is applied to merge the current control sequence with the following one. The integrity check condition is rechecked after the merge, and the position of the control sequence is advanced. In contrast, horizontal separations are detected by applying the look-ahead technique and merging the currently detected sequence with the following sequence. Then, the merged sequence is matched with the current position of the control sequence. Thereupon, the current position of the detected sequence is brought forward.

Sequences that failed control sequence validation are returned with corresponding positions as bounding boxes according to the validation process. These failed sequences are highlighted at their occurrence in the document. However, the expected text needs to be shown to the verifying user to inspect the failed sequences manually. Moreover, line spacing can vary depending on the document's layout and the used font size, which makes it challenging to draw expected text without overlapping other areas in the document. Therefore, a technique to optimize the placement of the text near adjacent failed areas is implemented. The technique determines all intersecting rectangles for each failed rectangle position. Then, the height of the failed rectangle is adjusted by subtracting it from the intersecting rectangle's height. Moreover, the height of the failed rectangle is reduced to half, and the vertical height position is centered. The system's default font was selected to show the control sequence at font size 25. Finally, the control sequence is shown above the rectangle's position at 25% of the rectangle's height.

Evaluation

Our experimental setup consists of documents with the five different fonts and three different sizes. Further we evaluate the robustness and usability by two specific documents with the charter font. Figure 3 shows that the different fonts have roughly identical error rates at a font size of 10.5. While the error rates for Arial, Helvetica and Times New Roman increase with decreasing font size, the error for Verdana remains about the same for all three font sizes and Charter has the lowest error rate across all font sizes. The results led us to continue further evaluations with the charter font.

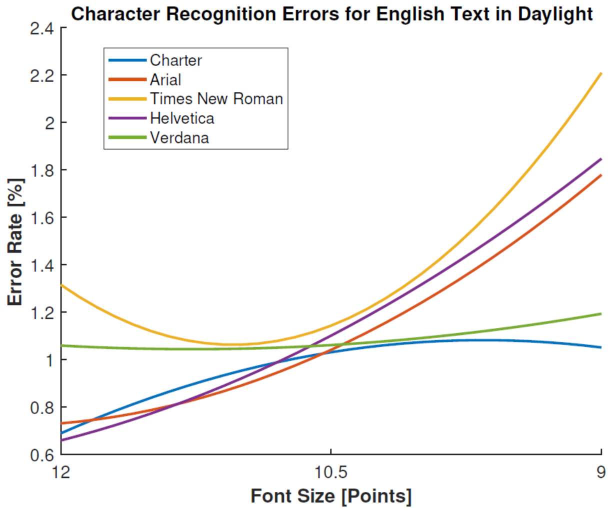

Figure 3. *Results for the error rate of the OCR for different fonts depending of the font sizes.*

With the language, it was noticeable that the error rates with English text were only about half as high as with German text. When rotating the camera to the document, one can see that the OCR used has high error rates if the document is not aligned before the OCR process. The document recognition time was 1.26 seconds on average.

The figure 4 shows how, in case of a difference, the too digitally stored information is inserted above the text. This means that the user only has to check at these points whether it is an OCR error or a forgery, as in figure 4, for example, the year of birth and the registration number have been changed. As you can see, the method for displaying the differences is not yet optimal.

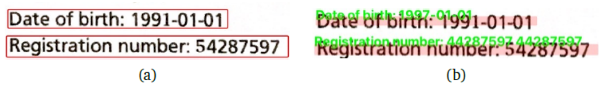
Date of birth: 1991-01-01
Registration number: 54287597
Date of birth: 1997-01-01
Date of birth: 1991-01-01
Registration number: 44287597 44287597
Registration number: 54287597
(a)
(b)

Figure 4. *Example of integrity failure visualization of (a) failed rectangular positions and (b) visualized control sequence at failed positions with large vertical line spacing.*

Conclusion and future work

In this work, an approach for integrity verification of digitally sealed printed documents using a smartphone application was presented. The proposed approach was implemented according to the defined requirements to achieve robustness, usability and privacy. Integrity verification scheme using machine-readable text for paper-based documents. Unlike most existing approaches, this approach provides automatic integrity verification of documents.

Due to the limitations in the state of the art implementation, the proposed concept was implemented as a smartphone application. The robustness and the usability of the state of the art implementation were compared to the implemented approach. In future work, the image thresholding technique should include a light intensity parameter to adjust the thresholding intensity and reduce the background noise. Although the proposed text alignment technique corrected horizontal over-segmentation errors, it is unclear under which conditions vertical over-segmentation errors occur. Therefore, a comprehensive analysis should be performed in future work using various vertical and horizontal line spacings, fonts, fonts sizes, and different text lengths.

References

[1] Jie Feng Lin, Wen Lue Chen, Zhi Hua Hu, Jian Kun Shun, New Transform Based on Solution of Singular Value for Electronic Seal, Electrical Insulating Materials and Electrical Engineering, 2012.

[2] A. Masoud, S. Ibrahim, and M. Salleh, Printed Document Authentication Using Watermarking Technique", In: 2010 Second International Conference on Computational Intelligence, Modelling and Simulation. Oct. 2010, pp. 367–370.

[3] Q. B. Sun, P. R. Feng and R. Deng, An optical watermarking solution for authenticating printed documents, Proceedings International Conference on Information Technology: Coding and Computing, 2001, pp. 65-70.

[4] C. Winter, W. Berchtold, J. N. Hollenbeck, Securing Physical Documents with Digital Signatures, In: 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP). Vol. 09. 2019, pp. 1–6.

[5] Zahrah Yahya et al. A New Academic Certificate Authentication Using Leading Edge Technology, ICEEG 2017: Proceedings of the 2017 International Conference on E-commerce, E-Business and E-Government 2017, pp. 82–85.

[6] Liu, Vicky, Caelli, William, Foo, Ernest, & Russell, Selwyn (2004) Visually Sealed and Digitally Signed Documents. In Estivill-Castro, V (Ed.) Computer Science 2004. Proceedings of the Twenty-Seventh Australasian Computer Science Conference (ACSC2004). Australian Computer Society, Sydney, New South Wales, pp. 287-294.