Extracting Handwritten Annotations from Printed Documents Via Infrared Scanning

Andreas Schmid
University of Regensburg
Regensburg, Germany
andreas.schmid@ur.de

Lorenz Heckelbacher
University of Regensburg
Regensburg, Germany
lorenz.heckelbacher@t-online.de

Raphael Wimmer
University of Regensburg
Regensburg, Germany
raphael.wimmer@ur.de

ABSTRACT

Despite ever improving digital ink and paper solutions, many people still prefer printing out documents for close reading, proofreading, or filling out forms. However, in order to incorporate paper-based annotations into digital workflows, handwritten text and markings need to be extracted. Common computer-vision and machine-learning approaches require extensive sets of training data or a clean digital version of the document. We propose a simple method for extracting handwritten annotations from laser-printed documents using multispectral imaging. While black toner absorbs infrared light, most inks are invisible in the infrared spectrum. We modified an off-the-shelf flatbed scanner by adding a switchable infrared LED to its light guide. By subtracting an infrared scan from a color scan, handwritten text and highlighting can be extracted and added to a PDF version. Initial experiments show accurate results with high quality on a test data set of 93 annotated pages. Thus, infrared scanning seems like a promising building block for integrating paper-based and digital annotation practices.

CCS CONCEPTS

• **Applied computing → Document scanning**; **Annotation**.

KEYWORDS

annotation extraction, multispectral imaging, computer vision

ACM Reference Format:
Andreas Schmid, Lorenz Heckelbacher, and Raphael Wimmer. 2022. Extracting Handwritten Annotations from Printed Documents Via Infrared Scanning. In *CHI Conference on Human Factors in Computing SystemsExtended Abstracts (CHI '22 Extended Abstracts), April 29-May 5, 2022,New Orleans, LA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3491101.3519872

1 INTRODUCTION

Despite an abundance of digital tools for annotating documents, a significant number of people prefer paper over screens for proofreading or annotating complex texts. Twenty years ago, Sellen and Harper [19] argued that the unique properties of paper "will likely continue to make it the preferred medium for certain work tasks in the foreseeable future". Indeed, in an informal survey among 65 students at our university - most of them studying computer science - 83.1% stated that they annotated on paper while only 70.8% did this on a computer.

To leverage the unique advantages of physical paper and digital documents, document-centered workflows should allow seamless transfer of digital annotations onto paper printouts and vice versa.

Efficient extraction of annotations on paper documents is especially important considering the sheer amount of such documents in existence [18]. Use cases for extracting handwritten text from printed documents reach from automatically parsing filled-out forms [22] to cleaning scanned documents [15], and digitizing notes or proofreading annotations [20].

Most approaches to annotation extraction either require a digital version of the printed document or use a machine learning approach that relies on a large training data set.

In this paper, we present an alternative method for accurately extracting handwritten annotations from printed documents using an off-the-shelf flatbed scanner to which we added an infrared (IR) light source. By scanning documents in two passes - RGB and IR - we can distinguish between IR-absorbing black toner and IR-transparent colored inks on a sheet of paper. Simple image filters deliver one bitmap containing the original printed document and another one which only contains all annotations. Extracted annotations can also be inserted into a PDF version of the document.

2 RELATED WORK

2.1 Annotation Extraction

Identifying and extracting handwritten annotations from printed documents is a well-explored challenge. In 1998, Stevens et al. [20] developed a method for matching annotations to text segments in a scanned document. Users could tab through annotated words with a plugin for the Emacs[1] text editor. The document's original text in ASCII format was required for this system to work.

Zheng et al. [21] propose a method for extracting handwritten text which identifies word segments in a document image. For each segment, a Fisher classifier determines whether it contains printed or handwritten text. Adjacent handwritten segments are merged into larger zones. Even though handwritten and printed text could be distinguished with an accuracy of 97.3%, only text-based annotations, such as notes, could be recognized. Also, strong image noise would get mistakenly recognized as annotations.

Chen et al. [4] extended this approach by using an alpha shape tree for classification on the pixel level. Their model classifies pixels in the document as *machine-printed*, *handwritten*, *noise* or *unsure*. Handwritten annotations were recognized with an accuracy of 96.4% and a false positive rate of 11.1%.

[1] https://www.gnu.org/software/emacs/

CHI '22 Extended Abstracts, April 29-May 5,2022, New Orleans, LA, USA

Nakai et al. [16, 17] propose a method for extracting annotations by aligning the scanned image with the original document and then subtracting one from the other. To align the images, they are split into color clusters with the k-means algorithm. The centroids of connected components are used as feature points for a *RANSAC* feature matcher. After alignment, the image of the original document is subtracted from the annotated document image so that only annotations remain. This method achieves a precision of 85.6% and a recall of 81%. It works for all kinds of documents and annotations, but a clean original document is required. In 2009, David Bargeron from Microsoft [1] received a patent on a very similar method which he called *"annotation lifting"*.

Kölsch et al. [12] compared different machine learning classifiers based on fully convolutional neural networks and different data augmentation strategies to locate handwritten annotations in historic documents. The best model in their comparison – *FCN-8s* with *Inception*-style data augmentation – achieved an *intersection over union* (IOU) score of 95.6%.

Banjlaidel et al. [2] combine a *Gabor filter* with *Fourier descriptors* and *Hue's moments invariant* to classify text regions as machine-printed or handwritten. Even though this combined approach achieved an accuracy of 98.5%, it is not suited for inserting extracted annotations back into the original document as annotations are extracted with their axis-aligned bounding boxes.

Nagabhushan et al. [15] propose a method for removing in-line and between-line annotations from document images. Printed text lines are identified as peaks in a histogram; regions between them are cleaned. Remaining handwritten text is separated from printed text by identifying large vertical edges at the character level. Even though 93.49% of handwritten annotations in a data set of 170 documents could be removed, annotations overlapping printed text can not be detected and removed with this method.

2.2 Multispectral/Hyperspectral Imaging

Multispectral imaging describes the process of capturing not only visible light intensity at RGB wavelengths but also at other wavelengths, such as infrared or ultraviolet light. With hyperspectral imaging, narrow spectral bands over a continuous spectral range are captured. A popular real-world use case of multispectral imaging is the removal of visible dust and scratches from film slides using infrared light. This is done by capturing an image of the film slide under red, green, blue and infrared light. As the film's colors are transparent in the infrared spectrum, the IR image only contains physical wear like dust and scratches. This image can be subtracted from all color images to remove such imperfections.

Another application of multispectral imaging is the enhancement and restoration of historical documents. Kim et al. propose a method for cleaning scans of old documents with a hyperspectral camera [7]. By combining images captured at different frequency bands, ink bleed, ink corrosion and brown spots caused by foxing[2] can be removed and details can be recovered.

As different inks respond at different frequency spectra, multispectral imaging can be used in document forensics, for example to detect counterfeits. Kaur et al. [8] investigated whether it is possible to derive the application sequence of different pens, typewriter, and printer ink based on their absorption spectra. Even though their results were negative, they provide a good overview over those spectra, indicating that colored pens are not visible at 900 nm and above.

Another application of hyperspectral imaging in forensic document analysis is analyzing documents for mismatching ink. Khan et al. [9, 10] were able to precisely distinguish between similar looking inks on the pixel level based on their frequency response combined with feature selection algorithms, even if they were not distinguishable in an RGB image.

Khan et al. [11] propose a method of converting an off-the-shelf pass-through scanner into a multispectral scanner by connecting multiple LED's with different colors to the scanner's light guide. A document can be scanned multiple times while manually switching between LEDs to capture images at different frequency spectra. This low-cost alternative to expensive multispectral cameras allows for applications like forgery detection "at home". However, as a document feeder was used instead of a flatbed scanner, pixel-perfect alignment of individual scans is not guaranteed.

Malik et al. [13] captured a data set with patches of 100 document images using a hyperspectral camera. They found that printer ink is visible along the camera's whole spectrum whereas pens disappeared on some frequency bands. Therefore, they suggest that hyperspectral imaging might be used to extract handwritten signatures from printed documents.

We build on these findings and describe a robust and versatile method that combines multispectral scanning with a custom computer-vision algorithm.

3 METHOD

We present a method for extracting handwritten annotations from printed documents with a modified flatbed scanner (*Canon LiDe CanoScan 30*[3], up to 1200 × 2400 dpi). An infrared LED (*SFH 4346*[4]) was affixed to the scanner's light guide on the opposite side of the RGB LED that is normally used to illuminate the document (Fig. 1, 2). By switching between those light sources with optocouplers, our device can produce scans illuminated only with infrared light, as well as normal RGB scans. As most inks used in pens are invisible in the infrared spectrum [8], handwritten annotations are only visible in the RGB scan. Therefore, annotations can be extracted from the RGB scan by processing both images and finding the difference between them.

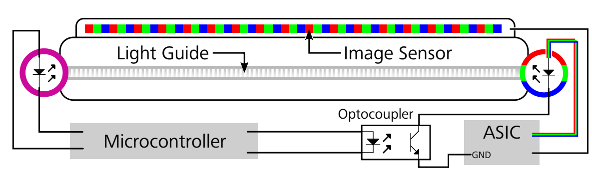

Figure 1: Schematic of the modified scanner. An infrared LED (left) was added to the light guide at the opposite end off the RGB LED (right). A microcontroller can switch both LED's to allow for RGB or IR scans.

[2]https://en.wikipedia.org/wiki/Foxing

[3]https://www.canon-europe.com/scanners/flatbed-scanners/canoscan_lide_30/

[4]940 nm, radiant intensity 90 mW/sr, φ 20° emission angle

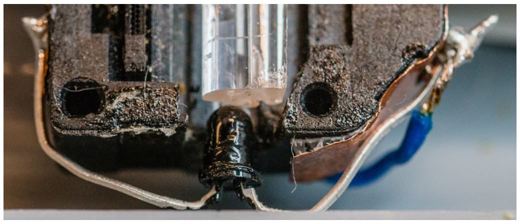

Figure 2: Infrared LED attached to the scanner's light guide.

As pixel-perfect alignment of both scan passes is necessary for our method, we used a flatbed scanner instead of a pass-through scanner like Khan et al. [11] which would require an additional step of aligning the images similar to Nakai et al.'s [16, 17]. Another approach that would also work with pass-through scanners would be to switch between IR and RGB illumination for each scan line. We plan to explore this in the future.

Both LED's can be switched with a *Wemos D32* microcontroller[5] which is controlled by a custom scan program via pyserial[6]. The scan program is implemented in *Python* and uses the *python-sane*[7] library to control the scanner. When the scan program is started, it switches the scanner to IR mode and performs an 8 bit grayscale scan. Afterwards it switches to RGB mode and performs an 24 bit color scan. Both scan passes are saved as PNG files. Currently, the whole process takes 73 seconds per page for scans with 300 dpi.

3.1 Image Processing

To extract annotations, both RGB scan (Fig. 3a) and IR scan (Fig. 3b) have to be processed beforehand for best results. Additionally, an IR scan of the empty scanner is used as a bias image (Fig. 3c) to compensate uneven illumination of the IR image. This bias image has to be captured only once.

After processing the scans, extracted annotations are saved as PNG files with a transparent background. These can be optionally inserted into a PDF document. Image preprocessing, annotation extraction, and inserting into a PDF is implemented in Python using the OpenCV 4 computer-vision library [3] and the pdf-annotate[8] library. Image processing takes about 80 seconds for a 10 page document and a scan resolution of 2539×3507 pixels[9]. As our unoptimized pipeline spends two thirds of that time with image loading and PDF generation, we see considerable room for improvement.

3.1.1 Annotation Extraction. The following annotation extraction process is used to generate an IR image with white text and black background and an RGB image with white background and preserved colors.

As the IR scan is illuminated unevenly, the bias image is inverted and overlaid with `cv2.addWeighted()` using 50% alpha for both images to normalize illumination across the image. The result is then binarized with `cv2.threshold()` and inverted for the text to be white. We use a threshold value of 110. This value has to be tailored to the scanner's exposure – too low values lead to artifacts such as visible borders around printed text, too high values lead to bright annotations getting lost. As soft borders around printed text get lost in the binarization step, one iteration of `cv2.dilate()` with a 3×3 kernel is performed. White pixels are stretched vertically by shifting the whole image up and down by two pixels each and adding together the results.

As the scanner we used captures one color at a time while moving across the document, color channels are not aligned perfectly, resulting in color fringing on the top (red) and bottom (blue) edges of letters of the RGB image. We reduce this fringing by extracting the red and blue channel of the RGB image, shifting them by one pixel in opposite vertical directions and blending them together with `cv2.addWeighted()`.

The RGB image is then processed so that the background is plain white but the original color of annotations is preserved. To this end, the RGB image is converted to HSV color space so that the background can be masked by selecting all pixels with low saturation (less than 20) and high brightness value (more than 200) with `cv2.inRange()`. These pixels are then turned white.

As inline annotations such as *highlighter* or *strikethrough* overlap with printed text, they are subtracted from the processed IR image. To achieve this, regions with saturation over 100 and a brightness value between 10 and 245 are masked from a copy of the cleaned RGB image. Matching pixels are subtracted from the processed IR image to yield the final mask which is then added to the cleaned RGB image. The result contains only handwritten annotations on a plain white background. The white background is turned transparent before the image is saved as a PNG file (Fig. 3d).

3.1.2 Inserting Annotations into a PDF File. As an optional step after extraction, annotations can be inserted into the original PDF one page at a time as an image overlay. First, the desired PDF page is loaded as a *NumPy* array using the *pdf2image* Python library[10]. The brightness of this page's IR scan is normalized using the bias image and its black and white points are set to minimum and maximum brightness to increase contrast.

To align PDF page and IR scan, bounding boxes around the text are calculated. Noise is removed using gaussian blur with a 9×9 kernel followed by an opening operation with a 21×21 kernel. A binarization of both images with `cv2.adaptiveThreshold()` removes possible remaining brightness gradients. Then, contours around individual text elements are calculated with `cv2.findContours()` and the minimum and maximum x and y values of those contours are used as a bounding box. Annotations are transformed into the PDF page's coordinate system by calculating a homography between both bounding boxes and using `cv2.warpPerspective()` on the extracted annotations.

Finally, aligned annotations are inserted into the PDF with the pdf-annotate[11] library. A more detailed description of the algorithm can be found on our project website[12] and the source code is available under an open source license[13].

[5]https://www.wemos.cc/en/latest/d32/d32.html
[6]https://github.com/pyserial/pyserial
[7]https://python-sane.readthedocs.io/
[8]https://github.com/plangrid/pdf-annotate
[9]measured on a *HP EliteBook 850 G4* (*Intel i7* CPU with 2.7 GHz, *Intel HD Graphics 620*, 16 GB RAM)

[10]https://pypi.org/project/pdf2image/
[11]https://github.com/plangrid/pdf-annotate
[12]https://hci.ur.de/projects/infrared_scan
[13]https://github.com/PDA-UR/ir-annotation-extraction

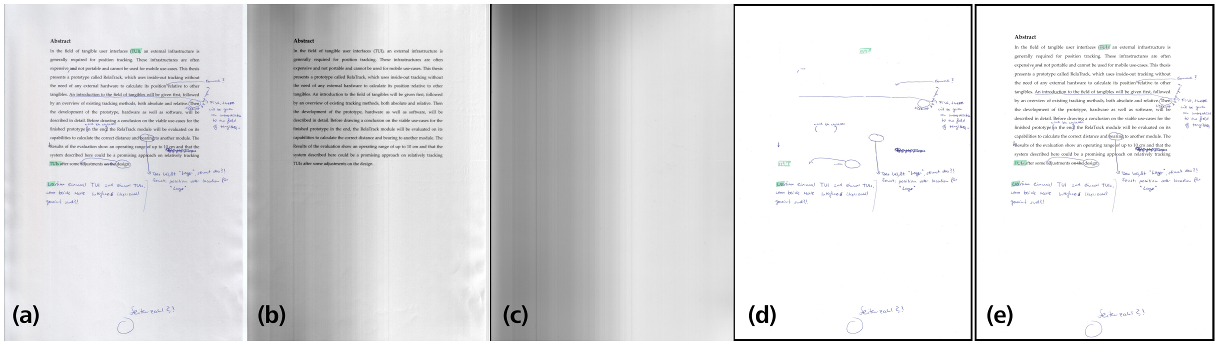

Figure 3: (a) RGB scan of a document. (b) IR scan of the same document. Annotations are invisible due to the absorption spectrum of the used pens. (c) IR scan of the empty scanner that is used to compensate for uneven illumination. (d) Result of the extraction process. (e) Extracted annotations inserted into the original PDF document.

4 EVALUATION

As different pens have different absorption spectra, [8–10], we compared scans of 42 different pens[14], as well as printed text in different colors, to determine how well they can be seen under infrared light (Fig. 4). Only black toner, graphite pencils and some black felt-tip pens were visible. Additionally, dark board markers (blue and green) can be recognized very faintly. Indents in the paper caused by ballpoint pens can be seen in the infrared scan. As non-black printer ink and toner are also invisible under infrared light, our method for extracting annotations does not work for color prints, as colored parts would be recognized as annotations.

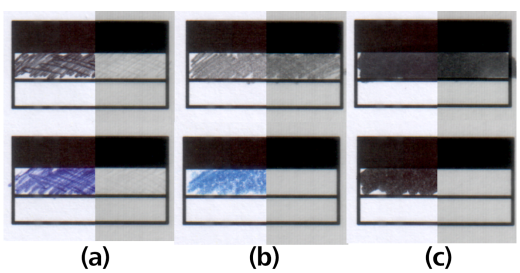

Figure 4: Comparison of different pens scanned under RGB (left) and IR (right) light. (a) Ballpoint pens vanish, but they leave visible traces on the paper. (b) Graphite pencils are visible under IR light, but color pencils vanish completely. (c) Some black felt-tip pens are visible, others are not.

4.1 Real-world Data Set

To evaluate our method for extracting annotations, we gathered a data set of annotated documents by letting ten participants (4 male, 6 female, aged 24–36) annotate ten pages of a real bachelor's thesis. Participants were asked to annotate the document as if they would proof-read a friend's thesis just before the deadline. As we knew our system does not work with color prints or pencil annotations, the document was printed in grayscale and participants could only use pens provided by us (blue and black ballpoint pen (generic brand); blue, black, red, and green sharpie (*edding 89 office liner*); yellow, pink, blue, and green highlighter (*Stabilo Boss*)).

All annotated pages were scanned with our modified scanner at 300 dpi (2539 × 3507 pixels) and annotations were counted and manually classified in terms of used pen, annotation type (categories: highlight, (border) note, underline, strikethrough, inline (note), arrow, other) and extraction quality (good/medium/bad – see Fig. 5 for examples). This classification was done by one person to counteract confounding effects due to different raters.

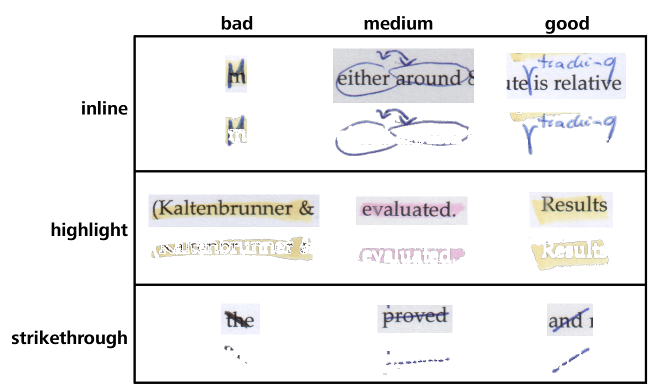

Figure 5: Extraction quality categories for different annotation types. Illegible annotations were classified as *bad*, legible annotations with some artifacts as *medium*, and (almost) perfectly extracted annotations as *good*. Top: RGB scan, bottom: extracted annotations.

4.2 Results

As seven pages without annotations were discarded, our data set consists of 93 annotated pages and contains a total of 1038 annotations. We considered extraction quality as *good* if there were no or only minor artifacts, as *medium* if there were some artifacts but the annotation was still recognizable, and as *bad* if significant parts of the annotation were missing or it was illegible. 83.6% of annotations were extracted with *good quality*, 13.8% with *medium quality* and 2.6% with *bad quality*. Extraction quality is impaired by annotations

[14] 14 sharpies, 18 felt-tip pens/highlighters, 6 pencils, 3 ballpoint pens, 2 other

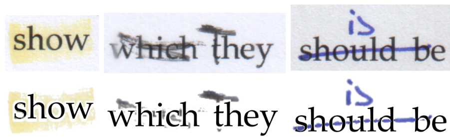

Figure 6: Left: Bright areas in highlighter annotations can be wrongly classified as background. Center/Right: Masked region around text can impair the quality of strikethrough annotations significantly while it works well in other cases. Top: RGB scan. Bottom: Annotations inserted into original PDF.

Table 1: Extraction quality in relation to annotation type. Annotations overlapping with text, such as highlight and strikethrough, were extracted significantly worse than others.

Type	total	good	medium	bad
highlight	106	59.43%	38.68%	1.89%
note	279	96.77%	3.23%	0.0%
underline	129	95.35%	4.65%	0.0%
strikethrough	126	26.19%	54.76%	19.05%
inline	248	93.55%	6.05%	0.4%
arrow	113	98.23%	1.77%	0.0%
other	37	97.3%	2.7%	0.0%

Table 2: Extraction quality in relation to pen type. Black ballpoint pen and green sharpie were not used. The black sharpie was only used by one annotator who did a lot of strikethrough annotations, thus the high *bad* percentage for this pen.

Pen	total	good	medium	bad
Ballpoint (blue)	655	87.63%	10.23%	2.14%
Sharpie (blue)	139	95.68%	4.32%	0.0%
Sharpie (black)	106	67.92%	21.7%	10.38%
Sharpie (red)	3	100.0%	0.0%	0.0%
Highlighter (yellow)	107	64.49%	33.64%	1.87%
Highlighter (pink)	9	66.67%	33.33%	0.0%
Highlighter (green)	13	76.92%	23.08%	0.0%
Highlighter (blue)	6	16.67%	83.33%	0.0%

overlapping with printed text, as the text mask (Section 3.1.1) is cut out of the annotation (Fig. 6). Therefore, the quality of *strikethrough* and *highlight* annotations was worse than for other annotation types (Table 1).

Inserting annotations into the original PDF has worked for all 93 pages. In some cases, there was slight misalignment in vertical direction with a maximum offset of 11 pixels, which is about half the height of a lowercase letter. Even though all annotations in our data set were still recognizable and assignable to their correct location, this could lead to a strikethrough annotation being misrepresented as underline in the worst case.

5 LIMITATIONS AND FUTURE WORK

Even though we have shown that infrared scans can be used to extract handwritten annotations from printed documents, our method still has some limitations. Due to their absorption spectra, our method does not work with either color prints or annotations with certain pens, such as graphite pencils.

Currently, extracted annotations are inserted into the original PDF as one image overlay per page. By implementing annotation segmentation and classification [5, 6, 14] into our system, highlight, underline, and strikethrough annotations could be automatically inserted as digital annotations into the PDF. This would also solve the problem of artifacts in inline annotations (Fig. 6) due to slight misalignment or masking borders around printed text.

A further possible application of our method which we did not discuss in this paper is document cleaning. For example, handwritten annotations could be removed from scans to recover the state of the original document [15]. It is also possible to remove certain types of stains from scanned documents, if the substance is invisible under infrared light.

Another extension of our basic method would be to augment paper documents with information printed using ink that is only visible in the infrared spectrum. This would allow for printing invisible markers on documents to augment them with machine-readable information, such as unique ID's or the position of form fields.

6 CONCLUSION

In this paper, we presented a method for using infrared light to extract handwritten annotations from printed documents. It requires a modified flatbed scanner which can be built using off-the-shelf components and basic knowledge of electronics. We extend existing research on annotation extraction by describing a robust method that produces high-quality results without relying on a training data set [4, 12, 21] or the original PDF document [1, 17, 18]. In addition to the use cases described so far, our method could also be used to generate accurate training data sets for machine learning models. While there are still practical limitations, such as problems with extracting annotations overlapping with printed text, most of them will be addressed in future iterations, for example by refining the pre-processing pipeline.

ACKNOWLEDGMENTS

We thank Thomas Fischer for providing his thesis, and the ten annotators for helping create the data set. This project is funded by the Bavarian State Ministry of Science and the Arts and coordinated by the Bavarian Research Institute for Digital Transformation (bidt).

REFERENCES

[1] David M. Bargeron. 2009. Lifting ink annotations from paper. https://patents.google.com/patent/US7526129/en
[2] Benjlaiel, Mohamed, Mullot, Rémy, and Alimi, Adel M. 2014. Multi-oriented Handwritten Annotations Extraction from Scanned Documents. https://doi.org/10.1109/DAS.2014.17
[3] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).
[4] Jindong Chen, Eric Saund, and Yizhou Wang. 2008. Image objects and multi-scale features for annotation detection. In *2008 19th International Conference on Pattern Recognition*. 1–5. https://doi.org/10.1109/ICPR.2008.4761932 ISSN: 1051-4651.

[5] J.K. Guo and M.Y. Ma. 2001. Separating handwritten material from machine printed text using hidden Markov models. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*. 439–443. https://doi.org/10.1109/ICDAR.2001.953828

[6] Mallikarjun Hangarge, K C Santosh, Srikanth Doddamani, and Rajmohan Pardeshi. 2012. Statistical Texture Features based Handwritten and Printed Text Classification in South Indian Documents. (2012). https://arxiv.org/pdf/1303.3087.pdf

[7] Seon Joo Kim, Fanbo Deng, and Michael S. Brown. 2011. Visual enhancement of old documents with hyperspectral imaging. *Pattern Recognition* 44, 7 (July 2011), 1461–1469. https://doi.org/10.1016/j.patcog.2010.12.019

[8] Ridamjeet Kaur, Komal Saini, and N.C. Sood. 2013. Application of Video Spectral Comparator (absorption spectra) for establishing the chronological order of intersecting printed strokes and writing pen strokes. *Science & Justice* 53, 2 (June 2013), 212–219. https://doi.org/10.1016/j.scijus.2012.10.001

[9] Zohaib Khan, Faisal Shafait, and Ajmal Mian. 2013. Hyperspectral Imaging for Ink Mismatch Detection. In *2013 12th International Conference on Document Analysis and Recognition*. 877–881. https://doi.org/10.1109/ICDAR.2013.179 ISSN: 2379-2140.

[10] Zohaib Khan, Faisal Shafait, and Ajmal Mian. 2015. Automatic ink mismatch detection for forensic document analysis. *Pattern Recognition* 48, 11 (Nov. 2015), 3615–3626. https://doi.org/10.1016/j.patcog.2015.04.008

[11] Zohaib Khan, Faisal Shafait, and Ajmal Mian. 2019. Converting a Common Low-Cost Document Scanner into a Multispectral Scanner. *Sensors* 19, 14 (Jan. 2019), 3199. https://doi.org/10.3390/s19143199

[12] Andreas Kölsch, Ashutosh Mishra, Saurabh Varshneya, Muhammad Zeshan Afzal, and Marcus Liwicki. 2018. Recognizing Challenging Handwritten Annotations with Fully Convolutional Networks. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 25–31. https://doi.org/10.1109/ICFHR-2018.2018.00014

[13] Muhammad Imran Malik, Sheraz Ahmed, Faisal Shafait, Ajmal Saeed Mian, Christian Nansen, Andreas Dengel, and Marcus Liwicki. 2015. Hyper-spectral Analysis for Automatic Signature Extraction. https://hal.univ-antilles.fr/hal-01165888

[14] Andrea Mazzei, Frederic Kaplan, and Pierre Dillenbourg. 2010. Extraction and Classification of Handwritten Annotations.

[15] P. Nagabhushan, Rachida Hannane, Abdessamad Elboushaki, and Mohammed Javed. 2015. Automatic Removal of Handwritten Annotations from Between-text-lines and Inside-text-line Regions of a Printed Text Document. *Procedia Computer Science* 45 (2015), 205–214. https://doi.org/10.1016/j.procs.2015.03.123

[16] Tomohiro Nakai, Kazumasa Iwata, and Koichi Kise. 2008. Accuracy Improvement and Objective Evaluation of Annotation Extraction from Printed Documents. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*. IEEE, Nara, Japan, 329–336. https://doi.org/10.1109/DAS.2008.80

[17] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. 2007. A Method of Annotation Extraction from Paper Documents Using Alignment Based on Local Arrangements of Feature Points. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Vol. 1. IEEE, Curitiba, Parana, Brazil, 23–27. https://doi.org/10.1109/ICDAR.2007.4378669

[18] Tomohiro Nakai, Nobuyuki Kondo, Koichi Kise, and Keinosuke Matsumoto. 2008. Analysis of annotations on documents for recycling of information. *Electrical Engineering in Japan* 165, 2 (2008), 60–68. https://doi.org/10.1002/eej.20516

[19] Abigail J. Sellen and Richard Harper. 2002. *The myth of the paperless office*. MIT Press, Cambridge, Mass.

[20] J. Stevens, A. Gee, and C. Dance. 1998. Automatic Processing of Document Annotations. In *Procedings of the British Machine Vision Conference 1998*. British Machine Vision Association, Southampton, 44.1–44.11. https://doi.org/10.5244/C.12.44

[21] Yefeng Zheng, Huiping Li, and David Doermann. 2002. The Segmentation and Identification of Handwriting in Noisy Document Images. *Document Analysis Systems V* 2423 (2002), 95–105. https://doi.org/10.1007/3-540-45869-7_12 Series Title: Lecture Notes in Computer Science.

[22] Guangyu Zhu, Yefeng Zheng, David Doermann, and Stefan Jaeger. 2009. Signature Detection and Matching for Document Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 11 (Nov. 2009), 2015–2031. https://doi.org/10.1109/TPAMI.2008.237