23^{rd} *Computer Vision Winter Workshop*
Zuzana Kúkelová and Júlia Škovierová (eds.)
Český Krumlov, Czech Republic, February 5–7, 2018

Physical Layout Analysis of Partly Annotated Newspaper Images

Thomas Lang, Markus Diem, Florian Kleber
Computer Vision Lab
TU Wien
Vienna, Austria
{tlang@cvl.tuwien.ac.at, diem@cvl.tuwien.ac.at, kleber@cvl.tuwien.ac.at}

Abstract. *Even though modern newspapers are born-digital, layout information is often not stored or not distributed along with the documents. Techniques for region segmentation and classification have been developed that use incomplete region information extracted from digital newspapers available in the PDF format. Extracted image regions which are too large or overlap each other can be used along with the document image to obtain region bounding boxes that fit the visible images in the document. Known and manually labeled document regions are used to train a random forest classifier using HOG features, in order to classify the final region bounding boxes, resulting in classification error rates of 0.05 for text/table regions and 0.1 for image/chart regions.*

1. Introduction

Physical (or *geometric*) layout analysis is defined by Kise [10] as the localization and classification of semantically homogeneous regions in a document image, in our case especially text, image, chart and table regions. It is therefore to be distinguished from *logical* layout analysis, which is concerned with the further classification of document regions into headers, footers etc. and out of the scope of this paper.

Modern newspapers are "born-digital", meaning that they are created in digital form. This allows newspaper creators to store additional meta-information alongside the digital document files. However, if this is not done or the added information is not distributed along with the newspapers, it is not available for readers and institutions that receive them. For this reason, it can be necessary to perform layout analysis on the documents, treating them as ordinary document images. This can be a difficult problem, since modern newspapers often have complex "non-manhattan" layouts, including overlapping document regions.

1.1. Related Work

For this problem, Baird *et al*. proposed versatile methods for pixel-wise (instead of region- or superpixel-wise) training and classification of documents using k-NN classifiers and k-d trees [3] [4]. They have been complemented by a post-classification step for merging neighboring pixels to uniform regions [1]. In 2009, Ray Smith published a method for detecting text and non-text regions [14] as part of the Tesseract OCR engine [1] that takes advantage of the fact that publishing systems use tab-stops for laying out documents.

In the context of the biannual ICDAR (International Conference on Document Analysis and Recognition) competitions on layout analysis, metrics for the evaluation of layout analysis algorithms [7] and a data format for ground-truth information called PAGE [13] have been devised by PRImA Research[2]. They have also developed a system for ground-truthing document images [6].

In addition to these general approaches, there are also methods for detecting specific regions in document images, which could be used in combination to produce a complete image segmentation and classification. In 2013, a competition on table recognition was held [9], which included the task of table location. The best-performing method in this sub-competition was submitted by Nurminen [11]. Chen *et al*. developed a technique for locating text areas based on the extraction of whitespace rectangles [5], which had the best results in the ICDAR competition on historical newspaper layout analysis in 2013 [2].

[1] https://github.com/tesseract-ocr/
[2] http://www.primaresearch.org/

1.2. Partly Annotated Newspaper Images

Having access to a dataset of digital newspapers stored as PDF files, we try to take advantage of the layout information that can be recovered from them, and propose techniques for segmentation and classification of document regions based on this information. Using specialized software, rectangular bounding boxes of text and image regions can be extracted from PDF files, which can then be used for further processing. After inspection of the extracted layout information, we found that the region rectangles correctly described the text regions, but the bounding boxes of the image regions were only partly correct (the region information is described in detail in section 2). In section 3.1, we propose a technique for the segmentation of these incomplete or incorrect image regions in order to produce bounding boxes that fit the content. To be able to distinguish between more than just two types, we trained a classifier for four region types: text, image, chart and table. The classification is described in section 3.2.

2. Extracted Document Regions

PDF files of born-digital newspapers contain all the necessary information for printing and rendering them in arbitrary sizes. Therefore, the locations and sizes of text and image objects can be extracted directly from the content of such files. For example, the locations of included raster images, which are stored as *XObject*s, can be inferred from modifications of the *current transformation matrix* (CTM) just before they are drawn on the screen by the *Do* operator [3]. Alternatively, specialized software can be used for this task. The open-source program Inkscape [4], for example, is able to load PDFs and convert them to SVG files, which are based on XML.

The authors had access to a digital repository of Austrian and German daily newspapers. Every page could be accessed as a PDF or JPEG image file and already included an additional XML file containing the bounding boxes of text and image regions, which were automatically extracted from the PDF files by the provider of the repository. Other content, like paths describing vector graphics, was not included in the XML files. These extracted bounding boxes will also be referred to as XML regions in the rest of this paper, in order to distinguish them from the actual regions in the documents.

For complex documents like newspapers, especially if they contain many embedded images, such extracted region descriptions are only partly correct. Text contained in the document is always described by a corresponding XML text region, except if the text is part of an image, meaning that it is not explicitly stored as text in the PDF. An example of both cases can be seen in Figure 1. Since tables contain text, they are also described as text regions. Image region descriptions, however, do not always match the actual visible images in the document, resulting in different problems:

- A single visible image may be described by several overlapping or neighboring regions.
- Image region boundaries can be larger than the actual images and include surrounding page content.
- When text surrounds the visible image area, there may be separate image regions for the text.
- Images that are parts of charts or diagrams may not be labeled as images at all, leaving only free-floating text regions.

Examples of these problems can be seen in Figure 2. Despite these flaws, the extracted regions can be used as a starting point for further layout analysis. The XML image regions at least roughly reveal the areas of images in the document and can be corrected by using the available information from the text regions in combination with the document image. Layout elements like separator lines can clearly be distinguished from other images and text, since they are represented by a different kind of object (*paths*) in PDF files and, in our case, not included in the XML descriptions.

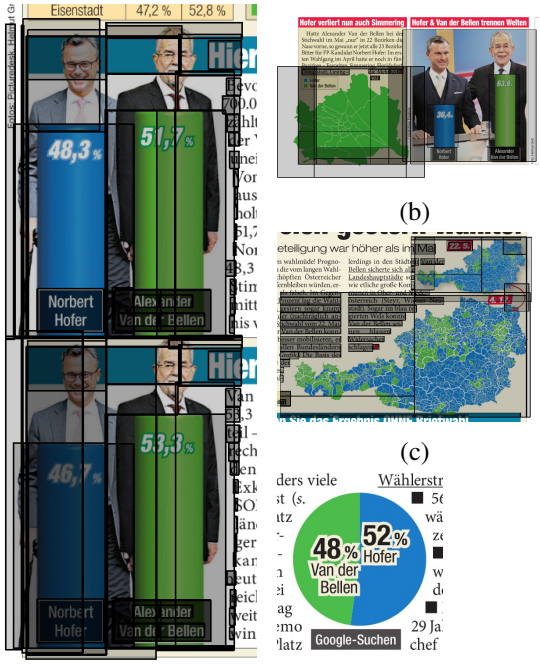
3. Methodology

The main task is to localize and identify all text, image, chart and table regions in the image. While it would be interesting to have exact region boundaries for non-manhattan and overlapping layouts like those of newspapers, it was decided that in order to reduce the complexity of the problem, this work only focuses on obtaining *rectangular* bounding boxes of each document component. This reduces the problem to two sub-tasks:

[3] see the PDF specification at `http://www.adobe.com/devnet/pdf/pdf_reference.html`

[4] `https://inkscape.org/`

Figure 1: Text regions described in the source XML, visualized by rectangles. The text in the image on the left is not recognized as such, because it is part of the image itself, which depicts an advertisement.



(a) (b) (c) (d)

Figure 2: Figure a shows an image with a chart that is roughly described by overlapping region bounding boxes. Figure b shows a case where the regions are bigger that the actual images and overlap into the neighboring depictions. Figure c shows an example of text described by image regions. Figure d shows a diagram that is not labeled by image regions at all.

1. The extracted rectangular image region boundaries have to be corrected. Based on the available region boxes, the document image must be segmented to find the actual boundaries of visible image objects in each page.

2. The final "text" and "image" regions have to be classified according to their actual type.

Afterwards, both tasks can be combined to a complete layout analysis system, taking the document images with incomplete region descriptions as input and producing text, image, chart and table regions.

3.1. Image Region Segmentation

The first step in the image region segmentation process is to take advantage of the known text regions and remove all text to obtain a document image which only contains non-text elements. The known partly-correct image regions can then be used to look for visible components that might depict an image or a chart. The input image with width W and height H is assumed to be in the RGB format using 8 bit for each channel c.

3.1.1 Text Removal

Text can occur in different colors and sizes in the image, therefore we locate and remove the text for every region individually. Based on visual inspection of the XML regions, we assume that all text regions in all newspaper pages always contain a single paragraph or heading of a single font size and color. This assumption can then be used to find text pixels based on the average foreground connected component (CC) properties in every region, considering components as non-text if they deviate from these properties. The results of the text removal step do not have to be perfect, because remaining letters and other imperfections can be easily removed or ignored afterwards. For this reason, the procedure was developed with the main focus on simplicity and runtime performance. Specific values for thresholds etc. have been determined experimentally. Where the number of bins for histograms is not specified, all 256 possible values are used as bins. The first step is to find a threshold T on the gray-scale image I_{gray} using the Otsu method [12]. The larger of the resulting complementary sets of pixel coordinates is considered the set of probable background pixels

$$M_{pbg} = \underset{M_i \in \{M_{low}, M_{high}\}}{\arg\max} |M_i| \,, \tag{1a}$$

$$M_{low} = \{(x, y) : I_{gray}(x, y) < T\} \,, \tag{1b}$$

$$M_{high} = \{(x, y) : I_{gray}(x, y) > T\} \,. \tag{1c}$$

Histograms are calculated to find the most frequent values v_R, v_G, v_B at the coordinates of M_{pbg} in each color channel I_R, I_G, I_B. The sets of foreground and background pixels are then computed based on the

deviation from these values:

$$M_{bg} = \{(x,y) : |v_c - I_c(x,y)| \leq 50 \quad \forall c \in \{R,G,B\}\} \tag{2}$$

$$M_{fg} = \{(x,y) \notin M_{bg}\}. \tag{3}$$

Connected component analysis is performed on the binary foreground image masked by M_{fg} and the areas of all connected components (CCs) are computed. Those that are larger than 4 times the upper quartile $Q_{0.75}$ are considered non-text components (if this threshold is chosen too low, merged letters, which have greater pixel areas, may be falsely removed). All other CCs are possible text components. The RGB image is then converted to the L*a*b* color space and k-means clustering with $k = 3$ is performed on the a* and b* channels to find the dominant text color $c_{dom} = (a*, b*)$, which is the cluster center with the most pixels assigned to it. Cluster centers c_i with $\|c_{dom} - c_i\| > 15$ are treated as deviating colors. All CCs that consist of more than 65% pixels assigned to clusters of deviating colors are considered non-text components. Afterwards, a binary image B_{text} is created, containing all pixels that are part of the remaining text components. This text mask is then dilated with a circle-shaped structuring element of diameter 3 to take smooth text edges into account, which may have been classified as background in previous steps. The last step is to replace all pixels in I_{RGB} masked with $B_{text,\,dilated}$ with the background color (v_R, v_G, v_B). An example of an image with removed text can be seen in Figure 3.

3.1.2 Image Region Merging and Separation

Because image regions are often clustered, meaning that they are overlapping or laid out like tiles (see Figure 2), the first step is to merge all regions belonging to the same region cluster. Regions are considered part of the same cluster if they have significant overlap or if they are adjacent to each other. Since the regions are rectangular, they are defined by four coordinates: $x_{left}, x_{right}, y_{top}, y_{bottom}$. All pairs of regions in the image are tested for three criteria:

1. The area of their intersection is greater than 5% of the area of the smaller of both regions.

2. The horizontal distance separating them is $\leq 1\,px$ and they have vertical overlap > 0.

3. The vertical distance separating them is $\leq 1\,px$ and they have horizontal overlap > 0.

If one of these criteria is fulfilled, they belong to the same region cluster. The result is a set of region sets, representing the region clusters, including possible clusters of only one isolated region.

The next step is to separate these regions where background-color segments run through them. Since the document foreground must be recognized by the reader, we can assume that it is clearly separated from the background in terms of brightness. Also, because of the spacing in documents, it is assumed that the total number of foreground pixels is always less than the number of background pixels. That is why the background value v_{bg} is determined as the most frequent value in the histogram of all pixel values of the gray-scale image I_{gray}. The foreground mask is created as

$$B_{fg}(x,y) = \begin{cases} 1 & \text{if } I_{gray}(x,y) \geq v_{bg} + T_{bg} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

if $v_{bg} < 127$, meaning that the background is dark and the foreground is bright, where T_{bg} is an additional tolerance. Otherwise it is created using the inverse operation with the threshold $v_{bg} - T_{bg}$. For T_{bg}, we choose a value of 10.

For every region cluster, a binary image mask $B^*_{cluster}$ of all pixels inside the regions contained in the cluster is created. To account for the tolerance of $1\,px$ in the region adjacency test, a binary closing with a quadratic structuring element of size 3 is performed afterwards, resulting in the region cluster mask $B_{cluster}$. The cluster foreground mask is then defined by

$$B_{cluster,fg} = B_{cluster} \wedge B_{fg}. \tag{5}$$

Connected component analysis is performed on $B_{cluster,fg}$ and all CCs which are contained in the bounding box of another CC are discarded (this can happen when CCs are non-convex). The bounding boxes of the remaining components are the boundaries of the new image regions. An example is depicted in Figure 4. It also shows a weakness of this method: because the background value v_{bg} is estimated for the whole image, text regions with deviating background colors may be regarded as images.

3.2. Region Classification

Since the XML dataset described in section 2 contains only text and image region rectangles, in order

(a) (b)

Figure 3: Figure 3a shows the text regions in the original image, 3b shows the result after the text removal step. Some components with larger size or different colors are not considered text by the algorithm and remain in the image. Text removal fails for the date in the upper-right corner because of the separator lines below it, which connect all the letters, making it a single connected component.

to distinguish between more than these two types, region classification is necessary. Histograms of oriented gradients (HOG) [8] are used as feature descriptors for classification. After testing and comparing different parameters, a window size of 64×64 pixels, cell size of 8×8 pixels and block size of 8×8 cells gave the best results. Using these parameters, each feature descriptor has the size 576.

For each of the labeled pages in the training set, an image pyramid is computed. The feature computation routine iterates over every region in every page and computes at least one feature per region. The only exception are regions that are smaller than the window size, which are discarded and not classified. For every region, the highest image pyramid level (where level 0 contains the original size) is selected where the downscaled region can still fit the HOG window. This step is made to approximately bring all regions to the same scale. The resulting two sides of the region in the down-sampled image are

$$64 \leq dim_1 < 128 \quad (6)$$

$$64 \leq dim_2, \quad (7)$$

where dim_1 and dim_2 are either width or height respectively. To use as much of the region as possible for feature computation, we let the HOG window slide over the region in each dimension with a step size of 32. This allows $\frac{dim_i - 64}{32}$ shifts in each dimension i, resulting in $(1 + \frac{dim_1 - 64}{32}) \times (1 + \frac{dim_2 - 64}{32})$ features for the whole region. After all features for all labeled regions in the page images contained in the dataset have been computed, they are used to train a random forest classifier.

For the classification, the feature descriptors are computed the same way for all known regions in the test set. However, the classification will return one label per feature descriptor, which can result in more than one label per region. For such cases, we use the random forest classifier to our advantage and add the votes of all the bagged trees to find the label with the most overall votes in the region.

3.3. Layout Analysis System

After the source XML layout description extracted from PDF files has been parsed, the layout analysis system consists of three main steps to obtain the final document regions:

1. Text regions described in the source XML are classified as *text* or *table* regions.
2. Image regions described in the source XML are segmented to obtain non-intersecting regions.
3. The resulting segmented regions are classified as *image* or *chart*.

By combining these steps, layout analysis on PDF files can be performed, resulting in layout descriptions containing text, image, chart and table regions.

(a) The original XML regions.

(b) The result of the image region segmentation.

Figure 4: After the partly incorrect image regions have been recalculated, every image in the result shown in b has a single region bounding box. Because of its background color, the whole box on the left side, including the text, was detected as an image/chart region.

4. Evaluation

Since the known regions only describe text and images, manual labeling was performed to add information about chart and table regions. For this purpose, 6211 pages of the publications “Heute”, “Kleine Zeitung”, “Kronen Zeitung”, “Kurier”, “Die Presse”, “Der Standard” and “Süddeutsche Zeitung” have been annotated. In order to be able to work with existing tools, the XML files have been converted to the PAGE format [13] for this task. The annotation resulted in 891 pages containing at least one table or chart. These pages represent the final dataset: 891 images with corresponding PAGE XML files, containing rectangular region descriptions of text, images, charts and tables, with the image regions being incomplete, redundant or false in some cases.

The classification was evaluated using 70% of the dataset for training and the remaining pages as test set (624 training images, 267 test images). In the training step, feature descriptors were computed for all original text and image regions from the source XML to be used as text and image features. The manually labeled chart and table regions were used to compute the respective chart and table feature descriptors. To avoid an unbalanced training set, the number of feature descriptors for each class was reduced to the minimum class size. The random forest predictor included in OpenCV 3.3 [5] was used with a tree depth limit of 25 and a maximum number of trees of 150. For the evaluation, we reduced the number of classified regions for each class to the minimum class size, in order to better be able to compare the results. The classification results, taking all classes into account (351 regions per class), were:

	Text	Image	Chart	Table	Re.
Text	320	23	3	5	0.91
Image	27	306	18	0	0.87
Chart	14	34	285	18	0.81
Table	29	10	6	306	0.87
Pr.	0.82	0.82	0.91	0.93	

The average recall and precision both are 0.87, the rate of wrongly classified regions is 0.13. However, in the final layout analysis system, only the distinction between text/table and image/chart regions is needed. When we train models for this purpose, the results for the text/table classification (1084 regions per class) are

	Text	Table	Re.
Text	1048	36	0.97
Table	63	1021	0.94
Pr.	0.94	0.97	

with an average recall of 0.95 and an average precision of 0.96. Out of 2168 regions, 99 were wrongly classified, resulting in an overall error rate of 0.05. The results for the image/chart classification (351 regions per class) were

	Image	Chart	Re.
Image	328	23	0.93
Chart	50	301	0.96
Pr.	0.87	0.93	

[5] http://opencv.org/

with an average recall and precision of 0.9. Out of 702 regions, 73 were wrongly classified, resulting in an overall error rate of 0.1.

5. Conclusion

We proposed techniques for using incomplete layout information extracted from digital PDF documents to obtain bounding boxes labeled as text, image, chart or table regions. We used HOG features to train a random forest classifier to distinguish between text and table regions and between image and chart regions. We also proposed a simple segmentation procedure to merge overlapping and adjacent image regions and to separate them according to background areas in the image.

At this point, we were not yet able to quantitatively evaluate these segmentation results, which could be done by manually labeling all image regions in a number of images and calculating the overlap of the labeled regions with the ones resulting from our segmentation procedure. The oversized region in Figure 4b shows that there is room for improvement of the segmentation. Calculating the background color locally may help to avoid this problem. In the current implementation, the classifier is trained using the image regions from the PDF, which are partly incorrect and often overlapping, as we have seen. The classification could perhaps be improved by first segmenting the image regions before using them for training. This would also make the classification dependent on the performance of the segmentation algorithm, adding another reason why it should be additionally evaluated.

We saw that the random forest classification method with HOG feature descriptors achieved low error rates, especially for the classification between text/table and image/chart region types. This leads us to conclude that this method is a viable option for the classification of document regions. Although the segmentation technique is yet to be evaluated, visual inspection of the results showed that it significantly reduces the number of image regions by merging overlapping and adjacent ones.

References

[1] C. An, H. Baird, and P. Xiu. Iterated document content classification. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 252–256. IEEE, 2007. 1

[2] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. Icdar 2013 competition on historical newspaper layout analysis (hnla 2013). In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1454–1458. IEEE, 2013. 1

[3] H. S. Baird and M. R. Casey. Towards versatile document analysis systems. In *International Workshop on Document Analysis Systems*, pages 280–290. Springer, 2006. 1

[4] H. S. Baird, M. A. Moll, J. Nonnemaker, M. R. Casey, and D. L. Delorenzo. Versatile document image content extraction. In *Electronic Imaging 2006*, pages 60670R–60670R. International Society for Optics and Photonics, 2006. 1

[5] K. Chen, F. Yin, and C.-L. Liu. Hybrid page segmentation with efficient whitespace rectangles extraction and grouping. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 958–962. IEEE, 2013. 1

[6] C. Clausner, S. Pletschacher, and A. Antonacopoulos. Aletheia-an advanced document layout and text ground-truthing system for production environments. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 48–52. IEEE, 2011. 1

[7] C. Clausner, S. Pletschacher, and A. Antonacopoulos. Scenario driven in-depth performance evaluation of document layout analysis methods. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1404–1408. IEEE, 2011. 1

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 5

[9] M. Göbel, T. Hassan, E. Oro, and G. Orsi. Icdar 2013 table competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1449–1453. IEEE, 2013. 1

[10] K. Kise. Page segmentation techniques in document analysis. In *Handbook of Document Image Processing and Recognition*, pages 135–175. Springer, 2014. 1

[11] A. Nurminen. Algorithmic extraction of data in tables in pdf documents. Master's thesis, Tampere University of Technology, 2013. 1

[12] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. 3

[13] S. Pletschacher and A. Antonacopoulos. The page (page analysis and ground-truth elements) format framework. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 257–260. IEEE, 2010. 1, 6

[14] R. W. Smith. Hybrid page layout analysis via tab-stop detection. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 241–245. IEEE, 2009. 1