A Modular Region and Text Line Layout Analysis System

Benjamin Kiessling

A Modular Region and Text Line Layout Analysis System

Benjamin Kiessling
Digital Humanities
École Pratique des Hautes Études, Université PSL
Paris, France
benjamin.kiessling@psl.eu

***Abstract*—High quality document layout analysis is fundamental to the accurate processing of handwritten textual material, on both the level of individual lines and higher order zones demarking textual and non-textual content. We present an artificial neural network based approach to prediction of either that is implemented as part of a libre optical character recognition package and highly reconfigurable for a variety of tasks. Experiments on different openly available datasets show competitive results to state-of-the-art methods.**

***Index Terms*—layout analysis, region detection, historical document analysis, artificial neural networks**

I. Introduction

Over the last decades tremendous amounts of historical handwritten documents have been digitized by archives, libraries, and other institutions engaging in the preservation of cultural heritage. Nevertheless the vast volume of scanned images, often with lack of metadata, results in the majority of this material being inaccessible in any meaningful way to scholars and the wider public. Optical Character Recognition[1] and Keyword Spotting aim to be technical solutions to the exploitation of large amounts of scanned textual data.

Current OCR systems operate largely on *line*-level data, i.e. the module in the OCR pipeline performing conversion into text does so one line image at a time. Therefore, a prior method is needed to extract these line images from whole document images. In addition, many documents require higher level understanding of how lines relate to each other for meaningful interaction. The usual way these higher level relations are modelled is through zoning, i.e. splitting a page into regions such as main text, marginalia, headings, illustrations, etc. Importantly, the nature of those regions can vary considerably between applications and material; they can often overlap, lines might extend across them or not be in any region at all. Consequently, text line extraction and region detection are arguably the most important part of an OCR system apart from the actual text recognizer.

As such, robust and accurate historical and handwritten document image analysis remains an open issue despite the recent advances facilitated by deep learning methods. Highly curved lines, variable orientation, interlinear notes, and multiple texts on the same page remain challenging to even state of the art layout analysis systems. Further, cultural bias in the conception of methods and data models continues to be a persistent problem: [1] shows the large amount of adaptation necessary to apply a seemingly script-neutral line model to Arabic manuscripts.

For our purposes we consider layout analysis along two principal axes. The *geometric axis* deals with the location, shape, and relations of found entitites, e.g. the by now obsolete character segmentation, text line extraction, and region detection. Text line extraction refers to the locating of individual text lines in the document images. In most modern LA systems text lines are the smallest unit of output, albeit for specialized tasks like scene text recognition subdivisions into words is also widespread. Region detection aims to find higher level, almost exclusively structural, zones, both textual and not, in document images.

The *semantic axis* concerns itself with the functional nature of detected entities, such as titles, illustrations, apparatus criticus, ... While not strictly necessary for most applications and often neglected outside of tools tailored for specific input data, enriching with semantic information can both boost raw metrics through allowing better incorporation of domain knowledge and aid in human understanding by improving output structuring, such as suppressing certain ancillary textual components.

Of note is that the focus of most methods is limited to a single or a subset of the tasks and axes. For example, no method could be found that allows semantic classification of both region and text line detection output simultaneously. In contrast, our method admits geometric and semantic classification on both text lines and regions while not requiring either.

Our method is implemented as part of a free OCR engine[2] which exposes the full customizability of the method's layout analysis features to end users. Hence, we are referring to the system as modular; it is possible to perform a wide range of tasks, ranging from simple text line extraction to highly specialized analysis like writing surface defect detection in a unified software package.

A. Related work

As a well established task in computer vision research a number of comprehensive surveys of document layout analysis

[1] As methods have converged considerably we do not distinguish between recognition of printed (OCR) and handwritten text (HTR)

[2] `http://kraken.re`

exist [2]–[4].

B. Text line extraction

The capabilities of text line extraction methods in the literature is to some extent driven by existing datasets. A variety of formulations for line extraction can be found in published datasets. These range from polygons [5]–[7], to subword bounding boxes [8], down to explicit pixel labeling [9]. Some others such as [10], [11] also include extensive metadata like reading order, text order, or full transcriptions. A recent model [12] reduces text line detection to the extraction of baselines, i.e. imaginary polylines upon which the text rests or hangs from. These polylines in combination with a bounding polygon can be ingested by line-based text recognizers with minimal adaptation while at the same time requiring only modest effort for manual annotation, encouraging the creation of substantial training datasets for machine learning based methods.

The methods employed for text line extraction are just as varied as the the data models employed. [13], [15] use connected components combined with filtering to perform pixel labeling. A common paradigm utilizes projection profiles in one way or another such as [16] for bounding box extraction, [17], [18] in combination with seamcarving for polygonal output, or RNN-based artificial projection profile generation for in-paragraph line splitting in [19]. A common drawback of the previously mentioned methods is that they operate on binarized input images which can be difficult to obtain for degraded historical material. [20]–[22] bypass this requirement through clustering of superpixels that can be obtained directly from color or grayscale image data to calculate polygons and baselines respectively. A number of deep learning based schemes have been proposed as well: [23]–[25] apply variants of the U-Net architecture for semantic segmentation.

C. Region detection

Region detection is almost always performed across both the geometric and semantic axis although they vary in the variety of zone labels they can yield. The most basic methods such as [26], [27] only distinguish between text and non-text regions while [28] can in principle be extended to all textual regions determinable solely by layout relations, and [24], [25], [29], [30] are able to distinguish arbitrary, non-overlapping regions with appropriate training data.

Like for text line extraction [24], [25], [29] variants of convolutional encoder-decoder networks are popular albeit pixel classifiers on handcrafted features [26], [27] exist. [28] performs clustering of text lines with convolutional conjugate graph networks. Definite clause grammars on a feature vocabulary as part of a user-driven interactive segmentation system are shown in [31].

II. Method

This section describes the proposed method for joint text line and region layout analysis. Our method can be divided into three main stages: multi-label pixel classification, baseline extraction and polygonization, and region extraction.

The first stage comprises of an Artifical Neural Network which outputs the probability of one or more classes (baselines, regions, and auxiliary classes) being present for each pixel of the input image. The second stage consists of the postprocessing extracting baselines from the auxiliary and baseline classes heatmaps, followed by a seam-carving step incorporating the original image to compute the bounding polygons required for inclusion of our method in a fully functional OCR pipeline. The final step extracts the regions from their respective class heatmaps through a contour finding algorithm. Notably, baselines are not restricted to regions, i.e. they can occur outside of regions and cross region boundaries.

A. R-BLLA - Architecture

The overall pixel labeling network neural network is described in Fig 1. Instead of conventional semantic segmentation encoder-decoder networks whose output is at the same scale as the input, our architecture decodes the learned representations at the downsampled scale of the last layer as the spatial information of regions and baselines can be recovered with sufficient accuracy at this reduced resolution. This architecture roughly halves the memory requirements in comparison to an equivalent U-Net with a Resnet-50 backbone.

Our network is composed of a convolutional feature extractor, utilizing atrous convolutions (3×3 kernel size with 2×2 dilation, ReLU activation) to increase receptive field without increasing filter size or a more memory intensive deeper decoding network. This convolutional stack is followed by consecutive unidimensional LSTM layers as proposed for the ReNet architecture [32]. In this configuration the feature maps from the previous layers are swept by a bidirectional 1D LSTM layer in one direction (vertical or horizontal), followed by a second sweep over the output by a LSTM layer in the other direction, attaining similar performance to more complex multidimensional RNNs. The final decoding layer is a 1×1 convolution with $|\tau|$ filters and a sigmoid activation function that results in per class probability maps. Regularization is performed with group normalization ($G = 32$) [33] after each convolutional layer.

The output of the network is a stack of probability maps $\hat{y} \in \mathbb{R}^{w/n \times h/n \times |\tau|}$ for an input image $I \in \mathbb{R}^{w \times h \times c}$ with height h, width w, c channels, a downsampling factor n, and $|\tau|$ different classes $\{\text{start\_sep}, \text{end\_sep}, bl_0, \ldots, bl_k, reg_0, \ldots, reg_l\}$ for k and l different baseline and region types. The special classes start_sep and end_sep are placed at the beginning and end of each baseline respectively and serve two purposes. First, by explicitly encoding line bounds at locations where lines can be minimally separated such as multi-column texts we avoid inadvertent baseline merging during postprocessing. Second, introducing separate indicator classes for the beginning and end of a line allows the system to determine the orientation of lines. These auxiliary classes are shared across all possible baseline classes $\{bl_0, \ldots, bl_k\}$. As our method is intended to work with most scripts, including multi-script documents,

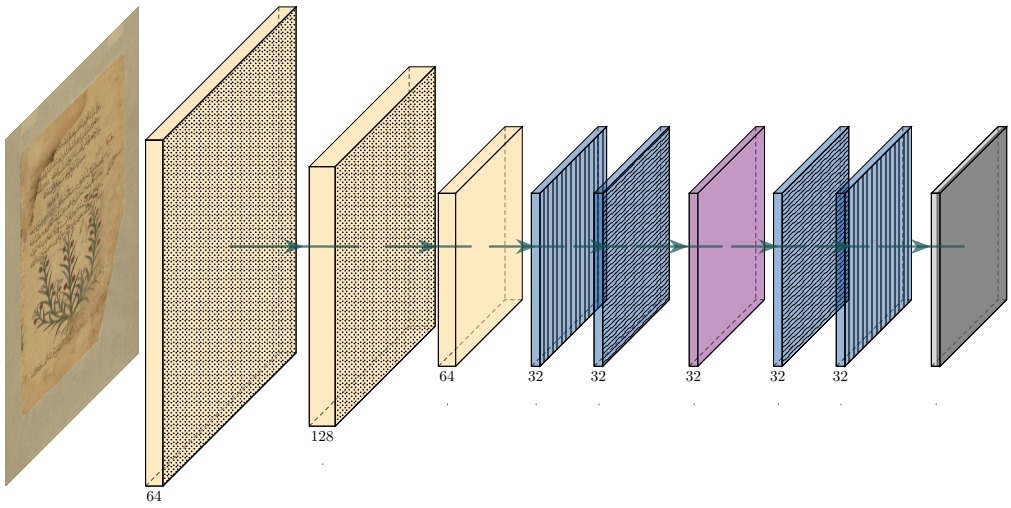


Fig. 1: Architecture of the pixel labelling network. Group normalization layers are omitted. (salmon: 3x3 convolutional layers, dotting indicates dilation by 2x2; purple: 1x1 convolution, blue: bidirectional LSTM blocks, striping indicates row/column time axis; grey: 1x1 convolution with $|\tau|$ filters + sigmoid)

the beginning and end of each line is not determined by the reading direction of the script. Instead we treat all scripts as canonically left-to-right, i.e when following the baseline from the start marker the upper part of the text line is always on the left-hand side (Fig. 2a. This scheme makes processing of arbitrarily oriented lines and mixed script pages with different reading direction without additional domain knowledge possible. The generated ground truth for the baselines classes are simple polylines drawn with a width. Regions are encoded as filled polygons. Thus the ground truth y is a multi-hot encoded tensor.

B. Training

The network is trained in a supervised manner with binary cross-entropy loss $L(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(y \cdot \log(\hat{y}_i) + (1-y) \cdot \log(1-\hat{y}_i))$. We adopt the Adam optimizer with moderate weight decay ($\alpha = 20^{-5}, \beta_1 = 0.9, \beta_2 = 0.999, w = 10^{-6}$). Input data are whole RGB color images scaled to a height of 1200 pixels.

In line with conventional practice, data augmentation is applied to the training set. With a probability of 0.5 a set of randomly parametrized transformations such as rotation, flipping, blurring, shifting, elastic transformations and hue changes are applied to each image [34].

We train for a fixed number of epochs, per default 50.

C. Baseline vectorization

Baseline vectorization refers to the extraction of baselines from the probability map output of the model. This task consists of multiple substeps: superpixel calculation, triangulation filtering, and interpolation.

As the process is identical for each baseline type we define the output of the neural network for an arbitrary baseline type $H = \hat{y}_{:,:,n}, n \in \{bl_0, \ldots, bl_k\}$. $P = \hat{y}_{:,:,\text{start\_sep}}, Q = \hat{y}_{:,:,\text{end\_sep}}$. Further we define a combined separator map $C = P + Q$.

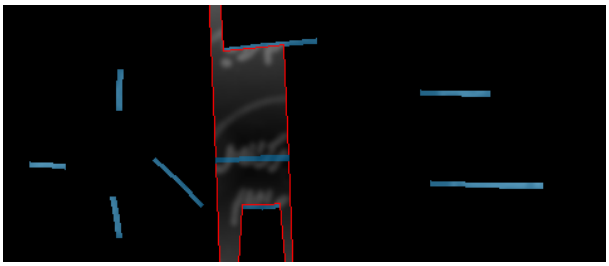
In a first step we reduce the number of pixels to be considered for baseline clustering through calculating a subset T of all image pixels. Elements of this subset are called superpixels (SPs). Determining SPs is largely identical to the algorithm proposed in [35]. For an arbitrary probability map H the map is binarized with a threshold 0.2 producing H_b and skeletonized with a medial axis transformation that also returns a distance transform of H_b, resulting in the skeleton H_s and the average diameter d_{cc} of each uneroded baseline. All foreground pixels in H_s are projected onto H and sorted in descending order by their probability (S). T is iteratively filled by removing elements from T as long as their distance exceed a minimum ($d_{min} = 10$) from all other pixels in S.

Algorithm 1 Triangulation filter

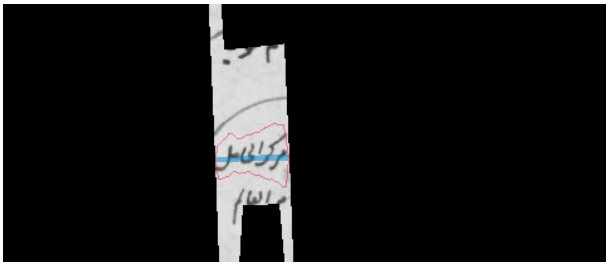
Input: $DT(T), H, C$
$E = \emptyset$
for $e_{p,q} \in DT(T)$ **do**
  if $\mu(H, e_{p,q}) \geq 0.4 \wedge \sigma^2(H, e_{p,q}) \leq 0.05$ **then**
    if $\mu(C, e_{p,q}) \leq 0.125 \wedge \max(C, e_{p,q}) \leq 0.25$ **then**
      $E \leftarrow e_{p,q}$
    end if
  end if
end for
return E

The following step of the vectorization algorithm filters the Delaunay triangulation $DT(T)$ of T to subdivide it into a set of baseline clusters. An edge between two SPs $p, q \in DT(T)$

(a) Ground truth overlay for a page with different line orientations (blue: baseline class, red: start_sep auxiliary class, green: end_sep auxiliary class)



(b) Region of Interests of the distance biased energy map for a sample line of the same image. (red: border of RoIs, blue: baselines)



(c) Computed seam/bounding polygon on the masked input image crop. (red: combined upper and lower half-seams, blue: baselines)

Fig. 2: Examples of the data model and intermediate representations for a page from the BADAM [1] dataset

is denoted by $e_{p,q}$. As a prerequisite of the filtering algorithm we also define a number of edge metrics. Given the discrete line coordinates produced by a line drawing method $l(e_{p,q})$ between the SPs p, q we define for an arbitrary map $I \in \{H, P, Q\}$:

$$\begin{aligned} \mu(I, e_{p,q}) &= \tfrac{1}{|p-q|_2} \sum I[l(e_{p,q})] \\ \sigma^2(I, e_{p,q}) &= \tfrac{1}{|p-q|_2} \sum (I[l(e_{p,q})] - \mu(I, e_{p,q}))^2 \\ \max(I, e_{p,q}) &= \max(I[l(e_{p,q})]) \end{aligned}$$

The output of the filtering algorithm Alg. 1 is a set of edges E defining a euclidean distance-weighted graph graph $G_E = G(V, E, w)$ where $V = \bigcup\{p, q\}, \forall e_{p,q} \in E, w(e_{p,q}) = |p - q|_2$, with a set of components C^{G_E}. Each component $C_n^{G_E}$ is treated as a separate baseline cluster. The remaining task is to create a directed polyline representation of each cluster. For each cluster we calculate the pairwise distances of all vertices and select the two most distant nodes a, b as the extrema of the baseline. The polyline approximation of each cluster is the shortest path $\gamma_{a,b}$ between the extrema in $C_n^{G_E}$. A slight correction of the line coordinates is necessary to compensate for the erosion incurred through the skeletonization prior to superpixel selection. The adjusted polyline path $\gamma_{a',b'}$ of $\gamma_{a,b}$ is obtained by elongating the initial and last edges by d_{cc}.

Due to the unknown orientation of each line we inspect each line end's affinity to the difference between the separator classes. As the separators are placed beyond the end of the line, the values of the separator maps at those points are commonly close to 0. By preprocessing P, Q with a maximum filter of size $2 \cdot d_{min}$ resulting in maps P', Q' containing sufficiently dilated separators the correct line orientation is such that:

$$L(\gamma_{a',b'}, P', Q') = \begin{cases} \gamma_{a',b'} & \text{if} (P' - Q')_p > 0.2 \wedge \\ & (Q' - P')_q > 0.2 \\ rev(\gamma_{a',b'}) & \text{if} (P' - Q')_p > 0.2 \wedge \\ & (Q' - P')_q > 0.2 \end{cases}$$

otherwise

$$L(\gamma_{a',b'}, P', Q') = \begin{cases} \gamma_{a',b'} & a'_x \leq b'_x \\ rev(\gamma_{a',b'}) & a'_x > b'_x \end{cases}$$

The final baselines for each baseline class is the set of all paths $\Gamma_m = \{\gamma_1, \ldots, \gamma_o\}, m \in \{bl_0, \ldots, bl_k\}, o \in \mathbb{N}$ determined as above.

D. Polygonization

For recognition by an HTR engine the vectorized baselines have to be supplemented by full polygons. A baseline with polygon can then rectified to produce a normalized line image with suppression of non-line content by projection onto a straight baseline through a piecewise affine transformation, allowing recognition of even highly curved lines by text recognition models.

Our polygonization algorithm consists of a line-wise seam carving [36] biased by distance from the baseline. The initial energy map is the derivative of the smoothed grayscale input image I^σ:

$$E(I^\sigma) = \left| \frac{\partial(I^\sigma)}{\partial x} + \frac{\partial(I^\sigma)}{\partial(y)} \right|$$

The primary purpose of the smoothing is to prevent the seam from crossing below disconnected line components such as diacritics and tonal marks. A gaussian filter with $\sigma = 2.5$ is sufficient for this purpose. Our implementation estimates the gradient with the Sobel operator.

Let $\{\Gamma_0, \ldots, \Gamma_n, \ldots, \Gamma_m\}, 0 < n < m, m \in \{bl_0, \ldots, bl_k\}$ be the baselines of all classes and $\gamma \in \Gamma_n$ be an arbitrary baseline. To calculate the bounding polygon we first extract two regions of interest (RoI) $E(I^\sigma)[r_{\text{left}}]$ and $E(I^\sigma)[r_{\text{right}}]$ around γ; these RoIs contain the energy map area between the left- and righthand side of γ and $\{\Gamma_0, \ldots, \Gamma_n \setminus \gamma, \ldots, \Gamma_m\}$ as

shown in Fig. 2b. The seams through r_{left} and r_{right} will form the respective halves of the bounding polygon.

Depending on the layout of the document the RoIs can vary considerably in size. Especially for baselines bordered only by the energy map boundaries, a distance bias has to be added to the energy map to ensure sufficiently tight boundary polygons. The biased energy map $E'(I^\sigma)[r_l], l \in \{\text{left}, \text{right}\}$ is computed through a euclidean distance transform D from the baseline with a scaling factor: $E'(I^\sigma)[r_l] = E(I^\sigma)[r_l] + D \cdot \overline{E(I^\sigma)[r_l]} \cdot 0.01$.

Requiring a rectangular area and principal direction for seam calculation, the RoIs need to be rotated. We rotate each RoI patch by the magnitude-weighted average direction. The energy-minimizing seam for each patch is then calculated using dynamic programming as described in [36]. Afterwards, the seams are rotated back into the original image coordinate system and concatenated to form the final bounding polygon for a line. Fig. 2c shows the result for a single line.

TABLE I: Baseline recognition metrics on cBAD 2019, BADAM, OHG, and Bozen

	P-val	R-val	F-val
cBAD			
Planet	0.937	0.926	0.931
DMRZ	0.925	0.905	0.915
UPVLC	0.911	0.902	0.907
TJNU	0.852	0.885	0.868
DMRZ-2017	0.773	0.743	0.758
proposed[3]	0.867	0.945	0.904
BADAM			
[1]	0.941	0.901	0.924
proposed	0.932	0.957	0.944
OHG			
[24]	0.962	0.971	0.966
[24][4]	0.984	0.977	0.980
proposed	0.978	0.973	0.975
proposed[4]	0.909	0.919	0.914
Bozen			
[24]	0.958	0.991	0.974
[24][4]	0.945	0.989	0.966
proposed	0.972	0.982	0.977
proposed[4]	0.936	0.949	0.942

E. Region extraction

Regions are extracted from the network output for each region type separately by thresholding at 0.5 and then extracting the contours around high-valued regions using the marching squares algorithm [37].

III. Evaluation

We evaluate the performance of the proposed method on 4 publicly available datasets: cBAD 2019 [38], Bozen [39], OHG [40], and BADAM [1]. Bozen and OHG are Latin script datasets with both region and baseline annotations, cBAD consists largely of Latin script annotated on the baseline line, while BADAM is an exclusively Arabic script baseline dataset.

For datasets providing both region and line data models we evaluate models trained solely on baselines and combined baseline and region detection models.

[3]Calculated on random 200 pages of the test set.

[4]Combined region and baseline model

TABLE II: Metrics for the region detection task of the OHG and Bozen datasets

	Mean acc	Mean IU	fw IU
OHG			
[24]	0.789	0.727	0.872
proposed	0.988	0.486	0.912
Bozen			
[24]	0.933	0.827	0.913
proposed	0.988	0.81	0.915

A. Metrics

Baseline measurements for precision, recall, and F1-score are calculated as defined in [41] with the default tolerance parameters. For region segmentation the standard metrics mean accuracy, mean intersection-over-union, and frequency-weighted intersection over union are reported:

mean accuracy	$(1/n_{cl}) \sum_i n_{ii}/t_i$
mean IU	$(1/n_{cl}) \sum_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$
frequency weighted IU	$\sum_k t_k)^{-1} \sum_i t_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$

where n_{ij} is the number of pixels of class i predicted to belong to class j, where there are n_{cl} different region classes and $t_i = \sum_j n_{ij}$ is the total number of pixels of class i [42].

Two aspects of the proposed method are not evaluated as there are no available datasets or widely accepted metrics: the orientation of the baseline (orientation is disregarded by [41] and only one dataset contains rotated lines) and the polygonization. According to [43] the size of the environment extracted around the baseline is not crucial to recognition accuracy as long as the line contents are contained in the rectified line image.

Results are reported in table I and II for baseline and region detection respectively.

IV. Conclusion

In this work we presented a flexible machine learning based method for text line and region layout analysis for historical documents including procedures for postprocessing which enable its use in a typical OCR workflow without further adaptation. The experimental results show its competitiveness with the current state of the art on a number of historical document layout analysis benchmarks.

References

[1] B. Kiessling, D. S. B. Ezra, and M. T. Miller, "Badam: A public dataset for baseline detection in arabic-script manuscripts," in *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, 2019, pp. 13–18.

[2] G. M. Binmakhashen and S. A. Mahmoud, "Document layout analysis: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–36, 2019.

[3] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, "A comprehensive survey of mostly textual document segmentation algorithms since 2008," *Pattern Recognition*, vol. 64, pp. 1–14, Apr. 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01388088

[4] G. Nagy, "Twenty years of document image analysis in pami," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, pp. 38–62, 2000.

[5] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, "Transcription alignment of latin manuscripts using hidden markov models," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*. ACM, 2011, pp. 29–36.

[6] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold, "Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 471–476.

[7] C. Clausner, A. Antonacopoulos, N. Mcgregor, and D. Wilson-Nunn, "Icfhr 2018 competition on recognition of historical arabic scientific manuscripts–rasm2018," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 471–476.

[8] M. Kassis, A. Abdalhaleem, A. Droby, R. Alaasam, and J. El-Sana, "Vml-hd: The historical arabic documents dataset for recognition systems," in *Arabic Script Analysis and Recognition (ASAR), 2017 1st International Workshop on*. IEEE, 2017, pp. 11–14.

[9] B. Gatos, N. Stamatopoulos, and G. Louloudis, "Ichfr 2010 handwriting segmentation contest," in *2010 11th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2010, pp. 737–742.

[10] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "Icdar2015 competition on recognition of documents with complex layouts-rdcl2015," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 1151–1155.

[11] ——, "Historical document layout analysis competition," in *Document Analysis and Recognition (ICDAR), 2011 11th International Conference on*. IEEE, 2011, pp. 1516–1520.

[12] M. Diem, F. Kleber, S. Fiel, T. Grüning, and B. Gatos, "cbad: Icdar2017 competition on baseline detection," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 1355–1360.

[13] N. Ouwayed and A. Belaïd, "A general approach for multi-oriented text line extraction of handwritten documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 4, pp. 297–314, 2012.

[14] N. V. Borse and I. R. Shaikh, "Language independent text-line extraction algorithm for handwritten documents," *International Journal*, vol. 4, no. 11, 2014.

[15] M. Diem, F. Kleber, and R. Sablatnig, "Text line detection for heterogeneous documents," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 743–747.

[16] N. Arvanitopoulos and S. Süsstrunk, "Seam carving for text line extraction on color and grayscale historical manuscripts," in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 726–731.

[17] R. Saabni, A. Asi, and J. El-Sana, "Text line extraction for historical document images," *Pattern Recognition Letters*, vol. 35, pp. 23–33, 2014.

[18] B. Moysset, C. Kermorvant, C. Wolf, and J. Louradour, "Paragraph text segmentation into lines with recurrent neural networks," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 456–460.

[19] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke, "Binarization-free text line segmentation for historical documents based on interest point clustering," in *2012 10th IAPR International Workshop on Document Analysis Systems*. IEEE, 2012, pp. 95–99.

[20] B. Ahn, J. Ryu, H. I. Koo, and N. I. Cho, "Textline detection in degraded historical document images," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 82, 2017.

[21] T. Gruuening, G. Leifert, T. Strauss, and R. Labahn, "A robust and binarization-free approach for text line detection in historical documents," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 236–241.

[22] O. Mechi, M. Mehri, R. Ingold, and N. E. B. Amara, "Text line segmentation in historical document images using an adaptive u-net architecture," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 369–374.

[23] L. Quirós, "Multi-task handwritten document layout analysis," *arXiv preprint arXiv:1806.08852*, 2018.

[24] S. A. Oliveira, B. Seguin, and F. Kaplan, "dhsegment: A generic deep-learning approach for document segmentation," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 7–12.

[25] M. Baechler and R. Ingold, "Multi resolution layout analysis of medieval manuscripts using dynamic mlp," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1185–1189.

[26] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 488–493.

[27] H. Déjean, J.-L. Meunier *et al.*, "Versatile layout understanding via conjugate graph," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 287–294.

[28] Y. Soullard, P. Tranouez, C. Chatelain, S. Nicolas, and T. Paquet, "Multi-scale gated fully convolutional densenets for semantic labeling of historical newspaper images," *Pattern Recognition Letters*, vol. 131, pp. 435–441, 2020.

[29] P. Kaddas and B. Gatos, "A deep convolutional encoder-decoder network for page segmentation of historical handwritten documents into text zones," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 259–264.

[30] A. Lemaitre, J. Camillerapp, and B. Coüasnon, "Multiresolution cooperation makes easier document structure recognition," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 11, no. 2, pp. 97–109, 2008.

[31] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, "Renet: A recurrent neural network based alternative to convolutional networks," *arXiv preprint arXiv:1505.00393*, 2015.

[32] Y. Wu and K. He, "Group normalization," *CoRR*, vol. abs/1803.08494, 2018. [Online]. Available: http://arxiv.org/abs/1803.08494

[33] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125

[34] T. Grüning, G. Leifert, T. Strauß, J. Michael, and R. Labahn, "A two-stage method for text line detection in historical documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, no. 3, pp. 285–302, 2019.

[35] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM SIGGRAPH 2007 papers*, 2007, pp. 10–es.

[36] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[37] D. Markus, K. Florian, and G. Basilis, "ICDAR 2019 Competition on Baseline Detection (cBAD)," Feb. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3568023

[38] A. Toselli, V. Romero, M. Villegas, E. Vidal, and J. Sánchez, "Htr dataset icfhr 2016," Feb. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1164045

[39] L. Quirós, V. Bosch, L. Serrano, A. H. Toselli, and E. Vidal, "From hmms to rnns: computer-assisted transcription of a handwritten notarial records collection," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 116–121.

[40] T. Grüning, R. Labahn, M. Diem, F. Kleber, and S. Fiel, "Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 351–356.

[41] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[42] V. Romero, J. A. Sanchez, V. Bosch, K. Depuydt, and J. de Does, "Influence of text line segmentation in handwritten text recognition," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 536–540.