TOWARD A DATASET-AGNOSTIC WORD SEGMENTATION METHOD

Gregory Axler[1], *Lior Wolf*[1,2]

[1]The School of Computer Science, Tel Aviv University, Israel
[2]Facebook AI Research

ABSTRACT

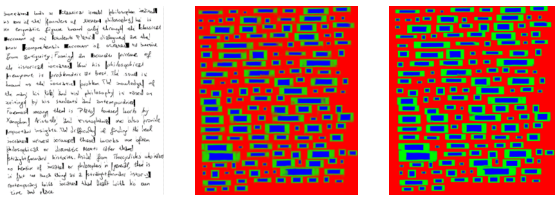
Word segmentation in documents is a critical stage towards word and character recognition, as well as word spotting. Despite recent advancements in word segmentation and object detection, detecting instances of words in a cluttered handwritten document remains a non-trivial task that requires a large amount of labeled documents for training. We present a flexible and general framework for word segmentation in handwritten documents, which incorporates techniques from the recent object detection literature as well as document analysis tools. Our method utilizes information that is relevant for word segmentation and ignores other highly variable information contained in a handwritten text, thus allowing for efficient transfer learning between datasets and alleviating the need for labeled training data. Our approach efficiently detects words in a variety of scanned document images, including historical handwritten documents and modern day handwritten documents, presenting excellent results on existing benchmarks. In addition, we demonstrate the usefulness of our approach by achieving state-of-the-art results for segmentation-free word spotting tasks.

Index Terms— Object Detection, Document Analysis, Transfer Learning

1. INTRODUCTION

Segmentation of individual words in a document image is a crucial task for performing word spotting, full text transcription and document clustering. Many of the recent word spotting and word recognition techniques assume access to segmented documents. This constitutes a limiting factor, since the word segmentation problem remains challenging in the case of unconstrained handwritten documents. This is due to large variations in writing styles and languages, overlapping and touching text parts, existence of accents, punctuation marks and decorative letters, local text skew and slant. The challenge is particularly striking in historical documents, where document and text degradation is common.

Classical word segmentation methods are based on connected components, where segmentation is performed by classifying gaps between connected components as being an inter-word or an intra-word gap [1]. Gap classification methods suffer from several drawbacks, including the need for preprocessing and limited utilization of the available information [2]. These issues severely limit the usability of such methods to cluttered or corrupt handwritten documents. See also [3, 4, 5, 6, 7]



(a) (b) (c)

Fig. 1: (a) A sample document from the ICDAR dataset and (b) A heatmap generated by the heatmap network.(c) A smooth heatmap generated by the smoother network.

Segmentation methods based on deep learning were recently introduced. These methods treat word segmentation as a special case of object detection. Although deep learning based methods allow a much better utilization of available information and added versatility in the types of documents that can be segmented, bounding box proposal generation still remains a bottleneck and some sort of connected component method must be used to augment the box proposal generation process. This is due to imperfect suitability of existing bounding box proposal generation processes to handwritten text documents [8, 9]. Moreover, deep learning based methods depend on training data availability, which in a setting of historical manuscripts might be a limiting factor.

To address the above issues, we propose a fully convolutional neural network based method that implicitly incorporates gap classification into the proposal generation process, and produces a box proposal generation process that is compatible with the structure of text documents.
This is achieved by predicting a heatmap (probability mask) that indicates whether an image pixel belongs inside a word bounding box or is a background pixel. Gap classification problem is solved implicitly, since, in order to classify a non-word pixel, a decision must be made as to whether a pixel belongs to an inter-word gap or an intra-word gap. To produce high quality, language- and style-agnostic word bounding boxes we propose a dynamic bounding box regression mechanism. The mechanism uses the heatmap and utilizes the

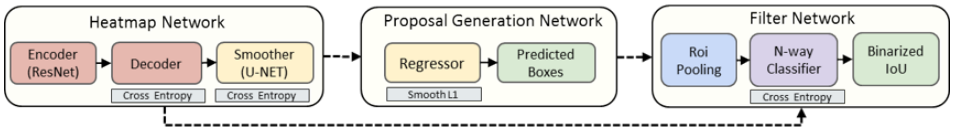


Fig. 2: Our architecture consists of a Heatmap Network that translates document image into a heatmap. The heatmap is fed into a Proposal Generation Network and to a Filter Network. The Proposal Generation Network learns to generate proposal bounding boxes based on a document's heatmap while the Filter Network learns to estimate how well the proposed bounding box envelopes a word (measured as an IoU with a ground truth bounding box), based on the heatmap as well as the proposed bounding box dimensions. Finally, non-maximal suppression is performed to produce word bounding box predictions.

relatively low overlapping among words and the linear structure common in most documents. Box proposals are then filtered to identify those that tightly envelope words. Heatmap production, bounding box regression and box filtering are implemented using a set of fully convolutional neural networks with specific losses and architectures that suit text document structure. Also, use of heatmaps is conductive for preserving information relevant for word segmentation and discarding other highly variable information contained in a handwritten text, thus allowing for efficient transfer learning between datasets and alleviating the need for labeled training data.

Our method can easily be trained for different kinds of documents, including cluttered historical documents, does not require preprocessing and generalizes well to a variety of languages. We evaluate its performance using established error metrics, previously used in competitions for word segmentation. The method achieves state of the art results with a large margin on ICDAR 2013 handwriting segmentation database of Latin-based and Indian languages. We also demonstrate the usefulness of our method by applying it to a segmentation-free word spotting task and achieve a state-of-the-art result. We make our source code available at: https://github.com/gaxler/dataset_agnostic_segmentation

2. PROPOSED METHOD

Given a document image, we want to identify rectangle bounding boxes that tightly envelope each of the document's words. To solve this problem, we propose a network architecture that consists of three parts: (i) A heatmap network that transforms a document image into a per-pixel "written vs background" probability map; (ii) A bounding box proposals generation network and (iii) Proposal filtering network. The architecture is depicted schematically in Fig. 2.

The heatmap network is formulated as a three class classification problem with positive (well inside a bounding box), periphery (near the edge) and negative (background) classes (See Fig.1). Using a heatmap implicitly incorporates gap classification into the proposal generation process and helps to "declutter" the document by reducing word overlaps. The network is implemented via encoder-decoder CNN architecture based on ResNet [10] that encodes an image into a feature map with size reduced by a factor of 8 and decodes the feature map into a heatmap with same size as the original document (See Fig.3). To increase generalization and transfer-

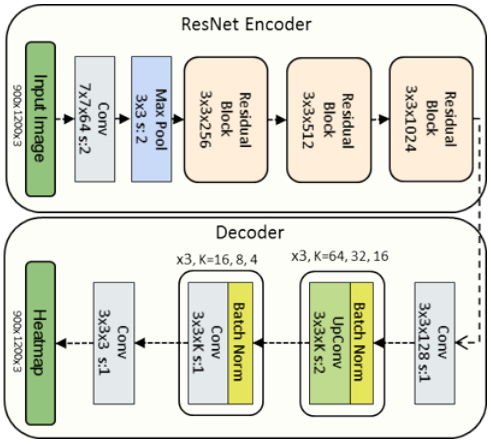


Fig. 3: The Encoder-Decoder architecture of the heatmap net.

ability between datasets, we apply a "smoother" network that learns to correct the mistakes of the heatmap network based on the noisy heatmap and the input image. The Smoother Network is based on U-NET architecture [11] with five down and five up layers. For each document, a ground truth mask is produced based on word bounding box locations and used as training labels for the Heatmap network. Outer 20% of a box are defined as periphery. The network is trained with the cross entropy loss. Since documents contain an order of magnitude more background pixels than foreground pixels, we introduce class weighting and assign a larger weight to the foreground and periphery pixels. For further details please refer to the source code.

Proposal Generation Network uses a heatmap to produce bounding box coordinates. The coordinates are parametrized as the left-top and right-bottom image locations of the bounding box (LTRB).
With the heatmap as an input and since our objects of interest are words, we can expect very low overlapping among objects. Unlike general objects, where, e.g., a bounding box of a sofa might fully contain a bounding box of a person. This allows us to incorporate an inductive bias that reduces the need to tune hyper-parameters and makes the learning task easier. We propose a dynamic mechanism: instead of anchors with predefined shapes (i.e. [12, 13]), we use uniformly spread center points along an image and learn a location invariant parametrization of a bounding box relative to each of the cen-

(a) (b)

Fig. 4: (a) The center points are equally distributed in 8 pixel gaps. (b) Box parametrization relative to a center point. Red arrows represent shifts to be predicted.

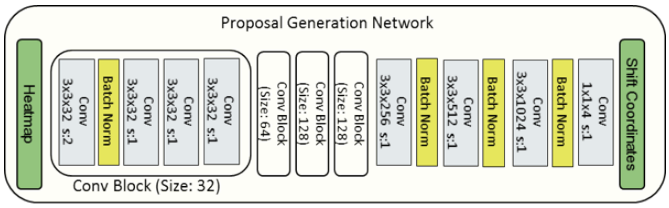


Fig. 5: The architecture of the proposal generation network

ter points. This way, the need to tune hyper-parameters for anchor size, aspect-ratio and ground truth comparability threshold is reduced. This adds robustness, since these factors often vary between different documents due to writing style, language, and preservation status. If a center point (i_c, j_c) is contained inside a bounding box b = $[i_L, j_T, i_R, j_B]$, we have the following parametrization: $T(i_c, j_c) = [i_c - i_L, i_R - i_c, j_c - j_T, j_B - j_c]$. We assign a zero-sized box for center points outside of a bounding box. In case a center point is inside the intersection of multiple bounding boxes, we ignore it during model fitting. See Fig.4 for illustration.

The Proposal Generation Network is modeled as a fully convolutional network that maps a heatmap to a regression feature map with size reduced by a factor of 8 and a regression function that maps each feature of the feature map to a location invariant box parametrization T (See Fig.5). Center points are spread on a grid such that each point is 8 pixels apart and each regression feature is assigned to a center point. The proposal generation network is trained with a smooth L1 loss, as defined in [14], weighted to adjust for the abundance of zero-size boxes.

The Filter network is used to identify box proposals that contain words and ignore proposals that cover no words or partial words. The filter network is implemented in the spirit of RPN's [12] anchor classification branch. We use a classifier to gauge the objectness (wordness in our case) of a proposal and extend the binary classifier used in [12] to an n-way classification to predict a discretized Jaccard index of the proposal with the closest (in terms of Jaccard index) ground truth box (See Fig.6).

Jaccard index discretization takes place by dividing the range $[0, 1]$ into $n = 5$ bins, where the first bin includes all boxes below 0.45 IoU with ground truth and the rest are uniformly spread in $[0.45, 1.0]$. The filter network is trained with a cross-entropy loss between the predicted probabilities on the IoU bins and the ground truth discretized IoU.

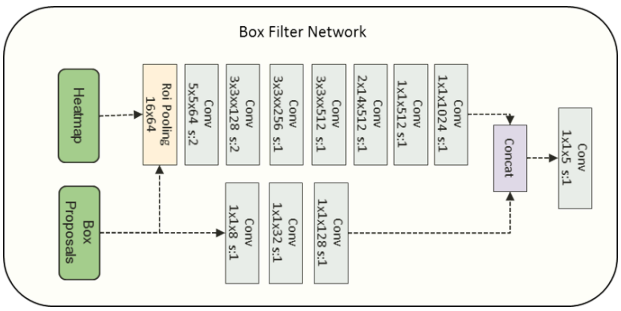


Fig. 6: The architecture of the filter network, which classifies each proposal as a word or a background

Word Embedding network is used to assess the performance of our segmentation method in a segmentation free word spotting task. We embed word images into a Pyramidal Histogram Of Characters (PHOC) [15] representation. We follow [9] and use 540-dimensional PHOC embedding (26 English characters, 10 digits and five levels in the PHOC). The embedding network is a CNN that takes the feature map of the heatmap generation network as an input. Training of the embedding network is cast as a multi-label classification problem and the training is done with a sigmoid cross-entropy loss (see [16, 9, 17] for further discussion)

3. EXPERIMENTS

We evaluate our pipeline on a handwriting segmentation task and perform a generalization analysis of our method to unseen or partially seen data. We also evaluate the usefulness of our method in a segmentation free word spotting scenario.

For **handwriting segmentation** we evaluate our pipeline on the ICDAR 2013 handwriting segmentation contest dataset [18] (ICDAR). **Generalization analysis** is performed by evaluation of segmentation performance measures on previously unseen or partially seen data. Our system is trained on four data regimes: (1) a source dataset; (2) a source dataset with 20 documents of the target dataset; (3) a source dataset with 10% of the target dataset; (4) a source and target datasets combined. Performance is evaluated on the validation set of the target dataset. For generalization analysis, we utilize two additional datasets, IAM Handwritten Database (IAMDB) [19] and pages from the Transcribe Bentham project (BENTHAM) [20]. For **segmentation free word spotting,** we use two of the biggest and most challenging datasets,IAMDB and Botany in British India (BOTANY) [21].

Implementation Details: The full architecture is trained in two stages. In the first stage the Heatmap Network is trained (along with word embedding network for word spotting task) followed by smoother network training (keeping the rest fixed). In the second stage the box proposal generation and the filter networks are trained (keeping the rest fixed). During training box proposals are randomly sampled and augmentations are randomly applied. For further imple-

	M	o2o	DR (%)	RA (%)	FM (%)
ILSP [23]	23,409	20,686	87.93	88.37	88.15
Students-t [7]	23,150	20,791	88.38	89.81	89.09
NCSR [4]	22,834	20,774	88.31	90.98	89.62
Golestan [18]	23,322	21,093	89.66	90.44	90.05
DTP [8]	-	-	87.09	93.82	90.33
Ryu et al. [2]	-	-	90.50	91.55	91.03
Our Method	23,551	22,043	**93.70**	**94.40**	**94.05**

Table 1: ICDAR benchmark results: M is the total proposals generated, o2o is the number of correct proposals, DR represents recall, RA accuracy and FM the F_1 score.

mentation and architecture details, please refer to the source code at:
https://github.com/gaxler/dataset_agnostic_segmentation

Performance Measures: To be comparable with previous results, we follow the word segmentation performance measure introduced in [22, 18]. The performance is evaluated by precision (RA), recall (DR), and F_1 (FM) scores. A proposal is considered a match for ground truth if IoU is greater than 0.9 for ICDAR dataset (see [22]). For IAMDB and BENTHAM we set the IoU threshold to 0.6 following [8] (due to differing definitions of IoU - see [8]). In word spotting, our protocol follows the usual segmentation-free Querby by String (QbS) word spotting protocols [9, 17]. A detection is considered as relevant if its IoU with a ground truth regions exceeds a threshold (0.25 or 0.5) and the retrieved proposal contains a word relevant to the query. Following [9], we remove stopwords from the set of queries, queries that come from lines that are marked as containing segmentation errors are removed and ground truth boxes that are so small that they collapse to a width or height of zero when downsampled by a factor of 8 are also removed.

Results: Segmentation task results on the ICDAR dataset are reported in Tab. 1. As can be seen, our method outperforms all literature baselines by a significant margin. We note that the previous state of the art on ICDAR was not obtained by the previous attempt [8] to employ an adaptation of FR-CNN, but with a classical method [2]. Thus, the novelties we have introduced play an important role.

As evident from table 2, using a heatmap as an input instead of document image greatly improves segmentation performance when using only a small fraction of the target dataset for training. In all examined cases, training on a full target dataset, using raw document images as inputs, provides significantly lower segmentation performance than utilizing heatmaps as inputs and using a small fraction of the target dataset for training.
Word spotting task results are presented in Tab.3. As can be seen, our method outperforms IAMDB literature benchmarks by significant margin and performs comparably on BOTANY.

	Heatmap Inputs			Image Inputs		
	DR	*RA*	*FM*	*DR*	*RA*	*FM*
IAMDB						
Source Only	65.77	29.15	40.40	53.53	38.15	44.55
Source + 20 Doc.	83.72	91.26	87.33	80.17	65.00	71.79
Source + 10%	85.22	92.54	88.73	81.14	72.87	76.78
Source + Target	85.75	93.73	89.56	80.40	71.12	75.47
BENTHAM						
Source Only	29.83	34.75	31.10	17.03	27.06	20.90
Source + 20 Doc.	66.62	73.58	69.93	60.83	60.83	60.83
Source + 10%	67.27	73.94	70.40	63.62	62.84	63.23
Source + Target	72.63	79.29	75.82	65.00	64.76	64.88

Table 2: Word segmentation results on the IAMDB and BENTHAM datasets when, using the ICDAR as the source training data, and when performing adaptation using a subset of the target dataset.

	IAMDB		BOTANY	
	25%	*50%*	*25%*	*50%*
BG-INDEX	-	48.6	-	-
Ctrl-F-Net (PHOC) [9]	80.8	78.8	-	-
Ctrl-F-Net (DCToW) [9]	82.5	80.3	-	-
Rothacker et al. [17]	-	-	**85.3**	78.8
Ours	**85.6**	**85.4**	78.7	**79.0**

Table 3: IAMDB & BOTANY word spotting results.

4. DISCUSSION

We introduced a novel fully convolutional neural network based method for word segmentation of a document image. Our architecture was designed to address the drawback of existing word segmentation methods by utilizing text image structure as a prior for the architecture and implicitly incorporating gap classification information as part of the architecture. Furthermore our method is designed to utilize information that is relevant for word segmentation and abstracts away other highly variable information contained in a handwritten text, thus increasing similarity among datasets and alleviating the need for segmented training data. Our method outperforms previous state of the art results on word segmentation benchmarks by a significant margin. Additionally, we demonstrate our method's usefulness in segmentation-free word spotting task and achieve a state-of-the-art or comparable result for word spotting benchmarks.

Acknowledgments Research supported in part by Grant #282601852 from the Deutsch-Israelische Projektkooperation (DIP), Grant #133014 of the Israel Science Foundation (ISF), Grant (#I-145-101.3-2013) from the German-Israeli Foundation for Scientific Research and Development (GIF), and a grant from the Blavatnik Family Fund.

5. REFERENCES

[1] Angelos P Giotis, Giorgos Sfikas, Basilis Gatos, and Christophoros Nikou, "A survey of document image word spotting techniques," *Pattern Recognition*, vol. 68, pp. 310–332, 2017.

[2] Jewoong Ryu, Hyung Il Koo, and Nam Ik Cho, "Word segmentation method for handwritten documents based on structured learning," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1161–1165, 2015.

[3] Giovanni Seni and Edward Cohen, "External word segmentation of off-line handwritten text lines," *Pattern Recognition*, vol. 27, no. 1, pp. 41–52, 1994.

[4] Georgios Louloudis, Basilios Gatos, Ioannis Pratikakis, and Constantin Halatsis, "Text line and word segmentation of handwritten documents," *Pattern Recognition*, vol. 42, no. 12, pp. 3169–3183, 2009.

[5] Tamas Varga and Horst Bunke, "Tree structure for word extraction from handwritten text lines," in *Int. Conf. on Document Analysis and Recognition.*, 2005.

[6] Chen Huang and Sargur N Srihari, "Word segmentation of off-line handwritten documents.," in *DRR*, 2008, p. 68150E.

[7] Georgios Louloudis, Giorgos Sfikas, Nikolaos Stamatopoulos, and Basilis Gatos, "Word segmentation using the student's-t distribution," in *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 78–83.

[8] Tomas Wilkinson and Anders Brun, "A novel word segmentation method based on object detection and deep learning," in *International Symposium on Visual Computing*. Springer, 2015, pp. 231–240.

[9] Tomas Wilkinson, Jonas Lindström, and Anders Brun, "Neural ctrl-f: Segmentation-free query-by-string word spotting in handwritten manuscript collections," *arXiv preprint arXiv:1703.07645*, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *ECCV*, 2016.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "SSD: Single shot multibox detector," in *ECCV*. Springer, 2016, pp. 21–37.

[14] Ross Girshick, "Fast r-cnn," in *CVPR*, 2015.

[15] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny, "Word spotting and recognition with embedded attributes," *TPAMI*, 2014.

[16] Sebastian Sudholt and Gernot A Fink, "Phocnet: A deep convolutional neural network for word spotting in handwritten documents," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 277–282.

[17] Leonard Rothacker, Sebastian Sudholt, Eugen Rusakov, Matthias Kasperidus, and Gernot A Fink, "Word hypotheses for segmentation-free word spotting in historic document images," in *ICDAR*, 2017.

[18] Nikolaos Stamatopoulos, Basilis Gatos, Georgios Louloudis, Umapada Pal, and Alireza Alaei, "Icdar 2013 handwriting segmentation contest," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1402–1406.

[19] U-V Marti and Horst Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[20] Mauricio Villegas, Joan Puigcerver, Alejandro Héctor Toselli, Joan-Andreu Sánchez, and Enrique Vidal, "Overview of the imageclef 2016 handwritten scanned document retrieval task.," in *CLEF (Working Notes)*, 2016, pp. 233–253.

[21] Praveen Krishnan, Kartik Dutta, and CV Jawahar, "Deep feature embedding for accurate recognition and retrieval of handwritten text," in *Frontiers in Handwriting Recognition*, 2016.

[22] Ihsin T Phillips and Atul K Chhabra, "Empirical performance evaluation of graphics recognition systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 849–870, 1999.

[23] Vassilis Papavassiliou, Themos Stafylakis, Vassilis Katsouros, and George Carayannis, "Handwritten document image segmentation into text lines and words," *Pattern Recognition*, vol. 43, no. 1, pp. 369–377, 2010.

[24] Matthias Zimmermann and Horst Bunke, "Automatic segmentation of the iam off-line database for handwritten english text," in *Pattern Recognition*. IEEE, 2002, vol. 4, pp. 35–39.

A. ARCHITECTURE DETAILS

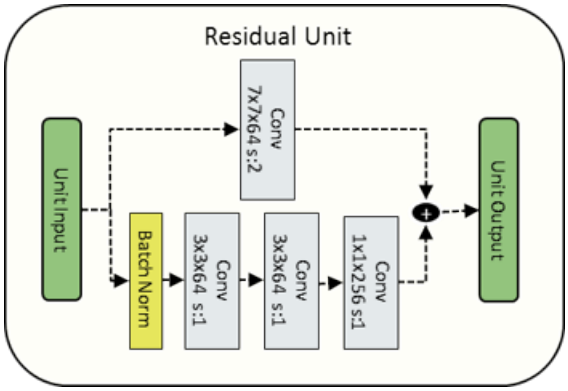


Fig. 7: Residual Unit architecture (sizes compatible with first residual block).

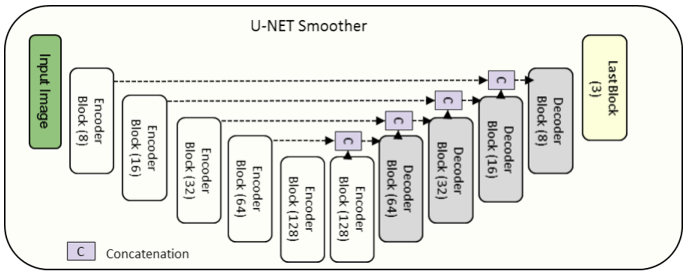


Fig. 8: The Smoother Network Architecture: U-NET with five down blocks and five up blocks

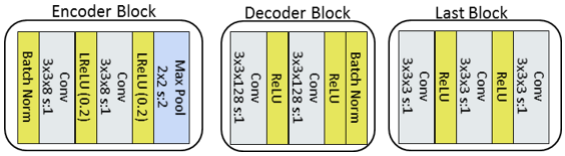
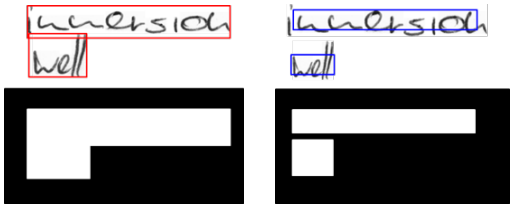


Fig. 9: U-NET Blocks: with five down layers and five up layers



(a) (b)

Fig. 10: (a) Full bounding box and ground truth heatmap. (b) Trimmed (10%) bounding boxes and ground truth heatmap.

B. DETAILED EXPERIMENTAL SETUP

B.1. Datasets

The ***ICDAR*** 2013 dataset contains images of handwritten documents that were produced by several writers in English, Greek and Bangla. The training data consists of 200 document images (total of 29,423 annotated word instances) while the test data consists of 150 document images (total of 23,525 annotated word instances). The document images are binary and do not include any non-text elements such as lines, drawings, etc.

IAMDB [24] contains pages of handwritten English text. 657 writers contributed samples of their handwriting that are available as 1,539 scanned pages. The pages contain 115,320 annotated word instances. We use the documents proposed for the Large Writer Independent Text Line Recognition Task for the train, validation and test splits.

The ***BENTHAM*** dataset consists of original scanned color page images of manuscripts by Jeremy Bentham, an 17th century English philosopher. The page images were divided into three sets: a training set of 363 pages, a development set containing 433 pages, and a test set of size 200. The training set contains 75,132 annotated word instances. No word annotations are available for the development and test sets. To produce an annotated validation and test set, we created a random split of the 363 train pages into a 217 page training set, two validation sets of 36 pages each, and a test set of 74 pages. For exact split details and page ids, please refer to the source code. Botany in British India or ***BOTANY*** is from the India Office Records and provided by the British Library. The dataset is partitioned into three parts. Each partition contains 10, 29 and 112 documents and 1684, 3611, 16686 English word instances (respectively). For word spotting task we follow [17] and use the last partition as a training set. For test BOTANY features 20 test document images containing 3,318 annotated word regions.

B.2. Implementation Details

The full architecture is trained in two stages dubbed below as Heatmap Training and Box Training. First, we perform Heatmap Training for 50K iterations, that is split into two stages. In the first stage we train the feature map and the heatmap for 30K iterations followed by 20K iteration of smoother network training. The Heatmap Training is followed by 50K iterations of Box Training.

For the Heatmap Training, the loss is as follows:

$$L_{heatmap} = \eta_1 \times L_{hmap} + \eta_2 \times L_{phoc} \quad (1)$$

with L_{hmap} and L_{phoc} being the average losses for the heatmap and the embedding network respectively. We set $\eta_1 = 50$, $\eta_2 = 150$ in word spotting setting.

For the Box Training, the loss is follows:

$$L_{box} = \eta_1 \times L_{proposal} + \eta_2 \times L_{filter} \quad (2)$$

with $L_{proposal}$ and L_{filter} being the average losses for the proposal generation network and the filter network respectively. We set $\eta_1 = 1$, $\eta_2 = 1$

For Heatmap Training and Box Training, we use the ADAM optimization algorithm with an initial learning rate of 10^{-4} and $\beta_1 = 0.5$, and $\beta_2 = 0.999$. Learning rate is divided by 10 after 10K iterations.

Training is done in mini-batches, using a single document image per mini-batch. All document images are resized to a uniform size of (900, 1200) pixels, preserving the aspect ratio of the original image and using border replication padding, where necessary.

The L_{hmap} loss class weights for negative class and positive class are set to 0.33 and 0.67 respectively, $L_{proposal}$ weights for the positive and negative classes are set to 100 and 2 respectively. Furthermore, the training regression predictions and targets are scaled as $s(x) = Cx$ where $C \in \mathbb{R}$, and we use C = 1500.

Box proposals for training the filter network are randomly sampled. The random sampling is done in two stages. First, we sample U random boxes around each of the W words in a document. Next, we sample $W \times U$ boxes uniformly across the image. For each random box, the IoU is calculated with ground truth boxes and to each random box, a label is assigned according to its maximal IoU with the ground truth boxes. Filter network loss is calculated on a uniform subsample (with replacement) of 400 boxes (50 for the first two IoU classes and a 100 for the last three IoU classes, where we set n=5 for n-way box classification).

Following data augmentation techniques were applied: (i) a multiplicative Gaussian noise; (ii) random resize of an image; (iii) dilation and slant correction for randomly selected words. These augmentations were applied at random and not for every training sample: the Gaussian noise was applied with an application probability of 0.1, random resize was applied for half of the training samples, dilation and slant correction were applied on a page with probability 0.2 and within a page, a word was augmented with probability 0.5.

In addition, L_2 regularization was applied on all model weights, except for the ResNet encoder of the Heatmap Generation Network, with a regularization rate of 0.01.

At test time, prediction is performed as follows: (i) The "Wordness" score is calculated for each proposal by summing over predicted probabilities of the proposal network, for IoU values of larger than 0.6; (ii) Proposals are sorted according to the score of the filtering network score and non-maximal suppression is applied with IoU threshold of 0.1; (iii) Proposals with filtering score lower than 0.5 are discarded. When in word spotting setting, in addition to the above, we predict a PHOC embedding for each of the proposals.

The evaluation is performed in the following fashion: (i) IoU is calculated for each prediction with each ground truth box; (ii) Per ground truth box, maximal IoU prediction is assigned as a prediction for ground truth box (Skipping previously used predictions); (iii) If a prediction is above a threshold α (0.9 for ICDAR and 0.6 for IAMDB and BENTHAM), we count it as a correct prediction.

Hyper-parameters settings for loss weighting, learning rate and box prediction targets transformation function were based on the training set. The architecture selection, weight regularization rate, IoU threshold for non-maximal suppression and probability detection for box filtering was based on the ICDAR 2013 validation set, were obtained by IID sampling 20% of the ICDAR 2013 training set.

For further implementation and architecture details, please refer to the source code at:
https://github.com/gaxler/dataset_agnostic_segmentation

B.3. Performance Measures

To be comparable with previous results, we follow the word segmentation performance measure introduced in [22, 18]. The measure is based on counting the number of matches between words detected by an algorithm and the words in the ground truth. Detection of words is based on a measure, based on the Jaccard Index of word foreground pixels. For a ground truth region G_i and foreground pixels of a region proposed by the algorithm R_j the measure is defined as $S_{ij} = \frac{|G_i \cap R_j|}{|G_i \cup R_j|}$.

A region j is considered a match for ground truth region i if $S_{ij} \geq \alpha$. Where $\alpha = 0.9$. Detection Rate (Recall) and Recognition Accuracy (Accuracy) are given by $DR = \frac{o2o}{N}$ and $RA = \frac{o2o}{M}$.

Where N is number of ground truth words in a document and M is the number of regions detected by the algorithm, o2o is the number of matched regions. A performance metric $FM = \frac{2DR \times RA}{DR + RA}$. The document performance metric is defined as the average FM value over all documents.

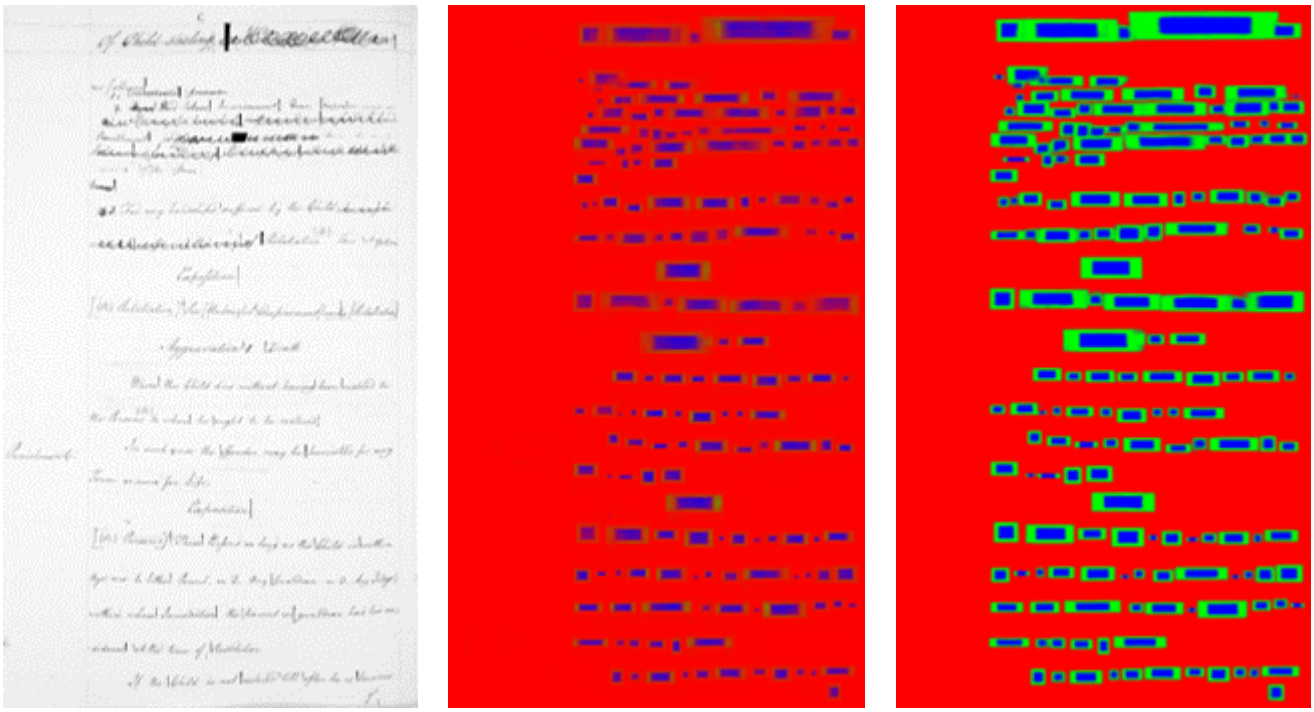
For the more challenging datasets (IAMDB, BENTHAM), we calculate the IoU for instance detection based on foreground and background box pixels as opposed to foreground only. Following [8] we set a lower threshold for detection ($\alpha = 0.6$) when all pixels are counted.

In word spotting, our protocol follows the usual segmentation-free Querby by String (QbS) word spotting protocols: Each query string is used to retrieve a list of regions The list is sorted according to a cosine distance between proposal and query PHOC embedding. The retrieval lists for all queries are scored by mean average precision (mAP), where average precision is defined as $AP = \frac{\sum_k^N P(k) \times r(k)}{|R|}$, and

$P(k)$ is the precision on the first k regions retrieved and $r(k)$ is and indicator function of whether the region on rank k is relevant. R is the set of all relevant queries. A detection is considered as relevant, if the IoU of a retrieved proposal with

a ground truth region is greater than a given overlap threshold (0.25 or 0.5) and the retrieved proposal contains a word relevant to the query. Following [9], we remove stopwords from the set of queries, queries that come from lines that are marked as containing segmentation errors are removed and ground truth boxes that are so small that they collapse to a width or height of zero when downsampled by a factor of 8 are also removed. $mAP = \frac{\sum_{q \in Q} AP(q)}{|Q|}$, where Q is the set of all queries.

C. DOCUMENT & HEATMAP SAMPLES



(a) (b) (c)

Fig. 11: (a) A document from the BENTHAM dataset. (b) Raw heatmap generated by the heatmap network. (c) Smooth heatmap generated by the smoother network. Red, green and blue colors represent the negative, periphery and positive classes of the heatmap