

dhSegment: A generic deep-learning approach for document segmentation

Sofia Ares Oliveira^{†*}, Benoit Seguin^{†*}, Frederic Kaplan[†]

[†]Digital Humanities Laboratory, EPFL, Lausanne, VD, Switzerland
^{{sofia.oliveiraares, benoit.seguin, frederic.kaplan}@epfl.ch}

*These authors contributed equally to this work

Abstract—In recent years there have been multiple successful attempts tackling document processing problems separately by designing task specific hand-tuned strategies. We argue that the diversity of historical document processing tasks prohibits to solve them one at a time and shows a need for designing generic approaches in order to handle the variability of historical series. In this paper, we address multiple tasks simultaneously such as page extraction, baseline extraction, layout analysis or multiple typologies of illustrations and photograph extraction. We propose an open-source implementation of a CNN-based pixel-wise predictor coupled with task dependent post-processing blocks. We show that a single CNN-architecture can be used across tasks with competitive results. Moreover most of the task-specific post-processing steps can be decomposed in a small number of simple and standard reusable operations, adding to the flexibility of our approach.

Index Terms—document segmentation, historical document processing, document layout analysis, neural network, deep learning

I. INTRODUCTION

When working with digitized historical documents, one is frequently faced with recurring needs and problems: how to cut out the page of the manuscript, how to extract the illustration from the text, how to find the pages that contain a certain type of symbol, how to locate text in a digitized image, etc. However, the domain of document analysis has been dominated for a long time by collections of heterogeneous segmentation methods, tailored for specific classes of problems and particular typologies of documents. We argue that the variability and diversity of historical series prevent us from tackling each problem separately, and that such specificity has been a great barrier towards off-the-shelf document analysis solutions, usable by non-specialists.

Lately, huge improvements have been made in semantic segmentation of natural images (roads, scenes, ...) but historical document processing and analysis have, in our opinion, not yet fully benefited from these. We believe that a tipping point has been reached and that recent progress in deep learning architectures may suggest that some generic approaches would be now mature enough to start outperforming dedicated systems. Also with the growing interest in digital humanities research, the need for simple-to-use, flexible and efficient tools to perform such analysis increases.

This work is a contribution towards this goal and introduces dhSegment, a general and flexible architecture for pixel-wise segmentation related tasks on historical documents. We present the surprisingly good results of such a generic architecture across tasks common in historical document processing, and show that the proposed model is competitive or outperforming state-of-the-art methods. These encouraging results may have important consequences for the future of document analysis pipelines based on optimized generic building blocks. The implementation is open-source and available on Github¹.

II. RELATED WORK

In the recent years, Long et al. [1] popularized the use of end-to-end fully convolutional networks (FCN) for semantic segmentation. They used an ImageNet [2] pretrained network, deconvolutional layers for upsampling and combined final prediction layer with lower layers (skip connections) to improve the predictions. The U-Net architecture [3] extended the FCN by setting the expansive path (decoder) to be symmetric to the contracting path (encoder), resulting in an u-shaped architecture with skip connections for each level.

Similarly, the architectures for Convolutional Neural Networks (CNN) have evolved drastically in the last years with architectures such as Alexnet [4], VGG [5] and ResNet [6]. These architecture contributed greatly to the success and the massive usage of deep neural networks in many tasks and domains.

To some extent, historical document processing has also experienced the arrival of neural networks. As seen in the last competitions in document processing tasks [7]–[9], several successful methods make use of neural network approaches [10], [11], especially u-shaped architectures for pixel-wise segmentation tasks.

III. APPROACH

A. Outline

The system is based on two successive steps which can be seen in Figure 1:

- The first step is a Fully Convolutional Neural Network which takes as input the image of the document to be

¹dhSegment implementation : <https://github.com/dhlab-epfl/dhSegment>

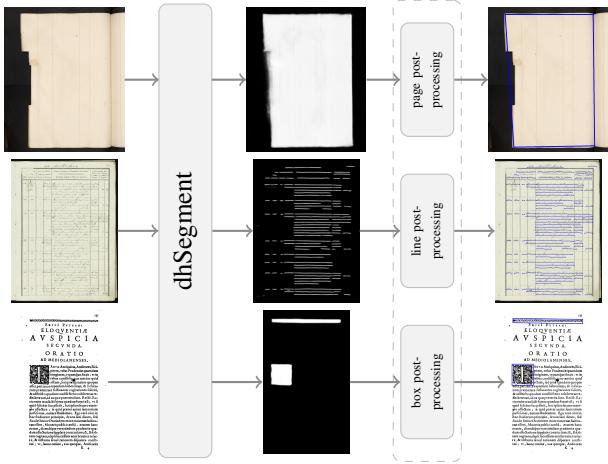


Fig. 1. Overview of the system. From an input image, the generic neural network (dhSegment) outputs probabilities maps, which are then post-processed to obtain the desired output for each task.

processed and outputs a map of probabilities of attributes predicted for each pixel. Training labels are used to generate masks and these mask images constitute the input data to train the network.

- The second step transforms the map of predictions to the desired output of the task. We only allow ourselves simple standard image processing techniques, which are task dependent because of the diversity of outputs required.

The implementation of the network uses TensorFlow [12].

B. Network architecture

The architecture of the network is depicted in Figure 2. dhSegment is composed of a contracting path², which follows the deep residual network ResNet-50 [6] architecture (yellow blocks), and a expansive path that maps the low resolution encoder feature maps to full input resolution feature maps. Each path has five steps corresponding to five feature maps' sizes S , each step i halving the previous step's feature maps size.

The contracting path uses pretrained weights as it adds robustness and helps generalization. It takes advantage of the high level features learned on a general image classification task (ImageNet [2]). For simplicity reasons the so-called “bottleneck” blocks are shown as violet arrows and downsampling bottlenecks as red arrows in Figure 2. We refer the reader to [6] for a detailed presentation of ResNet architecture.

The expanding path is composed of five blocks plus a final convolutional layer which assigns a class to each pixel. Each deconvolutional step is composed of an upscaling of the previous block feature map, a concatenation of the upscaled feature map with a copy of the corresponding contracting feature map and a 3x3 convolutional layer followed by a rectified linear unit (ReLU) [13]. The number of features channels in step $i = 4$ and $i = 5$ are reduced to 512 by a 1x1 convolution before concatenation in order to reduce the

²We reuse the terminology ‘contracting’ and ‘expanding’ paths of [3]

number of parameters and memory usage. The upsampling is performed using a bilinear interpolation.

The architecture contains 32.8M parameters in total but since most of them are part of the pre-trained encoder, only 9.36M have to be fully-trained.³

C. Post-processing

Our general approach to demonstrate the effectiveness and genericity of our network is to limit the post-processing steps to simple and standards operations on the predictions.

Thresholding: Thresholding is used to obtain a binary map from the predictions output by the network. If several classes are to be found, the thresholding is done class-wise. The threshold is either a fixed constant ($t \in [0, 1]$) or found by Otsu's method [14].

Morphological operations: Morphological operations are non-linear operations that originate from mathematical morphology theory [15]. They are standard and widely used methods in image processing to analyse and process geometrical structures. The two fundamental basic operators, namely the erosion and dilation, can be combined to result in opening and closing operators. We limit our post-processing to these two operators applied on binary images.

Connected components analysis: In our case, connected components analysis is used in order to filter out small connected components that may remain after thresholding or morphological operations.

Shape vectorization: A vectorisation step is needed in order to transform the detected region into a set of coordinates. To do so, the blobs in the binary image are extracted as polygonal shapes. In fact, the polygons are usually bounding boxes represented by four corner points, which may be the minimum rectangle enclosing the object or quadrilaterals. The detected shape can also be a line and in this case, the vectorization consists in a path reduction.

D. Training

The training is regularized using L2 regularization with weight decay (10^{-6}). We use a learning rate with an exponential decay rate of 0.95 and an initial value in $[10^{-5}, 10^{-4}]$. Xavier initialization [16] and Adam optimizer [17] are used. Batch renormalization [18] is used to ensure that the lack of diversity in a given batch is not an issue (with $r_{min} = 0.1$, $r_{max} = 100$, $d_{max} = 1$). The images are resized so that the total number of pixels lies between $6 \cdot 10^5$ and 10^6 . Images are also cropped into patches of size 300×300 in order to fit in memory and allow batch training, and a margin is added to the crops to avoid border effects. The training takes advantage of on-the-fly data augmentation strategies, such as rotation ($r \in [-0.2, 0.2] rad$), scaling ($s \in [0.8, 1.2]$) and mirroring,

³Actually one could argue that the 1.57M parameters coming from the dimensionality reduction blocks do not have to be fully trained either, thus reducing the number of fully-trainable parameters to 7.79M. Indeed, they are initialized as random projections, which is a valid way of reducing dimensionality, so they can also be considered as part of the fine-tuning of the pre-trained network.

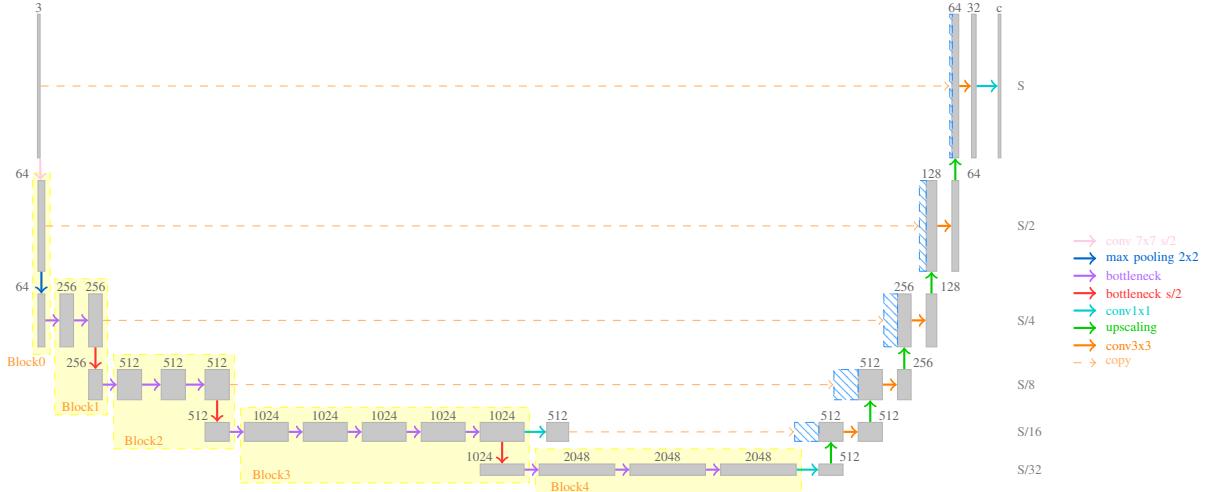


Fig. 2. Network architecture of dhSegment. The yellow blocks correspond to ResNet-50 architecture which implementation is slightly different from the original in [6] for memory efficiency reasons. The number of features channels are restricted to 512 in the expansive path in order to limit the number of training parameters, thus the dimensionality reduction in the contracting path (light blue arrows). The dashed rectangles correspond to the copies of features maps from the contracting path that are concatenated with the up-sampled features maps of the expanding path. Each expanding step doubles the feature map's size and halves the number of features channels. The output prediction has the same size as the input image and the number of features channels constitute the desired number of classes.

since data augmentation has shown to result in less original data needs.

In practice, setting up the training process is rather easy. The training parameters and choices are applicable to most experiments and the only parameter that needs to be chosen is the resizing size of the input image. Indeed, the resolution of the input image needs to be carefully set so that the receptive field of the network is sufficiently large according to the type of task.

All trainings and inferences run on a Nvidia Titan X Pascal GPU. Thanks to the pretrained weights used in the contracting path of the network, the training time is significantly reduced. During the experiments, we also noted that the pretrained weights seemed to help regularization since the model appeared to be less sensitive to outliers.

IV. EXPERIMENTS

In order to investigate the performance of the proposed method and to demonstrate its generality, dhSegment is applied on five different tasks related to document processing. Three tasks consisting in page extraction, baseline detection and document segmentation are evaluated and the results are compared against state-of-the art methods. Two additional private datasets are used to show the possibilities and performance of our system on ornaments and photograph extraction. For each task, the best set of post-processing parameters are selected according to the performance on the respective evaluation set.

A. Page extraction

Images of digitized historical documents very often include a surrounding border region, which can alter the outputs of document processing algorithms and lead to undesirable results. It is therefore useful to be able to extract the page

document from the digitised image. We use the dataset proposed by [19] to apply our method and compare our results to theirs in Table I. Our method achieves very similar results to human agreement.

The network is trained to predict for each pixel if it belongs to the main page, essentially predicting a binary mask of the desired page. Training is done on 1635 images for 30 epochs, using a batch size of 1. Since the network should see the image entirely in order to detect the page, full images are used (no patches) but are resized to $6 \cdot 10^5$ pixels and their aspect ratio is kept. The training takes around 4 hours.

To obtain a binary image from the probabilities output by the network, Otsu's thresholding is applied. Then morphological opening and closing operators are used to clean the binary image. Finally, the quadrilaterals containing the page are extracted by finding the four most extreme corner points of the binary image.

TABLE I
RESULTS FOR THE PAGE EXTRACTION TASK (MIOU)

Method	cBAD-Train	cBAD-Val	cBAD-Test
Human Agreement	-	0.978	0.983
Full Image	0.823	0.831	0.839
Mean Quad	0.833	0.891	0.894
GrabCut [20]	0.900	0.906	0.916
PageNet [19]	0.971	0.968	0.974
dhSegment (quads)	0.976	0.977	0.980

B. Baseline detection

Text line detection is a key step for text recognition applications and thus of great utility in historical document processing. A baseline is defined as a “virtual line where most characters rest upon and descenders extend below”.

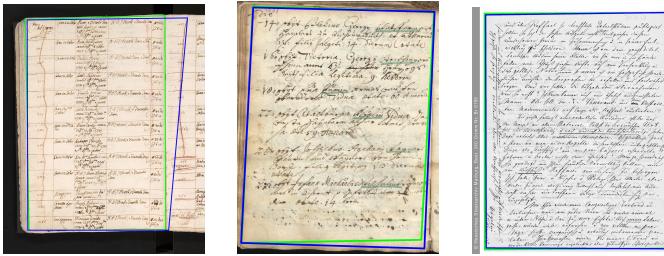


Fig. 3. Example of page detection on the cBAD-test set. Green rectangles indicate the ground-truth pages and blue rectangles correspond to the detections generated by dhSegment. The first extraction is inaccurate because part of the side page is also detected. The second one is slightly inaccurate according to the groundtruth but the entire page is still detected. The last example shows a correct detection.

The READ-BAD dataset [21] has been used for the cBAD: ICDAR2017 Competition [8].

Here the network is trained to predict the binary mask of pixels which are in a small 5-pixel radius of the training baselines. Each image is resized to have 10^6 pixels, training is done for 30 epochs, taking around 50 minutes for the complex track of [21].

The probability map is then filtered with a gaussian filter ($\sigma = 1.5$) before using hysteresis thresholding⁴ ($p_{high} = 0.4$, $p_{low} = 0.2$). The obtained binary mask is decomposed in connected components, and each component is finally converted to a polygonal line.

TABLE II

RESULTS FOR THE CBAD : ICDAR2017 COMPETITION ON BASELINE DETECTION [8] (TEST SET)

Method	Simple Track			Complex Track		
	P-val	R-val	F-val	P-val	R-val	F-val
LITIS	0.780	0.836	0.807	-	-	-
IRISA	0.883	0.877	0.880	0.692	0.772	0.730
UPVLC	0.937	0.855	0.894	0.833	0.606	0.702
BYU	0.878	0.907	0.892	0.773	0.820	0.796
DMRZ	0.973	0.970	0.971	0.854	0.863	0.859
dhSegment	0.943	0.939	0.941	0.826	0.924	0.872

C. Document layout analysis

Document Layout Analysis refers to the task of segmenting a given document into semantically meaningful regions. In the experiment, we use the DIVA-HisDB dataset [22] and perform the task formulated in [7]. The dataset is composed of three manuscripts with 30 training, 10 evaluation and 10 testing images for each manuscript. In this task, the layout analysis focuses on assigning each pixel a label among the following classes : text regions, decorations, comments and background, with the possibility of multi-class labels (e.g a pixel can be part of the main-text-body but at the same time be part of a decoration). Our results are compared with the participants of the competition in Table III.

⁴Applying thresholding with p_{low} then only keeping connected components which contains at least a pixel value $p \geq p_{high}$

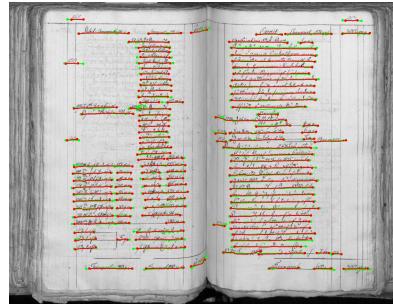
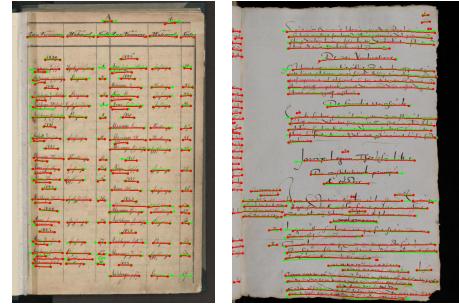


Fig. 4. Examples of baseline extraction on the complex track of the cBAD dataset. The ground-truth and predicted baselines are displayed in green and red respectively. Some limitations of the simple approach we propose can be seen here, for instance detecting text on the neighboring page (top right), or merging close text lines together (bottom and top left). These issues could be addressed with a more complex pipeline incorporating, for instance, page segmentation (as seen in Section IV-A) or by having the network predict additional features, but this goes beyond the scope of this paper.

For each manuscript, a model is trained solely on the corresponding 30 training images for 30 epochs. No resizing of the input images is performed but patch cropping is used to allow for batch training. A batch size of 8 and patches of size 400×400 are used for manuscripts CSG18 and CSG863 but because the images of manuscript CB55 have higher resolution (approximately a factor 1.5) the patch size is increased to 600×600 and the batch size is reduced to 4 to fit into memory. The training of each model lasts between two and four hours.

The post-processing consists in obtaining a binary mask for each class and removing small connected components. The threshold is set to 0.5 and the components smaller than 50 pixels are removed. The mask obtained by the page detection (Section IV-A) is also used as post-processing to improve the results, especially to reduce the false positive text detections on the borders of the image.

D. Ornament detection

Ornaments are decorations or embellishments which can be found in many manuscripts. The study of ornaments and discovery of unexpected details is often of major interest for historians. Therefore a system capable of filtering the pages containing such decorations in large collections and locate their positions is of great assistance.

⁶https://github.com/DIVA-DIA/DIVA_Layout_Evaluator

TABLE III
 RESULTS FOR THE ICDAR2017 COMPETITION ON LAYOUT ANALYSIS
 FOR CHALLENGING MEDIEVAL MANUSCRIPTS [7] - TASK-1 (IOU)

Method	CB55	CSG18	CSG863	Overall
System-1 (KFUPM)	.7150	.6469	.5988	.6535
System-6 (IAIS)	.7178	.7496	.7546	.7407
System-4.2 (MindGarage-2)	.9366	.8837	.8670	.8958
System-2 (BYU)	.9639	.8772	.8642	.9018
System-3 (Demokritos)	.9675	.9069	.8936	.9227
System-4.1 (MindGarage-1)	.9864	.9357	.8963	.9395
dhSegment	.9757	.9322	.9130	.9403
dhSegment + Page	.9783	.9317	.9205	.9435
System-5 (NLPR)	.9835	.9365	.9271	.9490

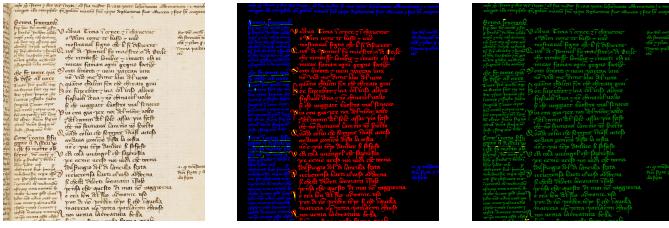


Fig. 5. Example of layout analysis on the DIVA-HisDB test test. On the left the original manuscript image, in the middle the classes pixel-wise labelled by the dhSegment and on the right the comparison with the ground-truth (refer to the evaluation tool⁶for the signification of colors)

The private dataset [23] used for this task is composed of several printed books. The selected pages were manually annotated and each ornament was marked by a bounding rectangle. A total of 912 annotated pages were produced, with 612 containing one or several ornaments. The dataset is split in the following way :

- 610 pages for training (427 with ornaments)
 - 92 pages for evaluation (62 with ornaments)
 - 183 pages for testing (123 with ornaments)

The original images are resized to $8 \cdot 10^5$ and the model is trained for 30 epochs with batch size of 16. The training takes less than two hours.

To obtain the binary masks, a threshold of $t = 0.6$ is applied to the output of the network. Opening and closing operations are performed and a bounding rectangle is fitted to each detected ornament. Finally, very small boxes (those with areas less than 0.5% of the image size) are removed.

The evaluation of ornaments detection task is measured using the standard Intersection over Union (IoU) metric, which calculates how well the predicted and the correct boxes overlap. Table IV lists the precision, recall and f-measure for three IoU thresholds as well as the mean IoU (mIoU) measure. Our results are compared to the method implemented in [23] which uses a region proposal technique coupled with a CNN classifier to filter false positives.

E. Photo-collection extraction

A very practical case comes from the processing of the scans of an old photo-collection. The inputs are high resolution scans of pieces of cardboard with an old photograph stuck in the

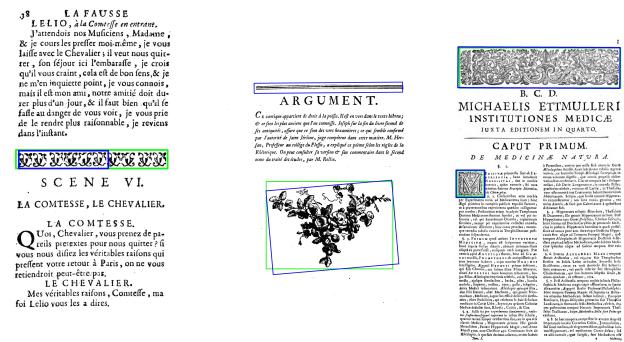


Fig. 6. The left image illustrates the case of a partially detected ornament, the middle one shows the detection of an illustration but also a false positive detection of the banner and the right image is a correct example of multiple ornaments extraction.

TABLE IV
ORNAMENTS DETECTION TASK. EVALUATION AT DIFFERENT IOU THRESHOLDS ON TEST SET

Method	IoU	F-val	P-val	R-val	mIoU
[23]	0.5	-	0.800	0.430	-
	0.5	-	0.470	0.600	-
dhSegment	0.7	0.941	0.969	0.914	
	0.8	0.874	0.847	0.902	0.870
	0.9	0.510	0.374	0.803	

middle, and the task is to properly extract the part of the scan containing the cardboard and the image respectively.

Annotation was done very quickly by directly drawing on the scans the part to be extracted in different colors (background, cardboard, photograph). Leveraging standard image editing software, 60 scans per hour can be annotated. Data was split in 100 scans for training, 20 for validation and 150 for testing. Training for 40 epochs took only 20 minutes. The network is trained to predict for each pixel its belonging to one of the classes.

The predicted classes are then cleaned with a simple morphological opening, and the smallest enclosing rectangle of the corresponding region is extracted. Additionally, one can use the layout constraint that the area of the photograph has to be enclosed in the area of the piece of cardboard. We compare the extracted rectangle with the smallest rectangle coming from the groundtruth mask and display relevant metrics in Table V.

TABLE V
PHOTO-COLLECTION EXTRACTION TASK. EVALUATION OF MIoU ON TEST SET, AND SOME RECALL AT IOU THRESHOLDS OF 0.85 AND 0.95

Method	Cardboard mIoU	Photo		
		mIoU	R@0.85	R@0.95
Predictions-only	0.992	0.982	0.980	0.967
+ layout constraint	0.992	0.988	1.000	0.993

V. DISCUSSION

While using the same network and almost the same training configurations, the results we obtained on the five different



Fig. 7. Example of extraction on the photo-collection scans. Note that contrary to the ornaments case, the zone to be extracted is better defined which allow for a much more precise extraction.

tasks evaluated are competitive or better than the state-of-the-art. Despite the genericity and flexibility of the approach, we can also highlight the speed of the training (less than an hour in some cases), as well as the little amount of training data needed, both thanks to the pre-trained part of the network.

The results presented in this paper demonstrate that a generic deep learning architecture, retrained for specific segmentation tasks using a standardized process, can, in certain cases, outperform dedicated systems. This has multiple consequences.

First, it opens an avenue towards simple, off-the-shelf, programming bricks that can be trained by non-specialists and used for large series of document analysis problems. Using such generic bricks, one just has to design examples of target masks to train a dedicated segmentation system. Such bricks could easily be integrated in intuitive visual programming environments to take part in more complex pipelines of document analysis processes.

Eventually, although the scenario discussed in this article presents how the *same* architecture can be trained to perform efficiently on separate segmentation tasks, further research should study how incremental or parallel training over various tasks could boost transfer learning for segmentation problems. In other words, it is not unlikely that even better performances could be reached if the same network would try to learn various segmentation tasks simultaneously, instead of being trained in only one kind of problem. From this perspective, the results presented in this paper constitute a first step towards the development of a highly efficient universal segmentation engine.

ACKNOWLEDGMENT

This work was partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943 (READ Recognition and Enrichment of Archival Documents)

REFERENCES

- [1] 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*, pp. 3431–3440, 2015.
- [2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.
- [3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [7] F. Simistira, M. Bouillon, M. Seuret, M. Würsch, M. Alberti, R. Ingold, and M. Liwicki, "Icdar2017 competition on layout analysis for challenging medieval manuscripts," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 1361–1370, IEEE, 2017.
- [8] M. Diem, F. Kleber, S. Fiel, T. Gruning, and B. Gatos, "cbad: Icdar2017 competition on baseline detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 1355–1360, Nov. 2017.
- [9] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos, "Icdar2017 competition on document image binarization (dibco 2017)," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 1395–1403, IEEE, 2017.
- [10] K. Chen, M. Seuret, J. Hennebert, and R. Ingold, "Convolutional neural networks for page segmentation of historical document images," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 965–970, IEEE, 2017.
- [11] Y. Xu, W. He, F. Yin, and C.-L. Liu, "Page segmentation for historical handwritten documents using fully convolutional networks," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 541–546, IEEE, 2017.
- [12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.
- [13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.
- [14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [15] J. Serra, *Image analysis and mathematical morphology*. Academic Press, Inc., 1983.
- [16] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.
- [17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [18] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Advances in Neural Information Processing Systems*, pp. 1942–1950, 2017.
- [19] C. Tensmeyer, B. Davis, C. Wigington, I. Lee, and B. Barrett, "Pagenet: Page boundary extraction in historical handwritten documents," in *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, pp. 59–64, ACM, 2017.
- [20] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, pp. 309–314, ACM, 2004.
- [21] T. Gruning, R. Labahn, M. Diem, F. Kleber, and S. Fiel, "Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents," *arXiv preprint arXiv:1705.03311*, 2017.
- [22] 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*, pp. 471–476, IEEE, 2016.

- [23] F. Junker, “Extraction of ornaments from a large collection of books,” Master’s thesis, EPFL, 2017.