Fully Convolutional Neural Networks for Newspaper Article Segmentation

Benjamin Meier, Thilo Stadelmann, Jan Stampfli, Marek Arnold, and Mark Cieliebak
benjmeier@bluewin.ch, {stdm, stmf, arnd, ciel}@zhaw.ch
Zurich University of Applied Sciences, Winterthur, Switzerland

***Abstract*—Segmenting newspaper pages into articles that semantically belong together is a necessary prerequisite for article-based information retrieval on print media collections like e.g. archives and libraries. It is challenging due to vastly differing layouts of papers, various content types and different languages, but commercially very relevant for e.g. media monitoring.**

We present a semantic segmentation approach based on the visual appearance of each page. We apply a fully convolutional neural network (FCN) that we train in an end-to-end fashion to transform the input image into a segmentation mask in one pass. We show experimentally that the FCN performs very well: it outperforms a deep learning-based commercial solution by a large margin in terms of segmentation quality while in addition being computationally two orders of magnitude more efficient.

***Index Terms*—semantic segmentation, deep learning, CNN**

I. Introduction

Incredibly large volumes of information – news, opinion, reports etc. – fill the shelves of large media archives and libraries. Many of them come from classical print media (e.g. newspapers, magazines, and journals), and many more are added daily [1]. In order to open up these volumes to content-based information retrieval, digitized print media pages have to be segmented into their semantically connected components, i.e. articles. Only retrieval based on articles (instead of whole pages) allows for more complex types of queries that are related to a single artifact: restrictions to e.g. a certain author or other meta data, guaranteed co-occurrence of search terms in the same semantic context etc. Such use cases frequently occur for example in media monitoring, where copyright restrictions often even prohibit to send out entire pages, but only single articles.

The challenge in semantic segmentation of newspaper pages lies in the diversity of the medium: different "genres" of papers have vastly different layouts (weekly papers e.g. use different typography and imagery than tabloid press). Newspapers also cover different genres of content, from regular articles to ads and weather reports. Layout differs with time and culture, and different languages are used. Nevertheless, print segmentation cannot be neglected even in the presence of respective online media, because (a) web publications suffer from a similarly challenging segmentation problem due to ad placement, comment sections, dynamic content etc.; and (b) newspapers publish different content online and in print (even in case of dual publication, online articles may change afterwards). Print media thus needs to be searchable to include all information. Then, optical character recognition (OCR) based digitization is insufficient, and semantic level approaches to segmentation are necessary to identify single articles.

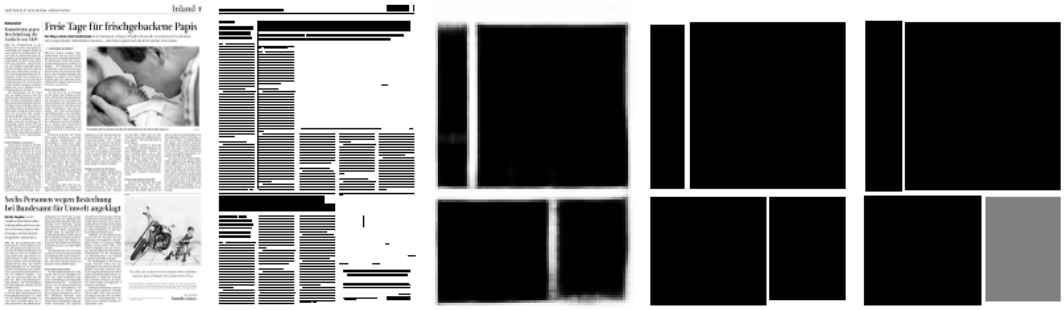
Based on the observation that humans generally appear able to detect the parts that belong to an article (including pictures) with high confidence even in languages they do not understand, several authors suggested newspaper segmentation systems based on visual clues alone. Section II gives an overview of relevant approaches and also reviews the influence of the recent success of deep learning methodologies in computer vision on the segmentation task. One such system is the state of the art commercial in-house solution of the media monitoring company *ARGUS DATA INSIGHTS Schweiz AG*, the industrial partner of this study. It classifies the pixels of a newspaper page into belonging either to an article or a border between articles, using a CNN [2]. We present this system together with the related work.

In this paper, we improve this system using a fully convolutional neural network (FCN) that transforms the input page into a complete segmentation mask in one pass [3]. We improve the segmentation quality of the baseline system by more than 46% on a dataset containing approximately $5,500$ pages from the largest newspapers of Switzerland. Our FCN approach also improves the computational performance by a factor of 30, i.e., the runtime to generate the segmentation mask is reduced to only 2.9% of the original runtime.

We outline our approach in Section III and explain its training process in Section IV before reporting on the experimental evaluation in Section V, giving all necessary details to make the presented work reproducible. Section VI concludes the paper with discussions and an outlook to future work.

II. Related Work

Palfray et al. [4] focus on the challenge of digitizing antique newspapers. Their approach not only performs segmentation but also extracts the reading order. The method uses a conditional random field (CRF) to perform pixel classification and works with an accuracy of 85.84% on respective pages, achieving state of the art results. However, this approach has only been tested on old newspapers and the article extraction is based on a rule set, which is less dynamic than a neural network. Our novel approach focuses on contemporary newspaper layouts and may also be used more widely. It is trained on a specific dataset of newspapers, but it is also possible to use another dataset to train the solution for other types of newspapers.



(a) original page (b) OCR result (c) raw output (d) postprocessed (e) ground truth

Fig. 1: Images (a) and (b) show input examples for our FCN. Image (c) shows the raw output of the network and (d) the detected polygons after postprocessing. Image (e) shows the actual labels (gray indicates an ad).

Gatos et al. [5] propose to segment newspaper pages into articles by first identifying lines in the layout, and then text and images. In a third step, headings are identified, and finally a set of rules is used to recognize the articles. The method reaches a recall of 75.20% and a precision of 77.15%. However, their rule set is much less dynamic than a trained neural net.

Following their success on the ImageNet 2D image classification task in 2012 [6], deep learning methods and especially convolutional neural networks (CNNs) have disrupted almost all areas of pattern recognition. They have been used e.g. for classification tasks on 3D images [7], for the analysis of text [8] and for audio data [9]. But even since the nineties they have been applied to document recognition by LeCun and his colleagues [10]. Noh et al. [11] generalized the approach to perform semantic segmentation in one pass by coupling a CNN architecture with subsequent deconvolution [12] and unpooling layers to create an output of the same dimensionality as the input. The FCN architecture by Long et al. [3] uses a similar idea, but performs upsampling exclusively with deconvolutional layers that combine their own information content with the finer resolution of details in the lower layer of identical dimension through element-wise addition. This yields the advantage of a very precise segmentation of finer structures without getting blurry.

Fakhry et al. [13] use a deconvolutional neural network for biological image segmentation. Their method uses unpooling layers, and the network is specifically designed to not lose any location information in the convolutional and pooling layers. This allows it to arrive at a very detailed segmentation. Their network achieves state of the art results, however, it is very memory intensive during training. They used an Nvidia K80 GPU with 24 GB memory, which was only able to hold a batch size of 15 pages. Our method requires less memory, which makes the training much easier.

Recurrent neural networks (RNNs) form another approach for semantic segmentation [14] [15]. Generally, RNNs can make better use of the global information for local classification. However, the training is computationally very expensive compared to FCNs.

The baseline approach used in this study is an industrial-strength in-house solution built on the work of Ciresan et al. [2]. It takes the image of a newspaper page as input and returns the segmentation map. The input is preprocessed using an OCR software to replace any detected text or image with a uniformly colored box ("label") of the the same size, and then rescaled to 100 pixels in height. Such an OCR-preprocessed page is shown in Figure (1b). The approach then slides a classification window ("patch") of size 25×25 pixels over every possible location of the input. Each patch is input to a CNN to classify its central pixel as belonging to an article or border. The network output is postprocessed to get polygons that represent articles. The Hough transform is used to obtain these polygons from the classification mask. The results are often of good quality (see Section V), and the system is in production use in the industrial partner's processes.

III. Proposed Model Architecure

The architecture of our proposed FCN is depicted in Figure (2), and each layer is described in Table I. The network begins with several convolutional and max-pooling layers. There are no densely connected layers in the model. The network is built of three logical parts. Initially, there is a feature extraction part that does the convolutions and the max-pooling. This part represents a standard CNN (up to layer `conv7-1`). The second part of the network performs upscaling and is trained to do the segmentation. These first two parts (up to layer `transposed_conv11-1`) are based on the FCN approach [3]. We augment them with a very small refinement network to correct some artifacts in the output of the second part. This is mainly useful because the expected output usually only contains rectangular objects. The refinement network considerably eases the post processing task while additionally decreases training time. Finally, a sigmoid layer is used to do a binary classification into article pixels (black) or background (white).

The network input has a shape of $256 \times 256 \times 2$ pixels: the raw image and its OCR-preprocessed version are fed into separate channels, each resized to 256 pixels height and width according to the original aspect ratio. The input

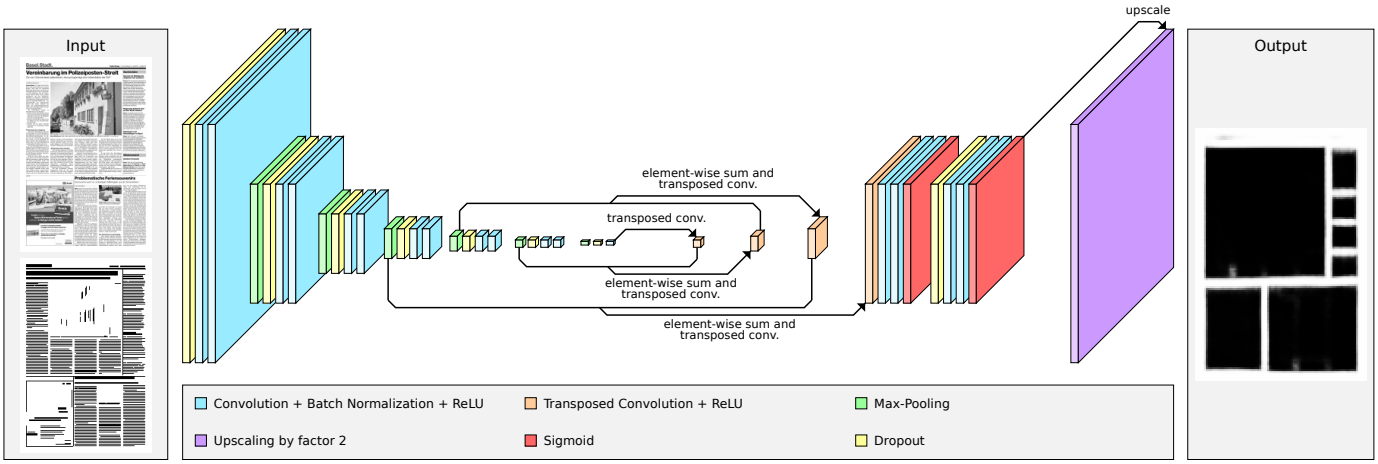


Fig. 2: The proposed network architecture (feature map count not shown): all parameters are described in Table I.

size is chosen to maximize the model's quality: we found that larger inputs do not increase the segmentation quality of newspaper articles. Grayscale pixel values are scaled to the range $[0, 1]$. The network inputs are shown in Figures (1a) and (1b). The OCR result is an abstract version of the page, indicating where text or images are found, and it also contains any horizontal or vertical line that appears in the plain newspaper page. Preliminary experiments indicate that the quality of the segmentation highly improves with the OCR input (irrespective of the actual OCR system used, as long as it marks texts, lines and images). An example OCR input is shown in Figure (1b).

Table I indicates that the width and height of the upscaling network is not equal to the width and height of the feature extraction network. For a pixel-wise classification it would be required that these two dimensions are identical. But the higher resolution is not required for segmentation, because the newspaper articles have a very coarse, often rectangular outline. This allows us to work internally with a lower resolution and still get highly accurate results, thus decreasing evaluation runtime, parameter count and training time. This is another advantage of the FCN architecture compared to a patch-based approach: fine details can be processed in the first layers, but subsequently lower resolutions of a page may be used internally. The patch-based approach requires to process each pixels if all information shall be used.

The general idea behind the novel refinement network is to automatically learn some simple postprocessings like correcting article borders to be rectangular. Therefore, the bottleneck layer `conv12-3-sigmoid` forces the network to compress and pre-classify any previous information. The network learns how to locally refine borders and remove small holes in the foreground. For our use case where we almost only have rectangular objects, this additional part of the network reduces the training time and also improves the output quality.

Finally, classification is performed on the lower resolution internal representation in layer `conv13-1-sigmoid` to save computational time before the we use simple upscaling by a factor of 2 to again match the result to the input resolution. The network produces values near to 1 if a pixel is classified as background/border and values near to 0 if a pixel is classified as foreground/article. Different threshold values have been tested to binarize this output. For the results presented later, we fixed the threshold at a low value of ≥ 0.35 to classify as background after extensive experiments. This takes into account that detecting article borders has priority for the subsequent post processing.

In general, the presented architecture may be used for any input image size in the range of $64n \times 64m$ with $n, m \in \mathbb{N}_{>0}$, because the complete network only contains max-pooling and convolutional layers.

IV. Training

We train and evaluate our approach on a data set curated for research purposes by *ARGUS DATA INSIGHTS Schweiz AG*. It consists of approximately $5,500$ newspaper pages of the most influential Swiss newspapers from the year 2016. All pages contain some labels by professional annotators (marked borders around articles), but only 426 pages are fully labeled. Figure (1e) illustrates a labeled page: while the gray labels would in principle allow us to treat advertisements separately, we do not distinguish between articles and ads in our approach. All images that are wrongly labeled or that have highly non-rectangular shapes, like the images shown in Figure (3), were removed from the training set. The resulting dataset improves the quality of the neural network. It contains $4,135$ pages. We use 80% of the partially labeled pages and 80% of fully labeled pages of those for training. The remaining 20% of the data is in the test set.

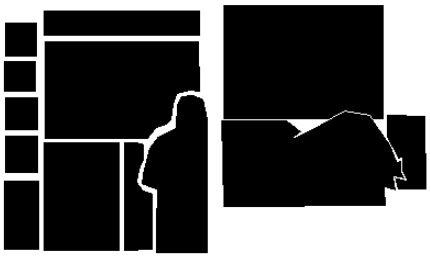
To learn from the partially labeled newspaper pages, we preprocess respective pages: the network would otherwise not be able to discriminate articles from background unless all articles are labeled. Thus, after resizing every page to the input size of our network, we change any pixel that is more than 3 pixels away from any article label to white (background), both in the plain input as well as in the OCR result. This

TABLE I: Layer details: `add` layers perform element-wise addition of their input and the given layer; p in `dropout` layers describes the probability of setting a value to 0.

	Name	Kernel size	Stride	Pad	Output size
Feature extraction					
	input	-	-	-	$256 \times 256 \times 2$
	dropout1 ($p = 0.3$)	-	-	-	$256 \times 256 \times 2$
	conv1-1	5×5	1	2	$256 \times 256 \times 32$
	conv1-2	3×3	1	1	$256 \times 256 \times 16$
	pool-1	2×2	2	0	$128 \times 128 \times 16$
	dropout2 ($p = 0.3$)	-	-	-	$128 \times 128 \times 16$
	conv2-1	5×5	1	2	$128 \times 128 \times 16$
	conv2-2	3×3	1	1	$128 \times 128 \times 16$
	pool2	2×2	2	0	$64 \times 64 \times 16$
	dropout3 ($p = 0.5$)	-	-	-	$64 \times 64 \times 16$
	conv3-1	3×3	1	1	$64 \times 64 \times 16$
	conv3-2	3×3	1	1	$64 \times 64 \times 16$
	pool3	2×2	2	0	$32 \times 32 \times 16$
	dropout4 ($p = 0.5$)	-	-	-	$32 \times 32 \times 16$
	conv4-1	3×3	1	1	$32 \times 32 \times 64$
	conv4-2	3×3	1	1	$32 \times 32 \times 64$
	pool4	2×2	2	0	$16 \times 16 \times 64$
	dropout5 ($p = 0.5$)	-	-	-	$16 \times 16 \times 64$
	conv5-1	3×3	1	1	$16 \times 16 \times 64$
	conv5-2	3×3	1	1	$16 \times 16 \times 128$
	pool-5	2×2	2	0	$8 \times 8 \times 128$
	dropout6 ($p = 0.3$)	-	-	-	$8 \times 8 \times 128$
	conv6-1	5×5	1	1	$8 \times 8 \times 128$
	conv6-2	3×3	1	1	$8 \times 8 \times 256$
	pool-6	2×2	2	0	$4 \times 4 \times 256$
	dropout7 ($p = 0.3$)	-	-	-	$4 \times 4 \times 256$
	conv7-1	5×5	1	1	$4 \times 4 \times 256$
Upscaling					
	transposed_conv8-1	2×2	2	0	$8 \times 8 \times 128$
	add8 (layer = pool5)	-	-	-	$8 \times 8 \times 128$
	transposed_conv9-1	2×2	2	0	$16 \times 16 \times 64$
	add9 (layer = pool4)	-	-	-	$16 \times 16 \times 64$
	transposed_conv10-1	2×2	2	0	$32 \times 32 \times 16$
	add10 (layer = pool3)	-	-	-	$32 \times 32 \times 16$
	transposed_conv11-1	4×4	4	0	$128 \times 128 \times 16$
Refinement					
	conv12-1	5×5	1	2	$128 \times 128 \times 32$
	conv12-2	5×5	1	2	$128 \times 128 \times 32$
	conv12-3-sigmoid	1×1	1	0	$128 \times 128 \times 8$
	dropout12 ($p = 0.3$)	-	-	-	$128 \times 128 \times 8$
	conv12-4	5×5	1	2	$128 \times 128 \times 32$
	conv12-5	3×3	1	1	$128 \times 128 \times 16$
Classification					
	conv13-1-sigmoid	1×1	1	1	$128 \times 128 \times 1$
	upscale13 (factor = 2)	-	-	-	$256 \times 256 \times 1$
	output	-	-	-	$256 \times 256 \times 1$

removes any non-labeled content from the input page. The 3 pixel border is very important for the neural network to learn how borders of articles look like (they may e.g. contain very helpful straight lines between articles, see for example the long vertical and horizontal lines in Figure (1b)). On the other hand, these 3 pixel borders add some noise by sometimes containing parts of other articles or images. Learning to ignore these additional parts would not always be correct. Another problem is that many newspapers use text and logos in their headers and footers, but such content does not belong to any article and therefore is removed in this step. This means the neural network cannot learn the concept of headers and footers using the partially labeled data.

With the given labeling, borders between articles may be as small as 1 pixel, which is not optimal for training. We thus shrink all labels by 2 pixels, which makes the minimum border size equal to 5 pixels. This makes the training and especially the postprocessing much easier: while the FCN architecture generally allows to produce a fine output, 1 pixel lines are not always clearly detected. The shrinked article labels also enable us to use the lower-resolution internal representation as the basis to perform the final segmentation on.



(a) (b)

Fig. 3: Two examples of non-rectangular labels.

To increase the variability of the training examples, the newspaper pages are scaled down to $n \times 256$ pixels, and then the content is placed at a random x position within the 256×256 target frame (note that pages are usually higher than wide, so that translation on the x axis is possible without loosing content). The newspaper pages are also randomly mirrored along the y axis with a probability of $p = 0.5$.

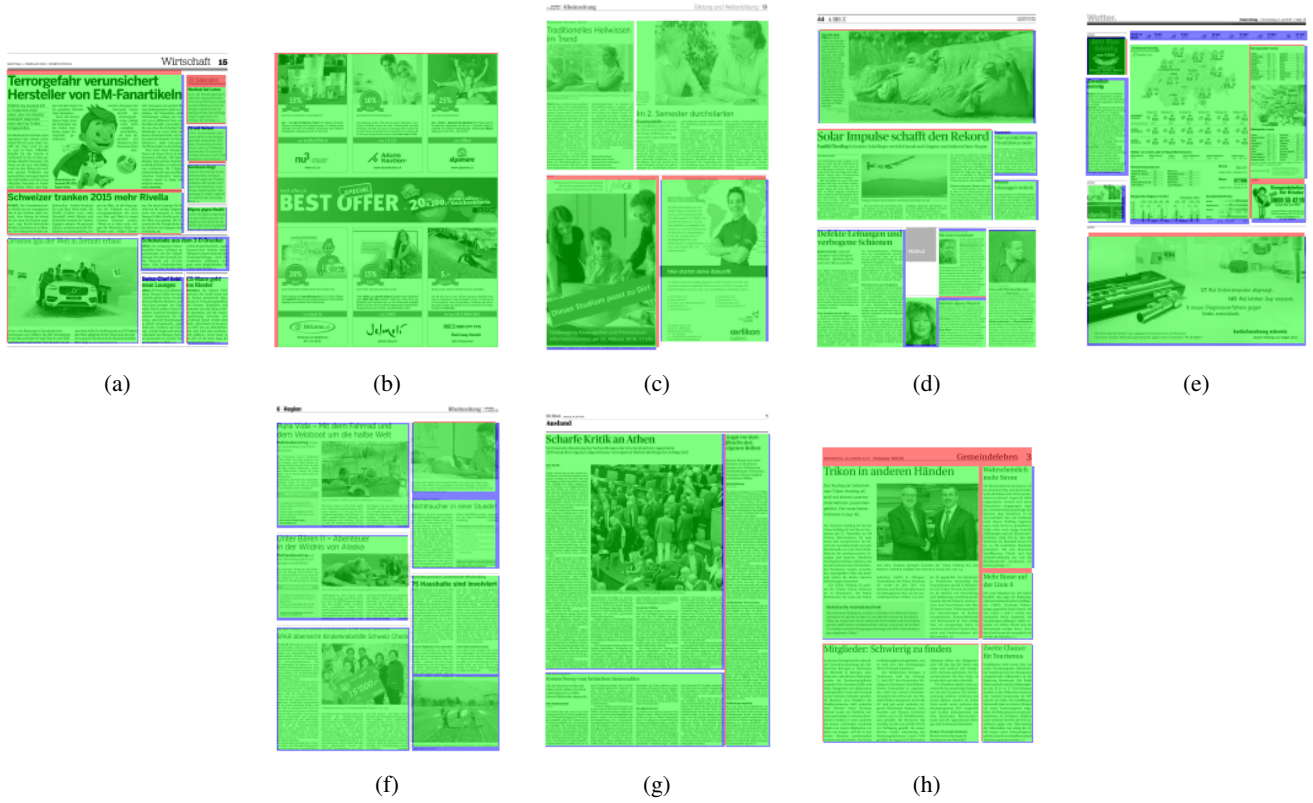
We attach different costs for the two types of error within the cross-entropy loss function: as the borders between the articles are small in terms of the number of involved pixels, but constitute the most important part, it is most important to classify them correctly. Thus pixels that are wrongly classified as article are weighted by a factor of 1.8; all other classification errors have a weight of 1.0.

To make the training more efficient and to avoid overfitting, the network makes heavy use of dropout regularization [16]. It is first used directly after the input to add some noise. The network also uses L_2 regularization with a weight decay of 0.0001. Due to the internal covariate shift problem, batch normalization is used [17] for every convolutional layer, which allows the network to converge to a better optimum. Nesterov momentum is used for optimization with a learning rate of 0.01. The batch size is 16. The FCN architecture has the advantage that end-to-end training is possible, i.e. all aspects are optimized by the same training loop. The training is therefore coneptually simple and also efficient.

P. Luc et al. [18] proposed a network for semantic segmentation that uses an adversarial network [19] for regularization. We evaluated this regularization approach in preliminary experiments together with the other design choices and parameters mentioned above. The training time increased because of the additional discriminator network, but no improvement in quality could be observed. The GAN approach is thus not included in our final architecture.

V. Experimental results

Our experiments use a Geforce GTX 780 GPU to train the network. The computer has 128 GB of RAM and an Intel Xeon E5-2620 v2 CPU, but neither training nor the classification



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

(f) (g) (h)

Fig. 4: Examples of input images that are used for the benchmark. The green area shows the correctly classified article pixels, the blue area the non-detected article pixels and the red area the pixels that are classified as article but should be background.

TABLE II: Detailed benchmark results for the 8 images shown in Figure (4). The best results are in **bold**.

Approach	Image	DER score	Completeness score	Classification time $[s]$
FCN	a	**0.1888**	**0.3000**	**0.0550**
Baseline	a	0.7244	0.0000	1.8310
FCN	b	**0.0183**	**1.0000**	**0.0460**
Baseline	b	0.0986	**1.0000**	1.8730
FCN	c	**0.0236**	**0.7500**	**0.0440**
Baseline	c	0.3729	0.5000	1.5130
FCN	d	**0.0744**	**0.8750**	**0.0470**
Baseline	d	0.5573	0.1250	1.6150
FCN	e	**0.2249**	**0.5000**	**0.0530**
Baseline	e	0.5328	0.0000	1.6070
FCN	f	0.2388	**0.4000**	**0.0440**
Baseline	f	**0.2203**	**0.4000**	1.5260
FCN	g	**0.0338**	**1.0000**	**0.0470**
Baseline	g	0.4158	0.0000	1.6700
FCN	h	0.2550	0.4000	**0.0440**
Baseline	h	**0.1150**	**0.6000**	1.5140

processes require that much memory. The training is done in two stages. First, we train using the full training set, and all input images are preprocessed as described in Section IV, for 210 epochs. Then, the dataset is limited to only the fully labeled pages. The neural network now has the possibility to learn how to deal with headers and footers. They are not part of any article and should be ignored for the segmentation process. The training on the fully labeled data is done for further 150 epochs. After these two stages, the network starts overfitting, and training thus is ended.

Our new approach is compared with the baseline approach described in Section (II) with regard to computational efficiency and segmentation quality. The required training time of our system with its $1,435,065$ parameters is ca. 5 h, whereas the baseline approach only requires 1.25 h with $252,706$ parameters. However, our model computes the complete classification of a newspaper page in about 47.6 ms on average, while the baseline approach, having to execute the network for each pixel, requires about $1,655$ ms on average for one page (see Table III).

To compare the results of the baseline approach and the new implementation, we use two scores. The diarization error rate (DER) [20] score, which is known from speaker diarization, is adopted for our use case. The best value is 0 and higher values are worse. A second score, which is called "completeness score", was defined by the industrial partner and reflects how the segmentations looks like for an end-user. It is defined as the fraction of almost perfect matches of article polygons to article labels, where 1 is best and 0 is the worst.

Figure (4) shows some test images. The scores for these images are listed in Table II. Table III summarizes the results for all 81 fully-labeled newspaper pages in the test set. Our new approach outperforms the baseline approach according to the DER score, to the completeness score, and also to the

TABLE III: Benchmark results for the fully labeled part of the test set (81 images). The best results are in **bold**.

Metric ↓, approach →	FCN	Baseline
Avg. DER score	**0.1378**	0.2976
Avg. completeness score	**0.5444**	0.2079
Min. DER score	0.0051	**0.0000**
Max. DER score	**0.5920**	1.0028
Min. completeness score	**0.0000**	**0.0000**
Max. completeness score	**1.0000**	**1.0000**
DER score σ	0.1343	0.2265
completeness score σ	0.3464	0.3011
Avg. classification time [s]	**0.0476**	1.655

required processing time.

VI. Conclusions and Future Work

We presented a FCN-based approach to learn newspaper article segmentation. To demonstrate the improvements with this architecture, we compared it with a proprietary CNN-based system that is in productive use at *ARGUS DATA INSIGHTS Schweiz AG* for media monitoring. We showed experimentally that the segmentation of newspaper articles works very well and also very efficiently with FCNs. Specifically, we improved the average segmentation quality as measured by DER by 46.3%, and the perceptual completeness score even by a factor of 2.62. At the same time, we improved the classification runtime per page by a factor of 34.8 to only a few milliseconds. Furthermore, our model architecture can cope with inputs of differing sizes without retraining or input scaling.

Currently, we mostly detect articles of rectangular shape. In future work this could be extended to arbitrary shapes: Montoya-Zegarra et al. [21] propose a method for multi-class semantic segmentation of urban areas, reaching high accuracy. The method might be modified and used for newspaper article segmentation, as well (articles correspond to buildings and article borders correspond to streets, thus capturing also arbitrary shapes). Future work might also differentiate between the found article types, e.g. articles and advertisements. We are also currently working on a text-based approach which takes the content of the article into account to reach a better segmentation quality. A fully operational system should combine both visual and textual clues for segmentation (as humans do).

Acknowledgments

This research was funded by CTI grant 17719.1 PFES-ES *PANOPTES*. We are grateful for the data set and benchmark provided by our industrial partner *ARGUS DATA INSIGHTS Schweiz AG*.

References

[1] R. Meyer. (2016) How many stories do newspapers publish per day? [Online]. Available: https://www.theatlantic.com/technology/archive/2016/05/how-many-stories-do-newspapers-publish-per-day/483845/

[2] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[3] 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.

[4] T. Palfray, D. Hebert, S. Nicolas, P. Tranouez, and T. Paquet, "Logical segmentation for article extraction in digitized old newspapers," in *Proceedings of the 2012 ACM symposium on Document engineering*. ACM, 2012, pp. 129–132.

[5] B. Gatos, S. Mantzaris, K. Chandrinos, A. Tsigris, and S. J. Perantonis, "Integrated algorithms for newspaper page decomposition and article tracking," in *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*. IEEE, 1999, pp. 559–562.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3d data," *arXiv preprint arXiv:1604.03265*, 2016.

[8] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[9] Y. Lukic, C. Vogt, O. Dürr, and T. Stadelmann, "Speaker identification and clustering using convolutional neural networks," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*. IEEE, 2016, pp. 1–6.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[12] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[13] A. Fakhry, T. Zeng, and S. Ji, "Residual deconvolutional networks for brain electron microscopy image segmentation," *IEEE Transactions on Medical Imaging*, 2016.

[14] B. Wonmin, T. Breuel, F. Raue, and M. Liwicki, "Scene labeling with lstm recurrent neural networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 3, 2015.

[15] P. H. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling." in *ICML*, 2014, pp. 82–90.

[16] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[18] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[20] M. Kotti, V. Moschou, and C. Kotropoulos, "Speaker segmentation and clustering," *Signal processing*, vol. 88, no. 5, pp. 1091–1124, 2008.

[21] J. Montoya-Zegarra, J. Wegner, L. Ladickỳ, and K. Schindler, "Semantic segmentation of aerial images in urban areas with class-specific higher-order cliques," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 3, p. 127, 2015.