Historical handwritten document segmentation by using a weighted loss

Historical handwritten document segmentation by using a weighted loss

Samuele Capobianco, Leonardo Scommegna, and Simone Marinai

University of Florence, via di Santa Marta, 3, Firenze, Italy
{samuele.capobianco,simone.marinai}@unifi.it
leonardo.scommegna@stud.unifi.it

Abstract. In this work we propose one deep architecture to identify text and not-text regions in historical handwritten documents. In particular we adopt the U-net architecture in combination with a suitable weighted loss function in order to put more emphasis on most critical areas.We define one weighted map to balance the pixel frequency among classes and to guide the training with local prior rules. In the experiments we evaluate the performance of the U-net architecture and of the weighted training on one benchmark dataset. We obtain good results using global metrics improving global and local classification scores.

Keywords: Convolutional Neural Networks · Page Segmentation · Loss Functions

1 Introduction

Understanding handwritten historical documents is a challenging task that includes several sub-problems. One of the first steps is to segment and extract text lines which could be recognized in subsequent phases to understand the document content. The layout analysis of handwritten documents can be very difficult, because of the variable layout structure, the presence of decorations, different writing styles and degradations due to the aging of the document. In the last years different techniques have been proposed to address this task [3, 10, 13]. In particular, to extract text lines from handwritten documents we can consider two types of related problems. Considering the page segmentation task the target is to split a document image into regions of interest [3]. On the other hand the text line extraction stage allows to localize and extract the text lines directly from the document image [1]. These approaches extract regions of interest which are considered as text lines and often provide similar results when considering handwritten documents. To clarify the goal of text extraction from historical document we show in Figure 1 one example from one benchmark dataset together with the ground truth of the page.

Among several solutions proposed to solve this task some use assumptions to simplify the approach. In [8] the authors assume that for each text line there is one path from one side of the image to the other that crosses only one text

line. Based on this assumption, they trace the text line after the blurred image transformation extracting directly the text lines.

Later, it has been proposed another solution [5] where the authors are able to extract text line from handwritten pages using Hough transform and the page structure as prior knowledge .

In the last years, many different CNN architectures have been presented to solve several computer vision tasks.One important task is the semantic segmentation of images whose goal is to classify pixels from different categories and subsequently to extract homogeneous regions. One interesting solution adopts Fully Convolutional Networks [12] composed only by convolution and pooling operations used to learn representations based on local spatial input to compute pixel-wise predictions. The FCNs with respect to CNNs architectures do not use fully connected layers and use upsampling layers as deconvolution operations.

In this paper we address the page segmentation using one Fully Convolutional Network with a weighting of the pixels used to compute the training loss designed to address our task. In this way, we aim at classifying with better results some areas of the image that are more critical to perform the text line extraction, without using dedicated post processing techniques. The main contributions of this paper are the use of the FCN to perform text segmentation and the design of the weighting schema.

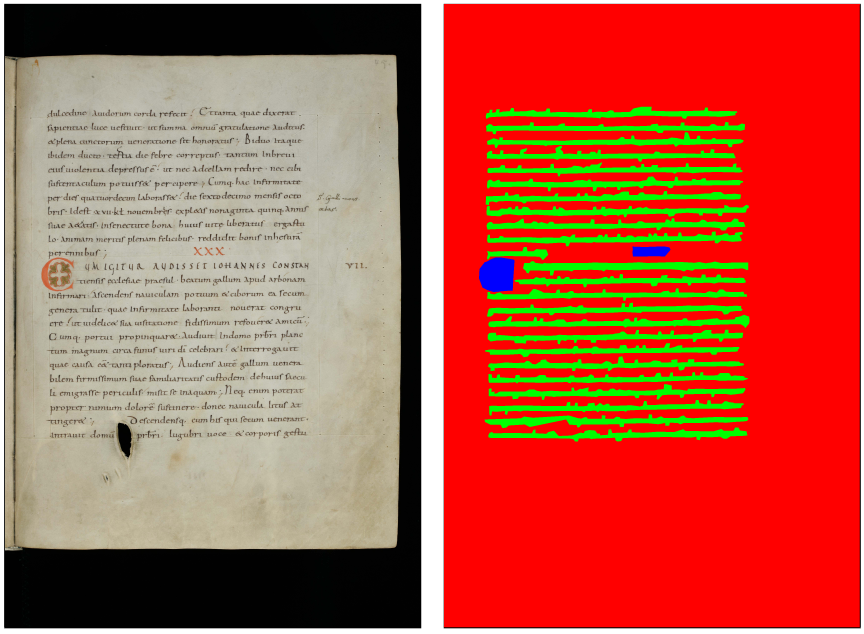
The rest of the paper is organized as follows. In Section 2 a brief review of related work in the fields of semantic and page segmentation tasks. Then, in Section 3 we describe the architecture used to perform page segmentation. The proposed weighting is presented in Section 4. Experimental results are discussed in Section 5 and concluding remarks are in Section 6.

2 Related works

In the field of document analysis, page segmentation task has gained a lot of attentions during the time. Several solutions use artificial neural networks as well as Convolutional Neural Networks which have been applied successfully to this task showing best results compared to handcrafted features solutions ([9], [3]). In the work [9] the authors use CNNs to extract text lines from historical documents classifying the central pixel from extracted image patch. After one suitable post-processing phase, using the watershed transform, it is possible to extract the text lines and also provide a page segmentation.

Instead, in [3] the authors propose to use a Convolution Networks for the pixel labeling task. Using a superpixel algorithm to extract coherent patches, they are able to perform page segmentation using a trained CNN model to predict the semantic class for each extracted patch.

We recently proposed one solution [1] to detect text lines on the basis of the assumption that for each text line it is possible to define one separator line and one median line. The median line is the middle line between the top profile of the text and the bottom profile, while the separator line is the middle line



(a) Original page (b) Original ground truth

Fig. 1: One document example from the dataset and its ground truth image. The background area is in red, decoration in blue and text in green.

between two consecutive median lines in the text area. In [1] we used one suitable Convolutional Network to separate the text line areas from the document background.

Fully Convolutional Networks are largely adopted in semantic segmentation field. One adapted version of FCN named *U-net* [11] has been applied to biomedical image segmentation outperforms existing methods for cell tracking challenge by a large margin. Always in the biomedical research area, it has been proposed a novel deep contour-aware network [2] to solve the gland segmentation task. This model is able to segment gland and separate the clustered objects into individual ones training a unified multi-task learning framework.

In scene parsing, a novel approach [15] has been proposed to solve the task. The authors propose pyramid scene parsing network which is able to merge together local and global representation defining a pyramid pooling module, in this way they prove as a global prior representation could improve the final segmentation result.

A Fully Convolution Network has been also used for page segmentation [14] where the FCN is used to provide a pixel-wise classification followed by post processing techniques to split a document image into regions of interest. The main focus is not the extraction of text lines, but the pixel classification and therefore the metrics used are related to semantic segmentation.

Recently, a very challenging dataset has been introduced by [7] where the authors evaluate various end-to-end FCN architectures to segment handwritten annotation in historical documents.

3 Network architecture

In this work we address the page segmentation using one Fully Convolution Network that is trained directly on document image (large) patches to learn a pixel-wise classification model which is able to segment different regions of interest in the input image. The documents addressed in our experiments have three different semantic classes: background, text and decoration. An example of one input image with its ground truth can be seen in Figure 1.

Several architectures have been proposed to address the semantic segmentation. One model that gained attention in biomedical image segmentation is the *U-net* [11]. In this work we propose a neural network which is strongly inspired by the *U-net* model. By inspecting the architecture in Figure 2 we can notice the U-shaped model where the first part consists in a contracting path and the second consists in an expansive path.

The contracting path consists of many encoding operations composed by convolution operators with kernel 3×3, stride 1, and max-pooling operator with kernel 2×2 stride 2, respectively. In this way the model is able to learn a data representation based on many local transformations computed by sequential convolution and pooling operations. In particular, for each transformation layer, we have two convolution operation followed by a pooling operation. The number of filters for each transformation layer is variable and we adapted these values to our problem. In particular, in the first layer we have 32, in the second 64, in the third 128, in the forth 256, and in the last 512 filters.

The expansive path consists of several decoding operations composed by upsampling and convolution operators. Having a look to Figure 2, for each decoding step the features are concatenated with the computed feature maps from the contracting path (with the same shape). Still in the same decoding layer two convolution operations with kernel 3×3 and stride 1 are applied to the previously computed features. The expansive path proposes the same number of filters for each decoding layer, but in reverse order with respect to the contracting path. All the convolution operators use Rectified Linear Units (ReLUs) as activation function. In the final layer one single 1×1 convolution linear operator is used to map the last features into the number of desired output channels.

In order to map the features into a classification score we use the Softmax operator to predict the probability score related to the semantic segmentation. In particular, we compute pixel-wise classification scores to determine a class for each input pixel. In the basic approach we use the cross-entropy loss function to train the model from random weights initialized using the technique proposed by [6]. This loss function is then modified in order to take into account the peculiarities of the problem addressed in this paper.

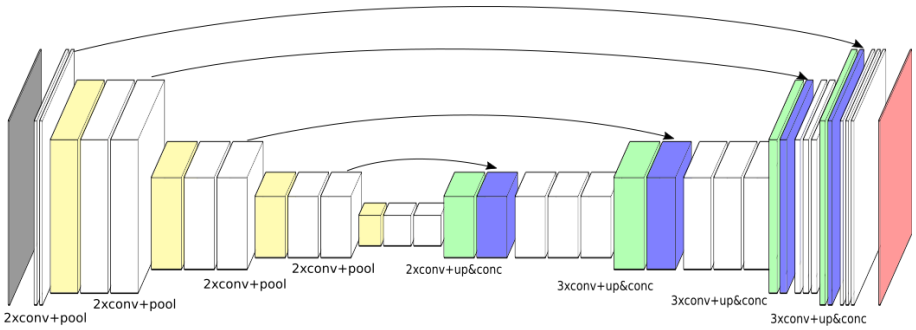


Fig. 2: The model architecture. Different transformations are depicted in different colors. The Input Layer is identified in gray, Convolutional Layers in white, Max Pooling in yellow, green for Upsampling Layers, blue for Combination Layers and red for the Softmax.

To build the training set we randomly crop several patches with a fixed shape from each document image. To maximize the differences between training patches the maximum overlap between patches is set to 25%. Like in [1] during the test phase, we systematically extract document patches from the input image with an overlap of 50%. For each pixel the final prediction is the average of the probability scores computed by the neural network for all the overlapping probability maps as illustrated in Figure 3.

4 Weighting the loss

In page segmentation there are several issues which make it difficult to obtain good performances. One significant problem is the unbalanced pixel class distribution. Having a look to Figure 1 we can see that the pixel distribution is highly unbalanced for background pixels with respect to the foreground pixels (considering foreground as text and decoration parts). We can notice also that some background pixels are very important to segment text lines. Often the text lines are very close to each other and in this case some misclassification errors of pixels between two text lines could give rise to significant problems for properly segmenting contiguous text lines.

The model is trained using a categorical cross entropy. One possibility to give different values to the input during the training is to add one weighted map to the loss function

$$WCE = -\sum_{x \in \Phi} w(x) \log p_{q(x)}(x) \tag{1}$$

where $\Phi \subset \mathbb{Z}^2$ is the set of pixel positions, $q : \Phi \to 1, \ldots, K$ maps input pixels to the class label of the predicted distribution p (K is the number of classes), $w : \Phi \to \mathbb{R}^+$ is the weight function that maps each pixel x to a suitable weight.

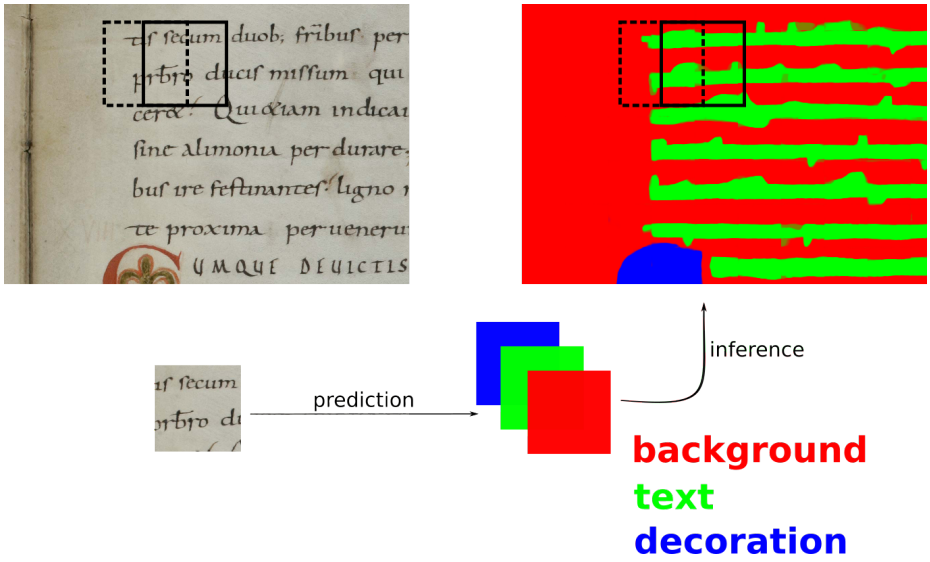


Fig. 3: Moving a sliding window over the input image, we compute a pixel-wise classification score for each patch. The results are combined by averaging the scores of overlapping patches.

Considering Equation 1, we define a weighted map function $w(x)$ which assigns a cost to each pixel considering the class frequency and the contribution which could provide in the segmentation task. In particular, considering the set of pixels Φ in the training mini-batch that are used to compute the loss function, we define a weight map to balance the class frequency and also to put more attention in specific areas which are useful to segment different regions properly. The weight map therefore includes two aspects of the document, the background and foreground areas. Formally, the weighted map assigns to a pixel x one weight balancing the pixel class frequency with a factor α and managing the background pixels with a predefined weight mask $\beta(x)$ (described in Section 4.1), such that:

$$w(x) = \begin{cases} \alpha & x \in \Phi_f \\ \beta(x) & x \in \Phi_b \end{cases} \tag{2}$$

where the foreground pixels $\Phi_f \subset \Phi$ represent the text and the decoration areas, while $\Phi_b \subset \Phi$ represent the background pixels.

Considering Φ the set of pixels for each mini-batch, the foreground pixel frequency is a variable number (usually $|\Phi_f| < |\Phi_b|$). In order to balance the foreground areas we apply a factor α as $\frac{|\Phi_b|}{|\Phi_f|}$ computed for each mini-batch. Having a pixel weight related to the class frequency we can balance the loss function improving the training.

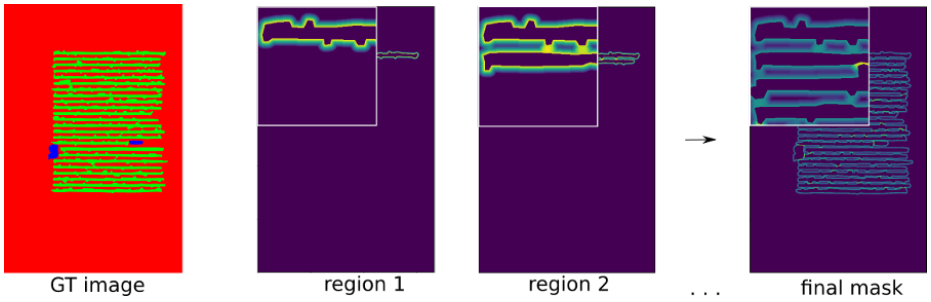


Fig. 4: Creation of the weighted mask of a ground truth page. In the third image, after merging the GT for region 1 and region 2 we can see pixels closer to both regions give a larger contribution to the weight mask.

As previously mentioned, not all background pixels have the same importance with respect to the overall performance. In particular, misclassification errors between contiguous text lines could give rise to improper segmentation of the text lines. To address this problem, we define one training rule weighting more the background pixels between different regions (text lines or decorations). This topological constraint is a rule which could be defined directly into the weighted map (Equation 2) defining a weighed mask $\beta(x)$ for the background area as described in the following.

4.1 Weighting background pixels

The weighted mask $\beta(x)$ gives more emphasis on background pixels considering the distance between two contiguous lines. The background pixels have a classification cost inversely proportional to the distance between two contiguous text lines. To this purpose, the weight mask assigns to each background pixel one value considering the distance to the nearest line (a larger distance gives a smaller value and vice versa). For the others background pixels, the weight mask $\beta(x)$ returns a fixed (neutral) weight value.

To compute the weight mask $\beta(x)$ we first transform the ground truth image from three class to a two class representation by merging text and decorator as foreground and the rest are background pixels. Considering this representation, taking a text region per time, we compute the *distance transform* which designs level curves from the region borders to a defined maximum distance d. An example of these level curves is shown in Figure 4 (region one) where the level curve value (in false colors) decreases when increasing the distance from the region border.

These level curves encode one information useful to consider the distance to the nearest regions. Iteratively, computing a level curve for each region and summing-up these values we can produce an overall weight mask. In this way, when the regions are close each other, the level curves are summed providing a larger value when the regions are closer. The largest value is obtained when

the distance between two regions is only one pixel. We force the range of values for the level curves to be between 0 and 1. By using a factor λ to multiply α (Section 4) we obtain mask values larger than foreground weights.

Considering a binary representation I of the ground truth image, for each region r_i at time i, we compute the level curves on the basis of the distance transform $dist_d(r_i)$ limiting this representation until a max distance d. We can consider the area around all the region borders with a maximum distance d as a dilation operation with kernel d.

In this way, the mask for an image with N regions is:

$$\beta(x) = \begin{cases} 1 + \frac{\lambda\alpha}{2d} \cdot (\sum_i^N dist_d(r_i)) & x \in dilate_d(I) \\ 1 & otherwise \end{cases} \tag{3}$$

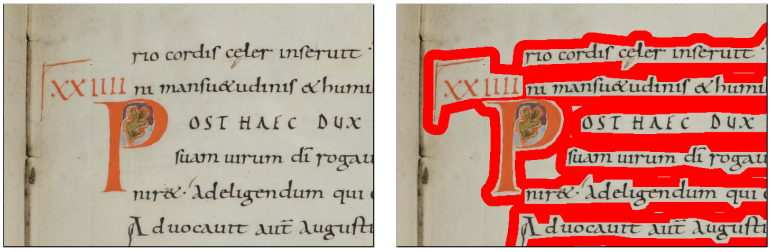
where $dilate_d$ is the morphological dilation operator with kernel d useful to consider the area where the weight mask has a variable number. For the remaining pixels in the page the weight mask maps pixels to a neutral value.

We illustrate in Figure 4 the approach to compute the weight mask. Starting from a ground truth image we compute a binary representation with foreground regions and background. For each region, we compute the distance curve levels as $dist(r_i)$ which are sequentially summed with the next region representations. The final result is the computed mask for all the pixels $x \in dilate_d(I)$ which are the critical pixels where we want to put more emphasis during the training to learn background representation. To provide a better idea about the critical pixels, in Figure 5 we highlighted in red the critical pixel areas.

5 Experiments

In this section we describe the experiments performed to test the proposed model to segment historical document images. The tests have been made on the *Saint Gall* dataset that consists of handwritten manuscript images that contain the hagiography *Vita sancti Galli* by Walafrid Strab. The manuscript has been most likely written by one single hand in Carolingian script with ink on parchment. Carolingian minuscules are predominant, but there are also some upper script letters that emphasize the structure of the text and some richly ornamented initials. Each page is written in a single column that contains 24 text lines. Altogether, the Saint Gall database includes 60 manuscript pages [4]. The database is freely downloadable and it is provided with layout descriptions in XML format. The document images in the original dataset have an average size of 3328×4992 pixels.

We evaluate the model performance using four metrics applied to semantic segmentation in previous works [**?**]. These measures are based on pixel accuracy and region intersection over union (IU). In particular, we evaluate the performance using: pixel accuracy, mean pixel accuracy, mean IU, and frequency weighted IU (f.w. IoU).



(a) input (b) crical pixels

Fig. 5: Given a input page, we can define the critical pixel areas (red) around the semantic regions found into the ground truth representation.

Let n_{ij} be the number of pixels of class i predicted to belong to class j (in total there are n_{cl} classes), and $t_i = \sum_j n_{ij}$ be the total number of pixels of class i. We can express the measures as:

– Pixel accuracy

$$pix.acc. = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{4}$$

– Mean accuracy

$$mean.acc = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i} \tag{5}$$

– Mean IoU (Intersection over Union)

$$mean.IoU = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i + \sum_j (n_{ji} - n_{ii})} \tag{6}$$

– Frequency weighted IoU

$$f.w.IoU = \frac{1}{\sum_k t_k} \sum_i \frac{t_i n_{ii}}{t_i + \sum_j (n_{ji} - n_{ii})} \tag{7}$$

The previous metrics are used to define a global evaluation for whole pages. To better evaluate the performance, we also define one local pixel accuracy considering only the area around the foreground regions. In Figure 5 we depict in red the area around foreground regions where the local pixel accuracy is computed. This area is important to extract text lines because misclassification pixels in it could give rise to a wrong layout analysis.

In the experiments we trained the proposed model using the patches extracted from the original training pages. The training dataset if composed by several patches of size 256×256 pixels randomly extracted from the input pages. Overall the training dataset contains $299,756$ patches. The different methods are compared evaluating the models on the test set pages.

In Table 1 we report the results for page segmentation on the *Saint gall* dataset and compare with previous results on the same dataset reported by

Model	Loss	pix.acc.	mean.acc.	mean.IoU	f.w.IoU
Baseline	CE Loss	98.07	95.12	90.80	96.27
Baseline/BN	CE Loss	98.07	94.47	90.85	96.25
VGG-FCN8s	CE Loss	98.09	95.11	91.03	96.31
Baseline	WCE Loss	98.03	94.75	90.88	96.18
Chen et al. [3]	98	90	87	96	

Table 1: Results for overall performance measures.

Model	Loss	Critical Pixel accuracy	DR	RA	FM
Baseline	CE Loss	95.65	77.81	83.48	80.55
Baseline/BN	CE Loss	96.42	83.08	85.57	84.31
VGG-FCN8s	CE Loss	95.07	67.27	79.64	72.93
Baseline	WCE Loss	96.25	81.28	86.18	83.65
Baseline/BN	WCE Loss	96.53	85.71	89.44	87.54

Table 2: Results for critical pixel classification.

Chen et al. [3]. The proposed model obtains good results with respect to [3] also by using the standard cross entropy loss. We improve only the mean IoU by using the proposed weighted loss. These metrics evaluate the page segmentation globally, but as we previously mentioned some misclassification errors have more importance in the final segmentation results.

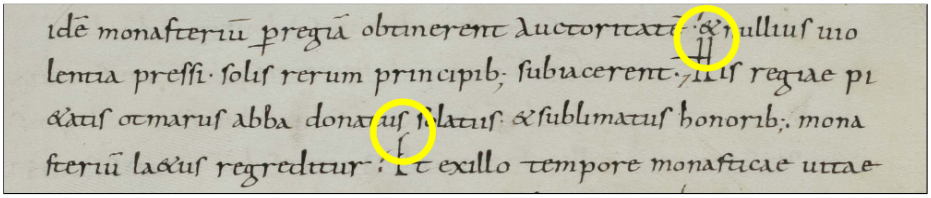
The results reported in Table 2 detail the critical pixel accuracy. This measure is useful to evaluate the model behavior after the training done using different losses. Using the weighted loss we can obtain better results which could be useful to extract text line directly after the page segmentation.

For a qualitative evaluation of results we show in Figure 6 one part of one page and two results, one from a model trained with cross entropy loss and the other from a model trained with the proposed weighted loss. We can notice that the model trained with the weighted map is able to better segment different text lines.

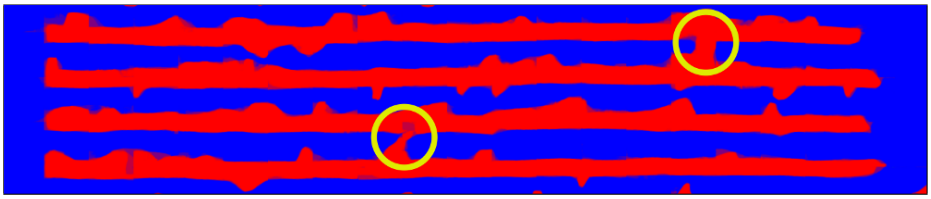
In order to evaluate the trained models with the measure proposed by [9], the model trained by weighted loss obtains Detection Rate (DR) and Recognition Accuracy (RA) respectively better than the model trained by cross entropy loss. Comparing these scores we considerably extract more accurate text lines using the proposed approach.

6 Conclusions

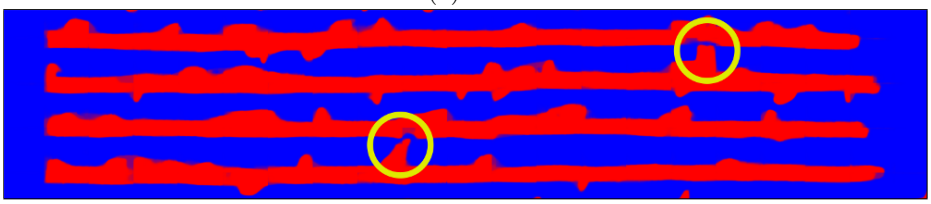
In this work we addressed the segmentation of handwritten historical documents by means of deep architectures. We presented one approach to weight a cross entropy loss to improve the results in particular in critical regions. By weighting



(a) Input



(b) CE



(c) Weighted CE

Fig. 6: Different results obtained with one model trained using cross entropy loss and one with weighted cross entropy loss,

the pixels to obtain a balanced loss and putting more emphasis on the background pixel around text lines, we obtained better classification results more suitable to extract text line as a post-processing of the classification based on neural networks.

In the future research we will from one side address more challenging datasets and from the other side we will exploit the pixel classification produced by the model discussed in this paper. In particular, we will compare the performance using other FCNs architecture to explicitely extract the text-lines from document pages.

References

1. Capobianco, S., Marinai, S.: Text line extraction in handwritten historical documents. In: Digital Libraries and Archives - 13th Italian Research Conference on Digital Libraries, IRCDL 2017, Modena, Italy, January 26-27, 2017, Revised Selected Papers. pp. 68–79 (2017)
2. Chen, H., Qi, X., Yu, L., Dou, Q., Qin, J., Heng, P.: DCAN: deep contour-aware networks for object instance segmentation from histology images. Medical Image Analysis **36**, 135–146 (2017)

3. Chen, K., Seuret, M., Hennebert, J., Ingold, R.: Convolutional neural networks for page segmentation of historical document images. In: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017. pp. 965–970 (2017)
4. Fischer, A., Frinken, V., Fornés, A., Bunke, H.: Transcription alignment of latin manuscripts using hidden markov models. In: Proceedings of the 2011 Workshop on Historical Document Imaging and Processing. pp. 29–36. HIP '11, ACM, New York, NY, USA (2011)
5. Gatos, B., Louloudis, G., Stamatopoulos, N.: Segmentation of historical handwritten documents into text zones and text lines. In: 14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, Crete, Greece, September 1-4, 2014. pp. 464–469 (2014)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics (2010)
7. Kölsch, A., Mishra, A., Varshneya, S., Liwicki, M.: Recognizing challenging handwritten annotations with fully convolutional networks. CoRR **abs/1804.00236** (2018)
8. Nicolaou, A., Gatos, B.: Handwritten text line segmentation by shredding text into its lines. In: 10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009. pp. 626–630 (2009)
9. Pastor-Pellicer, J., Afzal, M.Z., Liwicki, M., Castro-Bleda, M.J.: Complete system for text line extraction using convolutional neural networks and watershed transform. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS). pp. 30–35 (April 2016)
10. Renton, G., Chatelain, C., Adam, S., Kermorvant, C., Paquet, T.: Handwritten text line segmentation using fully convolutional network. In: First Workshop of Machine Learning, 14th IAPR International Conference on Document Analysis and Recognition, WML@ICDAR 2017, Kyoto, Japan, November 9-15, 2017. pp. 5–9 (2017)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III. pp. 234–241 (2015)
12. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (Apr 2017)
13. Vo, Q.N., Kim, S., Yang, H.J., Lee, G.: Text line segmentation using a fully convolutional network in handwritten document images. IET Image Processing **12**(3), 438–446 (2018)
14. Xu, Y., He, W., Yin, F., Liu, C.: Page segmentation for historical handwritten documents using fully convolutional networks. In: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017. pp. 541–546 (2017)
15. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 6230–6239 (2017)