dhSegment: A generic deep-learning approach for document segmentation
(2018)

Sofia Ares Oliveira, Benoit Seguin, Frederic Kaplan
Resume

December 12, 2018

Abstract

The diversity of historical document processing tasks prohibits to solve them one at a time and shows a need for designing generic approaches in order to handle the variability of historical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 they address multiple tasks simultaneously such as page extraction, baseline extraction, layout analysis or multiple typologies of illustrations and photograph extraction. Based on a CNN for pixel-wise prediction coupled with task dependent post-processing blocks.

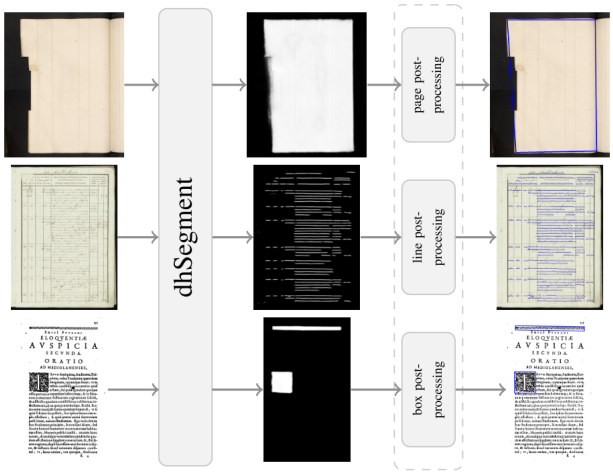
1 Introduction

When working with digitized historical documents, one is frequently faced with recurring needs and problems: how to cut out the page of the manuscript, how to extract the illustration from the text, how to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.

2 Proposed method

The system is based on two successive steps:

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



3 Network architecture

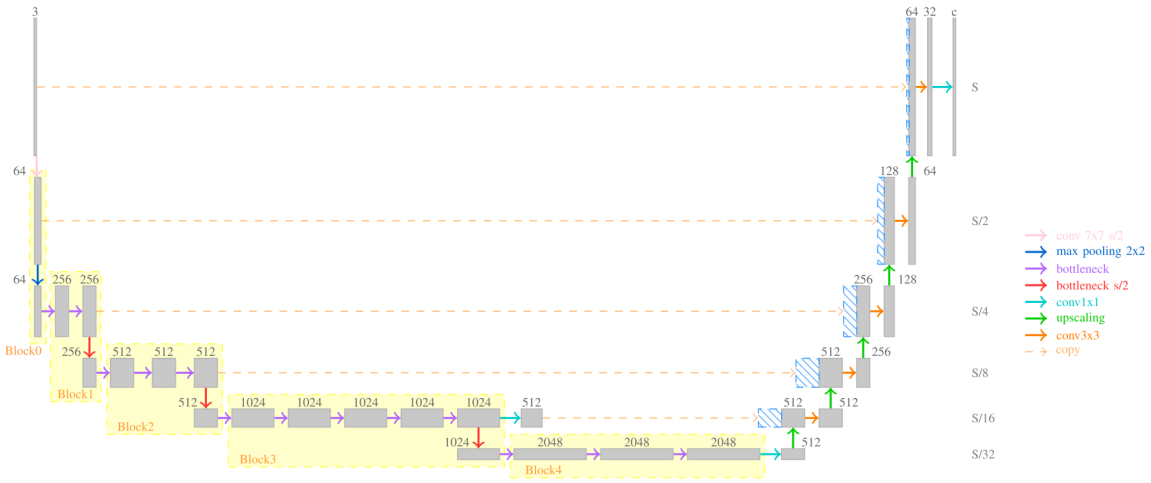


Figure 1: Network architecture of dhSegment. The yellow blocks correspond to ResNet-50 architecture which implementation is slightly different from the original for memory efficiency reasons. The number of features channels are restricted to 512 in the expansive path in order to limit the number of training 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 maps 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

dhSegment is composed of a contracting path, which follows the deep residual network ResNet-50 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). For simplicity reasons the so-called bottleneck blocks are shown as violet arrows and downsampling bottlenecks as red arrows.

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). 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 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.

4 - Post processing

The post processing is limited to simple and standards operations on the predictions.

- *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 Otsus method

- *Morphological operations* The two fundamental basic operations, erosion and dilation can be conbined resulting in closing and openning operations, these two operations are applied to the binary images.

- *Connected componenets analysis* Used in order to filter out small connected componenets that may remain after thresholding or morphological operations.

- *Shape vectorization* To transform the detected region into coordinates, a vectorisation step is performed, this done by extracting the blobs in the binary image as polygonal shapes. The polygones are generaly bounding boxes represented by four corners, the detected shape can also be a line, in this case the vectorisation is reduces to path reduction.

5 Training & Results

5.1 Training

The training is regularized using L2 regularization with weight decay (10^{-6}), a learning rate with an exponential decay rate of 0.95 and an initial value in [105, 104]. Xavier initialization and Adam optimizer are used. Batch renormalization is used to ensure that the lack of diversity in a given batch is not an issue (with rmin = 0.1, rmax = 100, dmax = 1).

The images are resized so that the total number of pixels lies between 6 105 and 106. Images are also cropped into patches of size 300300 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 [0.2, 0.2] rad), scaling (s [0.8, 1.2]) and mirroring.

5.2 Results

In order to investigate the performance of the proposedmethod 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.

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. The network is trained to predict for each pixel if it belongs to the main page.

To obtain a binary image from the probabilities output bythe network, Otsus thresholding is applied. Then morpho- logical 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.

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**

Baseline detection A baseline is defined as a virtual line where most characters rest upon and descenders extend below. Here the network is trained to predict the binary mask of pixels which are in a small 5-pixel radius of the training baselines.

The probability map is then filtered with a gaussian filter ($\sigma = 1.5$) before using hysteresis thresholding (phigh $= 0.4$, plow $= 0.2$, applying thresholding with p_{low} then only keeping connected components which contains at least a pixel value p p_{high}). The obtained binary mask is decomposed in connected components, and each component is finally converted to a polygonal line.

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**

Document layout analysis Refers to the task of segmenting a given document into semantically meaningful regions 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)

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

Ornament detection Ornaments are decorations or embellishments which canbe 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.

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	0.870
	0.8	0.874	0.847	0.902	
	0.9	0.510	0.374	0.803	

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

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

Method	Cardboard	Photo		
	mIoU	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