Pattern Recognition Letters
journal homepage: www.elsevier.com

Multi-scale sequential network for semantic text segmentation and localization

Michael Villamizar[a,**], Olivier Canévet[a], Jean-Marc Odobez[a,b]

[a]*Idiap Research Institute, Switzerland*
[b]*Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland*

ABSTRACT

We present a novel method for semantic text document analysis which in addition to localizing text it labels the text in user-defined semantic categories. More precisely, it consists of a fully-convolutional and sequential network that we apply to the particular case of slide analysis to detect title, bullets and standard text. Our contributions are twofold: (1) A multi-scale network consisting of a series of stages that sequentially refine the prediction of text and semantic labels (text, title, bullet); (2) A synthetic database of slide images with text and semantic annotation that is used to train the network with abundant data and wide variability in text appearance, slide layouts, and noise such as compression artifacts. We evaluate our method on a collection of real slide images collected from multiple conferences, and show that it is able to localize text with an accuracy of 95%, and to classify titles and bullets with accuracies of 94% and 85% respectively. In addition, we show that our method is competitive on scene and born-digital image datasets, such as ICDAR 2011, where it achieves an accuracy of 91.1%.

1. Introduction

Text localization in images has been an active field of research in the computer vision community for decades, including in the last years where people have extended the more traditional document analysis cases to other complex situations like localizing text in natural images [10, 11, 28] as well as oriented text [27]. This is a difficult task in which progresses have been relying on generating challenging datasets like artificial text in real world images [7] or slide datasets [24]. Text localization has many useful real-world applications and is a preliminary step to optical character recognition engines.

Besides localization, many applications require to recognize the semantic category associated with the text. For instance, in scene analysis, localized text needs to be classified as street name, street number, or directions for cars or pedestrians. In other applications such as business card digitizing, in addition to detecting text, we also need to recognize the semantic category of the text such as name, company, position, etc.

In this paper, we are interested in automatic text localization and semantic classification of presentation slides: we aim at detecting text and recognizing the title, the bullets and standard text (*i.e.* semantic information), so as to improve the indexation for better retrieval once uploaded to a website, for instance, through ontology analysis (*i.e.* detecting the main topics of the document). Fig. 1 shows an example of the proposed method in which the network detects text regions and provides a semantic label for each of them (*e.g.* title, bullet and standard text).

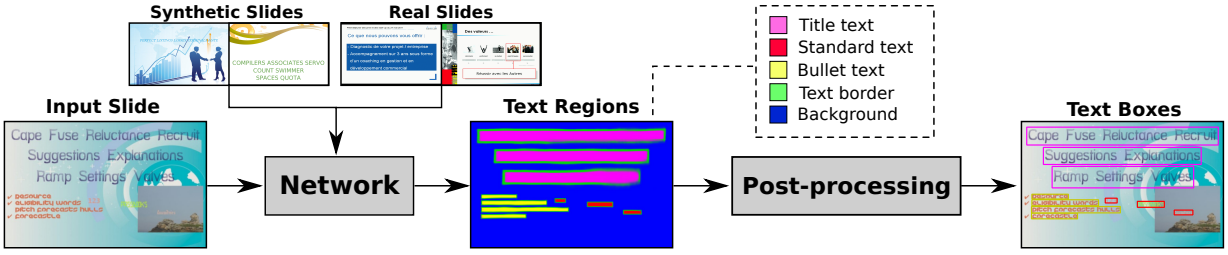
1.1. Related Work

Text localization has been addressed in many different ways according to the task definition. Early efforts were focused on the detection of characters or character components such as Maximally Stable Extremal Regions (MSER) [3, 16], Stroke-width Transform (SWT) [6] and text-like blocks [2]. However, these approaches typically involve complex and time-consuming pipelines which include many useful heuristics to provide text candidates. Additionally, these approaches are sub-optimal because they require tuning individually every pipeline component, which results in a difficult and tedious process [10].

In recent years, thanks to very deep and end-to-end trainable networks for object detection [8, 13, 17], works have addressed text localization as an object-like detection problem, relying on a single and fully-convolutional network to regress the coordinates of boxes containing text [7, 11]. They have shown good results both in accuracy and efficiency (*i.e.* complex pipelines are discarded), although requiring a good initialization.

**Corresponding author:
e-mail: michael.villamizar@idiap.ch (Michael Villamizar)

Fig. 1. We propose a Semantic Text Segmentation Network (STSN) that apart from localizing text in slide images with high accuracy also classifies text regions in different semantic categories such as title (magenta), bullets (yellow) or standard text lines (red). The STSN also predicts text borders (green) and background (blue). The network is trained with both synthetic and real data (slide images).



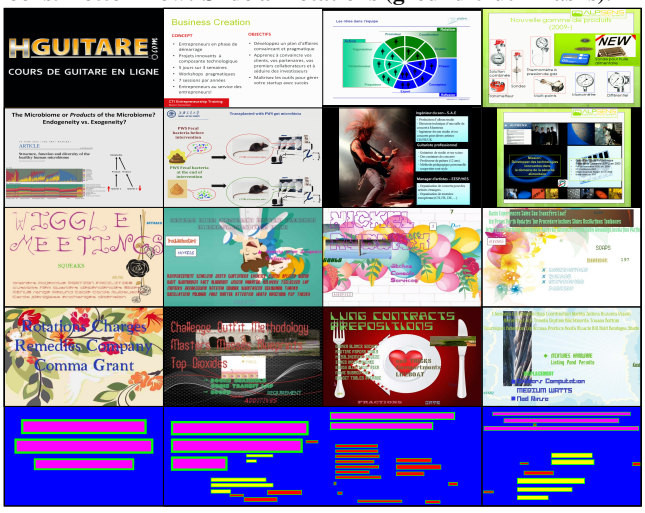
Other works have also proposed deep networks for text detection via image segmentation [28, 24] given the recent success of deep learning for semantic category segmentation [8, 14, 26] This kind of methods commonly use pyramidal architectures to provide more accurate segmentation results [18]. The network output is then text blocks or regions represented using pixel-wise maps. This results in a more natural way of localizing text than using bounding boxes. Nevertheless, most of the works only used two class memberships (text and background) since they mainly rely on an object localization paradigm. A notable exception is [24], in which the network also included text borders as an extra class in order to split blocks of text into multiple text lines, which is a common problem in text segmentation. This yielded better text detection scores.

1.2. Motivations & Contributions

Contrary to the above approaches, our first main contribution is a new method to perform semantic text segmentation. To this end, we propose a multi-scale, fully-convolutional, and sequential network that localizes text and predicts its semantic membership. In particular, the proposed network, called Semantic Text Segmentation Network (STSN), is focused on the problem of both detecting and classifying text on slide images. This also includes predicting text borders and background for better line identification, as shown in Fig. 1. STSN uses a feature pyramid network as feature extractor [12, 18], but we introduce a multi-scale prediction cascade that progressively refines and disambiguate the semantic text predictions in a bottom-up fashion, which results in more accurate text prediction maps. The STSN obtains an accuracy of 95% for detecting any type of text in slides, and accuracies of 94% and 85% for detecting particularly titles and bullets. STSN also shows competitive results in ICDAR dataset where it reaches an accuracy of 91.1%.

Note that achieving such classification is not trivial since, for instance, the main difference between bullets and standard text lines is a bullet symbol preceding text. Also, lines which do not start with a bullet symbol still need to be classified as a bullet line when it is the continuation of a text line preceded by a bullet symbol. In other words, the information about the presence of the bullet symbol needs to be propagated to other multiple lines. Another challenge of working with slides is the detection of text at very different sizes. Titles, for example, tend to have large sizes but text in footnotes or slide numbers is usually small. In addition, slides can also contain difficult backgrounds and text with compression artifacts (*e.g.* JPEG compression). This is usual in text associated with figures, diagrams and plots that have been imported and inserted into the slides.

Fig. 2. Examples of slide images used for training STSN. 1st-2nd rows: Real slides acquired from several conferences. 3rd-4th rows: Artificial slide images containing titles, bullets and multi-line text blocks. Bottom row: Slide annotations (ground-truth masks).



Our second contribution is the use of synthetic and real slides for training the network (see Fig. 1 and Fig. 2). Synthetic slide images are created artificially to reduce the cost of human annotation (*i.e.* text annotations are computed automatically) and to train the network with abundant data. Real slides are used for fine tuning the text segmentation network. These slides were acquired from several conferences and talks and manually annotated. Other recent works have also proposed synthetic text datasets [7, 24]. However, they are mainly focused on localizing single words [7] or computing artificial slides, but without semantic annotation [24].

The rest of this article is organized as follows: section 2 introduces the slide datasets, while section 3 describes the proposed STSN network and its main constituents. It is evaluated and compared in section 4. Conclusions are provided in section 5.

2. Slide Datasets

This section presents the datasets used to train the STSN.

Synthetic slides: Computing synthetic data enlarges the size of the training data and has shown to improve deep network learning [7, 15]. We followed this approach and designed a systematic framework to create artificial slides containing multi-line titles, bullets and text with different layouts, see Figure 2.

Fig. 3. General scheme of the Semantic Text Segmentation Network (STSN) whose blocks refer to small stacks of convolutional layers.

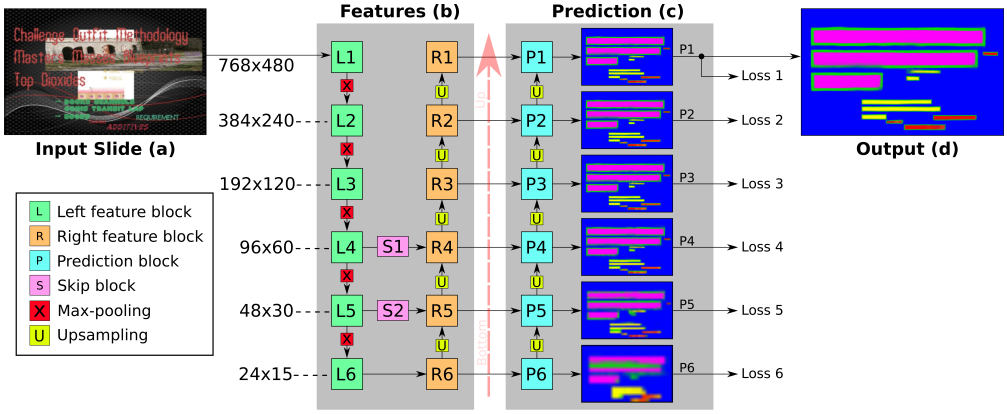


Table 1. Configuration of STSN for every block (B) in the network (see Figure 3). Each cell provides the number of convolutional filters (NF), filter size (FS) and layer operation. Note that blocks B_i, T_i and P_i for $i \in \{4, 5, 6\}$ have the same configurations.

B	Operation	NF	FS	B	Operation	NF	FS	B	Operation	NF	FS
L1	Conv+ReLU	16	7x7	R1	Conv+ReLU	8	13x13	P1	Conv+ReLU	8	7x7
	Conv+ReLU	16	1x1		Conv+ReLU	8	1x1		Conv+Softmax	5	13x13
L2	Conv+ReLU	32	5x5	R2	Conv+ReLU	12	9x9	P2	Conv+ReLU	8	5x5
	Conv+ReLU	32	1x1		Conv+ReLU	12	1x1		Conv+Softmax	5	9x9
L3	Conv+ReLU	48	3x3	R3	Conv+ReLU	16	5x5	P3	Conv+ReLU	12	3x3
	Conv+ReLU	48	1x1		Conv+ReLU	16	1x1		Conv+Softmax	5	5x5
L4	Conv+ReLU	64	3x3	R4	Conv+ReLU	64	3x3	P4	Conv+ReLU	16	3x3
L5	Conv+ReLU	64	1x1	R5	Conv+ReLU	128	1x3	P5	Conv+Softmax	5	3x3
L6				R6	Conv+ReLU	64	1x1	P6			
S1	Conv+ReLU	64	1x1	S2	Conv+ReLU	64	1x1				

Artificial slides were computed from 2115 empty presentation templates downloaded from Internet. Vertical and horizontal mirror images were computed to obtain 8460 templates. Although a large portion of these slides have homogeneous background, many other slides have difficult patterns and drawings.

To obtain realistic and challenging images, we generated slides with large variability in position and text appearance. While title areas were drawn from uniform distributions to appear mainly in the top and center areas of slides, bullet lines and standard text were placed randomly, but avoiding overlap with other text areas. With regard to text shape and appearance, random text size, font and color were used per slide component. Each text line contains multiple and random words and numbers. In detail, we used 5714 text fonts and 32 different symbols for bullet lines. In addition, random crops from plots and landscapes images were added to resemble slide figures. Random image blurring and JPEG compression were also applied to get more realistic effects.

An annotation file was attached to every slide image. It contains the location of all text lines and their semantic labels: title, bullet or standard text. From these annotations, the training system can compute online the annotation masks (*i.e.* targets) seen in Figure 2 (bottom row). For the computation of title, bullet and text masks, we used bounding boxes around text lines. These boxes were tighten to get rid of the borders by reducing the text line height and width by 16% of the text height [24]. The annotation mask is then an array of size $M \times N \times 5$ where M and N denote the size of the image and 5 is the number of feature maps: text, title, bullet, background and text border.

Ultimately, we generated 100k slide images for training and 1k images for test and validation, respectively. All slides were created with a size of 768×480 pixels. About 10% of slides were empty slides (*i.e.* with only background but no text), acting as negative samples.

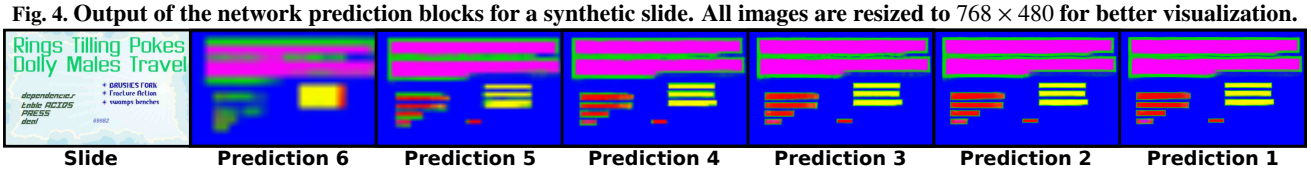
Real slides: To adapt the STSN trained with synthetic data to real images, we collected a dataset of slides from medical presentations. In detail, 1053 images of 1920×1200 pixels were gathered where 414 images are used for training and 639 for evaluation. These slides were resized to 768×480 during training and testing. The corresponding text boxes and semantics were manually annotated. This database is challenging given that many slides have highly textured and diverse backgrounds as well as figures, tables and logos. In addition, the text appears in various sizes as well as image artifacts as a consequence of image compression. Refer to the Fig. 2 to see some examples.

3. Semantic Text Segmentation Network (STSN)

This section describes the architecture and the computational aspects of the STSN. Figure 3 shows a general view of the proposed architecture and Table 1 presents more detailed information about the network configuration.

Architecture overview: We propose a novel network that comprises two modules for semantic text segmentation, as shown in Figure 3 (b, c). The first module uses a modified U-shape network (*a.k.a.* UNet) to extract text features at several resolutions and get a precise segmentation. UNets have shown remarkable results for object and text segmentation since the network combines features from multiple scales and performs the final segmentation at the image-level resolution [14, 18]. This feature module is also similar to feature pyramid networks used to improve object detection at multiple scales [12].

Fig. 4. Output of the network prediction blocks for a synthetic slide. All images are resized to 768×480 for better visualization.



The second module is a series of blocks that sequentially refines semantic text predictions using the features extracted by the first module and the predictions computed at lower resolutions, see Fig. 3 (c). This module is inspired from the Convolutional Pose Machines (CPM) network used for articulated human pose estimation [1, 15], but applied to multiple scales in order to refine predictions in a bottom-up fashion.

Actually, both network modules are designed to propagate features upwards (from low to high resolutions). This characteristic is suited for semantic analysis since the classification of text, title and bullets is done mainly at lower resolutions where the size of the convolutional filters covers large portions of the slide image. At these resolutions, filters encode spatial and feature relationships allowing to distinguish between title, text and bullet areas. This is shown in the prediction map at level 6 (see $P6$ in Figure 3) where the network detects semantic text regions with good accuracy. Subsequently, these predictions are enhanced in upper levels to detect finer details such as lines and text borders (breaking large text regions into text lines). This sequential approach improves the accuracy of the semantic text segmentation task, observe the predictions maps $P5$ and $P1$.

Next, we describe both network modules in further detail.

Feature module: The feature extraction module is based on an U-shape network [18] with 6 levels of depth, see Fig. 3 (b). It computes text features at multiple resolutions which is appropriate because text can appear in images at varying sizes.

In our implementation, the feature module has two types of blocks (stacks of convolutional layers): *left feature blocks* for top-down connections, and *right feature blocks* for bottom-up ones. The first type is focused on computing text features from images which in conjunction to max-pooling operations allows to obtain features at several resolutions. Each block has only two convolutional layers and a small number of filters (and feature maps) due to the simpler task, compared to more generic object recognition methods, and to keep efficiency.

The second type of blocks was designed to compute more discriminative text features. At lower resolutions (see $R4$, $R5$ and $R6$ in Table 1), they have three convolutional layers and a larger number of rectangular filters to capture horizontal patterns such as text lines and words [9, 11]. Note that the input to these blocks is the concatenation of features from *left feature blocks*, via skip blocks containing 1×1 convolutional layers ($S1$ and $S2$), and text features coming from lower resolutions. This approach allows combining and processing efficiently features from different scales and propagates features from coarser to finer resolutions for refinement ($R3$, $R2$ and $R1$). This contrasts to the method presented in [24] in which four different UNets are applied in parallel to cope with text size variations, therefore increasing the computational cost and the number of network parameters since the stacks of convolutional layers are replicated several times.

Prediction module: It consists of a set of prediction blocks (P) that progressively refines the prediction of semantic text classes as well as background and text borders, see Fig. 3 (c). As stated earlier, this idea comes from the CPM network [1, 15] which gradually estimates the location of human body joints through a series of prediction stages, but it differs in two main aspects. The first one is that the refinement of semantic text predictions is done at different feature resolution levels, instead of a single and low resolution in the CPM approach. With this novel approach, the network performs image segmentation from coarser to finer resolution levels. At low levels the network is focused on general aspects of the slide such as recognizing chunks of text and discerning between title, standard text, bullets and background. This is possible because the filters cover large areas of the input image which allows to capture geometrical and appearance relationships between semantic classes. Later, in next prediction blocks, these initial hypotheses are refined and enhanced to detect finer details such as lines and edges and produce more accurate segmentation results. This is seen in Figure 4 displaying the output of the prediction blocks.

The second difference lies in using different features at multiple resolutions as inputs to the prediction blocks. This allows to integrate features from different scales and to produce more precise prediction maps. By contrast, the CPM network shares the same features computed at low resolution [1].

Finally, Table 1 shows that each prediction block has two convolutional layers, the second one using Softmax.

General settings: All convolutional layers (41 layers), except the last prediction layers, comprise batch normalization and Rectified Linear Units (ReLU), showing good experimental results and faster training. Note that STSN is a fully-convolutional network involving only convolutional layers. This reduces the number of parameters and makes the network independent of the size of the input image [7, 11, 13, 14].

Training: The sequential and incremental nature of the STSN across scales allows to introduce partial losses to add intermediate supervision during training. This idea has shown good results in other works, especially for dealing with the problem of *vanishing gradients* in deep networks [20]. The use of partial losses also enforces the STSN to learn text features at different scales, allowing the refinement of semantic text estimations.

Thus, the training loss of STSN is calculated as a linear combination of partial losses, $L = \frac{1}{6}\sum_{i=1}^{6} L_i$, where L_i is the loss for the prediction block i which in turn is defined as the Mean Squared Error (MSE) between the prediction map provided by this block (P_i) and the ground truth (annotation mask) in the training dataset, observe Figure 2 (bottom row).

The STSN is trained for 50k iterations using the synthetic dataset and for 2k iterations to fine tune the network with real slides. The size of mini-batches is 8 samples. To optimize the network, we use Adam optimizer with default settings.

Fig. 5. Different network architectures for text detection: VGG-like network [19], UNet [18], multi-scale UNet [24].

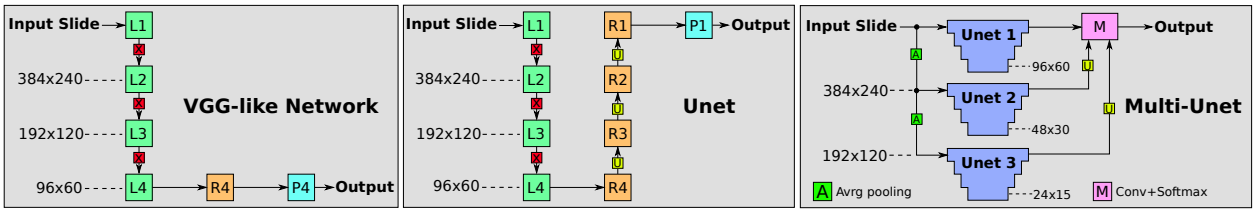


Table 2. Evaluation of the networks on the synthetic and real slide datasets, in terms of the recall (R), precision (P) and harmonic mean (H) rates. "Real slide dataset [Fine Tuning]" denotes the rates on the real data after the networks having been fine tuned with real slides.

	VGG-like			UNet			Multi-UNet [24]			STSN [P1]			STSN		
	R	P	H	R	P	H	R	P	H	R	P	H	R	P	H
Syntethic Slide Dataset															
Text	0.59	0.63	0.61	0.81	0.75	0.78	0.84	0.83	0.84	**0.88**	0.85	0.86	**0.88**	**0.92**	**0.90**
Title	0.61	0.43	0.51	0.75	0.53	0.62	0.85	0.76	0.81	0.90	0.82	0.86	**0.94**	**0.91**	**0.93**
Bullet	0.53	0.44	0.48	0.30	0.86	0.44	0.59	0.94	0.72	0.79	**0.90**	0.84	**0.83**	**0.90**	**0.87**
Real Slide Dataset															
Text	0.54	0.63	0.58	0.66	**0.64**	**0.65**	**0.60**	0.63	0.61	0.59	0.59	0.59	0.58	0.63	0.61
Title	0.54	0.59	0.56	0.63	0.73	0.68	**0.87**	0.73	0.80	0.77	0.81	0.79	0.79	**0.85**	**0.82**
Bullet	0.50	0.26	0.34	0.18	0.30	0.23	0.40	**0.53**	0.45	0.54	0.50	0.52	**0.67**	0.52	**0.58**
Real Slide Dataset [Fine Tuning]															
Text	0.74	0.85	0.79	0.92	0.94	0.93	0.92	0.95	0.93	0.91	**0.96**	0.93	**0.94**	**0.96**	**0.95**
Title	0.82	0.77	0.80	0.82	0.86	0.84	0.93	**0.93**	0.93	0.93	**0.93**	0.93	**0.95**	0.92	**0.94**
Bullet	0.66	0.47	0.55	0.49	0.68	0.57	0.84	0.75	0.79	**0.90**	0.73	0.81	0.89	**0.81**	**0.85**

Post-processing: When the network is tested on an input image, a set of post-processing steps are applied to the network output so as to enhance the quality of the text and to extract the bounding boxes (Figure 1). The first one consists of thresholding the prediction maps associated to title, bullet, and standard text to generate text areas (lines) and to discard poor predictions. A threshold of 0.2 was set experimentally. Then, small text regions are removed either if the retrieved text line height is below 1.5% of the image height or the area of the text region is lower than 0.1% of the image size. These steps are soft rules devoted to remove spurious unlikely text areas.

Sometimes, text line predictions present ambiguity about the class membership. This occurs mainly for hard cases where STSN predicts bullet and standard text for the same line, given the similarity between these text categories. In those cases, the text line is assigned to the class with larger area in the text line.

Ultimately, bounding boxes are computed for the predicted title, bullet, and standard text lines. These boxes are enlarged to consider the reduction of borders done during training.

4. Experiments

The presented method is validated in two different scenarios. The first scenario is focused on semantic text localization in slides (*i.e.* our main topic), whereas the second one is for text localization in standard benchmarks.

4.1. Semantic Text Localization

Evaluation protocol: Results are evaluated using the DetEval evaluation protocol [23] that measures the overlapping between the predicted text boxes and ground-truth ones. DetEval returns the recall, the precision, and the harmonic mean rates.

Tested models: Our STSN is compared against other conventional architectures to validate the proposed network combining the multi-scale feature and prediction modules. Figure 5 shows the evaluated architectures, which all use the same block configurations as the STSN to be consistent with features and to focus on evaluating the network structure. All networks are learned and evaluated with the same training and testing settings.

The first network is a VGG-like network [19] consisting of convolutional and max-pooling layers. In this network, the text prediction is done at low resolution (*i.e.* 96×60). The second network is an UNet network [18] including further convolutional layers and upsampling operations. The final prediction is done at the image resolution (*i.e.* 768×480). The third network is a multi-scale UNet having three UNets working in parallel to detect text at varying sizes. The final prediction is computed by concatenating and processing the predictions from all UNets. This network is similar to the network introduced in [24]. The fourth network is the proposed STSN without using the prediction module. That is, the network only predicts text at level 1 ($P1$) whereas blocks $P2$ to $P6$ are removed.

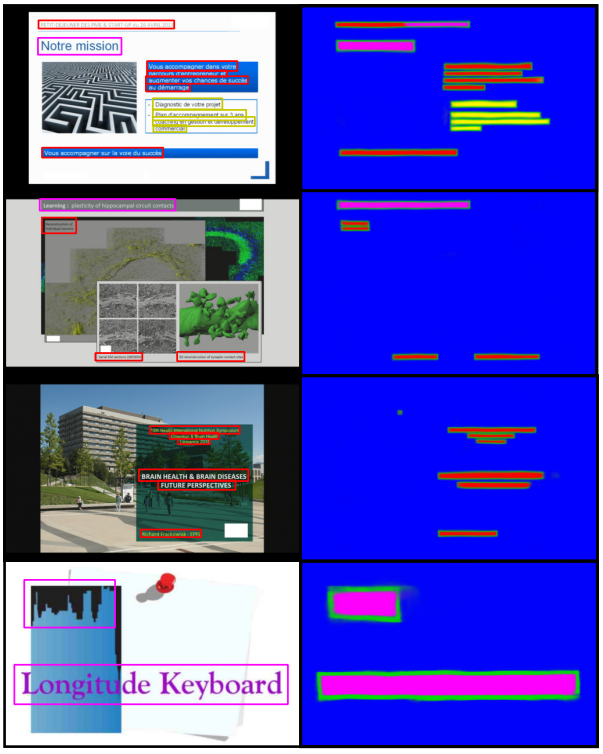
Results: Table 2 shows the detection rates for the synthetic and real datasets. Here, instead of reporting the rates for the standard text category, we computed the detection rates for all type of text in the slides (*i.e.* without semantic information). This was done combining the detections for the three disjoint text semantic categories. We see that in all cases STSN obtains better scores than the other network architectures, especially for titles and bullets. The exception is the harmonic mean score for the text category without fine tuning. Our method is exceeded by the UNet architectures. Yet, STSN obtains a larger rate for bullets and outperform all networks after fine tuning. This shows that the progressive refinement of features contributes to distinguish the different text categories, and the importance of fine tuning with real data to close the gap between data domains.

VGG-like network yields low detection rates since the text prediction is done at low resolution (similar case for the CPM [22]), whereas the UNet provides better scores because

Table 3. Detection rates of the STSN prediction blocks (*P6* to *P2*, see Figure 3).

	Prediction 6			Prediction 5			Prediction 4			Prediction 3			Prediction 2		
	R	P	H	R	P	H	R	P	H	R	P	H	R	P	H
Text	0.25	0.35	0.29	0.56	0.72	0.63	0.77	0.88	0.82	0.85	0.89	0.87	0.92	0.95	0.94
Title	0.38	0.49	0.43	0.84	0.88	0.86	0.92	0.91	0.92	0.94	0.92	0.93	0.95	0.93	0.94
Bullet	0.31	0.19	0.24	0.63	0.37	0.47	0.83	0.50	0.63	0.86	0.57	0.68	0.90	0.74	0.81

Fig. 6. Output of STSN on real and synthetic slides indicated via bounding boxes on slides and semantic prediction maps.



it uses a larger resolution for semantic text prediction. Similarly, the Multi-UNet [24] shows very remarkable results, but at the expense of a more complex and inefficient method since it computes three UNets in parallel, replicating blocks and features. Conversely, STSN uses a single UNet to compute multiple prediction maps. As a consequence, the network needs less parameters (see Table 1). We also see that the use of the prediction module is beneficial for the prediction of all text categories.

Fig. 6 shows the output of STSN for real and synthetic slides and failure cases (two bottom rows). Note that STSN is able to simultaneously localize text and perform semantic text segmentation with high accuracy. Particularly noteworthy is the detection of text in complex backgrounds and with large variations in the size of text. The failure cases correspond to an unusual title position in the slide (third row) that STSN identifies as regular text, and the wrong detection of multiple vertical edges as text (fourth row). Note that the STSN classifies this erroneous text as title given its size and proximity to the true title.

Fine tuning: Table 2 also provides the rates computed on the real slide dataset after having fine tuned the networks. We observe again that the STSN outperforms other networks and that fine tuning improves significantly the detection scores. This shows that while synthetic data are helpful for training the network, it is still necessary to perform fine tuning with real data to adapt the network to the real domain.

Table 4. Text localization scores on ICDAR 2011 and 2013 dataset.

	ICDAR 2011			ICDAR 2013		
Scores	R	P	H	R	P	H
Chen *et al.* [4]	0.89	0.92	0.90	–	–	–
Cho *et al.* [5]	0.91	**0.95**	**0.93**	0.79	0.86	0.82
Jadeberg *et al.* [10]	0.68	0.88	0.77	0.68	0.88	0.77
Gupta *et al.* [7]	0.75	0.91	0.82	0.75	**0.92**	0.83
Liao *et al.* [11]	0.82	0.89	0.86	**0.83**	0.89	**0.86**
Tian *et al.* [21]	–	–	–	0.76	0.85	0.80
Wu *et al.* [24]	**0.95**	0.91	**0.93**	0.78	0.91	0.84
Yin *et al.* [25]	0.87	0.94	0.90	–	–	–
Zhang *et al.* [27]	0.76	0.84	0.80	0.74	0.88	0.80
STSN	0.88	0.94	0.91	0.78	0.86	0.81

Prediction refinement: Table 3 shows the detection scores on real slides for the different predictions maps provided by STSN. Prediction rates for level 1 (*P1*) are given in Table 2. The detection scores are improved as the resolution increases.

4.2. Text Localization

Our main goal is to detect text in presentation slides (*i.e.* not text in the wild) and to extract the semantic of the text. However, to assess the detection performance of our STSN, we evaluated it on the standard ICDAR datasets which consist of digital images (ICDAR 2011) and text in the wild (ICDAR 2013).

Training: Since ICDAR does not have semantic text annotations, STSN is trained for text detection only. Specifically, it is trained for 50k iterations using the synthetic slide dataset, and for 10k iterations to fine tune the network using the corresponding ICDAR training data and artificial slides. Data augmentation is done to enlarge the training data with random image crops and rotations between ±15 degrees.

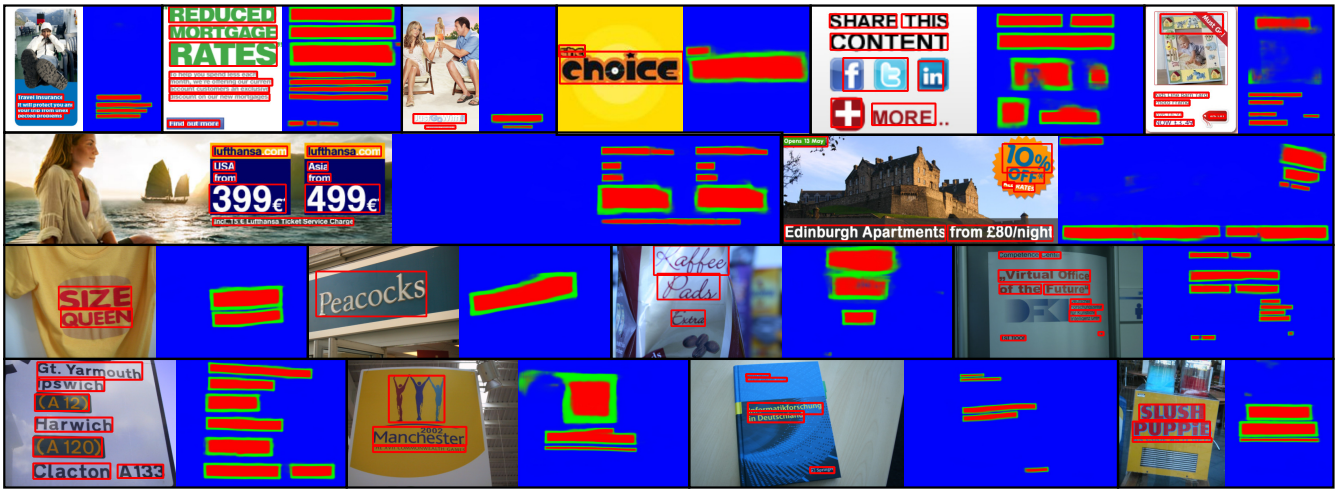
Results: Table 4 shows the results of our STSN and a comparison with other works in the state of the art. On ICDAR 2011 STSN reached a hmean of 0.91 (the best being 0.93), and of 0.81 on ICDAR 2013 (the best being 0.86). The main failure cases of our system are single digits or letters as well as rotated text and low contrasted text, which are all very seldom in presentation slides. This explains why STSN does not reach state-of-the-art performances, but still remains competitive.

Figure 7 shows some example images with the output of the STSN on the ICDAR dataset. We see that STSN is able to detect text in born-digital images as well as text in the wild under different and challenging imaging conditions.

5. Conclusion

We presented a Semantic Text Segmentation Network (STSN) to simultaneously detect text and to classify the detections in semantic categories. We have shown that our novel architecture which includes our proposed multi-scale prediction cascade is able to sequentially refine the semantic text predictions, thus achieving better performance for difficult semantic

Fig. 7. Illustration of STSN output on ICDAR 2011 (top rows) and ICDAR 2013 (bottom rows).



classes like bullets; fine tuning on a target dataset was also achieving better performance. We have shown that although the STSN was conceived for our purpose (presentation slides), it is competitive with other works on the standard ICDAR dataset for detecting text in scene and born-digital images. Future work includes the study of novel architectures such as hourglass blocks to refine even more the semantic text predictions, the use of temporal information to exploit the layout consistency between slides from the same talk, and the prediction of other semantic categories.

Acknowledgments: The work was supported by Innosuisse, the Swiss innovation agency, through the VIEW-2 (Visibility Improvement for Events Webcasting) project.

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2017.

[2] D. Chen, J.-M. Odobez, and H. Bourlard. Text detection and recognition in images and video frames. *Pattern recognition*, 37(3):595–608, 2004.

[3] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *in International Conference on Image Processing*, 2011.

[4] K. Chen, F. Yin, and C.-L. Liu. Effective candidate component extraction for text localization in born-digital images by combining text contours and stroke interior regions. In *In Document Analysis Systems (DAS), 2016 12th IAPR Workshop*, pages 352–357, 2016.

[5] H. Cho, M. Sung, and B. Jun. Canny text detector: Fast and robust scene text localization algorithm. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] B. Epshteini, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010.

[7] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *International conference on computer vision*, pages 2961–2969, 2017.

[9] T. He, W. Huang, Y. Qiao, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. *CoRR*, abs/1603.09423, 2016.

[10] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, Jan 2016.

[11] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, 2017.

[12] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. Ssd: Single shot multibox detector. In *Computer Vision - 14th European Conference, ECCV 2016, Proceedings*, pages 21–37, 2016.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[15] A. Martínez-González, M. Villamizar, O. Canévet, and J.-M. Odobez. Real-time convolutional networks for depth-based human pose estimation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 41–47. IEEE, 2018.

[16] L. Neumann. Real-time scene text localization and recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[17] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[18] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *in Proc. Med. Image Comput. Comput.-Assisted Intervention*, pages 234–241, 2015.

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan. Text flow: A unified text detection system in natural scene images. In *International Conference on Computer Vision*, pages 4651–4659, 2015.

[22] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] C. Wolf and J.-M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4):280–296, 2006.

[24] Y. Wu and P. Natarajan. Self-organized text detection with minimal post-processing via border learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[25] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):970–983, 2014.

[26] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[27] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[28] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.