Title

Robust and Accurate Text Stroke Segmentation

ISBN

978-1-5386-4886-5

Authors

Qin, Siyang
Ren, Peng
Kim, Seongdo
et al.

Publication Date

2018-03-01

DOI

10.1109/wacv.2018.00033

Peer reviewed

Robust and Accurate Text Stroke Segmentation

Siyang Qin, Peng Ren, Seongdo Kim, Roberto Manduchi
Computer Engineering Department, University of California Santa Cruz
{siqin,seongdo,manduchi}@soe.ucsc.edu, pren1@ucsc.edu

Abstract

We propose a new technique for the accurate segmentation of text strokes from an image. The algorithm takes in a cropped image containing a word. It first performs a coarse segmentation using a Fully Convolutional Network (FCN). While not accurate, this initial segmentation can usually identify most of the text stroke content even in difficult situations, with uneven lighting and non-uniform background. The segmentation is then refined using a fully connected Conditional Random Field (CRF) with a novel kernel definition that includes stroke width information. In order to train the network, we created a new synthetic data set with 100K text images. Tested against standard benchmarks with pixel-level annotation (ICDAR 2003, ICDAR 2011, and SVT) our algorithm outperforms the state of the art by a noticeable margin.

Figure 1. Text stroke segmentation is a challenging task due to large variance in text font, color, confounding background, poor contrast as well as different illumination conditions. Here we show several challenging cases and our results.

1. Introduction

Optical character recognition (OCR) has been one of the earliest success stories in computer vision. A fully electronic text reading system was demonstrated as early as in 1946 [21], while the first commercial OCR company, Intelligent Machines, was founded by Shepard and Cook in the early 50's. By the early 1980s, OCR of scanned documents was considered a solved problem. More recently, automatic text reading has received renewed interest in domains that were considered too challenging for traditional technology.

Scene text (or *text in the wild*) is a term often used to indicate text of any kind appearing in pictures or videos, often taken by hand or by a moving camera. As such, these images suffer from all sort of imperfections: blur, low resolution, poor exposure, reduced contrast. The text content itself is often very coincise (e.g., the name of a store), and not necessarily displayed on a straight line. Text may appear in front of a possibly multi-colored background. Specularities and cast shadows cutting across the text area are not unusual. Unlike scanned documents, which normally contain a large portion of well-structured text printed against a solid color background, detecting and localizing text areas in general scenes is challenging, especially when the scene contains visual clutter and the text itself occupies a small area. In addition, almost all applications involving scene text reading demand high frame rate processing. For this reason, considerable research effort went into algorithms for fast and robust scene text detection (or *spotting*). Once a text bearing region has been identified, its content can be processed by any standard OCR algorithm. Some text spotting algorithms specialize on separating individual words within the text area [30], thus further simplifying the job of subsequent modules.

Early attempts at text spotting considered an initial stage of *text stroke segmentation*, that is, segmentation of the regions corresponding to text strokes from the background. Widely used techniques include Maximally Stable Extremal Regions (MSER) [22], a fast technique for generic local segmentation that is robust against domain and photometric distortions; and the Stroke Width Transform (SWT) [4], which is specifically designed for the detection of stroke-like regions. More modern text spotting approaches skip the text stroke segmentation step altogether, relying instead on general object detection techniques based on convolutional

neural networks (CNNs).

While text stroke segmentation may not be needed for text detection, it still has an important role in improving the performance of OCR [15] and other specific applications of interest. For example, binarization allows for operations such as text removal (and possibly substitution), text color change, and contrast enhancement. This type of operations are often required for stock photography processing (e.g., license plate number removal from Google StreetView images [7]), augmented reality (e.g., substitution of original text with its translation in a different language [6]), and assistive technology (e.g., to increase text readability for people with low vision [12]). Precise stroke segmentation is needed in these applications in order to preserve the naturalness of processed images.

Our main contribution is a novel algorithm for text stroke segmentation that produces accurate results in the face of adversarial conditions such as cast shadows and cluttered background. The algorithm operates on image areas that have been previously identified as containing text (by an appropriate spotting algorithm). It is structured as the cascade of two modules. The first module uses a fully convolutional network (FCN) trained to robustly discriminate pixel within a stroke from those in the background. This results in a segmentation that reliably identifies text stroke areas; however, due to the multi-scale nature of FCN, this segmentation is often not accurate (see e.g. Fig.4). The second module is in charge of refining the earlier segmentation, in order to ensure that the contour of the stroke regions is correctly preserved. It relies on a fully connected conditional random field (CRF) model that uses an innovative expression for the pairwise energy term, one that uses information from the estimated local stroke width. We show that, by adding the proposed stroke width term to the more traditional bilateral kernel, the accuracy of segmentation improves noticeably.

In order to train the FCN, a large amount of images labeled at the pixel level are necessary. Unfortunately, existing data with pixel-level labeling of text content is scarce. We therefore assembled a new synthetic data set, with 100,000 images, representing a wide variety of font and backgrounds, along with pixel-level ground truth labels. This is the second original contribution of our work. Note that another synthetic data set was created in prior work to facilitate training of text spotters in natural images [8]. However, this prior data set did not provide pixel-level labels, and thus could not be used for our purpose.

We also propose the use of our text stroke segmentation algorithm and image inpainting technique to generate realistic synthetic data (see Sec.5.4). This is our third contribution.

This paper is organized as follows. In Sec. 2 we review previous work on text stroke segmentation. Sec. 3 describes our new synthetic text data set with pixel level ground truth labels. Our algorithm for robust and accurate text stroke segmentation is described in Sec. 4, with experimental results presented in Sec. 5. Sec. 6 has the conclusions.

2. Related Work

Document image binarization has a long history. A number of algorithms, based on image brightness thresholding , have been developed, beginning with Otsu' seminal work[28, 27, 32, 35, 9]. These techniques achieve good performance on scanned documents, but often fail on scene text segmentation, due in part to the typical large variance in font, color, illumination that is typical in this type of imagery, as well as the to possible presence of complex background.

Early attempts at scene text segmentation tried to separate text strokes from background using local features such as edge [16] and color [40, 20], which were processed using simple thresholding or filtering. Later work used more sophisticated image models such as Markov Random Field (MRF) [24, 37, 23]. For example, Mishra *et al.* [24] used a MRF model where the unary energy term is described by a Gaussian mixture. The parameters of the color distribution within the text area were initialized using the stroke width transform (SWT [4]) Energy minimization was obtained via iterative graph cut [31]. A variant of this algorithm, proposed by Tian *et al.* [37], used the Stroke Feature Transform (SFT [10]) for initialization. SFT is more robust than SWT, resulting in more accurate initial color distributions, and thus avoiding the need for iterative graph cuts. Unfortunately, neither algorithm can cope with challenging situations, when local features become unreliable.

Maximally Stable Extremal Regions (MSERs) [22] have been used widely for scene text detection[25, 11, 29] and segmentation[25, 36, 26]. A classifier is trained to separate text from background based on the shape of each MSER region, along with other hand–drafted features. In order to achieve high recall rates, MSERs are often extracted from multiple color channels and using different thresholds; this, however, increases the computational load. Zhou *et al.* [42] proposed to use analysis-by-synthesis for text segmentation. A physical model was used to synthesize image given an initial set of rendering parameters and initial foreground/background labels. The parameters of the model were optimized using Expectation Maximization.

In recent years, convolutional neural networks (CNN) have been successfully applied to virtually all fields of computer vision, including scene text reading. In particular, fully convolutional networks (FCN) [17] are well suited for pixel-level segmentation. One problem with FCNs, however, is that, due to the large receptive field size of the cells in the network, the segmentation produced is often poorly localized (i.e., the contours of the detected regions may not closely follow the contours of the foreground regions

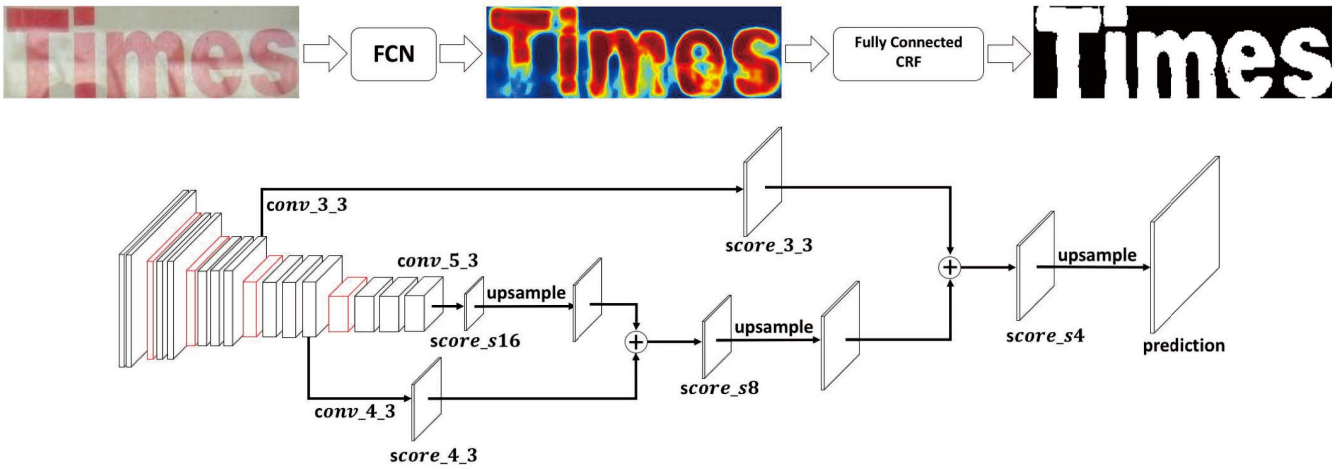

Figure 2. Our proposed framework.

in the image). Fully-connected conditional random field (CRF) [14] models are often used to overcome this limitation, and to refine local details of segmentation by minimizing a carefully designed global energy function. Chen *et al.* [3] fed the label assignment probabilities produced by an FCN to a fully-connected CRF, with the two modules trained separately. Zheng *et al.* [41] reformulated the mean-field algorithm for approximate inference for a fully-connected CRF as a Recurrent Neural Network (RNN), thus enabling end-to-end training.

3. A New Synthetic Text Data Set with Pixel-Level Labels

Training CNNs requires a large amount of data. Large data sets have been assembled for scene text detection (e.g. [38]). These sets are equipped with bounding box annotation identifying individual words. Unfortunately, data sets with *pixel-level* annotations are of a much smaller scale. For example, ICDAR 2003[19], ICDAR 2011[33], ICDAR 2015[1] and SVT[39] only contain a few hundreds word bounding boxes annotated at the pixel level as text stroke vs. background. While this size can be adequate for a test set, it is insufficient for training a network. We thus decided to generate a new, large scale data set with synthetic data. In the following, we describe how our new data set has been generated.

We began by sampling 100K words from an English corpus. These words were rendered using ImageMagick[1] onto a background. Each word was randomly assigned one of 264 different fonts, with height varying between 15 and 90 pixels. The font color could be white (25% of words), black (25%), grey (25%), or randomly chosen from a palette. Each word underwent one of a set possible geometric transformations (rotations, cylindrical projections, perspective transformations, wave distortion), with parameters sampled from a normal distribution. Words were then rendered against a background that could have a randomly chosen solid color (66% of words), or a portion of a "natural" image, randomly selected from the IAPR TC-12 Benchmark. The resulting images were corrupted with additive noise (Gaussian, impulse, and Laplacian), reflection and shadow effect. The resulting images have height of 112 pixels and variable width; the binary mask (text stroke vs. background) is provided for each image. We only use images from this set to train our algorithms (reserving a 10K subset for validation); the algorithms are then benchmarked on all available annotated real images.

Figure 3. Samples from our synthetic dataset.

4. Text Stroke Segmentation

Our proposed framework and FCN structure are shown in Fig.2. The resized input word patch (height is 112 pixels) is fed to the FCN to produce a coarse segmentation, more accurate text stroke mask is obtained with a fully-connected

[1] imagemagick.org

CRF refinement step.

4.1. Coarse Segmentation: FCN

The first step in our algorithm is a coarse pixel-level segmentation of text strokes from the background using a FCN. Thanks to their ability to use information at multiple scales, FCNs can segment text strokes even in challenging situations.

The network structure of the original FCN [17] was derived from the VGG 16-layer network[34], with the final classifier layer removed, and the fully connected layers converted to convolutional layers. We modified the original FCN scheme for our application as follows. First of all, we remove the last pooling layer (*pool5*) and all subsequent layers. This is justified by the observation that text stroke segmentation from an already cropped word patch is a simpler undertaking than generic semantic segmentation, which was the task addressed by [17]. The last convolutional layer (*conv_5_3*) is fed to a 1×1 convolutional layer with channel dimension of two, producing class prediction scores for text and background (*score_s16*). As suggested in [17], two skip layers are added, with the purpose to combine low resolution, highly semantic information with finer detail. The coarse prediction scores *score_s16* are upsampled by two before being combined with the prediction scores from *conv_4_3* to produce a finer scale prediction (*score_s8*). The same process is repeated for the second skip layer. The resulting prediction score *score_s4* is then upsampled by four to match the input image size. The upsampling layers are initialized with a bilinear interpolation kernel, whose weights are then learned during training.

Another difference with respect to the original FCN [17] is that the skip layers branch out at the end of a "block" of layers between two pooling layers, rather than at the beginning (layers labelled in red in Fig. 2). Skip layers are used to maintain information at higher resolution. The end layer of a block has the same resolution as the beginning layer, but may contain semantically richer information, and thus may prove a better candidate for a skip layer branching point.

Note that features at the coarsest scale (*score_s16*) have receptive field size of 192-by-192 , which is substantially larger than the height of input word block (112 pixels).

4.2. Refinement: Fully-Connected CRF with Stroke Width Kernel

The first stage FCN is able to segment out text strokes under a variety of font, color, illumination and background. However, as observed in Fig.4, the resulting segments are often not accurately localized. This is likely due to the large receptive field size of the nodes in the network, and to the fact that the result is upsampled from a low resolution map.

In order to refine the segmentation produced by FCN, we add a fully-connected CRF as a post-processing step. This produces a very noticeable improvement. A further improvement is obtained by modifying the the standard bilateral kernel [3] used to compute joint energy terms. Specifically, we propose to include in this term the estimated stroke width as a new text-specific feature. This is born by the observation that the stroke width is approximately constant within a text character. The standard bilateral kernel, which discourages assigning different labels to nearby pixels that have similar colors, fails to properly characterize the appearance of characters with large local color variations (e.g. as due to a cast shadow); background pixels with similar color as text region might be wrongly predicted as text (see Fig.5). By adding a measure of stroke width consistency in the joint energy term, CRF is more likely to correctly segment out whole characters and filter out background region with confounding color.

We define the following CRF energy function:

$$E = \sum_i -\log P(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{1}$$

$$\begin{aligned}\theta_{ij}(x_i, x_j) =& \mu(x_i, x_j) w \exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} \\ & - \frac{|I_i - I_j|^2}{2\theta_\beta^2} - \frac{|s_i - s_j|^2}{2\theta_\gamma^2}).\end{aligned} \tag{2}$$

where x_i and x_j are labels for pixels i and j, located at position p_i and p_j, with colors I_i and I_j, and associated stroke widths s_i and s_j. (The computation of stroke width is described later in Sec. 4.2.1.) In the unary energy term, $P(x_i)$ is the probability of pixel i having label x_i; this is computed from the score returned by FCN. More specifically, $P(t_i) = 1 - P(b_i)$ is the probability that pixel i belongs to a text stroke. $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$, zero otherwise. $\theta_{ij}(x_i, x_j)$ for $x_i \neq x_j$ is the cost of assigning different labels to pixels i and j, which depends on the distance between pixels, their difference in color, and their difference in associated stroke width. The hyper-parameter w controls the weight of the joint energy term, while θ_α, θ_β and θ_γ controls the scale of each feature.

4.2.1 Modified Stroke Feature Transform

The stroke width at each pixel (term s_i in equation (2)) is computed before CRF refinement, based on the original image with additional input from the FCN predictions. It is based on the Stroke Feature Transform (SFT) [10], which is a modification of the original Stroke Width Transform (SWT) algorithm [4]. In the SFT algorithm, edges are first extracted from the image (using Canny); then, a line is drawn from each edge pixel in the direction of the image gradient. The line is stopped as soon as it hits another edge pixel, or when the color of the current pixel differs from

Figure 4. The first row is input images, second row is raw FCN predictions and last row contains final results with CRF refinement.

Figure 5. The first row is input images, second row is results without stroke width kernel and last row contains results with stroke width kernel. In left example the text color has large variance due to shadow and for right image the background contains regions with similar color as text region. In these cases use bilateral kernel alone becomes unreliable.

the median color of the pixels in the current segment by a large margin. The segment is then accepted if the image gradients at its endpoints point in approximately opposite directions. In addition, after all segments have been drawn, any segment whose median gradient orientation or colors is significantly different from that of its neighbors is discarded. Finally, all pixels within a segment are assigned a *stroke width* value equal to the segment's length. Note that this algorithm may leave some small untouched "islands" of pixels within a character; these pixels are then assigned a value equal to the median of the value of their closest neighbors.

SFT was shown to be more robust than SWT, especially in situations in which edges may be difficult to compute reliably, or when the gradient at an edge pixel points away from the normal to the stroke edge. We further improve on the SFT algorithm (Modified SFT) by using information from the FCN output probability map. Specifically, we only keep a segment (as computed by SFT) when the average of $P(t_i)$ for pixels i within the segment is larger than a threshold. This helps ensuring that incorrect segments are not mistakenly accepted only because the image gradients at their endpoints happen to have approximately opposite directions. We run our modified SFT twice on two polarities, in order to find the stroke widths for dark text on light background as well for light text on dark background.

Note that, unlike SFT and SWT, we don't compute connected components of pixels with similar stroke width. We use the stroke width information solely as a feature in the kernel for the joint pairwise energy.

5. Experiments

5.1. Implementation Details

The coarse FCN segmentation component of our system is trained on the 100K synthetic images in our data set. As mentioned earlier, each image in the data set has fixed height (112 pixel) and variable width, depending on the word's aspect ratio. Due to the variable size of the samples, we set the batch size to one, and reshape the network at each forward pass. Cross-entropy loss is used during training.

The weights of our FCN are initialized from those of the network described in [30] (originally trained for scene text detection), and fine-tuned following the guidelines of [17]. We first fine-tune the model without skip layers for four epochs, with learning rate set to 10^{-9}, momentum set to 0.99, and weight decay set to 0.0005. We then add one skip layer at a time with reduced learning rate (10^{-11} and 10^{-12} respectively). The hyperparameters of the fully-connected CRF are determined by cross-validation on the validation set. We used the publicly available C++ implementation of the CRF's provided by the authors of [14].

Our system is implemented using Caffe [13] and runs on a workstation (3.3Ghz 6-core CPU, 32G RAM, Nvidia GTX Titan X GPU and Ubuntu 14.04 64-bit OS). At run tine, a 180 by 60 pixel input image is processed in about 0.2 seconds.

5.2. Quantitative Results

Data sets: We evaluated our algorithm against several popular document binarization methods [9, 27, 28], as well as against other state-of-the-art scene text segmentation techniques [18, 24, 5, 37, 36, 42]. We computed pixel-level precision, recall, and f-score for three popular scene text data sets: ICDAR 2003 [19] (1110 words); ICDAR 2011 [33] (716 words); and SVT [39] (647 words). For each data set, cropped rectangular regions containing individual word are available. Pixel level ground-truth labeling was generated by Kumar[15] using a publicly available semi-automated tool.

Polarity: For document binarization algorithms such as Niblack and Howe, correct polarity is not guaranteed. For fair comparison, we simply computed f-scores for each po-

larity, and reported the largest one. This is an optimistic measure: in practice, an automatic polarity check would be needed when using these algorithms, which may generate errors not considered by this measure.

Ablation study: We present results (1) using the full system (FCN+CRF/SFT), (2) removing the stroke width term form the CRF joint energy term (FCN+CRF), and (3) without using the fully connected CRF refinement step (FCN).

5.2.1 ICDAR datasets

Comparative results for the ICDAR 2003 and ICDAR 2011 data sets are shown in Table 1 and 2. Note that these sets are relative easy as compared with SVT. Many images have clean background and clear text with bimodal color distribution, which allows simple binarization algorithm such as Niblack[27] and Howe[9] to reach appreciable performance (assuming correct polarity). More modern scene text segmentation algorithms consistently outperform these simpler binarization algorithms, thanks to their enhanced ability to remove background. FCN produces lower score than most competitors, due to poor localization. However, using the fully-connected CRF refinement step (FCN+CRF/SFT), significant improvement is observed, with an increase in f-score by 6.75% on ICDAR 2003 and by 7.05% on ICDAR 2011, achieving the the highest precision and f-score. Note that Lu[18] reaches a higher recall, but much lower precision due to oversegmentation. The stroke width term in the CRF kernel contributes to the improvement in f-score by 0.87% and 1.71% respectively. Detailed analysis shows that even though the modified Stroke Feature Transform sometimes fails with extremely low contrast images, the algorithm can still produce good results thanks to the robust FCN output and the bilateral CRF kernel component.

Table 1. Pixel level segmentation evaluation on ICDAR 2003 dataset.

Method	P	R	F
Niblack[27]	71.10	81.72	76.04
Lu[18]	72.61	**95.48**	82.49
Howe[9]	81.08	87.92	84.36
Mishra[24]	85.20	88.60	86.86
Feild[5]	86.58	87.84	87.21
Tian[37]	87.45	90.63	89.01
Zhou[42]	88.06	90.35	89.19
Tian[36]	88.27	90.18	89.21
Ours (FCN)	83.11	85.11	83.75
Ours (FCN+CRF)	88.80	90.47	89.63
Ours (FCN+CRF/SFT)	**89.96**	91.04	**90.50**

Table 2. Pixel level segmentation evaluation on ICDAR 2011 dataset.

Method	P	R	F
Niblack[27]	77.39	90.33	83.36
Lu[18]	77.26	**95.37**	85.36
Howe[9]	82.50	89.28	85.76
Feild[5]	90.84	89.61	90.22
Tian[37]	87.24	93.85	90.42
Ours (FCN)	84.87	86.91	85.88
Ours (FCN+CRF)	90.03	92.45	91.22
Ours (FCN+CRF/SFT)	**92.04**	93.84	**92.93**

5.2.2 SVT dataset

Compared with the ICDAR 2003 and 2011, the SVT dataset is arguably more challenging. Its images, which are extracted from Google Street View images, tend to be affected by noticeable blur, low contrast, complex background, and large variation in illumination. As shown in Table 3, prior methods produce results with substantially lower quality on this data set. Our system (FCN+CRF/SFT) achieves an f-score of 86.36%, compared with 81.20% for its closest competitor.

Table 3. Pixel level segmentation evaluation on SVT dataset.

Method	P	R	F
Niblack[27]	57.59	78.56	66.46
Howe[9]	69.16	81.32	74.74
Zhou[42]	71.93	87.18	78.82
Tian[36]	76.74	86.22	81.20
Ours (FCN)	77.67	82.43	79.98
Ours (FCN+CRF)	83.01	85.36	84.17
Ours (FCN+CRF/SFT)	**85.34**	**87.40**	**86.36**

5.3. Qualitative Results

In Fig. 6 we show some results of our method (FCN+CRF/SFT) for some challenging cases. The raw FCN probability map output is also shown, along with the output from the classic Otsu binarization algorithm[28] and from the scene text segmentation method proposed by Zhou [42]. Our method produce cleaner segmentation with high recall. It can deal with uncommon font, complex background, reflections, different illumination conditions and poor contrast. With large receptive field and trained multi-scale features, the coarse FCN segmentation produces robust results. The CRF refinement steps allows local details to be captured more faithfully. Key to our approach is the fact that "local" features (color, stroke width) are used only to refine the FCN output, and not to segment text from background, as in traditional algorithms based on MSER or

Figure 6. In this figure we compare the result of Otsu binarization algorithm [28], Zhou's text segmentation algorithm [42] and our method with several challenging images. From top to bottom: input image, Otsu result, Zhou's result, our result and our raw FCN output probability map.

edges. Even when local features become unreliable, FCN, thanks to its large receptive field, gets the job done.

Failure cases includes images with extremely low contrast (Fig. 7, left), as well as images with background similar to the text color and containing pattern consistent with text strokes (Fig. 7, right).

Figure 7. Failure cases.

5.4. Application: Text Substitution

Text substitution [6] is the art of replacing visible text in an image with other text (using different content, font, color, size, or language), in such a way that the rendered image looks "natural". Accurate text stroke segmentation is an important component of text substitution. A typical computational pipeline for text substitution would follow these steps: (1) segment original text strokes; (2) remove text stroke content, substituting with background color or texture; (3) superimpose new text, possibly warped according to the surface orientation [8]. In Fig.8 we show examples of text substitution based on our text stroke segmentation algorithm. The segmentation was first morphologically dilated, then the resulting area was inpainted from nearby background using PatchMatch [2] (with 5×5 patch size).

One intriguing application of text substitution could be in the creation of natural-looking synthetic data sets for training convolutional neural network to perform text detection, segmentation and recognition. In [8], the author proposed a method to find "plain" surface on natural images to render text. However, the data set generated by [8] are not fully realistic, in that the distribution of background textures and context information on which text is superimposed may not match the distribution of real world scenario (see Fig.8 last column). By substituting text in regular scenes, we are able to generate new synthetic images (thus increasing the size of training data) while preserving the "natural" background. With our proposed method, large scale high quality synthetic dataset for multiple languages with character, word and pixel level ground truth labels can be generated.

6. Conclusion

We have presented a new algorithm for text stroke segmentation that produces state of the art results. The algorithms relies on FCN, a robust technique for pixel-level segmentation. FCN, however, cannot precisely localize the stroke edges, due to the large receptive fields of its cells and to its multi-resolution nature. The output of FCN is then refined by a fully connected CRF that uses the assignment probabilities from FCN as unary potentials. Results are further improved by adding to the standard joint energy term of the CRF information about the stroke width, which is computed using a modified Strike Feature Transform. The FCN is trained on a new data set with 100K synthetically gener-

Figure 8. The left column is original images, middle column is our result with text substituted and last column contains samples from dataset generated in [8]. Clearly our synthetic data is more realistic.

ated test images. When tested on standard benchmarks with pixel-level annotations (ICAR 2003, ICDAR 2011, SVT), our algorithm is shown to work very well, with quality (as measured by the f-score) exceeding the state of the art by a sizable margin. We also show promising applications of our algorithm in text substitution.

Acknowledgements

Research reported in this publication was supported by the National Eye Institute of the National Institutes of Health under award number 1R21EY025077-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

[1] Icdar focused scene text dataset. `http://rrc.cvc.uab.es/?ch=2`.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24–1, 2009.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2963–2970. IEEE, 2010.

[5] J. Feild and E. Learned-Miller. Scene text recognition with bilateral regression. *Department of Computer Science, University of Massachusetts Amherst, Tech. Rep. UM-CS-2012-021*, 2012.

[6] V. Fragoso, S. Gauglitz, S. Zamora, J. Kleban, and M. Turk. Translatar: A mobile augmented reality translator. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 497–502. IEEE, 2011.

[7] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent. Large-scale privacy protection in google street view. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2373–2380. IEEE, 2009.

[8] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[9] N. R. Howe. A laplacian energy for document binarization. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 6–10. IEEE, 2011.

[10] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1241–1248. IEEE, 2013.

[11] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *ECCV 2014*, pages 497–511. Springer, 2014.

[12] A. D. Hwang and E. Peli. An augmented-reality edge enhancement application for google glass. *Optometry and vision science: official publication of the American Academy of Optometry*, 91(8):1021, 2014.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[14] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[15] D. Kumar, M. Prasad, and A. Ramakrishnan. Benchmarking recognition results on camera captured word image data sets. In *Proceeding of the workshop on Document Analysis and Recognition*, pages 100–107. ACM, 2012.

[16] X. Liu and J. Samarabandu. Multiscale edge-based text extraction from complex images. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1721–1724. IEEE, 2006.

[17] 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*, pages 3431–3440, 2015.

[18] S. Lu, T. Chen, S. Tian, J.-H. Lim, and C.-L. Tan. Scene text extraction based on edges and support vector regression. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):125–135, 2015.

[19] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 682–687. IEEE, 2003.

[20] C. Mancas-Thillou and B. Gosselin. Color text extraction with selective metric-based clustering. *Computer Vision and Image Understanding*, 107(1):97–107, 2007.

[21] M. Mann. Reading machine spells out loud. *Popular Science*, 154:125–7, 1949.

[22] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[23] S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. Lempitsky. Image binarization for end-to-end text understanding in natural images. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 128–132. IEEE, 2013.

[24] A. Mishra, K. Alahari, and C. Jawahar. An mrf model for binarization of natural scene text. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 11–16. IEEE, 2011.

[25] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3538–3545. IEEE, 2012.

[26] L. Neumann and J. Matas. On combining multiple segmentations in scene text recognition. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 523–527. IEEE, 2013.

[27] W. Niblack. *An introduction to digital image processing*. Strandberg Publishing Company, 1985.

[28] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[29] S. Qin and R. Manduchi. A fast and robust text spotter. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2016.

[30] S. Qin and R. Manduchi. Cascaded segmentation-detection networks for word-level text spotting. *arXiv preprint arXiv:1704.00834*, 2017.

[31] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[32] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33(2):225–236, 2000.

[33] A. Shahab, F. Shafait, and A. Dengel. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1491–1496. IEEE, 2011.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] B. Su, S. Lu, and C. L. Tan. Binarization of historical document images using the local maximum and minimum. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 159–166. ACM, 2010.

[36] S. Tian, S. Lu, B. Su, and C. L. Tan. Scene text segmentation with multi-level maximally stable extremal regions. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2703–2708. IEEE, 2014.

[37] S. Tian, S. Lu, B. Su, and C. L. Tan. Robust text segmentation using graph cut. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 331–335. IEEE, 2015.

[38] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

[39] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1457–1464. IEEE, 2011.

[40] X. Wang, L. Huang, and C. Liu. A novel method for embedded text segmentation based on stroke and color. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 151–155. IEEE, 2011.

[41] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[42] Y. Zhou, J. Feild, E. Learned-Miller, and R. Wang. Scene text segmentation via inverse rendering. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 457–461. IEEE, 2013.