Localization of Text in Photorealistic Images

Grishkin Valery(✉), Ebral Alexander, and Iakushkin Oleg

Saint-Petersburg State University, St.Petersburg 199034, Russia
v.grishkin@spbu.ru

Abstract. Detection and localization of text in photorealistic images is a difficult, and not yet completely solved, problem. We propose the approach to solving this problem based on the method of semantic image segmentation. In this interpretation, text characters are treated as objects to be segmented. In this paper proposes the network architecture for text localization, describes the procedure for the formation of the training set, and considers the algorithm for pre-processing images, reducing the amount of processed data and simplifying the segmentation of the object "background". The network architecture is a modification of well-known DeepLabv3 network and takes into account the specifics of images of text characters. The proposed method is able to determine the location of text characters in the images with acceptable accuracy. Experimental results of assessing the quality of text localization by the IoU criterion (Intersection over Union) showed that the obtained accuracy is sufficient for further text recognition.

Keywords: Text localization · Semantic segmentation · Convolution neural network

1 Introduction

Photorealistic images containing text information comprise a large part of the multimedia content presented on the Internet. These include real photos and videos, in which there are titres, pictures with inscriptions, signs, ads, etc. Automatic detection and reading of text from photorealistic images is an important part of systems such as machine translation based in images, video indexing, robot control, internal navigation, autonomous driving, and other systems that process visual information.

For high-quality recognition of text placed over natural images of the scene, it is necessary to separate the text from a rather complex background, in other words, to segment the image. Normally, the problem of text recognition of a scene is solved in two stages—the stage of detection and localization of text and the stage of textual content recognition [1–3]. There are two approaches to localizing and reading text from photorealistic images.

In the first approach, heuristic features such as HOG [4], MSERs [5], wavelet features [6] or Stroke Width transforms [7] are used to localize text areas. Text recognition is carried out at the second stage using a sliding window and one

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11622, pp. 825–834, 2019.
https://doi.org/10.1007/978-3-030-24305-0_63

of classifiers such as SVM [8] or k-Nearest Neighbor [4,9]. This approach shows good results, but is based on heuristics and requires fine tuning of a large number of parameters depending on the properties of the processed images.

The second approach is based on the localization of text areas using deep neural networks. The paper [1] describes a text detection model that uses a fully convolutional deep neural network to identify text areas. Then the images of the identified areas of the text are passed on to the input of other deep neural networks that perform text recognition. A known system [3] detects text using a convolutional network within a sliding window, which is represented in several resolutions. Character-by-character text recognition is performed in a small window that slides over the identified text area using a CNN-based classifier. In a large number of papers on this topic, various modifications of convolutional neural networks are used, both for localization and for text recognition. In these works, the result of localization are the parameters of the rectangles that limit the text areas. In other words, the result of localization is a set of bounding boxes. At present, works have appeared in which systems are proposed that simultaneously detect and recognize text in images of natural scenes [10,11]. These systems successively localize the boundaries of the text areas and immediately recognize the text content of the found areas.

In this paper, we also use a deep convolutional neural network to localize the text. Unlike traditional text localization methods, we suggest using the semantic segmentation method. This method allows achieving high-quality segmentation of various objects on a complex background [12,13]. The methods of semantic segmentation are good in localizing objects representing closed and sufficiently large areas of the image. However, text characters in most cases consist of relatively thin lines and occupy a small area. Therefore, it is not possible to use neural networks designed for segmentation of large objects directly for text localization. However, it is possible to modify the architecture of well-proven deep convolutional neural networks for working with images containing text characters.

2 Convolution Neural Network for Text Localization

Quality of segmentation of objects is estimated by criterion of IoU (Intersection over Union). Currently, the DeepLabV3 neural network [14] shows the best segmentation results. For this network, IoU criterion reaches 77% on the PASCAL VOC 2012 data set [15]. Therefore, we chose architecture of this network as the basis for developing a new network for symbol segmentation.

2.1 Network Topology

Figure 1 shows the proposed topology of network for text localization. This topology is similar to the topology of DeepLabv3 network. It contains sequential convolutional layers with average pooling. Network output unit performs atrous spatial pyramid pooling (ASPP).

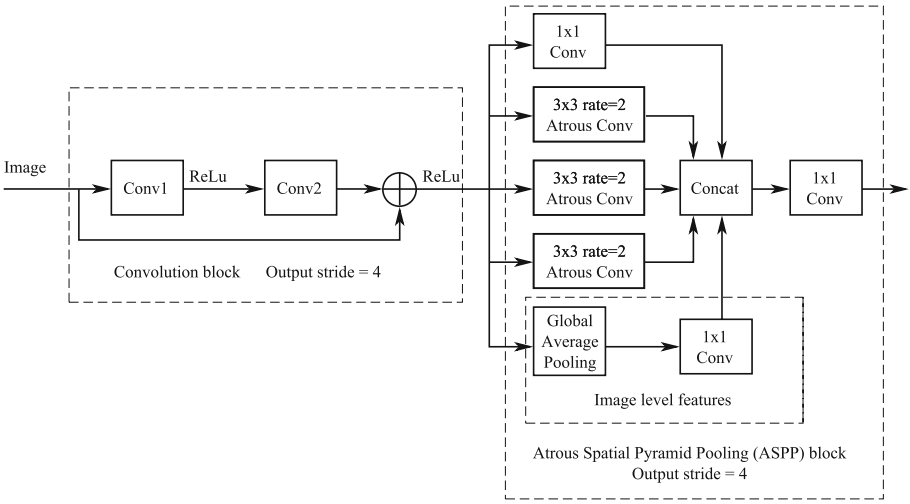

Fig. 1. Topology of network for text localization.

The ASPP block uses several parallel layers of extended convolutions with different spatial steps (atrous rate) between convolution weights. These expanded convolutions allow you to increase the perceptual field without reducing the spatial dimension. This block also calculates image level features. These low-level features are obtained by processing the output of the convolutional block using global average pooling. The result of the pooling is then processed by convolution 1×1. Output features of all five branches are combined and passed through a 1×1 convolution. The resulting image at the output of the ASP block is the segmentation mask.

Convolution of the original DeepLabv3 network is configured so that output stride i.e. the value of the ratio of the resolution of the input image to the resolution of the output image equals 16. This approach is not valid for text segmentation because the geometric structure of text symbols is not preserved. Considering these differences, as well as the fact that the text symbols are of a simpler geometric structure than the objects of the PASCAL VOC 2012 dataset, we suggest reducing the output stride to 4.

Each convolutional block of the original DeepLabv3 network contains two image paths: a short and a long one - through several consecutive convolutions. This allows us not to lose the data that turned into zero at convolutions. This property is useful for presenting text, since the loss of part of the fine lines that make up the text can affect the overall quality of segmentation. Usually for segmentation of complex objects of many classes several blocks of the structure shown in Fig. 2 are used. To ensure the output stride value of 4, we suggest using only one block of this type. The proposed block contains only two consecutive convolutions, to the outputs of which the activation function ReLu is applied, followed by averaging pooling with a 4×4 window and a shift parameter of two.

Each convolution in the proposed block is carried out with a 3×3 kernel. The total number of filters is chosen equal to 512. The dimension of the convolution kernels and the number of filters was determined experimentally during network training.

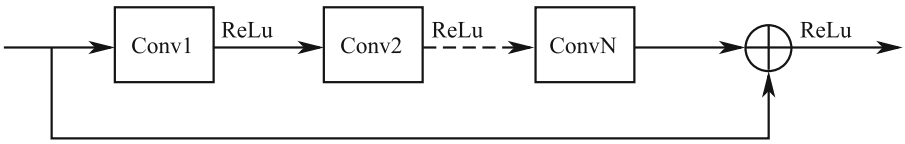

Fig. 2. Convolutional block of the original DeepLabv3 network.

The ASPP block is similar to the same DeepLab V3 block. It consists of four parallel convolutions—one convolution with a 1×1 core, three expanded convolutions with a 3×3 core, and has an image level features calculation unit. The difference is that for extended convolutions, the spatial steps between the convolution weights have been reduced. We chose the following values of atrous rates—2,4, and 6, while in the original network their values were—6,12, and 18. We apply triple reduction in rate because the text consists of relatively thin lines. Note that convolution with a 3×3 kernel and a maximum rate of six covers a line width of 15 pixels, which seems sufficient for text localization. Like in the original network, we combine the resulting feature maps and pass them through convolution 1×1.

2.2 Dataset

To train the proposed network, a pre-marked dataset of photorealistic images containing text is required. Specialized datasets, such as ICDAR2013 [16], SVT [17] or similar, are commonly used to develop and test text detection and recognition systems. However, these sets have no areas marked in the image that contain only text symbols. In addition, there are no Cyrillic characters in all these data sets. Currently, only one public dataset is known that is suitable for the subject area—Chars74K [18]. However, the quantity and quality of images, as well as annotations of the text in them, are quite low.

Thus, to apply the semantic segmentation method for text detection, we created our own dataset. This set is generated on the basis of a set of a sufficiently large number of existing fonts, which take into account various options for displaying characters of a given alphabet. When generating a set, we formed the text, which was then superimposed, with a certain level of transparency, in the photorealistic image. The text itself can be of any size, color, and angle of inclination, displayed in one or several fonts, and located anywhere in the image.

The text overlay algorithm to the original image is as follows: we divide the image into 4 blocks and superimpose the pre-formed text with the specified

transparency to each image block. The position and orientation of the text in the block is randomly selected.

To generate text, we used lower case letters of the Cyrillic alphabet, numbers, and common punctuation marks (‘.’, ‘,’, ‘;’, ‘-’, ‘:’, ‘!’)—49 characters overall. The text generation algorithm randomly selects characters from this alphabet and then forms a long string of several words. The length of each word is chosen randomly from the range of lengths of Russian words. The number of words in a long string is chosen randomly in the range of 1 to 40. The text block in each image is printed with one of 30 standard TrueType fonts. The font type and its size and color are all selected randomly. It should be noted that the text color is ensured to differ from the background in brightness. If a word in a long string is outside the block, it is truncated, and the new text line of the block begins with the next word.

As background images, 5000 photorealistic images that do not contain text are used. Usually text symbols shade the background and differ from it in brightness. The Fig. 3 an example of the generated image for our data set.



Fig. 3. An example of the generated image.

Together with the text overlay procedure, two masks are also created. These masks are used to markup the generated dataset. The first mask is a binary image of white text on a black background. The second mask is a grayscale image of the same text, where the brightness of the symbol corresponds to the position of the symbol in the alphabet. We use the first mask for binary segmentation between the text and the background, and the other mask for multi-class segmentation of character type vs. background.

2.3 Preprocessing

The main purpose of the preprocessing is to try to reduce the variability of the background and highlight areas in the image whose brightness is different from the background.

Before serving to the input of the neural network, we convert the original color image to grayscale. Then, using the Canny operator on this halftone image, we search for the boundaries of the brightness differences. Since the text is different in brightness from the background, this allows highlighting the borders of the characters. At the same time, all other rather sharp differences in brightness in the halftone image are also highlighted. The generated binary image of all boundaries is processed using the morphological dilation operation. Then, element-wise multiplication of the resulting binary mask by a halftone image is performed. This processing allows for filtering out most of the background, keeping just information about the brightness near the borders, which simplifies the task of separating the background borders from the text borders. Figure 4 shows the structure of the preprocessing algorithm. Figure 5 shows an example of image processing using the proposed algorithm.

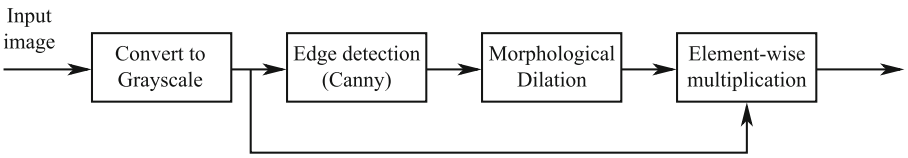

Fig. 4. The preprocessing algorithm.

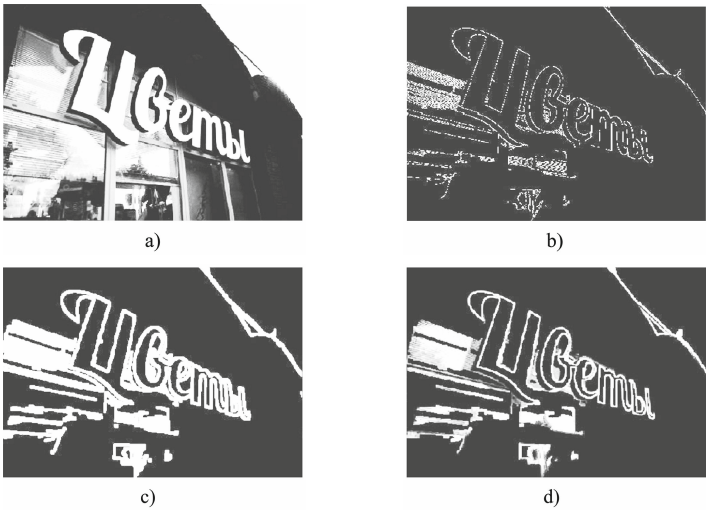

Fig. 5. The preprocessing stages: (a) the original image; (b) the mask of borders; (c) the mask of borders after morphological dilation; (d) the result of element-wise multiplication.

The proposed preprocessing algorithm simplifies the segmentation of the "background" class, which is usually diverse and therefore requires large computational costs for network training. Segmentation of a background consisting of zero-brightness pixels requires less computation.

3 Experimental Results

The proposed neural network is implemented in Python using the TensorFlow machine learning library for the GPU platform. Software for generating the necessary dataset is also implemented in the Python language. We trained our network model using 4000 training images and evaluated it against 1000 verification images from the generated dataset. The localization quality is evaluated by the metric IoU (Intersection over Union).

We explored two approaches to the segmentation of images containing text. In the first approach, the network was trained to recognize two classes—background and text. In the second approach, the network was trained to recognize 50 classes: 49 characters and background.

When training networks, we used the batch normalization procedure, and cross-entropy as a function of losses. The following condition serves as the criterion for stopping the training:

$$\mathrm{IoU}_i - \mathrm{IoU}_{i-1000} < 0.01,$$

where IoU_i the average metric value obtained during training, i is the current iteration number.

For binary segmentation (background vs. text), the learning rate was chosen experimentally and amounted to $5 \cdot 10^{-3}$. A larger coefficient led to a drop in accuracy, and a smaller one—to an increase in the training time, but at the same time, the accuracy did not increase.

For multi-class segmentation (background vs. symbols), it was found out that the learning rate used for binary classification is too high. Therefore, the new coefficient was taken an order of magnitude less, which increased the training time, but at the same time, the growth of the metric was more stable. The remaining parameters of training did not change.

Figure 6 shows the process of training the network for segmentation of the background vs. the text, while Fig. 7 shows training process for segmentation of background vs. symbols. All of these charts show the change of the IoU metric value during network training.

For binary segmentation, the maximum value of the metric over the training set for the background was 94.8%, while the value of the same metric for the text was 58.6%, with average of 76.7%. For multi-class segmentation, the average maximum value of the metric was much less and amounted to 47.6%.

Figures 8 and 9 show the results of binary and multi-class segmentation for the image shown in Fig. 5, respectively. Segmentation was performed using the proposed trained neural networks. Note that in the case of background vs. symbols segmentation, the accuracy drops significantly. However, both approaches can be combined and used to clarify the results of each other.

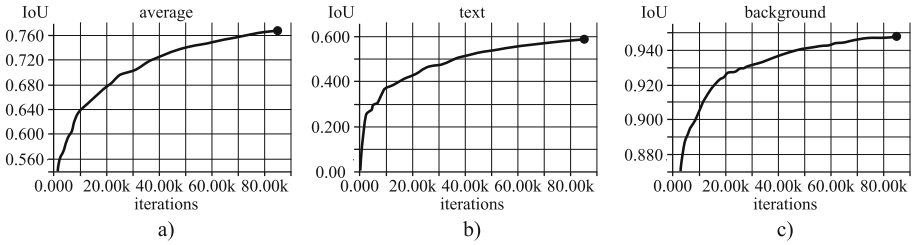

Fig. 6. Background vs. text segmentation. Changing IoU metrics during training: (a) the average value of the metric; (b) metric value for the text class; (c) metric value for the background class.

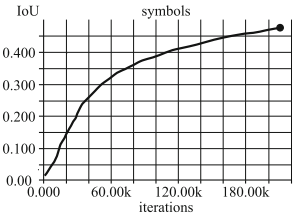

Fig. 7. Background vs. symbols segmentation. Changing average IoU metrics during training.



Fig. 8. An example Background vs. text segmentation: (a) real mask; (b) predicted mask.



Fig. 9. An example Background vs. symbols segmentation: (a) real mask; (b) predicted mask.

4 Conclusion

In this paper, we applied the semantic segmentation method to detect and localize text in photorealistic images. Deep neural networks that implement this method are designed to localize and recognize large objects in an image, but cannot be used for text segmentation. Therefore, we propose a modification of the well-known network DeepLabV3, which showed good results of semantic segmentation. Unlike the original network, our modified neural network keeps the geometric structure of text characters into the feature maps. To reduce the computational cost of network training, we also propose the image preprocessing algorithm that simplifies background segmentation.

Known datasets that are used for both semantic segmentation and text detection are not suitable for semantic segmentation of text characters. Thus, to apply the semantic segmentation method for text localization, we created our own dataset.

The proposed neural network is implemented using the TensorFlow machine learning library with support for GPU computing. The quality of the localization of the text characters and the background is evaluated experimentally by the metric IoU. With the validation data set, the value of this criterion reaches 78%, which is sufficient for further processing of the segmented image using OCR systems.

Acknowledgment. The authors acknowledge Saint-Petersburg State University for a research grant 39417687.

References

1. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2315–2324. IEEE, Las Vegas, NV (2016). https://doi.org/10.1109/CVPR.2016.254
2. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: TextBoxes: A fast text detector with a single deep neural network. In: 31st AAAI Conference on Artificial Intelligence, pp. 4161–4167. AAAI, San Francisco (2017)
3. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman A.: Reading text in the wild with convolutional neural networks. Int. J. Comput. Vis. **116**(1), 1–20 (2016). https://doi.org/10.1007/s11263-015-0823-z
4. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision, pp. 1457–1464. IEEE, Barcelona, Spain (2011). https://doi.org/10.1109/ICCV.2011.6126402
5. Neumann, L., Matas, J.: A method for text localization and recognition in real-world images. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6494, pp. 770–783. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19318-7_60
6. Grishkin, V.: Document image segmentation based on wavelet features. In: 2015 Computer Science and Information Technologies (CSIT), pp. 82–84. IEEE, Yerevan, Armenia (2015). https://doi.org/10.1109/CSITechnol.2015.7358255

7. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2963–2970. IEEE, San Francisco, CA (2010). https://doi.org/10.1109/CVPR.2010.5540041
8. Yao, C., Bai, X., Shi, B., Liu, W.: Strokelets: A learned multi-scale representation for scene text recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 4042–4049. IEEE, Columbus, OH (2014). https://doi.org/10.1109/CVPR.2014.515
9. Wang, K., Belongie, S.: Word spotting in the wild. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 591–604. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15549-9_43
10. Bartz, C., Yang, H., Meinel, C.: STN-OCR: A single neural network for text detection and text recognition. https://arxiv.org/abs/1707.08831. Accessed 20 Mar 2019
11. Busta, M., Neumann, L., Matas, J.: Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2223–2231. IEEE, Venice, Italy (2017). https://doi.org/10.1109/ICCV.2017.242
12. Chen, L., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4545–4554. IEEE, Las Vegas, NV (2016). https://doi.org/10.1109/CVPR.2016.492
13. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. https://arxiv.org/abs/1412.7062. Accessed 10 Mar 2019
14. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. https://arxiv.org/abs/1706.05587. Accessed 20 Mar 2019
15. Pascal VOC data set mirror. https://pjreddie.com/projects/pascal-voc-dataset-mirror/. Accessed 27 Feb 2019
16. Karatzas, D. et al.: ICDAR 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1484–1493. IEEE, Washington, DC (2013). https://doi.org/10.1109/ICDAR.2013.221
17. The Street View Text Dataset (SVT). http://tc11.cvc.uab.es/datasets/SVT_1. Accessed 3 Mar 2019
18. The Chars74K dataset. http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/. Accessed 10 Feb 2019