TEXT SEGMENTATION ON PHOTOREALISTIC IMAGES

Valery Grishkin [a], Alexander Ebral [b], Nikolai Stepenko [c], Jean Sene [d]

Saint Petersburg State University, 7–9 Universitetskaya nab., Saint Petersburg, 199034, Russia

E-mail: [a] valery-grishkin@yandex.ru, [b] aleksandr.ebr@gmail.com, [c] n.stepenko@spbu.ru, [d] senejeanvalery@yahoo.fr

The paper proposes an algorithm for segmentation of text, applied or presented in photorealistic images, characterized by a complex background. The algorithm is able to determine the exact location of image regions containing text. It implements the method for semantic segmentation of images, while the text symbols serve as detectable objects. The original images are pre-processed and fed to the input of the pre-trained convolutional neural network. The paper proposes a network architecture for text segmentation, 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 ResNet network and takes into account the specifics of text character images. The convolutional neural network is implemented using CUDA parallel computing technology at the GPU. The experimental results for evaluating quality of text segmentation with IoU (Intersection over Union) criterion have proved effectiveness of the proposed method.

Keywords: text segmentation, 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 pictures with the text applied to them later, the video in which there are titles, pictures with inscriptions, signs, ads, etc. For high-quality text recognition in such images, it is necessary to separate the text from a rather complex background, in other words, to segment the image. Two approaches to segmentation are possible.

The first approach is based on finding the image areas containing text and defining, for these areas, the parameters of the minimum bounding rectangles. Within this approach, classifiers using heuristic features are constructed [1-3]. Commonly classifiers like SVM and Random Forest is used. The second approach is based on semantic segmentation methods that localize regions containing objects that are present in the image. These methods use classifiers based on convolutional neural networks [4 - 6].

In this paper, we propose to use the second approach for text segmentation. It allows to achieve high quality segmentation of various objects on a complex background. 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 segmentation. However, it is possible to modify the architecture of well-proven deep convolutional neural networks for working with images containing text characters.

2. Convolutional neural network for text segmentation

Quality of segmentation of objects is estimated by criterion of IoU (Intersection over Union) Currently, the DeepLabV3 neural network [7] shows the best segmentation results. For this network, IoU criterion reaches 77% on the PASCAL VOC 2012 data set [8]. 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 segmentation. 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).

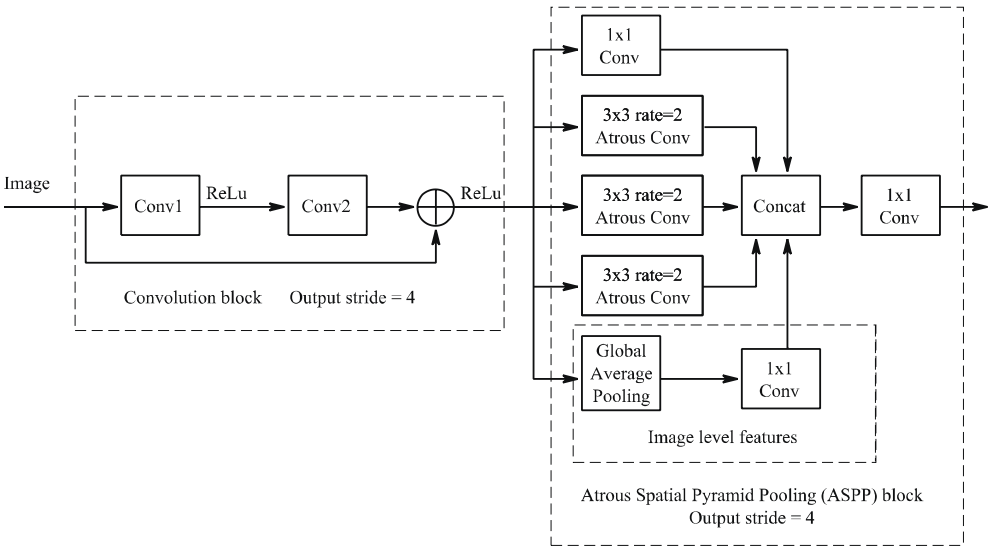

Figure 1. Topology of network for text segmentation

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. The 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 1x1. Output features of all five branches are combined and passed through a 1x1 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 paths of image passage: short and long — through several consecutive convolutions. This allows us not to lose the data that turned into 0 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 Figure 2 are used. To ensure the output stride value of 4, we suggest using only one block of this type. The proposed block contains two consecutive convolutions, to the outputs of which the activation function ReLu is applied, followed by averaging pooling with a 4x4 window and a shift parameter of two.

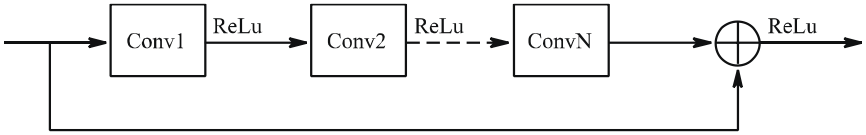

Figure 2. Convolutional block of the original DeepLabv3 network

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 segmentation. Like in the original network, we combine the resulting feature maps and pass them through convolution 1x1.

2.2. Dataset

To train the proposed network, a pre-marked set of photo-realistic images containing text is required. Currently, only one public dataset is known that is suitable for the subject area - Chars74K [9]. However, the quantity and quality of images, as well as annotations of the text in it, are quite low. Therefore, we have created our own data set to solve the problem. This set is generated on the basis of a collection of a sufficiently large number of existing fonts that take into account the various options for displaying the characters of an alphabet. When generating a set, we formed the text, which was then superimposed, with a certain level of transparency, on 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.

To generate text, we used lower case letters of the Russian alphabet, numbers, and common punctuation marks ('.', ',', ';', '-', ':', '!') – 49 characters overall. The algorithm randomly selects characters from this alphabet and then picks nouns of the Russian language from the dictionary. Then it applies the text to the image using one of 30 free TrueType fonts. The font type and its size and color are all selected randomly. As background images, 5,000 photorealistic images that do not contain text are used. The algorithm of applying text to the original image is like the following: we divide the image into 4 blocks and apply the pre-shaped text to each image block. Again, the position and orientation of the text in the block is randomly selected.

When applying text to a background image, two masks are also formed that serve to mark up the dataset being created. 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

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, the boundaries of the brightness differences are searched. Since the text is different in brightness from the background, this allows to highlight 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, leaving just information about the brightness near the borders, which simplifies the task of separating the background borders from the text borders. Figure 3 shows the structure of the preprocessing algorithm.

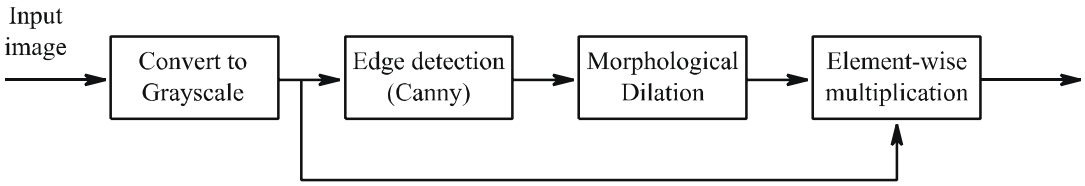

Figure 3. The preprocessing algorithm

The proposed preprocessing algorithm simplifies the segmentation of the “background” class, which is usually diverse and therefore requires large computational costs for network training. For segmentation of the background consisting of pixels of zero brightness, less computation effort is.

3. Experimental results

The proposed neural network implemented in the Python language using the TenorFlov 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 on 4000 training images and evaluated it on 1000 verification images from the generated dataset. The segmentation quality is evaluated by the metric IoU. Figure 4 shows the results of the background - text segmentation, and in Fig 5 the results of the background - symbols segmentation. All of these graphs show the dependence of the IoU metric value during network training.

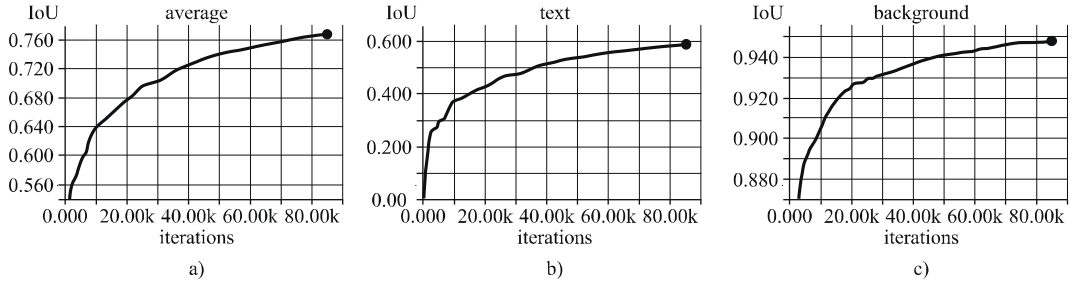

Figure 4. Background – text segmentation. Changing IoU metrics during training: a) the average value of the metric; b) metric value for the text class; c) metric value for the background class.

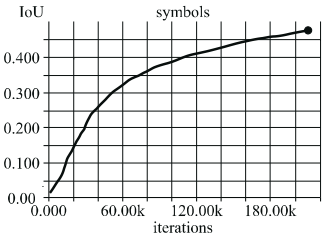

Figure 5. Background – symbols segmentation. Changing average IoU metrics during training

Note that in the case of background – symbols segmentation, the accuracy is significant falls. However, both approaches can be combined and used for clarify the results of each other.

4. Conclusion

We propose the algorithm for segmentation of text regions on photorealistic images. It consists of a preprocessing step, a recognition step, and a localization step. The second step uses the modified convolutional network DeepLabV3 for recognition. Unlike the original network, the modified neural network saves the geometric structure of text characters into the feature maps. The third step determines the exact localization of recognized text characters and finds areas of the image containing text. Experimental results show the effectiveness of the proposed algorithm. The segmentation quality is evaluated by the metric IoU and reaches 78%, which is sufficient for further processing of the image text using OCR systems. The use of parallel processing technology significantly reduces the processing time of large series of images

5. Acknowledgement

The authors acknowledge Saint-Petersburg State University for a research grant 11756691.

References

[1] Zhang Jing, Dong Wei, Zhang Youhui. An Algorithm for Scanned Document Image Segmentation Based on Voronoi Diagram // Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on(Volume:1), 2012, pp. 156–159.

[2] H. S. Baird, M. A. Moll, Chang An, Document Image Content Inventories // Proc. of SPIE/IS&T Document Recognition & Retrieval, 2007.

[3] Grishkin, V. Document Image Segmentation Based on Wavelet Features // CSIT 2015 - 10th International Conference on Computer Science and Information Technologies. pp. 82-84. 2015. - DOI:: 10.1109/CSITechnol.2015.7358255

[4] S. Chandra and I. Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs. Available at: https://arxiv.org/abs/1603.08368 (accessed 11.05.2018)

[5] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In CVPR, 2016.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In ICLR, 2015.

[7] L.-C.Chen, G.Papandreou, F.Schroff, and H.Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. Available at: https://arxiv.org/abs/1706.05587 (accessed 18.04.2018)

[8] Pascal VOC data set mirror. Available at: https://pjreddie.com/projects/pascal-voc-dataset-mirror/ (accessed 19.04.2018)

[9] The Chars74K dataset. Available at: http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/ (accessed 20.03.2018)