DeepLayout: A Semantic Segmentation Approach to Page Layout Analysis

Yixin Li, Yajun Zou, and Jinwen Ma(✉)

Department of Information Science, School of Mathematical Sciences and LMAM, Peking University, Beijing 100871, China
{liyixin, zouyj}@pku.edu.cn, jwma@math.pku.edu.cn

Abstract. In this paper, we present DeepLayout, a new approach to page layout analysis. Previous work divides the problem into unsupervised segmentation and classification. Instead of a step-wise method, we adopt semantic segmentation which is an end-to-end trainable deep neural network. Our proposed segmentation model takes only document image as input and predicts per pixel saliency maps. For the post-processing part, we use connected component analysis to restore the bounding boxes from the prediction map. The main contribution is that we successfully bring RLSA into our post-processing procedures to specify the boundaries. The experimental results on ICDAR2017 POD competition dataset show that our proposed page layout analysis algorithm achieves good mAP score, outperforms most of other competition participants.

Keywords: Page layout analysis · Document segmentation
Document image understanding · Semantic segmentation and deep learning

1 Introduction

Page layout analysis, also known as document image understanding and document segmentation, plays an important role in massive document image analysis applications such as OCR systems. Page layout analysis algorithms take document images or PDF files as inputs, and the goal is to understand the documents by decomposing the images into several structural and logical units, for instance, text, figure, table and formula. This procedure is critical in document image processing applications, for it usually brings a better recognition results. For example, once we get the structural and semantic information of the document images, we only feed the text regions into the OCR system to recognize text while the figures are saved directly. Thus, page layout analysis has become a popular research topic in computer vision community.

Most of the conventional methods [1–5] have two steps: segmentation and classification. Firstly, the document images are divided into several regions, and then a classifier is trained to assign them to a certain logical class. The major weakness of these methods is that the unsupervised segmentation involves lots of parameters that rely on experience, and one set of parameters can hardly fit all the document layout styles.

To tackle this problem, the most straightforward way is supervised localization or segmentation. The parameters in supervised learning can be tuned automatically during

D.-S. Huang et al. (Eds.): ICIC 2018, LNAI 10956, pp. 266–277, 2018.
https://doi.org/10.1007/978-3-319-95957-3_30

the training, which avoids the large amount of human-defined rules and hand-craft parameters. On the other hand, supervised segmentation provides semantic information which means we can perform segmentation and classification simultaneously.

Since the final output of the page layout analysis is a number of bounding boxes and their corresponding labels, this problem can be framed as an object detection or localization problem. Unfortunately, the state-of-the-art object detection approaches such as Faster R-CNN [6] and Single-Shot Multibox Detector (SSD) [7] have been proven not working very well in the page layout analysis case [1]. This is because the common object detection methods are designed to localize certain objects in real life such as dogs, cars and human, and most of them have a specific boundary unlike text and formula regions and are not likely to have an extremely aspect ratio like text line in page layout analysis case. Also, the error causing by the bounding box regression is inevitable, which is usually the final stage of common object detection networks.

To address this issue, we adopt semantic segmentation approach to classify each pixel into their semantic meaning like text, formula, figure or table. The semantic segmentation model is a deep neural network trained under supervised information where parameters are learned automatically during training. Moreover, the pixel level understanding from semantic segmentation is more precise than the bounding box level understanding from the conventional object detection methods.

In this paper, we propose a page layout analysis algorithm based on semantic segmentation. The pipeline of our proposed algorithm contains two parts: the semantic segmentation stage and the post-processing stage. Our semantic segmentation model is modified on DeepLab [8] to fit our problem. As for post-processing part, we get the bounding box locations along with their confidence scores and labels by analyzing the connected components on the probability map generated from our segmentation model and adopt the run length smoothing algorithm (RLSA) locally on the original image to modify the bounding boxes. It is demonstrated in the experiments that our proposed algorithm achieves both reasonable visualization results and good quantization results on the ICDAR2017 POD competition dataset [9].

The main contribution of this paper is three fold. First, we propose a powerful and efficient approach on page layout analysis. Also, we successfully contrast a coarse-to-fine structure by combining the supervised learning algorithm (DeepLab) and unsupervised algorithm (RLSA). Finally, though the optimization targeted on POD dataset may not simply be applied to other datasets, the ideas have good extension meaning and reference value.

The rest of the paper is organized as follow: we briefly review the page layout analysis and semantic segmentation algorithms in Sect. 2. The methodology of our page layout analysis algorithm is then presented in the next section. In Sect. 4, the datasets we used and the experiments we conducted are described in detail. And discussions of the limitation and running time analysis are also given in Sect. 4. Finally, we conclude the whole paper in the last section.

2 Related Work

2.1 Page Layout Analysis

Page layout analysis has been studied for decades and a great number of algorithms have been established and developed to solve this problem. Most of the page layout analysis approaches can be roughly divided into two categories by the type of segmentation algorithms. One of them is unsupervised segmentation by hand-craft features and human-defined rules, and the other is supervised segmentation by a learning based model and supervised training data.

Most of the conventional methods [1–5] adopt a two-step strategy: an unsupervised segmentation model and a learning based classification model. An unsupervised segmentation method is either starting from the pixels then merging them into high level regions (bottom-up) [2, 3] or segmenting the page into candidate regions by projections or connected components (top-down) [4, 5]. Both need a large amount of experience-depended parameters. And as for classification step, hand-craft features are extracted to train a classifier [3, 4] or a CNN is trained to classify the segmented regions [1]. Also, some algorithms are proposed to detect the specific type of boxes or regions like equations [10, 11] and tables [12] in the PDF files or document images. As we mentioned before, the two-step algorithms mostly contain lots of human-defined rules and handcraft features which involve a large parameter set.

In recent years, supervised segmentation is introduced to solve the page layout case [13, 14]. Supervised segmentation provides semantic information which allows us to perform segmentation and classification at the same time. Oyebade et al. [13] extracts textural features from small patches and trains a neural network (fully connected) to classify them to get the document segmentation result. Due to the patch classification, there is a so-called "mosaic effect" where the segmentation boundary is rough and inaccurate. Yang et al. [14] first introduce semantic segmentation to document segmentation, and an additional tool (Adobe Acrobat) is adopted to specify the segmentation boundary. By the power of deep learning, this type of methods is normally faster and stronger than two-step ones and is easier to generalize to other type of documents.

2.2 Semantic Segmentation

Semantic segmentation is a computer vision task that aims to understand the image in pixel level by performing the pixel-wise classification. A demonstration of the semantic segmentation task is shown in Fig. 1[1]. Semantic segmentation is a deeper understanding of images than image classification. Instead of recognizing the objects in image classification task, we also have to assign each pixel to a certain object class or background to lineate the boundary of each object. In the industry, semantic segmentation is widely used in a variety of computer vision scenarios such as image matting, medical image analysis and self-driving.

[1] http://cocodataset.org/#detections-challenge2017.

Before deep learning, the commonly used solutions are random forest based algorithms [15], this kind of algorithms are inaccurate and extremely slow. With CNN taking over computer vision, one of the first attempts on page layout analysis by CNN was patch classification [16] where the pixel class is assigned based on the classification result on a small image patch around it. The size of image patches need to be fixed due to the fully connected layer used in the network structure. And to keep the parameter size acceptable, the patch window which equals the receptive field needs to be small. Thus the segmentation result was still not ideal.

In 2014, Fully Convolutional Network (FCN) was proposed by Long et al. [17] which is a milestone of semantic segmentation. This model allows us to feed the images in any size to the segmentation model because no fully connected layer was used in the network structure. Also the end-to-end trainable structure makes it much faster and much stronger than the patch classification methods.

The next big problem of using CNN on segmentation is the pooling layers. Pooling layers increase the receptive field and robustness while weakening the spatial information. Spatial information is unimportant in image

Noticed that our proposed deep learning based page layout analysis algorithm takes only document image as input, unlike previous work taking benefits from structural information in PDF file [11] or applying additional commercial software to localize the logical units [14].

3.2 Semantic Segmentation Model

Fully Convolutional Network (FCN) [17] represents a milestone in deep learning based semantic segmentation. End-to-end convolutional structure is first introduced and deconvolutional layers are used to upsample the feature maps. However, loss of spatial information during pooling stage makes the upsampling produce coarse segmentation results which leaves a lot of room for improvement.

DeepLab [18] is proposed to overcome this problem and achieves state-of-the-art performance. So we choose DeepLab v2 structure as our segmentation model, and take ResNet-101 [19] as the backbone network. The key point in the network structure is that we use a set of dilated convolution layers to increase the receptive field without

Consistency Loss. The semantic segmentation model is designed to capture objects with any shapes but the logical units in document images are all rectangles. Inspired by Yang et al. [14], we implement the consistency loss to penalize the object shapes other than rectangle. The training loss is the segmentation loss combining with the consistency loss. And the consistency loss is defined as follow:

$$\mathcal{L}_{\text{con}} = \frac{1}{|gt|} \sum\nolimits_{p_i \in gt} (p_i - \bar{p})^2 \tag{1}$$

Where gt is the ground truth bounding box, $|gt|$ is number of pixels in the ground truth box, p_i is the probability given by the segmentation Softmax output, and $\bar{p}$ is the mean value of all the pixels in the bounding box.

Training Details. The segmentation model we use is based on DeepLab v2 [18]. All the layers except the last prediction layer are restored on the model pretrained on MSCOCO dataset [21]. And the last layer is random initialized to predict four classes: background, figure, formula, and table. Parameters like learning rate, weight decay and momentum are inherited from the Tensorflow implementation of DeepLab v2[2].

Connected Component Analysis (CCA). To restore the bounding boxes from the saliency map predicted by our segmentation model, we extract connected components on each class then take the bounding rectangles as candidate bounding boxes. Label of each candidate bounding box is the same as the connected component and the confidence score is calculated by the average of the pixel segmentation Softmax scores.

Run Length Smoothing Algorithm (RLSA). RLSA is widely used in the document segmentation to aggregate the pixels in the same logical unit for last few decades [2, 3]. The input of RLSA is binary image or array where 0 s represent black pixels and 1 s represent white pixels. The aggregation procedure is under the rule that 1 s are changed to 0 s if the length of adjacent 1 s is less than a predefined threshold C. An example of RLSA on 1d array (RLSA threshold C is set to 3) is shown in Fig. 3(a) and an example of RLSA document segmentation procedure is shown in Fig. 3(b).

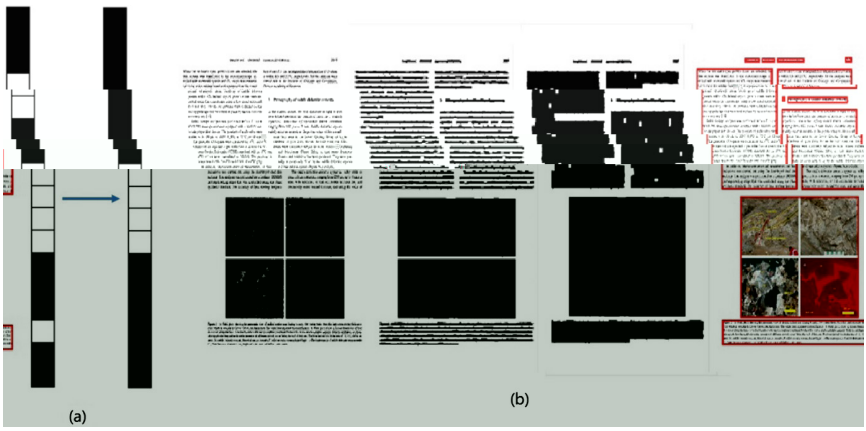

Fig. 3. (a) A 1d example of RLSA procedure. (b) A document segmentation example by RLSA.

RLSA is under the assumptions that the horizontal or vertical distance between black pixels in the same logical region is less than C while distance between different logical region is large than C. But there are several cases that do not meet these assumptions. For example, image captions are usually very close to the image, but they are different logical regions. Thus the determination of threshold C could be very hard and experience dependent.

In our case, since we have semantic meaning of each pixel by our segmentation model, we are able to apply RLSA on each connected component where pixels are in the same logical meaning. For the caption case, the figure and its caption are processed separately, they thus won't be aggregated together no matter how close they are. Semantic segmentation model gives us the logical class label of each pixel but the boundary is rough which is shown in Fig. 2, and local RLSA is adopted to gives us the exact boundary of each logical region.

Some Other Processing Steps. We investigate the ground truth of POD competition dataset [9] and design several rules to improve the mAP score. Note that this part is

designed specifically for the POD competition dataset, so it may not be able to generalize to all types of document images. We briefly introduce the problems instead of solutions, for this part is not the focus of this paper.

We noticed that each subfigure is annotated separately in the POD dataset and the segmentation model tends to predict a single figure, so we set series of rules to split the figure regions into several subfigures. Tables normally have a clear boundary, so besides removing small regions, there is no additional processing step for tables.

Standard of equation annotation is unclear in POD dataset. Most of equations are annotated as "equation line" in POD dataset where multiline equation is labeled as multiple equation annotations. But some equations are annotated in "equation block". Also, the equation number is annotated in the same box with the corresponding equation. Equation number may very far away from the equation which leaves the annotated box a large blank space. Therefore, some human-defined rules are designed to split equations into single lines and aggregate equation number and equation itself. The result is still far from ideal, for the splitting procedure creates a new problem (the start and stop indexes of "Σ" and "Π") which will be discussed in Sect. 4.

4 Experiments

4.1 Datasets

Since our segmentation model is deep learning based, we need annotated data for training. We choose the open dataset for ICDAR2017 Page Object Detection Competition [9]. The competition dataset contains 2,400 annotated document images (1600 for training and 800 for testing) and the extended dataset which is not open for the competition contains about 10,000 annotated images. The document images are annotated by bounding boxes with three classes: figure, table and formula. The competition dataset can be downloaded on the official website[3].

4.2 Experimental Results

The evaluation metric we use to quantize the segmentation performance is mean IoU. Mean IoU is a standard metric in semantic segmentation [8, 17, 18] which calculates the mean intersection over union metrics on all classes. Also, mean average precision (mAP) and average F1 measure are calculated to evaluate the final page layout analysis results. Mean average precision is a rough estimation of area under precision-recall curve and is the most common used evaluation on object detection [6, 7]. F1 measure is the harmonic mean value of precision and recall. In particular, mAP and F1 are also the evaluation metrics used by POD competition [9].

The segmentation models are trained on POD dataset and POD extended dataset to show the performance gain from more training data. And post-processing methods are applied to prediction maps generated from the segmentation model. The results of different training datasets and different post-processing steps are shown in Table 1.

[3] https://www.icst.pku.edu.cn/cpdp/ICDAR2017_PODCompetition/index.html.

The results are evaluated by official evaluation tool of POD competition. And IoU threshold is set to 0.8 in mAP and average F1 calculation.

Table 1. The segmentation and final detection performance of our proposed methods.

	Mean IoU	Average F1	Mean AP
Base	0.869	0.518	0.498
Base + RLSA	0.869	0.763	0.666
Base(+) + RLSA	**0.908**	**0.801**	**0.690**
Base(+) + CRF + RLSA	0.886	0.745	0.611
Base(+) + CL + RLSA	0.897	0.776	0.662

In Table 1, "base" represents our segmentation model, "RLSA" represents our whole post-processing steps, "(+)" means the extended dataset is used to train the segmentation model, "CRF" represents segmentation model with fully connected CRF layers [22] and "CL" represents consistency loss considered during training.

From Table 1 we can see that the best result comes from segmentation model trained on extended dataset, follow by our proposed post-processing procedure including RLSA. The most significant performance gain is from our proposed RLSA post-processing which boosts the average F1 for 0.21 and mAP for 0.12. In the segmentation network, a larger training dataset gives us a 4% gain, for the deep learning structure we use heavily relies on large amount of training data. And the fully connected CRF which usually improves the segmentation results on real life objects, does not work well on page layout case. The reason is that objects in natural images have a clear boundary while logical units in document images have holes and blanks which is inadequate for CRF post-processing. Also the consistency loss is supposed to penalize the predictions with shapes other than rectangle. But in our experiments, some predictions vanished to have a zero consistency loss, thus the segmentation and final results are not what we expected.

Then we compare our best method with the top results submitted in POD competition [9]. The overall performance and performances on three specific classes (figure, table and formula) are shown in Table 2. Noticed that we come to a good place but not the best of all. All the methods in POD competition [9] have not been presented by academic papers, so the algorithms of their approaches and the authenticity of the results are unknown.

Most of the figures, equations and tables in document images can be correctly recognized and localized by our proposed page layout algorithm. Some visualization results are shown in Fig. 4. Green boxes represent equations detected by our proposed page layout analysis approach, red boxes represent figures and blue boxes represent tables. The numbers under the bounding boxes are the confidence scores.

Table 2. Results on POD competition.

Team	F1-measure				Average precision			
	Formula	Table	Figure	Mean	Formula	Table	Figure	Mean
PAL	0.902	0.951	0.898	0.917	0.816	0.911	0.805	0.844
Ours	**0.716**	**0.911**	**0.776**	**0.801**	**0.506**	**0.893**	**0.672**	**0.690**
HustVision	0.042	0.062	0.132	0.096	0.293	0.796	0.656	0.582
FastDetectors	0.639	0.896	0.616	0.717	0.427	0.884	0.365	0.559
Vislint	0.241	0.826	0.643	0.570	0.117	0.795	0.565	0.492
SOS	0.218	0.796	0.656	0.557	0.109	0.737	0.518	0.455
UTTVN	0.200	0.635	0.619	0.485	0.061	0.695	0.554	0.437
Matiai-ee	0.065	0.776	0.357	0.399	0.005	0.626	0.134	0.255

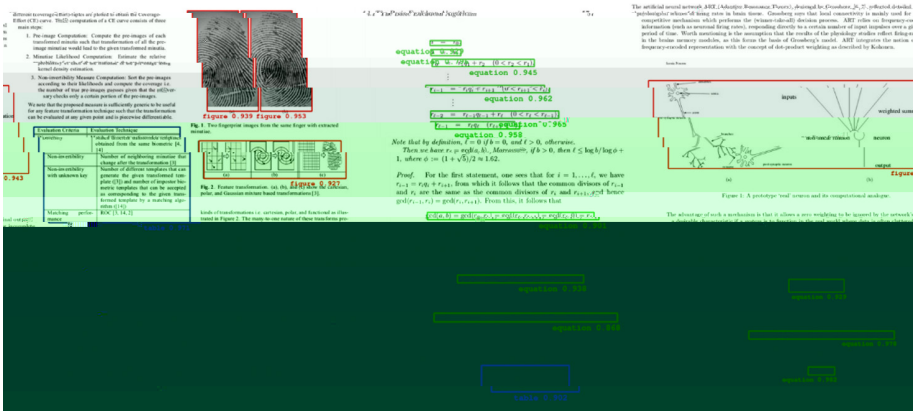

Fig. 4. Visualization results of our proposed page layout analysis algorithm. (Color figure online)

4.3 Limitations

There are still several scenarios our proposed algorithm might fail. As we can see in Table 2, the F1 measure and average precision score of equations is much lower than tables and figures. After analyzing the visualization results, we found that the evaluation scores are crippled by equations with "Σ", "Π", and equation number, for the reasons that the segmentation model tends to predict the equations with equation number into two separate equation regions, and RLSA separates the start and stop indexes of "Σ" and "Π" into three lines.

To tackle the equation number problem in the future work, one can increase the receptive field of semantic segmentation model to merge the equations and equation numbers in prediction map. As for the start and stop index problem, we trained a classification model to recognize "Σ" and "Π", and then merge the indexes. This procedure did bring us a precision gain on equations but is still not perfect. Therefore, there is still some room for the improvement of equation issue.

4.4 Running Time

Our proposed algorithm consists of two main parts: semantic segmentation and post-processing. Our segmentation is a deep neural network and a single inference takes 0.25 s on GPU (a single GTX1080). It should be at least twice faster if running on a decent GPU like Titan X. Our post-processing step can be efficiently done on CPUs (56 cores E5-2680v4) in 100 ms. In general, our whole system can process approximately 3 document images ($\sim$1300 * 1000) per second.

5 Conclusion

We have proposed a deep learning algorithm for page layout analysis, DeepLayout, which is capable of recognizing and localizing the semantic and logical regions directly from document images, without any help of PDF structural information or commercial software. We treat page layout analysis as a semantic segmentation problem, and a deep neural network is trained to understand the document image on pixel level. Then connected component analysis is adopt to restore bounding boxes from the prediction map. And we successfully bring local run length smoothing algorithm into our post-processing step which significantly improve the performance on both average F1 and mAP scores. Our semantic segmentation model is trained and experiments are conducted on ICDAR2017 POD dataset. It is demonstrated by the experiment results on POD competition evaluation metrics that our proposed algorithm can achieve 0.801 average F1 and 0.690 mAP score, which outperforms the second place of the POD competition. The running time of the whole system is approximately 3 fps.

Acknowledgement. This work was supported by the Natural Science Foundation of China for Grant 61171138.

References

1. Yi, X., Gao, L., Liao, Y., et al.: CNN based page object detection in document images. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 230–235. IEEE (2017)
2. Cesarini, F., Lastri, M., Marinai, S., et al.: Encoding of modified X-Y trees for document classification. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 1131–1136. IEEE (2001)
3. Priyadharshini, N., Vijaya, M.S.: Document segmentation and region classification using multilayer perceptron. Int. J. Comput. Sci. Issues **10**(2 part 1), 193 (2013)
4. Lin, M.W., Tapamo, J.R., Ndovie, B.: A texture-based method for document segmentation and classification. S. Afr. Comput. J. **36**, 49–56 (2006)
5. Chen, K., Yin, F., Liu, C.L.: Hybrid page segmentation with efficient whitespace rectangles extraction and grouping. In: International Conference on Document Analysis and Recognition, pp. 958–962. IEEE Computer Society (2013)
6. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)

7. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
8. Chen, L.C., Papandreou, G., Kokkinos, I., et al.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
9. Gao, L., Yi, X., Jiang, Z., et al.: Competition on page object detection. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR 2017, pp. 1417–1422. IEEE (2017)
10. Chu, W.T., Liu, F.: Mathematical formula detection in heterogeneous document images. In: Technologies and Applications of Artificial Intelligence, pp. 140–145. IEEE (2014)
11. Gao, L., Yi, X., Liao, Y., et al.: A deep learning-based formula detection method for PDF documents. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 553–558. IEEE (2017)
12. Hassan, T., Baumgartner, R.: Table recognition and understanding from PDF files. In: International Conference on Document Analysis and Recognition, pp. 1143–1147. IEEE (2007)
13. Oyedotun, O.K., Khashman, A.: Document segmentation using textural features summarization and feedforward neural network. Appl. Intell. **45**(1), 198–212 (2016)
14. Yang, X., Yumer, E., Asente, P., et al.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. arXiv preprint arXiv:1706.02337 (2017)
15. Shotton, J., Fitzgibbon, A., Cook, M., et al.: Real-time human pose recognition in parts from single depth images. In: Computer Vision and Pattern Recognition, pp. 1297–1304. IEEE (2011)
16. Ciresan, D., Giusti, A., Gambardella, L.M., et al.: Deep neural networks segment neuronal membranes in electron microscopy images. Adv. Neural. Inf. Process. Syst. 2843–2851 (2012)
17. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640 (2017)
18. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
19. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
20. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (2016). arXiv:1511.07122
21. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
22. Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: International Conference on Neural Information Processing Systems, pp. 109–117. Curran Associates Inc. (2011)