Instance Segmentation with BoundaryNet

Teodor Boyadzhiev[1,2(✉)] and Krassimira Ivanova[1]

[1] Institute of Mathematics and Informatics at the Bulgarian Academy of Sciences, Sofia, Bulgaria
{t.boadzhiev,kivanova}@math.bas.bg
[2] University of Library Studies and Information Technologies, Sofia, Bulgaria

Abstract. Instance segmentation is one of the key technology in many domains, such as medical image analysis, traffic and critical infrastructures monitoring, understanding of natural scenes. Recent methods for instance segmentation rely on bounding box regression, however the bounding boxes are not a natural representation for many domains.

We address the limitations of the bounding boxes with a new approach called BoundaryNet, in which we train a fully convolutional neural network to draw the boundaries around each object of each class. The boundaries allow for an easy bounding box and mask inference while still providing detailed information about the shape of the object. BoundaryNet avoids the restrictions of YOLO such as the number of bounding boxes, while it is more computationally efficient than the R-CNN methods.

The conducted experiments with the proposed neural network architecture BoundaryNet on the Common Object in Context (COCO) dataset show promising results in improving the instance segmentation process.

Keywords: Instance segmentation · Deep learning · BoundaryNet

1 Introduction

The Instance Segmentation is widely used in various fields such as medical image analysis, traffic monitoring, and remote sensing. The field of medicine has always been a primary source of image analysis tasks. Instance segmentation is extremely useful in histopathology for the detection of nuclei that can be used to diagnose dangerous diseases [9,18] or segmentation of organs or tumors in the organs from CT scans and MRI [1]. The combination between Semantic Segmentation and Instance Segmentation is often used in the recognition of complex street scenarios by self-driving cars [13] or by traffic management systems [24], as well as in the monitoring of critical infrastructures such as stations and airports [20]. The challenging tasks in the sphere of satellite and aerial imagery have also benefited the instance segmentation field. Such tasks include automated artificial object detection and building extraction from satellite images [21], evaluating building damage after a large-scale natural disaster from post-event aerial images [23], extracting geographical features (such as water bodies) from satellite maps using bounding boxes [3]. Of course, these areas do not

R. P. Barneva et al. (Eds.): IWCIA 2022, LNCS 13348, pp. 260–269, 2023.
https://doi.org/10.1007/978-3-031-23612-9_16

exhaust the field of application of instance segmentation - recently its use in more diverse tasks is increasing. A brief reference in Scopus on the keyword "instance segmentation" shows an exponential increase in the number of articles from 2016 so far.

In 2014 Girshick et al. proposed R-CNN for instance segmentation [5]. This approach uses a class agnostic region proposal method, based on a generic objectness score, to propose around 2000 regions for each image. Then from each region, a 4096-dimensional feature vector is extracted by a convolutional neural network, which was trained on the Image-Net challenge. Finally, each of these feature vectors is classified by a SVM.

This approach is later improved by Fast R-CNN for speed and accuracy by sharing the computations for feature extraction between the proposed regions. In the improved solution, feature maps are extracted by a convolutional neural network, then for each region proposal a feature vector is extracted, through a custom max-pool layer. For each of these feature vectors, a class and bounding box are predicted, using fully connected layers [4]. Further improvements in the same direction are made by Faster R-CNN by using a region proposal network and sharing the computations between this network and the feature extraction [17].

Based on these methods is proposed Mask R-CNN [8] which adds a third path to the Fast R-CNN to predict also the semantic mask for each bounding box. Gkioxari et al. [6] replace the mask branch in Mask R-CNN with a branch that predicts a 3D triangular mesh.

A different approach is used by Redmon et al. [14], which is called You Only Look Once (YOLO). YOLO splits the image in $S \times S$ grid and for each cell from the grid it predicts B bounding boxes and C class probabilities. For each bounding box is predicted also a confidence. The input image is processed by 24 convolutional layers, followed by 2 fully connected layers. The shape of the output tensor is $S \times S \times (B * 5 + C)$. This method predicts one category and its bounding box for each cell. YOLO is simpler and works much faster than the R-CNN pipeline, however, the performance is lower at 57.9% mAP. An improved version of YOLO is YOLO9000 which utilizes batch normalization, finer grid, relative to the cell centers bounding box regression, and is capable of detecting over 9000 object categories [15]. Further improvements were made in the third version [16].

Frequently bounding boxes are not a good approximation for the object boundaries in many domains. For instance, to overcome this problem Schmidt et al. [19] use star-convex polygons for the detection of cells in microscope images, while Loncomilla et al. [12] propose replacing bounding boxes with ellipses for detecting rocks. Other methods such as Mask R-CNN could overcome this problem by also predicting masks, however, this might become a problem for overlapping objects of the same category, due to crowding.

Here we propose a different approach to instance segmentation, called BoundaryNet, in which we train a fully convolutional neural network to draw the boundaries around each object of each class. This method is inspired by Yu

et al. [22] who use a neural network to draw boundaries around each category in semantic segmentation, to improve the performance of their model. Instead of drawing a boundary around each class, we draw boundaries around each instance of a class.

Drawing the boundaries does not impose hard restrictions on the number of bounding boxes, such as YOLO, does not have a complex pipeline such as the R-CNN architectures, and has great flexibility with respect to the object shape. The boundaries allow for easy bounding box and mask inference while still providing detailed information about the shape of the object.

This paper is organized in 5 sections. Section 2 describes the problem representation, the network architecture, and the error function. Section 3 provides details about dataset size, image resolution, data augmentation, training algorithm parameters, network size, etc. Section 4 shows the results from the experiments and Sect. 5 contains discussion, conclusion, and directions for further research.

2 BoundaryNet

The problem of instance segmentation is addressed by BoundaryNet by predicting the semantic masks for each class as well as the boundaries of each object. The outputs of the network are two tensors, one for the semantic segmentation and one for the boundaries. The semantic tensor has the shape $H \times W \times (C+1)$, where H and W are the height and the width of the image, and C is the number of categories. One more channel is used for the background category, which is considered everything else. The boundary tensor predicts whether each pixel is a part of a boundary or not. It has the shape $H \times W \times 2$. In general, it could be replaced with $H \times W$ and *sigmoid* activation, since it is a binary classification.

2.1 Labelling

For the semantic segmentation the class of each pixel is determined by the category of the object it belongs to. This is a multi-category classification at a pixel level. Therefore, the semantic label is a matrix, $L_s \in \mathbb{L}^{h \times w}$, where $\mathbb{L}$ is the set of the category labels, w is the width and h is the height of the input image. During training each such matrix is converted into one-hot notation, making it a 3D tensor. If there are less than 256 categories, the semantic label can be represented as a gray-scale image.

For the boundary output, the class of each pixel is "background", unless it is on the inside of an edge of an object, in which case it is assigned the label "boundary". The boundary label a matrix, $L_b \in \{0, 1\}^{h \times w}$, where w is the width and h is the height of the input image. The label can be represented as an image containing the edges between the instances of interest and the background, Fig. 1.

Fig. 1. The boundary label is a matrix, where each cell is either 0 or 1 depending on whether the pixel is a part of a boundary of an object of interest.

2.2 Segments Extraction

Once the boundaries and the semantic information is extracted from the network, each object of each class needs to be determined. The method consists of several steps:

1. The boundaries are used to extract several segments of connected background. Each segment is numbered with a different integer, creating a segment mask s.
2. For each category the semantic mask is extracted, c_k, by setting the pixels classified as this category to 1 and the rest to 0.
3. Each semantic mask is multiplied by the segments mask, element-wise

$$s_k = s \odot c_k \tag{1}$$

   where s_k is the segments, belonging to category k.
4. Finally for each category, k, the segments have to be grouped into objects. *This step is outside the scope of this paper and remains for further development.*

2.3 Network Architecture

The architecture of BoundaryNet is based on the architecture of UNet [18], (Fig. 2). It has one encoder and two independent decoders. Each skip connection from the encoders is connected to the corresponding level of both decoders. At each level the network has two convolutions with 3×3 kernel. Each convolution uses batch normalization and has *ReLU* activation. At each level of the encoder a 2×2 MaxPool operation is used. In the decoders UpSize operation with linear resampling is used. After the last convolution of each decoder, a 1×1 convolution is used with *softmax* activation as a classifier for each pixel.

The error for the network is a weighted sum of cross-entropy error for the semantic decoder and focal loss [10] for the boundary decoder

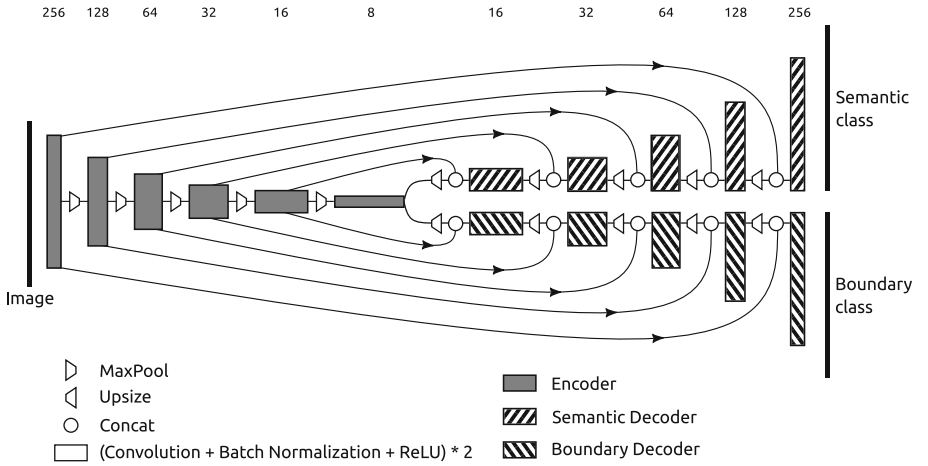


Fig. 2. Framework of BoundaryNet.

$$\mathcal{E} = \alpha \text{Focal} + (1 - \alpha)\text{CE} \quad \alpha \in [0, 1] \tag{2}$$

where

$$\text{Focal} = \sum_{k=1}^{2} (1 - pb_k)^{\gamma} b_k \log(pb_k) \tag{3}$$

and b_k is the one-hot notation for the boundary labeling, pb_k is the probability of category k in the boundary decoder, and γ is a parameter. Since for the boundary a binary classification is used, $k \in 1, 2$. This part can be substituted by a *sigmoid* and binary focal loss.

The filters in each level, $f(l)$, of the network are determined by

$$f(l) = \lfloor f 2^{\frac{l}{d}} \rfloor \tag{4}$$

where f is the number of frames in level 0, which is before the first MaxPool layer, l is the level number, and d is a divider. For example, if the network has 5 MaxPool layers, it will have 6 levels, numbered from 0 to 5.

3 Methods

The network was trained on the COCO dataset for people only [11]. The images were scaled to a resolution of 256×256. Objects which are composed of less than 256 pixels in total were removed, using image in-painting. In total 38027 images were extracted. The boundary labels were created by using an edge detection algorithm for each of the segments of objects of interest separately and then interpolated on top of each other. Then a dilation of 1 pixel in each direction was used to make them thicker and avoid small discontinuities.

The network was initialized using the Xavier method [7], where the weights were drawn from normal distribution and it was trained with the ADAM algorithm with learning rate 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-5}$ for 100 iterations. The training algorithm used batch size of 64 images. During the training no L-regularization or dropout was used.

The training and testing set was split randomly, using 80% of the data for training and the rest for testing. The training set was augmented using random zoom, horizontal flip, vertical flip, and rotation. The parameters for these operations are summarized in Table 1.

Table 1. Probability of augmentation and the parameters.

Operation	Probability	Parameters
Zoom	0.2	$\sim U(1, 1.5)$
Rotation	0.2	$\sim U\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$
Horizontal flip	0.2	
Vertical flip	0.2	

The parameter of the weighted sum of the error function is $\alpha = 0.1$. The parameter for the focal loss is $\gamma = 2$. The filters in the topmost layer are $f = 24$ and the divider is $d = 1.25$. The network has 5 MaxPool layers, meaning that in the lowest level it uses a feature map of 8×8 with 384 channels.

For the initialization and training algorithm were used the implementations provided by Wolfram Mathematica 13.0.

4 Results

Figure 3 shows the training and testing intersection over union (IOU) for the network trained for 100 iterations. The result shows that there is no over-fitting for the first 100 iterations. The semantic IOU reached 67.7% on the testing dataset and 71.6% on the training dataset. The boundary IOU reached 46.5% on the testing dataset and 46.4% on the training dataset. The network could reach better results if it is trained longer time since it has not fully converged.

Using IOU to measure the quality of the boundaries is not very appropriate, since it is too sensitive. For example, if the boundary generated by the network is half the thickness, the IOU would be 0.5, however, if it is unbroken and in the correct place, it can still be used to identify the object correctly.

Figure 4a shows examples from the validation set, where the network managed to find semantic mask and object boundaries with quality sufficient to distinguish between objects. Figure 4b shows examples from the validation set, where the network has made mistakes with the boundaries and the semantic segmentation.

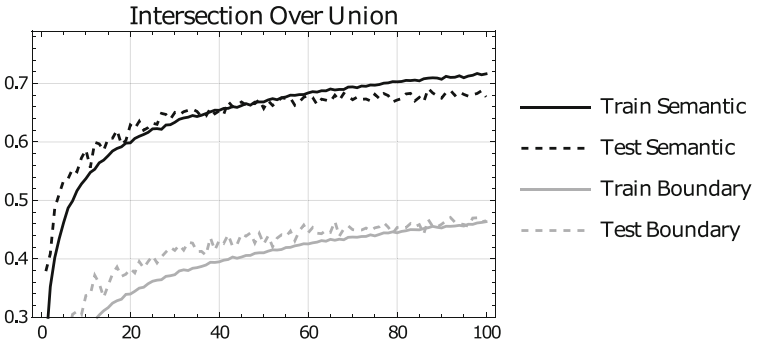


Fig. 3. Training and testing intersection over union (IOU) for the boundaries and the semantic segmentation.

Most of the mistakes for the boundaries are false negative, which will cause incorrect segment merging. For instance, in the first example of the mistakes (the left hand side of Fig. 4b), in the top right corner the contour of the palm of the person is broken, which can cause his hand to be identified as part of the person behind him.

In the second image (the center of Fig 4b), the boundary line of the child's elbow is broken. This will cause the child and the body of the man behind to be identified as the same instance, while the head of the man will become a separate instance.

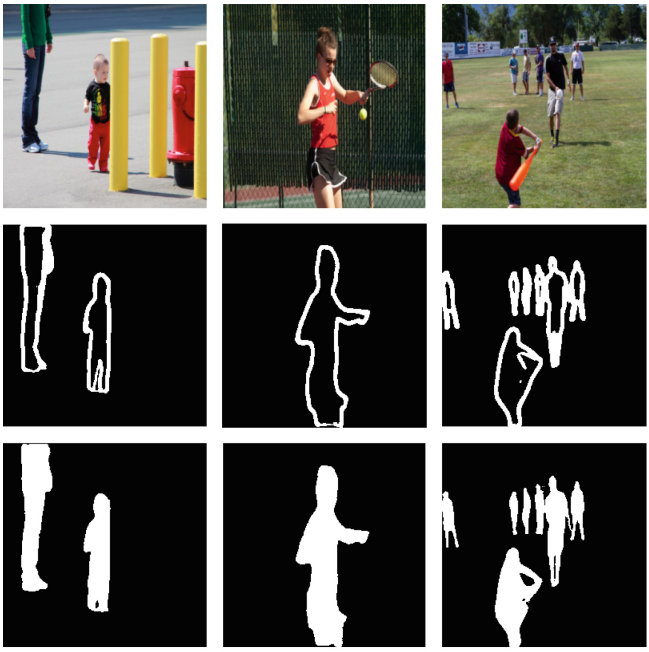
The haze in the third image (the right hand side of Fig. 4b), causes the boundary between the woman and the child to be broken and they will be merged into the same instance later. In this case the algorithm will identify two people instead of three.

5 Conclusion

In this paper we proposed a new approach for instance segmentation, based on drawing boundaries around each object of interest. We used a deep fully convolutional neural network, named BoundaryNet. The network, which is based on the UNet architecture, has one encoder and two parallel decoders. One of the decoders produces semantic masks and the other produces boundaries around each object of interest.

We demonstrated successful boundary identification for people, however, it is possible to have improvements in the quality of the boundaries. Further developments can include exploring different network architectures, such as channel attention blocks [22] or atrous pooling [2]. Other work in the same area is handling cases when the same object is covered by another object and therefore splitting it into two segments.

The results which we demonstrated here are inferior to other approaches such as YOLO and R-CNN, however, using transfer learning and exploring



(a) Examples of successful segmentation and finding of object boundaries.

(b) Examples of issues with finding the boundaries between objects.

Fig. 4. Examples for drawing boundaries.

other architectures could improve the quality of the results. Boundaries with improved quality can provide an alternative approach to instance segmentation with greater flexibility concerning the object shape. Such approach can be useful to domains, dealing with objects which are hard to be described by bounding boxes.

References

1. Altini, N., et al.: Liver, kidney and spleen segmentation from CT scans and MRI with deep learning: a survey. Neurocomputing **490**, 30–53 (2022)
2. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
3. Dhyakesh, S., et al.: Mask R-CNN for instance segmentation of water bodies from satellite image. In: Haldorai, A., Ramu, A., Mohanram, S., Chen, M.-Y. (eds.) 2nd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing. EICC, pp. 301–307. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-47560-4_24
4. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
5. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
6. Gkioxari, G., Malik, J., Johnson, J.: Mesh R-CNN. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9785–9795 (2019)
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
9. Hou, L., et al.: Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. Pattern Recogn. **86**, 188–200 (2019)
10. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
11. 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
12. Loncomilla, P., Samtani, P., Ruiz-del Solar, J.: Detecting rocks in challenging mining environments using convolutional neural networks and ellipses as an alternative to bounding boxes. Expert Syst. Appl. **194**, 116537 (2022)
13. Ojha, A., Sahu, S.P., Dewangan, D.K.: Vehicle detection through instance segmentation using mask R-CNN for intelligent vehicle system. In: 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 954–959. IEEE (2021)
14. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)

15. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
16. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28** (2015)
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. Schmidt, U., Weigert, M., Broaddus, C., Myers, G.: Cell detection with star-convex polygons. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11071, pp. 265–273. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00934-2_30
20. Tseng, C.H., Hsieh, C.C., Jwo, D.J., Wu, J.H., Sheu, R.K., Chen, L.C.: Person retrieval in video surveillance using deep learning-based instance segmentation. J. Sens. **2021**, 12, 9566628 (2021). https://doi.org/10.1155/2021/9566628
21. Vakalopoulou, M., Karantzalos, K., Komodakis, N., Paragios, N.: Building detection in very high resolution multispectral data with deep learning features. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1873–1876. IEEE (2015)
22. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1857–1866 (2018)
23. Zhan, Y., Liu, W., Maruyama, Y.: Damaged building extraction using modified mask R-CNN model using post-event aerial images of the 2016 kumamoto earthquake. Remote Sens. **14**(4), 1002 (2022)
24. Zhang, X.: A method to estimate position relationship between pedestrian and crosswalk based on YOLCAT++. In: 2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), pp. 38–42. IEEE (2021)