Seminar Image Processing and Content Analysis

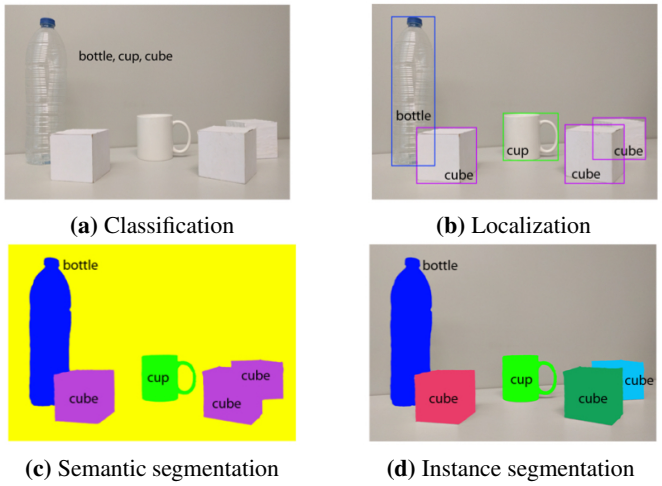
Semantic Instance Segmentation with a Discriminative Loss Function

Yuli Wu
Institute of Imaging & Computer Vision
Rheinisch-Westfälische Technische Hochschule (RWTH), 52056 Aachen
Email: yuli.wu@rwth-aachen.de

Abstract. Semantic instance segmentation is one of the most significant tasks in the field of computer vision, which requires machine not only to classify objects but also recognize them individually, even when they are incomplete, occluded or overlap with one another. A number of approaches have achieved decent results according to various benchmarks, among which a lot of them are proposal-based. In this document, the method using pixel embedding is reviewed, which shows that such a simple setup without bells and whistles is effective and can perform on-par with more complex methods.

1 Introduction

Object recognition is one of the most fundamental tasks in the field of image processing and computer vision. It can be referred to as localization, which usually results in a bounding box around the desired object. Another more difficult implement, namely semantic segmentation, recognizes which category the object (or stuff) belongs to. The most complex branch of this task mentioned in this document is instance segmentation, in which the objects are classified not only according to different categories, but also to the individuals inside each category. An example for this evolution of object recognition is shown in Fig. 1.



(a) Classification **(b)** Localization

(c) Semantic segmentation **(d)** Instance segmentation

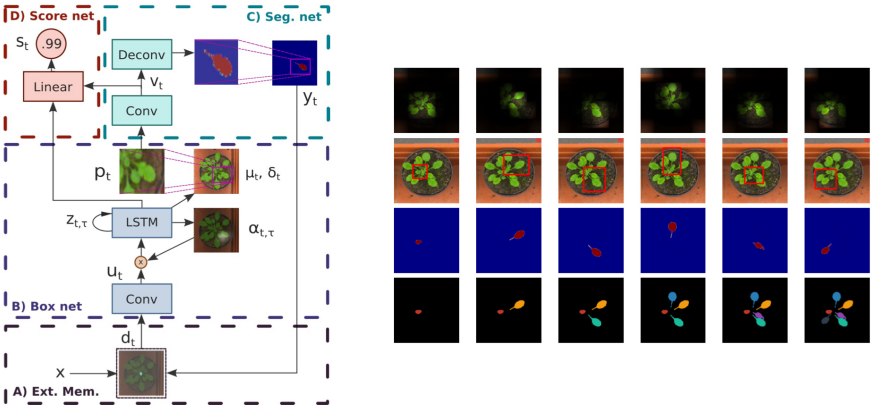
Fig. 1: Evolution of object recognition. Figures from [1].

In the following, some methods with different perspectives are briefly introduced in Section 2. After that, the method using discriminative loss function described in [2] is presented in Section 3. In the end of this document, the evaluation regarding this method and this paper is discussed and probable future works are presented. The focus of this document is on the different perspectives to solve the instance segmentation task. The detailed deep learning networks are beyond the scope of this document.

2 Related Work

A set of methods are talked in the paper. They are classified to several categories: proposal-based methods, recurrent methods, deep metric based methods and others. Among them, two methods [3,4] are briefly introduced in this section, which result in the best performance on the basis of CVPPP dataset and Cityscapes dataset respectively according to the listed comparison in the paper. Moreover, the method [5] with similar philosophy is also briefly introduced.

End-to-End Instance Segmentation with Recurrent Attention As shown in Fig. 2, this proposal-based model has four major components: an external memory that tracks the state of the segmented objects; a box proposal network responsible for localizing objects of interest; a segmentation network for segmenting image pixels within the box; and a scoring network that determines if an object instance has been found and also decides when to stop.

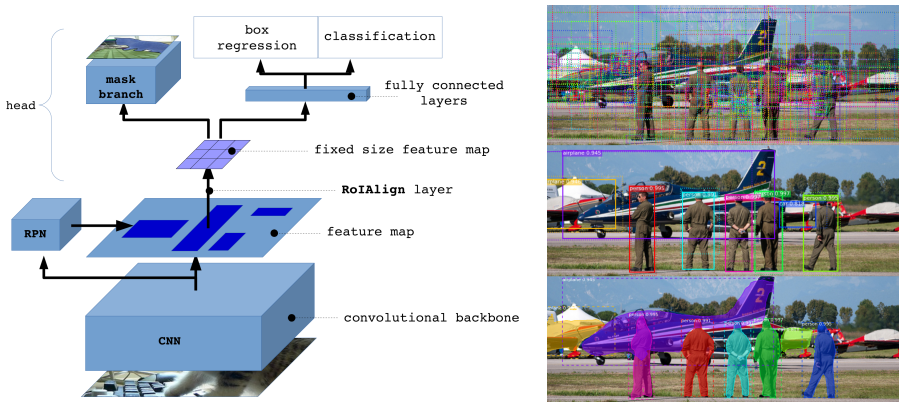


(a) Pipeline

(b) Row 1-4: soft attention of current glimpse, predicted box, current step segmentation, all segmentations.

Fig. 2: Left: end-to-end instance segmentation with recurrent attention model. Right: examples of outputs of different stages. Figures from [3].

Mask R-CNN This model has two main stages: Region Proposal Network(RPN) and a second stage, where in parallel to predicting the class and box offset, a binary mask for each RoI is also output. The main idea of this proposal-based model is similar to the former one, whereas the detailed networks and pipelines are not quite the same. Fig. 3(b) illustrates the outputs of each stage. Although the idea of this procedure is simple, the networks and the implementation are rather complex compared to the method with pixel embedding.



(a) Pipeline

(b) Top to bottom: RPN predictions, detections after Non-Max Suppression, final predictions.

Fig. 3: Left: Mask R-CNN model. Right: examples of outputs of different stages. Figures from [6].

Semantic Instance Segmentation via Deep Metric Learning In this approach, the idea of pixel embedding is also used. Unlike the main method to be introduced, this model predicts the embedding vector of each pixel, as well as a predicted class label for the mask centred at each pixel and a confidence score that this pixel would make a good seed for creating a mask. Furthermore, the loss function in the first part is different, which is introduced in the following.

First, the similarity between two pixels p, q is defined as sigmoid L2 distance between respective embeddings e_p, e_q in the feature domain.

$$\sigma(p,q) = \frac{2}{1+\exp(\|e_p - e_q\|_2^2)} \tag{1}$$

For pairs of pixels that are close in embedding space, it gives $\sigma(p,q) = \frac{2}{1+e^0} = 1$, and for pairs of pixels that are far in embedding space, it gives $\sigma(p,q) = \frac{2}{1+e^\infty} = 0$.

The loss function of embedding is defined as below:

$$L_e = -\frac{1}{|S|}\sum_{p,q\in S} w_{pq}[1_{\{y_p=y_q\}}\log(\sigma(p,q)) + 1_{\{y_p\neq y_q\}}\log(1-\sigma(p,q))] \tag{2}$$

where S is the set of pixels. w_{pq} are the weights of the loss, which are set to the values inversely proportional to the size of the instances p, q belong to, so the loss will not become biased towards the larger examples. $1_{\{y_p=y_q\}}$ means it equals 1, if p, q belong to the same instance, otherwise it equals 0.

The loss function of classification is defined as below:

$$L_{cls} = -\frac{1}{|S|} \sum_{p \in S} \sum_{c=0}^{C} y_{pc} \log \mathcal{C}_{pc} \tag{3}$$

where $\mathcal{C}_{pc}$ is the probability that the mask generated from seed pixel p belongs to class c. The joint loss is therefore denoted as

$$L = L_e + \lambda L_{cls}. \tag{4}$$

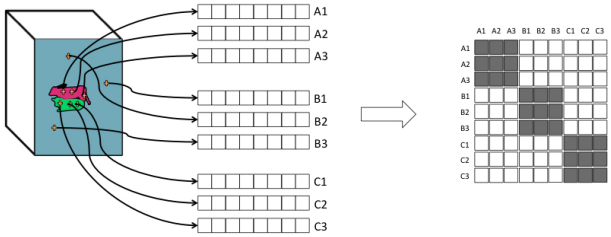


Fig. 4: Left: Pixels are mapped as embedding vectors. Right: Sigmoid cross entropy loss on the similarity between pairs of embedding vectors. Figures from [5].

3 Method

3.1 Discriminative Loss Function

Primitively, the output of a deep learning network for the tasks of classification and segmentation is usually a vote for labels. The label with the highest score represents the predicted category. In contrast, the deep learning network in [2] is used to map pixels into an abstract feature domain. The expected case is, the pixels belonging to the same individual object are clustered in the feature domain. That means, the mapped pixels, or the so-called embeddings, from different individuals in the feature domain should have larger distance(e.g. L1 or L2 norm) compared to those from same individuals.

To realize the idea, a discriminative loss function with three terms, namely variance term, distance term and regularization term, is introduced:

$$L_{var} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} \max(0, \|\mu_c - x_i\| - \delta_v)^2 \tag{5}$$

$$L_{dist} = \frac{1}{C(C-1)} \sum_{\substack{c_A=1 \\ c_A \neq c_B}}^{C} \sum_{c_B=1}^{C} \max(0,\, 2\delta_d - \|\mu_{c_A} - \mu_{c_B}\|)^2 \tag{6}$$

$$L_{reg} = \frac{1}{C} \sum_{c=1}^{C} \|\mu_c\| \tag{7}$$

$$L = \alpha \cdot L_{var} + \beta \cdot L_{dist} + \gamma \cdot L_{reg} \tag{8}$$

The number of clusters in the ground truth is denoted as C, an embedding is denoted as x_i, the number of embeddings and the mean embedding in cluster c are denoted as N_c and μ_c respectively. δ_v and δ_d represent the margins for the variance and distance loss.

The variance term L_{var} penalizes the embeddings, which are outside the user-designed margin δ_v of their cluster centre. This term can be regarded as an intra-cluster *pull-force* that draws embeddings towards the mean embedding. The distance term, analogously, penalizes the mean embeddings, which are inside the user-designed margin $2\delta_d$. As the parameter δ denotes radius, the margin is defined as doubled δ_d in the latter case. This term can be regarded as an inter-cluster *push-force* that pushes clusters away from each other. The last term is the regularization term. It penalized mean embeddings, if they are too far away from origin, to keep the activations bounded.

The first two terms are hinged. The goal is only to attract embeddings inside the margin and repulse the clusters one another. The definite locations of embeddings and clusters are not taken into account. Geometrically speaking, all embeddings are within a distance of δ_v from their cluster centre and all cluster centres are at least $2\delta_d$ apart, if L_{var} and L_{dist} both equal zero. It can be written as following:

$$\begin{cases} \|\mu_c - x_i\| \leqslant \delta_v, \forall c \leqslant C, i \leqslant N_c \\ \|\mu_{c_A} - \mu_{c_B}\| \geqslant 2\delta_d, \forall c_A, c_B \leqslant C \wedge c_A \neq c_B \end{cases} \tag{9}$$

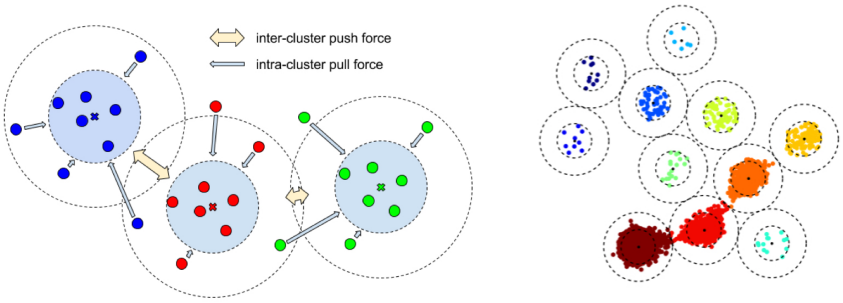
3.2 Post-processing

In the ideal cases, the embeddings belonging to the same object are well clustered, i.e. Equation 9 is fulfilled. If it is the case, it is very convenient to segment embeddings with user-designed margins for example $\delta_d = 0.5$ and $\delta_v = 1.0$: select any embedding and all other embeddings are clustered together, which have distance to the selected embedding less than $\delta_d = 0.5$.

As the loss is not minimized to zero in the real world, it is rational to take into account that some embeddings may be mapped outside of the inner margin, or even outside of the outer margin in the worst case. With this in mind, an effective clustering algorithm is applied, namely the mean-shift algorithm [7].

3.3 Experiments

Datesets Two datasets are tested using this method: CVPPP Leaf Segmentation and Cityscapes.



(a) Effect of variance term and distance term in a 2D feature domain. Radiuses of two dotted circles are δ_d and δ_v respectively. Crosses represent meam embeddings. This diagram is inspired by a similar one in [8].

(b) Results after optimazation. Embeddings with different colours represent pixels belonging to different individual objects.

Fig. 5: Visualization of pixel embeddings in a 2D feature domain. Figures from [2].

Setup The loss function introduced in the paper is easy to use with an off-the-shelf network. ResNet-38 network [9] is chosen in this experiment. In the CVPPP dataset, the images are augmented to increase the overall robustness. Networks are trained with margins $\delta_d = 0.5$ and $\delta_v = 1.5$ in the both datasets. The output have 16 and 8 dimensions in the CVPPP and Cityscapes respectively. The reasons for such a choice is not further explained. It is also unclear, if the dimension of the network's output plays a role in the performance.

Results Table 1 shows the results of the method using this loss function, compared with two top-performed methods. The method using loss function is competitive in the CVPPP, where it is close to the best scores. In the Cityscapes, however, it shows a large room for improvement.

	SBD	$\lvert DiC\rvert$	AP	$\text{AP}^{50\%}$	AP^{100m}	AP^{50m}
End-to-End	84.9	0.8				
Mask R-CNN			26.2	49.9	37.6	40.1
Loss Function	84.2	1.0	17.5	35.9	27.8	31.0

Table 1: Left: segmentation and counting performance on the test set of the CVPPP leaf segmentation challenge. Two metrics from [10] are reported: Symmetric Best Dice(SBD) and Absolute Difference in Count($\lvert DiC\rvert$). Right: segmentation performance on the test set of the Cityscapes instance segmentation benchmark. Accuracy is represented using 4 metrics from [11]: mean Average Precision(AP), mean Average Precision with overlap of 50%($\text{AP}^{50\%}$), objects within 100m(AP^{100m}) and 50m(AP^{50m}).

Apart from the loss function, there are two main factors which influence the results in this experiment:

1. *Semantic segmentation.* In Cityscapes, for example, the loss function is applied independently on each semantic class. Despite that the clusters from different classes

may overlap in the feature domain, they are treated separately and not pushed away from each other.

2. *Mean-shift clustering.* The step of clustering can also potentially influence the results. Unlike stated in the paper, the imperfect network architecture and the limitation of convergence during the loss function minimization are the critical reasons. Even with a small δ_d and a large δ_v , this a priori drawback cannot be compensated.

semantic seg.	clustering	AP	$AP^{50\%}$
resnet38 [9]	mean-shift	21.4	40.2
resnet38 [9]	centre threshold	22.9	44.1
ground truth	mean-shift	37.5	58.5
ground truth	centre threshold	47.8	77.8

Table 2: Effect of the semantic segmentation and clustering components on the performance.

Besides the routine results of experiments, Table 2 is shown to compare the effects of the semantic segmentation and clustering components on the performance. No surprisingly, the quality of the semantic segmentation has a big influence on the overall performance. Moreover, in the case of ground truth, results with different clustering strategies have a big gap. The clustering processing in this comparison is not described or visualized. It is unclear, what do the embeddings look like, such that thresholding is much more effective than mean-shift.

Speed Accuracy Trade-off It is stressed in the paper that their method using loss function is suitable for an off-the-shelf network. To evaluate the performance of the selection of different networks, accuracy and efficiency are tested based on the car class of the Cityscapes validation set. Table 3 shows that there exists a speed accuracy trade-off. ENet is advantageous, if speed is important. While ResNet-38 is competitive in accuracy, but requires some more memory.

	Dim	AP	AP_{gt}	fps	#p	mem
ENet	512x256	0.19	0.21	145	0.36	1.00
	768x384	0.21	0.25	94		1.03
	1024x512	0.20	0.26	61		1.12
Segnet	512x256	0.20	0.22	27	29.4	1.22
	768x384	0.22	0.26	14		1.29
	1024x512	0.18	0.24	8		1.47
Dilation	512x256	0.21	0.24	15	134.3	2.20
	768x384	0.24	0.29	6		2.64
	1024x512	0.23	0.30	4		3.27
ResNet38	512x256	0.24	0.27	12	124	4.45
	768x384	0.29	0.34	5		8.83

Table 3: AP: Average Precision, AP_{gt}: AP using gt segmentation labels, fps: speed of forward pass, #p: number of parameters ($\times 10^6$), mem: meemory usage (GB).

4 Discussion

4.1 Summary

From the aspect of machines, instance segmentation of human beings is the mapping of images in some abstract feature domain and the output can be regarded as a point, which represents that instance. All approaches are to simulate this *mapping*. As the deep learning network is still black-box-like, it is not easy to evaluate the methods using proposals and directly using loss function in a precise way of causality. In the following, a comparison between those approaches is tried to presented in an abstract level though.

Intuitively speaking, the method using pixel embedding makes this mapping less strict, compared to the one of human beings. All pixels of an image are mapped to a single point in the human's eyes. In this approach, however, it is enough that all pixels of an image are mapped closely with each other in the feature domain. Without constraint of the definite location of an embedding, the network can choose the mapping more freely.

Compared to proposal-based approaches, this method differs in different aspects. The selection of proposal plays a significant role in the performance. Confronting with complicated images, it is vulnerable that this step works not as expected. If a proposal is not detected, it results in an unrepairable defect. Second, in the case of single class, such as CVPPP, the procedure of proposal-based methods consists usually of two stages: selection of bounding box of individual objects and segment the instance from that region. In contrast, the method using loss function performs a per-pixel network to directly map them as embeddings in the feature domain, which makes this pipeline more simple. Furthermore, in the case of multi-class, such as Cityscapes, the semantic segmentation is applied in the method using loss function at first. After this step has been completed, the routine procedure is applied. It is advantageous for the method like Mask R-CNN, as the class, box and the mask are output in a parallel style.

As a solution, [5] uses one network to achieve both embeddings and the classifications. It differs from the main method in the format of loss function. [5] uses a sigmoid cross entropy loss. In the main method, a mixture of hinge loss and squared loss is performed. In addition, unlike the main method uses mean shift to cluster the embeddings, [5] uses a seediness model. Due to the limited dimension of this document, the details are not presented. Despite that this simple proposal-free approach overcomes various difficulties, it is still not as competitive as the proposal-based methods in some benchmarks.

4.2 Outlook

It is clear that the deep learning network is not perfect in semantic segmentation. A more accurate network is always expected. Moreover, together with [5], methods using pixel embedding is an inspiration of non-proposal-based approaches in this field. A single stage pipeline combining classification and instance segmentation is the promising perspective in the future work.

References

1. Garcia-Garcia Alberto, Orts-Escolano Sergio, Oprea Sergiu, Villena-Martinez Victor, Garcia-Rodriguez Jose. A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:170406857 2017;.
2. De Brabandere Bert, Neven Davy, Van Gool Luc. Semantic instance segmentation with a discriminative loss function. arXiv preprint arXiv:170802551 2017;.
3. Ren Mengye, Zemel RichardS. End-to-end instance segmentation with recurrent attention. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA; 2017. p. 21–26.
4. He Kaiming, Gkioxari Georgia, Dollár Piotr, Girshick Ross. Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE; 2017. p. 2980–2988.
5. Fathi Alireza, Wojna Zbigniew, Rathod Vivek, Wang Peng, Song HyunOh, Guadarrama Sergio, et al. Semantic instance segmentation via deep metric learning. arXiv preprint arXiv:170310277 2017;.
6. Abdulla Waleed. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. `https://github.com/matterport/Mask_RCNN`; 2017.
7. Fukunaga Keinosuke, Hostetler Larry. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on information theory 1975;21(1):32–40.
8. Weinberger KilianQ, Saul LawrenceK. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 2009;10(Feb):207–244.
9. Wu Zifeng, Shen Chunhua, Hengel Antonvanden. Wider or deeper: Revisiting the resnet model for visual recognition. arXiv preprint arXiv:161110080 2016;.
10. Scharr Hanno, Minervini Massimo, French AndrewP, Klukas Christian, Kramer DavidM, Liu Xiaoming, et al. Leaf segmentation in plant phenotyping: a collation study. Machine vision and applications 2016;27(4):585–606.
11. Cordts Marius, Omran Mohamed, Ramos Sebastian, Rehfeld Timo, Enzweiler Markus, Benenson Rodrigo, et al. The Cityscapes Dataset for Semantic Urban Scene Understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. .