

Instance Segmentation for Urban Street Scenes

Francesco Bari

francesco.bari.2@studenti.unipd.it

Eleonora Signor

eleonora.signor@studenti.unipd.it

Abstract

In this report we compared different existing instance segmentation techniques, on the specific task of Urban Street Scenes. Our interest in the topic was born out of the fact that the segmentation of instances is one of the fundamental tasks of the vision, however it is still complex and not fully explored.

1. Introduction

Image segmentation is the process of separating an image into several segments, in which each pixel is associated with an object type. There are two types of image segmentation: semantic segmentation and instance segmentation. The first marks objects of the same type with the equal class label; the second marks objects of the same type and belonging to distinct entities with different class labels. The idea we tried to develop was to compare different instance segmentation methods. The first technique we studied was Mask R-CNN [1], a two-stage approach. We chose this one in the light of the positive feedback it has received from the world of vision research, due to its conceptually simple and general framework, characterised by efficient image object detection, and the simultaneous generation of a high quality segmentation mask for each instance. The technique we decided to contrast with Mask R-CNN [1] was BlendMask [2]. Instead, this is a one-stage technique that has been shown to outperform Mask R-CNN [1] both in terms of mask prediction and training time, on the MSCOCO 2017 [3] and LVIS [4] datasets. We were interested in verifying whether this also applied to datasets, such as *Cityscapes* [5] and *WildDash* [6], belonging to the specific topic of *Urban Street Scenes*, featuring images from the streets around the world, with many difficult scenarios. Some of the aspects we tested involved backbone changes, depth of the ResNet [7] and number of frozen layers. The results obtained confirmed what had already been announced in previous works, generalising BlendMask [2] as one of the most promising staged approaches. At the end of our work and in order not to limit our analysis, we also made some considerations on other instance segmentation techniques, such

as SOLOv2 [8] and Deep Snake [9].

2. Related Work

The approach we used in our work was to use the following papers as guidelines [1], [2], [8] and [9], extending the analysis also to datasets of *Urban Street Scenes*.

2.1. Stage approach

Mask R-CNN [1] is a box-based two-stage approach with a Convolutional Neural Network, that is at the avant-garde of image segmentation. It is a variant of a Deep Neural Network that detects objects in an image and generates a segmentation mask for each instance. Mask R-CNN [1] is the next evolution of Faster R-CNN [10], a Region-based Convolutional Neural Network, which produces for each candidate object 3 outputs: the class label, the bounding box offset and the object mask. The architecture of the network, Figure 1, consists of a CNN (backbone), which processes the image and extracts the feature map. After that, thanks to the *Region Proposal Network*, proposals or ROI are presented, on which to make recognition of the bounding box and mask prediction (head). In addition, before generating the output, *RoIAlign* is applied to each ROI, which makes it possible to obtain a mask, where the layout of the object is maintained.

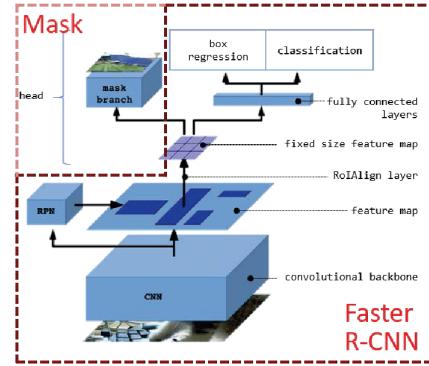


Figure 1. Mask R-CNN architecture. Source: [11], pag. 3.

The convolutional backbones (in the lower part of the figure) proposed in the paper are C4, C5 and FPN [12]. For heads (in the upper part of the figure) the backbone must include the 5-th stage of ResNet [7], "res5".

BlendMask [2] is derived from the limits of Mask R-CNN [1]. The authors of the paper define how Mask R-CNN [1] strongly constrains the speed and quality of mask generation at heads, thus making it difficult to deal with complicated scenarios and placing a limit on masks resolution. Furthermore, Mask R-CNN [1] presents itself as an inflexible framework for multi-task networks. They thus attempted to combine top-down and bottom-up search strategies in FCOS [13], anchor box-free one-stage approach, which seems able to outperform its two-stage counterparts in terms of accuracy. The architecture of BlendMask [2], Figure 2, consists of a detector network and a mask branch. The latter is partitioned in 3 parts: the bottom module that deals with predicting scores maps, called bases; the top layer composed of a single convolution layer and towers, as many as the input features, with the task of predicting attention instances; and a blender module that combines scores with attentions.

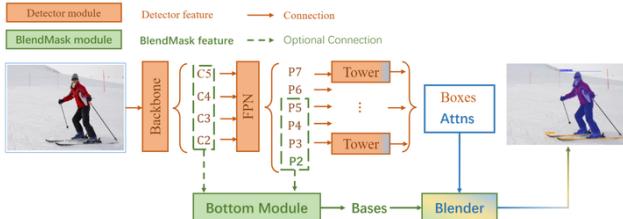


Figure 2. BlendMask architecture. Source: [2], pag. 3.

The *bottom module* (to left) uses C4/C5 or FPN [12], depending on whether it takes "C" or "P" functions as input, and produces the bases. In the *top layer* (to right) a single convolution layer is added above the towers, and this allows attention masks to be produced, together with each prediction of the bounding box. The *blender module* (lower part of the figure), for each instance combining bases linearly with the learned attention maps.

SOLOv2 [8] is a box-free one-stage approach, a successor of SOLO [14]. In this case each instance of an image is segmented dynamically, without detection of the bounding box. The mask generation, unlike Mask R-CNN [1], is decoupled into mask kernel prediction and mask feature learning. These two elements are responsible for generating convolution kernels and feature maps. SOLOv2 [8] also manages to achieve promising results through the use of the matrix *non-maximum suppression* (NMS) technique, proposed by the authors of the paper, which reduces duplicate predictions while gaining less inference overhead.

2.2. Contour-based approach

Deep Snake [9] is a contour-based approach, implementing the idea of snake algorithms with a learning-based approach. Deep Snake [9] consists of a two-stage pipeline: in a first instance there is an initial contour proposal on the object of an image; followed by the use of a Neural Network, which iteratively deforms this proposal until it

exactly matches the object's boundaries. For learning the structure of contour features, the authors of the paper propose the use of *Circular Convolution*.

3. Datasets

The datasets of *Urban Street Scenes*, which we used, belong to *Cityscapes* [5] and *WildDash* [6].

3.1. Cityscapes

Cityscapes [5] is a suite of benchmark and a large-scale dataset for semantic urban scene understanding. It is suitable for learning and testing pixel-level and instance-level semantic labelling methods. The images of *Cityscapes* [5] were created from a large and diverse set of video sequences, recorded in 50 different cities. These may be inclusive of high quality annotations, or/and coarse annotations; the latter allow the testing of methods employing large volumes of weakly labelled data. The annotations contained, which are fundamental for the evaluation of a model, are of a polygonal type.

The dataset we used, belonging to *Cityscapes* [5], is partitioned into two units: *gtFine* and *leftImg8* (Figure 3), which we used in pairs. *gtFine* consists of fine annotations for 3 475 train and val images, and 1 525 test set images. *leftImg8* from "row" images of urban traffic, with train set, test set and val set; a total of 5 000 images.



Figure 3. *Cityscapes* [5] images: *gtFine* to left and *leftImg8* to right.

In Figure 4, we report the classes and the number of occurrences in *gtFine train* and *leftImg8 train*. The total number of classes is 8.

| category | #instances | category | #instances | category | #instances |
|------------|------------|----------|------------|----------|------------|
| person | 17918 | rider | 1781 | car | 26963 |
| truck | 484 | bus | 380 | train | 168 |
| motorcycle | 737 | bicycle | 3675 | | |
| total | 52106 | | | | |

Figure 4. *Cityscapes*: definitions of train dataset classes.

The class considered "dangerous" for a good generalisation of a model is *train*, due to the small number of instances (168).

However of *Cityscapes* [5] although the datasets include several months and seasons, they are always images taken in good weather conditions. This aspect prompted us to analyse the behaviour of our instance segmentation techniques on the *WildDash* [6] dataset.

3.2. WildDash

WildDash [6] is a benchmark suite and dataset for semantic and instance segmentation for the automotive domain. The images, contained in the datasets, come from different sources from all over the world. In addition, they present scenarios, such as rain, darkness and road cover, which are real challenges for image recognition. This highlights the shortcomings of any instance segmentation technique.

The dataset we used is *public gt package* (Figure 5), consisting of 4 256 images, aimed specifically at solving instance segmentation tasks; however, was not divided into train, val and test set. Consequently, we decided to divide it up manually, reserving 3 405 images as train set and 851 images as test set. We did not consider it necessary to make a further partition in val set, in order to keep as many images as possible in the training; and assuming that our models, using weights already pretrained on *ImageNet* [15], were accurate enough not to require *Model Selection*.



Figure 5. *WildDash* [6] images: *rain scenario* to left and *road in the desert* to right.

In Figure 6, we report the classes and the number of occurrences in *public gt package train*. The total number of classes is 13.

| category | #instances | category | #instances | category | #instances |
|-------------|------------|----------|------------|----------|------------|
| ego vehicle | 1526 | person | 7943 | rider | 1626 |
| car | 14440 | truck | 1807 | bus | 825 |
| caravan | 46 | trailer | 65 | train | 74 |
| motorcycle | 1555 | bicycle | 536 | pickup | 531 |
| van | 928 | | | | |
| total | 31902 | | | | |

Figure 6. *WildDash*: definitions of train dataset classes.

The classes considered "dangerous" for a good generalisation of a model are *caravan* (46), *trailer* (65) and *train* (74).

4. Method

4.1. Architecture

4.2. Hyperparameters of configuration

4.3. Training with fine-tuning

Mask R-CNN La loss utilizzata durante il training è la seguente

$$\min(L) = \min(L_{cls} + L_{box} + L_{mask})$$

L_{cls} is the classification loss, L_{box} is the bounding-box loss and L_{mask} is the average binary cross-entropy loss.

BlendMask La loss utilizzata durante il training è la seguente

$$\min(L) = \min(\text{semantic loss}) [?]$$

4.4. Evaluation and inference

4.5. Metrics

- AP
- numero di istanze, tempo :: accuracy visiva.confidence-threshold

5. Experiments

In questa sezione descriviamo gli esperimenti che abbiamo eseguito per testare e valutare le tecniche oggetto di questo lavoro. Tali esperimenti gli abbiamo eseguiti al termine delle fasi di studio e codifica.

5.1. Backbone

La seconda serie di esperimenti, che abbiamo compiuto, riguarda la definizione della backbone. Tutte le tecniche a stadi, oggetto del confronto, sono dotate del suddetto modulo inferiore, per cui ci è risultato semplice uniformare le scelte architettoniche in modo da poter compiere una valutazione oggettiva. Le tecniche che abbiamo confrontato sono state Mask R-CNN e BlendMask.

Le configurazioni costanti delle reti sono image size ..., numero massimo di iterazioni, learning rate ..., step size a ... e fine-tuning esclusivamente agli ultimi 2 livelli.

| Method and architecture | Cityscapes AP | WildDash AP |
|--|---------------|-------------|
| Mask R-CNN + ResNet50 + C4 + Base-RCNN-C4 | | |
| Mask R-CNN + ResNet50 + DC5 + Base-RCNN-DilatedC5 | | |
| Mask R-CNN + ResNet50 + FPN + Base-RCNN-FPN | | |

Table 1. Backbone Mask R-CNN result.

Per BlendMask, oltre a settare le configurazioni costanti, avvalendoci dei risultati presentati in ... abbiamo settato $R = 56$, $M = 14$, $K = 4$, sampling method for bottom bases bilinear pooling, interpolation method for top-level attentions bilinear upsampling and semantic loss. Inoltre abbiamo deciso di testare vari tipi di decoder: ProtoNet and DeepLabv3+.

| Method and architecture | <i>Cityscapes AP</i> | <i>WildDash AP</i> |
|--|----------------------|--------------------|
| BlendMask with decoder ProtoNet + ResNet50 + FPN + Base-550 | | |
| BlendMask with decoder ProtoNet + ResNet50 + deformable convolution + FPN + Base-550 | | |
| BlendMask with decoder DeepLabv3+ + ResNet50 + FPN + Base-550 | | |
| BlendMask with decoder DeepLabv3+ + ResNet50 + deformable convolution + FPN + Base-550 | | |

Table 2. Backbone BlendMask result.

5.2. Deepness

Una terza serie di esperimenti ha riguardato lo studio della profondità delle reti ResNet.

I parametri di configurazione non definiti in modo esplicito, sono le medesime di quelle riportate nella sezione §5.1.

| Method and architecture | <i>Cityscapes AP</i> | <i>WildDash AP</i> |
|---|----------------------|--------------------|
| BlendMask with decoder ProtoNet + ResNet101 + FPN + Base-BlendMask | | |
| BlendMask with decoder ProtoNet + ResNet101 + deformable convolution + FPN + Base-BlendMask | | |

Table 3. Deepness BlendMask result.

5.3. Freeze levels

Per la quarta serie di esperimenti ci siamo voluti concentrare sul numero di layers da "scongelare" di ResNet durante il re-training dei pesi.

I parametri di configurazione non definiti in modo esplicito, sono le medesime di quelle riportate nella sezione §5.1.

| Method and architecture | <i>Cityscapes AP</i> | <i>WildDash AP</i> |
|--|----------------------|--------------------|
| Mask R-CNN + ResNet101 + FPN 1 layers freeze | | |
| Mask R-CNN + ResNet101 + FPN 3 layers freeze | | |
| BlendMask with decoder ProtoNet + ResNet101 + FPN + Base-BlendMask 1 layers freeze | | |
| BlendMask with decoder ProtoNet + ResNet101 + FPN + Base-BlendMask 3 layers freeze | | |

Table 4. Freeze layers result.

5.4. Own best models

Come ultima serie di esperimenti abbiamo cercato di individuare i modelli migliori, per ciascuna le due tecniche di instance segmentation in esame in questa sezione; tenendo conto della possibilità di allenare ciascun modello solo su una singolo macchina e 1 GPU.

I parametri di configurazione non definiti in modo esplicito, sono le medesime di quelle riportate nella sezione §5.1.

| Dataset | method and architecture | AP |
|---------|-------------------------|----|
| | | |

Table 5. Own best models result.

6. Conclusion

From our experiments we are able to say that one-stage and anchor box-free techniques, suitably modified, perform better than box-based two-stage methods (of which king is Mask R-CNN [1]), both in terms of AP precision and GPU time. These positive effects are probably caused by the hybridisation of top-down and bottom-up methods, and the detection of unanchored objects (as is the case for BlendMask [2]). In addition, recent studies reported in [8] and partly shown in Figure 7, prove that a box-free stage approach combined with matrix NMS, such as SOLOv2 [8], is demonstrated to be very competitive against BlendMask [2].

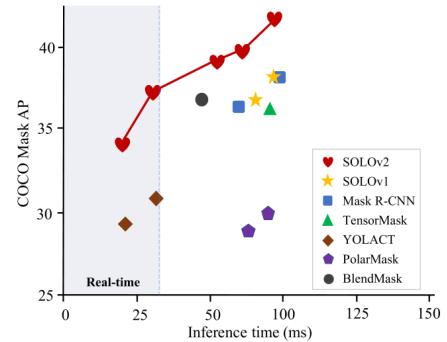


Figure 7. Comparison between SOLOv2 and other stage approaches. Source: [8], pag. 2.

From top to bottom: SOLOv2[8] and SOLO[14] are box-free one-stage approach; Mask R-CNN is box-based two-stage approach; TensorMask [17] is dense sliding-window one-stage approach; YOLACT [18] is one-stage approach predecessor of BlendMask [2]; PolarMask [19] anchor box-free, single shot instance segmentation one-stage approach; BlendMask [2] anchor box-free one-stage approach.

Moreover, stage approaches are not the only possible methods for solving instance segmentation tasks. For example, there is Deep Snake [9], a counter based approach that outperforms Mask R-CNN [1] both in terms of inference speed and AP precision, as shown by the table in Figure 8, and has the potential to be a good competitor to SOLOv2 [8].

| | training data | fps | AP [val] | AP | AP ₅₀ |
|------------------|---------------|-----|-------------|-------------|------------------|
| SGN [26] | fine + coarse | 0.6 | 29.2 | 25.0 | 44.9 |
| PolygonRNN++ [1] | fine | - | - | 25.5 | 45.5 |
| Mask R-CNN [18] | fine | 2.2 | 31.5 | 26.2 | 49.9 |
| GMIS [28] | fine + coarse | - | - | 27.6 | 49.6 |
| Spatial [31] | fine | 11 | - | 27.6 | 50.9 |
| PANet [27] | fine | <1 | 36.5 | 31.8 | 57.1 |
| Deep snake | fine | 4.6 | 37.4 | 31.7 | 58.4 |

Figure 8. Results on *Cityscapes* val (AP [val] column) and test (remaining columns) sets. Source: [9], pag. 7.

A possible future extension, in line with the results and

considerations we have made, could consist in search to further improve the performance of BlendMask [2] by trying to use Deep Snake [9] contour-based approach for detecting object bounding boxes of a image.

References

- [1] Kaiming He and Georgia Gkioxari and Piotr Dollár and Ross Girshick. Mask R-CNN. CoRR, 2018.
- [2] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang and Youliang Yan. BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation. CoRR, 2020.
- [3] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. CoRR, 2014.
- [4] Agrim Gupta, Piotr Dollár and Ross B. Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. CoRR, 2019.
- [5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. CoRR, 2016.
- [6] Zendel, Oliver and Honauer, Katrin and Murschitz, Markus and Steininger, Daniel and Dominguez, Gustavo Fernandez. WildDash - Creating Hazard-Aware Benchmarks. Proceedings of the European Conference on Computer Vision, (ECCV), 2018.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. CoRR, 2015.
- [8] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li and Chunhua Shen. SOLOv2: Dynamic, Faster and Stronger. CoRR, 2020.
- [9] Sida Peng, Wen Jiang, Huaijin Pi, Hujun Bao and Xiaowei Zhou. Deep Snake for Real-Time Instance Segmentation. CoRR, 2020.
- [10] Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. CoRR, 2015.
- [11] Bienias, Lukasz & n, Juanjo & Nielsen, Line & Alstrøm, Tommy. Insights Into The Behaviour Of Multi-Task Deep Neural Networks For Medical Image Segmentation. 2019.
- [12] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan and Serge J. Belongie. Feature Pyramid Networks for Object Detection. CoRR, 2016.
- [13] Zhi Tian, Chunhua Shen, Hao Chen and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. CoRR, 2019.
- [14] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang and Lei Li SOLO: Segmenting Objects by Locations. CoRR, 2019.
- [15] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
- [16] Xu, Jingyi and Zhang, Zilu and Friedman, Tal and Liang, Yitao and Van den Broeck, Guy. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. Proceedings of the 35th International Conference on Machine Learning, 2018.
- [17] Xinlei Chen, Ross B. Girshick, Kaiming He and Piotr Dollár. TensorMask: A Foundation for Dense Object Segmentation. CoRR, 2019.
- [18] Daniel Bolya, Chong Zhou, Fanyi Xiao and Yong Jae Lee. YOLACT: Real-time Instance Segmentation. CoRR, 2019.
- [19] Enze Xie, Peize Sun, Xiaoge Song, Wenhui Wang, Xuebo Liu, Ding Liang, Chunhua Shen and Ping Luo. PolarMask: Single Shot Instance Segmentation with Polar Representation. CoRR, 2019.