BlitzMask: Real-Time Instance Segmentation Approach for Mobile Devices

(Abstract) We propose a fast and low complexity anchor-free instance segmentation approach BlitzMask. For the first time, the approach achieves competitive results for real-time inference on mobile devices. The model architecture modifies CenterNet by adding a new lite head to the CenterNet architecture. The model contains only layers optimized for inference on mobile devices, e.g. batch normalization, standard convolution, depthwise convolution, and can be easily embedded into a mobile device. The instance segmentation task requires finding an arbitrary (not a priori fixed) number of instance masks. The proposed method predicts the number of instance masks separately for each image using a predicted heatmap. Then, it decomposes each instance mask over a predicted spanning set, which is an output of the lite head. The approach uses training from scratch with a new optimization process and a new loss function. A model with EfficientNet-Lite B4 backbone and 320×320 input resolution achieves 28.0 mask AP at 36.5 fps on Samsung S20. This sets a new speed benchmark for inference for instance segmentation on mobile devices.

Keywords: Instance Segmentation, Neural Network, Computer Vision, Anchor-free detector, Sigle-stage detctor

1 Motivation

We were unable to find fully comparable benchmarks with performance tested on regular mobile devices for instance segmentation. Why is porting an instance segmentation approach to mobile such a problem?

A naive solution is to simply use an existing instance segmentation approach with light backbones, such as MobileNet, as well as low-resolution input to achieve real-time speed on mobile. Suppose we are adapting well-known SOLO [1], or CenterMask [2], or a similar approach to mobile. Our results indicate that the operation number for the mask branch (which is part only a part of the model head) is 40 times larger than that of the entire MobileNetv2-SSDLite model [3] and therefore it is impractical to use such approaches on mobile. The authors of SOLOv2 [4] greatly reduced computational complexity of the model head using dynamic convolution. However, this is a non-standard layer and its adaption is currently not available for mobile; it is a complex task to produce such an adaptation.

It is a non-trivial task to change input resolution in YOLACT, which can dramatically decrease accuracy. For example, the authors [5] report $mAP = 29.8$ for 550×550, and $mAP = 24.9$ for 400×400 input, both with ResNet-101 backbone; our own YOLACT training for 320×320 input size gives only $mAP = 20.1$ with ResNet-101 backbone. The lightweight YOLACT modification named mobileYOLACT [6] has low input resolution 320×320 and is designed for mobile environments. However, the best result of the mobileYOLACT [6] is $AP^{mask}_{50} = 23.0$, which is too low, because AP^{mask} is usually near twice smaller than AP^{mask}_{50} and as we show below our approach is more than twice more accurate than mobileYOLACT and has even faster speed on the mobile device considered in the paper [6]. All these facts prove that using a lightweight backbone and low input resolution for YOLACT approach leads to too low accuracy. Furthermore, YolactEdge [7] uses TensorRT optimization for 550×550 input with MobileNetV2 backbone to achieve real-time calculation on Jetson AGX Xavier with accuracy $mAP = 20.8$. However, this model is still too heavy for mobile devices, and we show better accuracy for real-time application on mobile using a method descried below.

[1]Samsung RD Institute Ukraine

2 Method

2.1 Architecture

We propose a new single-stage anchor-free instance segmentation method named BlitzMask. Its architecture is based on CenterNet [8] and FPN [9]. When implemented for real-time inference on mobile devices, the architecture uses separable convolution [10] with bilinear upsampling instead of a combination of deformable [11] and transpose convolutions as in [8]. The reason is that, deformable convolution is not supported, and transpose convolution has high latency on mobile devices [12].

Model output consists of Spanning Set B, Coefficients Λ, Heatmap Y and Size D (see Fig. 1). The output spatial dimensions are four times less than the input spatial dimensions.

Each channel c of the ground truth for Heatmap Y contains probability density-like function located at the instance segmentation mask mass center for objects of class c. Ground truth for Size D contains distances from the object bounding box centers to its four sides $[w_l, h_t, w_r, h_b]$ similarly to Liu et al. [13].

We express an instance mask as a linear combination of Spanning Set B with corresponding Coefficients Λ. We only define ground truth for instance mask, and do not require ground truths for its components used in the derivation, i.e. Spanning Set and Coefficients.

2.2 Loss function

For each ground truth mask mass center i_n, j_n, $n = 1, \ldots, N$, we fix $\lambda_{n,k} = \Lambda^k_{i_n, j_n}$, $k = 1, \ldots, K$, where N is the number of instances and K is the number of channels in both Coefficients and Spanning Set outputs. Then, an instance mask is predicted as a linear combi-

Table 1: BlitzMask speed and accuracy of the instance segmentation on Samsung S20 Snapdragon 865

Method	Backbone	Resolution	Time(ms)	AP^{mask}	AP^{mask}_{50}
BlitzMask	MobileNetV2	320×320	10.1	21.9	35.8
BlitzMask	EfficientNet-Lite B4	320×320	27.4	28.0	47.1
mobileYOLACT	MobileNetV2	320×320	47.6	-	23.01
YolactEdge	MobileNetV2	550×550	28*	20.8	-
EOLO	MobileNetV2	320×320	33.3*	11.7	27.7

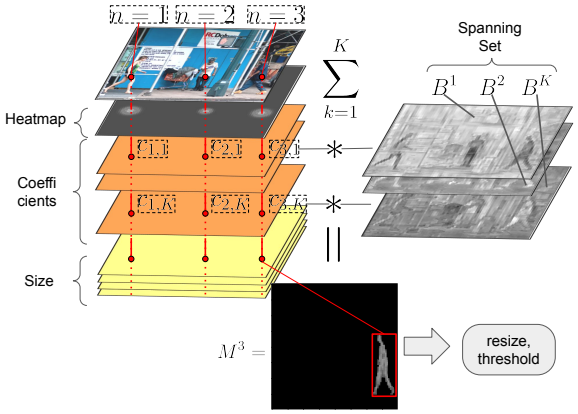

Figure 1: Instance mask reconstruction using Spanning Set B, Coefficients Λ, Heatmap Y and Size D for the case of $N = 3$ instance masks, class number $C = 1$. Operation "resize" means resize M^n, $n = 1, \ldots, N$ to original images size. The "threshold" operation assigns 1 to pixels for which $\sigma(M^n) > 0.5$ and 0, otherwise.

nation of the Spanning Set channels B^k, $k = 1, \ldots, K$ with coefficients $\lambda_{n,k}$ (also see Fig. 1):

$$M^n = \sum_{k=1}^{K} \lambda_{n,k} \cdot B^k \qquad (1)$$

The Heatmap Y and Size D outputs are compared to the corresponding ground truths using object detection loss function L_{box} that consists of focal loss L_{foc} for Heatmap and L_{size} for Size. The focal loss L_{foc} is a penalty-reduced pixelwise logistic regression loss [14], [2] with perturbation of the first polynomial coefficient of Taylor expansion. L_{size} penalizes bounding box predictions using combination of l_1-type loss and UnitBox loss [15]. We define the loss function for instance masks as combination of Cross-Entropy loss [16] and Dice loss [17], where we use normalization coefficient which allows small and large objects contributing equally to the loss function.

2.3 Training Details

The instance segmentation models are trained on MS COCO [18] train2017 dataset and evaluated on val2017. We follow [19, 20] recommendation to search for a flat (rather than sharp) minima on the loss surface, as a sharp minima determined on train data could result in a large loss function value on evaluation data. Our training process is a modification of cyclical learning rates approach [20], it uses Stochastic Gradient Descent (SGD) with momentum and has three phases: Ascent, Plateau, and Descent. In contrast to [20], we find the optimal learning rate and SGD momentum only during the training by creating model backups and "look forwards" to what happens if we change the learning rate or SGD momentum. Moreover, we use different learning rates ascent speeds and SGD momentum descent to get the best validation loss. It greatly differs from [20] and any other training process published in the literature.

3 Results

There are only a few results for real-time instance segmentation on mobile devices. In Table 1 we compare our results with mobileYOLACT [6], YolactEdge [7] and EOLO [21], where for YolactEdge inference time is measured on Jetson AGX Xavier, and for EOLO - on Raspberry Pi4 with Google Coral USB Accelerator. Given that YolactEdge input resolution is 550×550, we achieve better accuracy of 21.9 with a smaller input resolution 320×320, but the same backbone MobileNetV2 (see Table 1). Table 1 shows that our model has 2x better accuracy and 1.75x faster inference speed than mobileYOLACT [6], which is based on popular approach YOLACT [5]. One of the main BlitzMask advantages is its simplicity, which is very important for a further method development or/and re-implementation from scratch. Our experiments show that mixing of l_1 and UnitBox losses for the box regression part and Cross-Entropy and Dice losses for the segmentation part improves model performance. In addition, using center mass of masks as location of Gaussian peaks in Heatmap improves model accuracy. Simple and fast post-processing allows achieving real-time speed on mobile devices with sufficient accuracy. BlitzMask achieves $AP^{mask} = 28.0$ at 36.5 fps on Samsung S20. This sets a new speed and accuracy benchmark for instance segmentation inference on mobile devices.

References

[1] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. In *European Conference on Computer Vision*, pages 649–665. Springer, 2020.

[2] Yuqing Wang, Zhaoliang Xu, Hao Shen, Baoshan Cheng, and Lirong Yang. CenterMask: single shot instance segmentation with point representation. In *Proceedings of the IEEE/CVF conference on*

computer vision and pattern recognition, pages 9313–9321, 2020.

[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[4] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and fast instance segmentation. *arXiv preprint arXiv:2003.10152*, 2020.

[5] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019.

[6] Juwon Lee, Seungjae Lee, and Jong Gook Ko. Mobile yolact: Toward lightweight instance segmentation for mobile devices. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1456–1460. IEEE, 2021.

[7] Haotian Liu, Rafael A Rivera Soto, Fanyi Xiao, and Yong Jae Lee. YolactEdge: Real-time instance segmentation on the edge. *arXiv preprint arXiv:2012.12259*, 2020.

[8] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. doi: 10.1109/CVPR.2017.106.

[10] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[12] Cheng-Ming Chiang, Yu Tseng, Yu-Syuan Xu, Hsien-Kai Kuo, Yi-Min Tsai, Guan-Yu Chen, Koan-Sin Tan, Wei-Ting Wang, Yu-Chieh Lin, Shou-Yao Roy Tseng, et al. Deploying image deblurring across mobile devices: A perspective of quality and latency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 502–503, 2020.

[13] Zili Liu, Tu Zheng, Guodong Xu, Zheng Yang, Haifeng Liu, and Deng Cai. Training-time-friendly network for real-time object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11685–11692, 2020.

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[15] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.

[16] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[17] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context, 2014.

[19] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[20] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[21] Longfei Zeng and Mohammed Sabah. EOLO: Embedded object segmentation only look once. *arXiv preprint arXiv:2004.00123*, 2020.