Toward Efficient Simultaneous Detection and Segmentation

Chong Zhang[1,2], Zongxian Li[1], Qiong Liu[2], Yonghong Tian[1*], Wei Zeng[1], Yaowei Wang[3], Wenbai Chen[2]
[1] National Engineering Laboratory for Video Technology, School of EE&CS, Peking University, Beijing, China
[2] School of Automation, Beijing Information Science and Technology University, Beijing, China
[3] School of Information and Electronics, Beijing Institute of Technology, Beijing, China

***Abstract*—To solve the low-speed problem of two-stage based framework for object detection and instance segmentation, we creatively introduce the large separated convolution to the typical two-stage method. In our method, the two-branches separated large kernel convolution operation is applied before the ROI pooling layer, which is able to reduce the complexity of the follow-up process to a great extent and make the ROI pooling much more efficient. Furthermore, the subnet of region-based convolution network is carefully simplified and designed for obtaining better performances. Extensive evaluation experiments on Microsoft COCO datasets show that our method provides ∼2× speedup compared with the original Mask R-CNN method and results in a comparable detection and segmentation performances.**

***Index Terms*—Instance Segmentation, Objection Detection, Separated Convolution, Network acceleration**

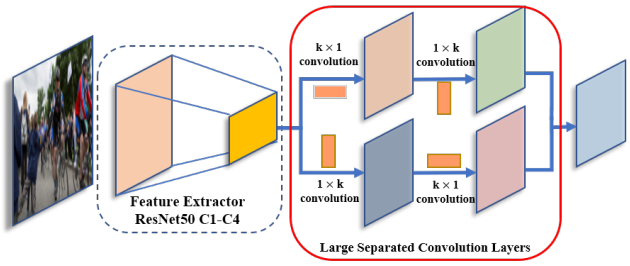

Fig. 1. The separated large kernel convolution. We deploy a k×1 followed by 1×k convolution and a k×1 followed by 1×k convolution on the ResNet50-C4 block. The feature maps are sumed as 256 channels which are fed to Region Proposal Network and ROI operation.

I. Introduction

The state-of-the-art CNN-based object detection approaches can be divided into two-stage based methods such as Faster R-CNN [1] and R-FCN [2]. and one-stage based methods like SSD [3] and YOLO [4]. One-staged based methods such as SSD, which predict the possible location of the target object during the network forward procedure and the object recognition and location regression are operated at the end of the network, which is able to achieve a substantial improvement in speed compared with the two-stage based method. However, this kind of one-stage based detector usually cannot achieve a satisfied performance when detecting tiny targets [5]. Due to its scalable detection performance, the Faster R-CNN based method has aroused considerable research interests in recent years in the field of detection and segmentation [2], [6], [7].

Mask R-CNN [6], the Faster R-CNN based method proposed by He et al, won the COCO2017 challenge by adding a segmentation branch after the ROI pooling operation. The same as Faster R-CNN, the ROI operation involves a complex, time-consuming computation, and the problem of low speed is still not resolved properly in the Mask R-CNN method. For the two-stage based architecture represented as Faster R-CNN, some previous works [1], [8] showed that the large number of feature map after the ROI warping will directly lead to the increase of the computation complexity and it will inevitably result in a long time cost in both training and testing phase. According to [9], by suitably factorizing convolutions and aggressive regularization, the time-consuming operation is broken down into the separated channels. Specifically, Peng et al. used this kind of separated convolution in the semantic segmentation task and gained a satisfied performance [10]. Motivated by this, we think of decomposing the convolution layer before the ROI Pooling operation into separated channels and decreasing the output channel of the feature map, which will lead to a decrease of the computation complexity in the follow-up process significantly.

In this paper, the separated large kernel convolution is introduced into the two-stage based object detection and instance segmentation method. The Mask R-CNN is considered as the principle baseline in our work, for the network modification and experiments comparison. Specifically, we modify the ResNet-50 based Mask R-CNN architecture from two aspects: 1) As shown in Fig 1, we decrease the number of feature maps by adding separated convolution channels on the ResNet-50-C4 block before the ROI Align, 2) We replace the C5 convolution block of the ResNet50 with two convolution layers with 3×3 kernel size to obtain features for final object detection and instance segmentation. Despite the proposed method is apparently simple and there is hardly any new technique in our method, the improved Mask R-CNN version by applying our method does gain ∼2× speed up and result in a comparable detection and segmentation on Microsoft COCO datasets [11] when compared with the original Mask R-CNN.

It is worthwhile to highlight the following contribution of our work on optimizing the Mask R-CNN:

1. Chong Zhang and Zongxian Li contribute equally to this paper and should be considered as co-first authors.

2. This work was done when Chong Zhang was an intern at the National Engineering Laboratory for Video Technology, Peking University.

* Corresponding author: Yonghong Tian (email: yhtian@pku.edu.cn).

978-1-5386-5321-0/18/$31.00 ©2018 IEEE

- A separated large kernel convolution is introduced to solve the low-speed problem of two-stage based detection and segmentation methods, aiming to reduce the input channel of the ROI operation, which can reduce the network redundancy and accelerate the follow-up detection and segmentation significantly.
- We gained $\sim 2\times$ speed up in simultaneously object detection and instance segmentation when compared with the original ResNet50 based Mask R-CNN model by using the largely separated convolution kernel and simplified subnet, which result in a comparable detection and segmentation performance.

II. Related Work

Object detection and instance segmentation are treated as two separate tasks for a long term. Mask R-CNN extends the Faster R-CNN detector by adding an ROI Align mechanism and a segmentation branch into the region proposal network, which implements the instance segmentation and the object detection at the same time. However, the efficiency problem of the two-stage based method still has not been solved properly.

A. Object Detection

R-CNN [12], proposed by Ross et al, first introduced the convolution neural network into the field of the object detection. He et al. proposed SPP-Net [13] to improve R-CNN using Space Pyramid Pooling. Fast R-CNN [8] was proposed to improve the efficiency of feature extraction with ROI pooling mechanism and multi-task training method. Considering the strong representation ability of the feature map extracted by the deep network, the Region Proposal Network (RPN) was introduced in Faster R-CNN [7]. Due to its scalable detection performance, the Faster R-CNN based method has aroused considerable research interests in recent year [2], [6], [7]. Unlike region-based detectors, Some one-stage based detectors like YOLO [4], SSD [3] greatly increase the speed of detection by abandoning region proposal phase. However, the performance is not satisfactory when detecting tiny targets.

B. Instance Segmentation

The abstract features extracted by CNN are very helpful for image recognition and classification [14], [15], and can be used to determine exactly what kind of objects are contained in an image. However, due to the detail information missing among the convolution and pooling operation, it is hard for the neural network to classify each pixel to get a precise segmentation performance.Some previous works [16]–[19] focused on segment candidates, and then classified by applying a general classifier. The FCN method [20] proposed by Long attempts to recover the useful information from the abstract features by applying the deconvolution method and then classify each pixel to its corresponding category. FCIS [21], proposed by Li et al, combining the segmentation method in [22] with region-based FCN detector [2], results in a significant performance.

C. Network acceleration

The purpose of network acceleration is to reduce the redundancy of the network and speed up network computing. There are two main methods of model compression:1) Speed up the convolution neural network forwarding by optimizing the network architecture. For instance, the well-known Inception [9], [23]–[25], ResNeXt [26], MobileNets [27] and etc. are proposed and carefully designed to reduce the time and space complexity of the convolution operation without the obvious performance drop or even obtain a better performance. 2) modification on a trained model. Pruning [28], [29], quantization [28], [30], binarized neural networks [31], [32], Teacher-student Framework [33] are also widely used.

III. Approach

A. Framework Overall

Our overall two-stage based framework is illustrated in Fig 2. The public released ResNet50 model pre-trained on ImageNet [15] as our base feature extraction network for extracting discriminative features. After the C4 block of the ResNet50 architecture, a separated large kernel convolution block is added before the region proposal network and ROI layer. The main purpose of our method is to reduce the input channel of the region proposal network and ROI operation, which can reduce the computation complexity of the head part of the network to a great extent without an obvious performance drop. After that, the subnet of the Mask R-CNN is carefully modified to get comparable performances in both detection and segmentation tasks and reduce the time cost.

B. Separated Large Kernel Convolution

Recent advances in two-stage object detection and instance segmentation methods are driven by region proposal methods and region-based convolutional neural networks. As [1], [7], [8] shows, creating more accurate feature maps and high-quality region proposal candidate boxes is a critical step for two-stage methods, and the two-stage method acceleration has become a bottleneck due to the huge computation complexity in processing the generated candidates.

Asymmetric Convolutions [9] shows that factorizing the traditional k$\times$k convolution into a 1$\times$k convolution followed by a k$\times$1 convolution only involves O($\frac{2}{k}$) computation cost when compared with the tradition k$\times$k convolution, which is able to save computational cost dramatically as k grows . As illustrated in Fig 1, and Fig 2, the separated large kernel convolution used in our framework is composed of two branches. Each branch is a large kernel convolution of the Asymmetric Convolutions. The input feature map of the Asymmetric Convolutions is obtained after the previous convolution layer, and two large kernel convolution with size k$\times$1 and k$\times$1 are followed. The large padding strategies are used among the convolution operation.

We take ResNet50 based Mask R-CNN framework as our baseline. The last convolution block of ResNet-50-C4 is denoted as C4. As shown in Fig 1, we adopt a large kernel separated convolution on C4. Specifically, we employ

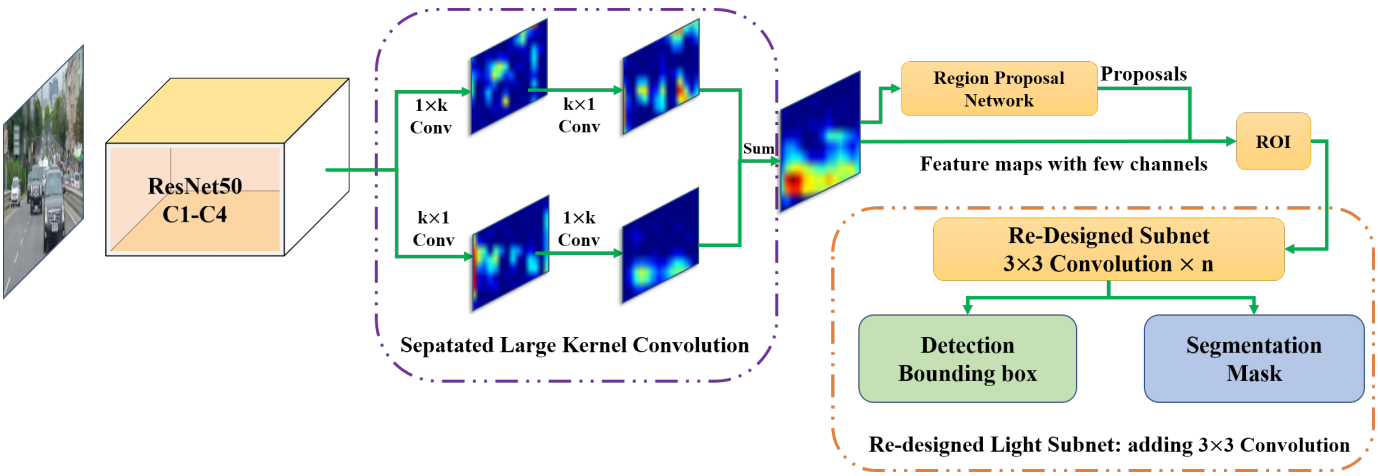

Fig. 2. The overview of the our framework. The separated large kernel convolution is deployed between ResNet50-C4 and ROI pooling operation. The original ResNet50-C5 block is replaced by our Redesigned region-based convolution network.

a combination of 1×k + k×1 and k×1 + 1×k convolutions. The k was set to 15 in our work, and the number of feature maps in the middle layer is 256. With padding employed, we can get 256 feature maps as large as input. After that, the redundancy of the feature maps fed into the RPN and ROI Align are reduced by separated large kernel convolution to a great extent, while the number of channels is four times less than that of original Mask R-CNN version.

C. Subnet of Region-Based Convolution Network

Besides having heavy computation in RPN and ROI Align part, the Mask R-CNN employs an expensive network for feature extraction, which leads to a serious impact on its computing speed. Similarly, Faster R-CNN has a more expensive subnetwork with continuous fully connected layers. Considering there are N channels feature maps generated from C4 block and fed into the ROI operation, there are N channels feature maps output after the ROI operation. C5 block of ResNet50. And such a heavy part is operated on the generated feature map, the computation cost can be defined as follows:

$$cost \sim O(M_l^2 K^2 C_{l-1} C_l) \tag{1}$$

The M_l refers to the size of the output feature map and the K means the convolution kernel size, and the C_{l-1} and the C_l refer to the number of the input and output channel of feature maps. In our work, the channels of the input feature maps are decreased in separated convolution layers significantly. On the other hand, the heavy ResNet C5 block is replaced by two 3×3 convolution layers with 256 and 512 channels feature map with size 14×14, which will lead to a decrease of the computation cost in another aspect. We found that the channel number of the final output has a great influence on the final segmentation performance and we will discuss it carefully in the ablation study part in experiment section. For the detection branch, we replace the global average pooling by an average pooling of 7×7 size with stride 7. For the segmentation branch, a Deconvolution layer is used to enlarge the 14×14 feature and recover the information from 28×28 feature maps for obtaining a better performance.

Overall, by reducing the input channels of the RPN and ROI operation and re-design the follow-up subnet, nearly 40 % parameters are decreased in our framework when compared with the original ResNet50 based Mask R-CNN, and leads to a comparable detection and segmentation performances.

IV. Experiments

In this section, we evaluate our method on the Microsoft COCO datasets as it is the benchmark widely used in object detection and segmentation tasks. For all evaluation, we test the trained model on the remaining 5k subset of val images and report the average over IoU thresholds (AP) at different scales. (AP, AP50, AP75, APs, APm, APl). Specifically, we make a comprehensive comparison with the ResNet50-based Mask R-CNN baseline and other state-of-the-art detection and segmentation methods in both precision and speed.

A. Implementation Details

The intersection-over-union (IoU) threshold is fixed at 0.7 and the ROI has IoU over 0.7 will be considered as positive. We resize the short edge of images to 800 pixels with the same ratio as the original and train the model for 360k iterations on 2 Nvidia 1080Ti GPU. The Learning rate is set to 0.01 and decreases by 10 at 240k and 320k. We use a weight decay of 0.0001 and momentum of 0.9. At test time, The IoU threshold is set to 0.5. Precision and time testing is implemented on a single Nvidia 1080Ti GPU at the same time.

B. Main Results

We mainly compare the proposed framework with the ResNet50 based Mask R-CNN and other state-of-the-art detection and segmentation methods. The results are shown in table 1. Since the Mask R-CNN is the only method which addresses the object detection and instance segmentation tasks into an integral pipeline and trained in an end to end way, we report

	Backbone	Test time(ms)	Segmentation: Mask						Detection: Bbox					
			AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
Mask R-CNN [6]	ResNet50-C4	209	29.06	46.29	30.44	14.35	32.38	42.77	33.95	53.32	34.17	20.87	37.51	43.7
MNC [18]	ResNet101-C4	-	24.6	44.3	-	4.7	25.9	-	-	-	-	-	-	-
FCIS [21]	ResNet50-C5	132	27.1	46.7	-	-	-	-	-	-	-	-	-	-
R-FCN [2]	ResNet101-C5	136	-	-	-	-	-	-	27.6	-	-	8.9	30.5	4.0
FPN [7]	ResNet50-C4	107	-	-	-	-	-	-	33.9	-	-	17.8	37.7	45.5
Faster-R-CNN [7]	ResNet50-C4	172	-	-	-	-	-	-	31.6	-	-	13.2	35.6	47.1
Our framework 1	ResNet-50-C4	**103**	28.28	**48.54**	28.98	10.28	31.50	**43.20**	31.85	52.55	34.10	16.16	36.00	42.40
Our framework 2	ResNet-50-C4	163	28.42	**48.76**	29.28	10.95	31.43	**43.84**	32.12	52.93	34.31	17.28	35.95	42.09

TABLE I
COMPARISONS OF OUR RESULTS ON COCO2014 MINIVAL.

Separated Convolution		Subnet	Test time(ms)
Conv1	Conv2		
64	64	C5	81
128	128	C5	120
256	256	C5	182
64	64	Re-Designed	71
128	128	Re-Designed	92
256	256	Re-Designed	103

TABLE II
RESULTS OF SPEED WITH DIFFERENT CHANNELS OF SEPARATED LARGE KERNEL CONVOLUTION AND R-CNN SUBNET DESIGN.

Output Channels	Pooling	Detection			Segmentation		
		AP	AP50	AP75	AP	AP50	AP75
256	14×14	30.31	51.25	31.55	27.57	47.33	28.04
256	7×7	31.46	52.30	33.17	27.81	47.60	28.34
512	14×14	30.82	51.54	32.29	27.72	47.55	28.60
512	7×7	31.85	52.55	34.10	28.28	48.54	28.98
1024	7×7	32.12	52.93	31.31	28.42	48.76	28.28

TABLE III
RESULTS OF PRECISION WITH DIFFERENT CHANNELS OF REGION-BASED CONVOLUTION OUTPUT FEATURE MAPS.

Fig. 3. Example of instance segmentation and detection results on Microsoft COCO 2014, using 512 output feature maps in R-CNN subnet and running at 10 fps, with 28.28 mask AP and 31.85 bounding box AP.

the segmentation and detection results of the other state-of-the-art methods separately besides the baseline method. For a fair comparison, the public released ResNet50 model pre-trained on the ImageNet is used to initialize the Mask R-CNN baseline and our framework.

For details, a separated large kernel convolution layer is added on the 4^{th} stage of ResNet50, reducing the number of feature maps from 1024 to 256. We remove the 5^{th} stage of ResNet50 and 2 convolution layers with 3×3 kernel size are added for further feature extraction with output 14×14 feature maps. For detection branch, a 7×7 large pooling layer is operated to map the 14×14 feature map to a 2048(2×2×512) dimension vector for final bounding box regression and category prediction. For segmentation branch, a deconvolution layer is applied to recover the spatial and semantic information from the 14×14 feature map and enlarge it to size 28×28. The results are shown in Table I. We test two kinds of method with different final output channels(one for 512 and the other one for 1024) and observe that the outputs with 512 channels results in a comparable result with 1024 channels in both segmentation and detection but much faster. It is obvious that ∼2× speed up and comparable detection and segmentation performances have gained when compared with the ResNet50 based Mask R-CNN. Our method is also outperforming a large margin on segmentation and detection when compared with the state-of-the-art FCIS [21] and R-FCN [2] method, even though they use a stronger backbone network than us.

C. Ablation Study

Speed Analysis. Table II shows the forwarding time by using separated convolution layer with different output channels and different subnets. Specifically, we observe the network forwarding time in the case of the different separated convolution and different subnets respectively. The subnet here are composed with 2 continuous 3×3 convolution with 512 final output channels and followed by a 7×7 pooling. We notice that the network forwarding time decreased significantly with the decline of number of the output channels, which further highlights the theoretical analysis in Sec III. By combining the separated convolution layers with the re-designed subnet, we finally gain a ∼2× speed up compared with the baseline and it is also faster than the other state-of-the-art methods.

Precision Analysis. The segmentation and detection precision with different subnets and pooling strategies are demonstrated in Table III. All ROI procedure of the frameworks receives feature maps with 256 channels from the separated large convolution layers. Two convolution layers with kernel size 3×3 are implemented for further feature extracting on ROIs. The segmentation and detection performances keep increasing as the outputs channel increase. More information will be reserved when applying 7×7 pooling for final output features, which will result in better performances in both segmentation and detection.

V. Conclusion

In this work, we introduced a large separated convolution and simplified subnet into the Mask R-CNN. We enable the ROI pooling operation much more efficient by decreasing the channel of input features with the large separated convolution. On the other hand, the Re-Designed light subnet also contributes a lot on the network speed up. We gain nearly $\sim 2\times$ speed up compared with the original ResNet50-based Mask R-CNN method and also result in a comparable object detection and instance segmentation results when evaluating our method on Microsoft COCO datasets. Although we mainly discuss the performance of the proposed method by using Mask R-CNN as a baseline in this paper, our method can be introduced to any other two-stage based methods for accelerating the speed of detection and segmentation.

Acknowledgment

This work is partially supported by grants from the National Key R&D Program of China under grant 2017YFB1002401, the National Natural Science Foundation of China under contract No. U1611461, No. 61650202, No. 61633002 and No. 61471402, the Open Topic of the Key Laboratory of Machine Perception under contract No. K-2018-8, the Program for the Outstanding Young Talents of Municipal Colleges and Universities of Beijing under contract No. CIT&TCD201804054, and the National University Student Science and Technology Innovation Program under contract No. 5111723300.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[2] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[5] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[8] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083*, 2015.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[10] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel mattersłimprove semantic segmentation by global convolutional network," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017, pp. 1743–1751.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *european conference on computer vision*. Springer, 2014, pp. 346–361.

[14] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 512–519.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 297–312.

[17] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3992–4000.

[18] ——, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[19] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 447–456.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[21] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2359–2367.

[22] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 534–549.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich *et al.*, "Going deeper with convolutions." Cvpr, 2015.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 5987–5995.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[28] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[29] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[30] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4820–4828.

[31] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in neural information processing systems*, 2016, pp. 4107–4115.

[32] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 525–542.

[33] B. B. Sau and V. N. Balasubramanian, "Deep model compression: Distilling knowledge from noisy teachers," *arXiv preprint arXiv:1610.09650*, 2016.