SegRPN: Scale Aware Joint Object Detection and Semantic Segmentation

Nakul Agarwal[1], Wei-Chih Hung[1], Yi-Hsuan Tsai[2], and Ming-Hsuan Yang[1,3]

[1]University of California, Merced
[2]NEC Laboratories America
[3]Google Cloud

***Abstract*— Object detection and semantic segmentation are two of the most fundamental tasks in computer vision. When addressed jointly, it can be applied to many fields such as autonomous driving. While most of the existing works mainly rely on sharing the deep convolutional features for jointly dealing with the two tasks, we exploit a much deeper connection between object detection and semantic segmentation. We observe that the feature maps used by Region Proposal Networks (RPN) resemble a dense class-agnostic (i.e., foreground/background) segmentation map. We exploit this observation to improve the functioning of RPN as well as improve performance for semantic segmentation. We propose a framework called SegRPN where we endow Faster-RCNN with a semantic segmentation branch using a shared Feature Pyramid Network (FPN) backbone. The semantic segmentation branch is facilitated by a class agnostic segmentation module, which serves two purposes: (i) it provides a scale specific objectness prior for semantic segmentation and (ii) it supports the RPN in the segmentation branch which improves the functioning of the RPN in the detection branch. Experimental results on Cityscapes dataset demonstrate that the proposed SegRPN is able to improve both object detection and semantic segmentation results.**

I. Introduction

The task of object detection is to identify all objects of predefined categories in an image and localize them via bounding boxes. Semantic segmentation operates at a finer scale where the goal is to assign a class label to each pixel. While there has been significant progress in the individual tasks of object detection [1], [2], [3], [4], [5], [6] and semantic segmentation [7], [8], [9], [10], [11], only few works have tackled them jointly. Existing approaches could benefit from solving these tasks jointly [12], [13], [14], [15]. For example, object detection should be easier if we know the semantics of the scene at the class-agnostic level, i.e. which pixels belong to the objects (e.g., car, person) and which belong to the background (e.g., sky, building). Conversely, semantic segmentation should be easier if we know where the object of interest is. In fact, it has been shown in the past that semantic segmentation usually used as a multi-task supervision can help object detection [16], [17], [18], and object detection used as a prior knowledge can improve semantic segmentation [19], [18]. Therefore, these two tasks are highly related.

Joint object detection and semantic segmentation has attracted a lot of attention in the past few years, leading to some interesting works. These works can be summarized into four categories: (i) a common encoder with parallel branches for object detection and semantic segmentation attached to the last layers of the encoder [20], [14], (ii) same as (i) but with features from semantic segmentation branch refining object detection [16], [21], (iii) encoder-decoder network where the concatenated feature map from each layer of the decoder is used for semantic segmentation, whereas each layer of the decoder focuses on multi-scale object detection [12] and (iv) encoder-decoder network where each layer of the decoder is simultaneously used for object detection and semantic segmentation [13]. Although the above methods have shown to be effective by jointly training the two tasks, we believe the performance of object detection and semantic segmentation can be further improved. One possible reason is that most of the above methods still mainly rely on simply sharing convolutional features for the two tasks, without focusing on any particular aspect of the pipeline responsible for semantic segmentation and object detection.

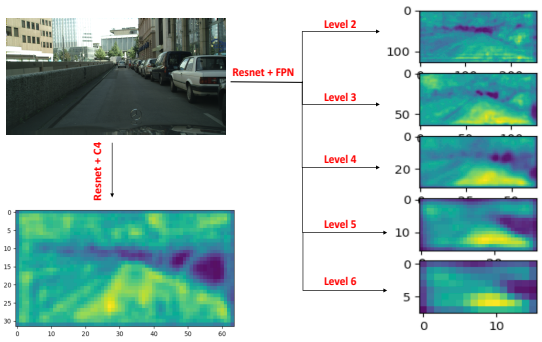

Fig. 1. RPN feature maps as class-agnostic segmentation maps. Feature maps are shown for two different backbone networks: Resnet+C4 and Resnet+FPN. In case of FPN, maps of each level (2-6) have been visualized. The darker regions have lower activation values while the brighter have higher activation values.

In this paper, we focus our attention on a much deeper connection between object detection and semantic segmentation. We observe that the feature maps used by RPN resemble a dense class-agnostic (i.e., foreground/background) segmentation map. We observe this across different backbone networks, two of which have been shown in Figure 1. We use this observation to improve the functioning of the RPN. The goal of the RPN is to produce proposals which have the potential of containing an object, which forms the first

part of a two stage object detector. These proposals are filtered based on probability score maps generated by RPN, as well as non-max suppression. They then are processed for final classification and regression, which forms the second part. Therefore, the RPN, and consequently the RPN input feature map as well as the RPN probability map play a very crucial role in object detection. If the first stage (i.e., RPN) itself misses out on good proposals, the second stage cannot recover them. Therefore, we propose a framework named SegRPN where we endow Faster-RCNN with a semantic segmentation branch using a shared Feature Pyramid Network (FPN) backbone. The semantic segmentation branch is composed of a multi-scale class agnostic segmentation module, which has another RPN (i.e. RPN_{mask}) attached to it. We improve the RPN in the detection branch (i.e. RPN_{det}) using RPN_{mask} in the segmentation branch in two ways: (i) we fuse the input feature maps of the RPN_{mask} score classifiers and bounding box regressors at different scale levels with the corresponding feature maps from RPN_{det}, (ii) we fuse the probability maps generated by RPN_{mask} score classifier at different scale levels with the corresponding ones in RPN_{det}. This improves the quality of proposals generated by RPN_{det} both in terms of precision and recall.

Since RPN_{mask} is attached to a mutli-scale class-agnostic segmentation module, it helps us learn scale specific features for class-agnostic segmentation. This means the higher resolution feature maps in the class-agnostic segmentation module focus on segmenting the lower scale objects, whereas the lower resolution feature maps focus on segmenting the higher scale objects. We then use these scale specific features to improve the semantic segmentation performance, which becomes possible since the class-agnostic segmentation module is attached to the class-specific module. The contributions of this work are summarized as follows:

- We exploit the observation that the feature maps used by RPN represent a dense class agnostic segmentation map, and treat it as a much deeper connection between object detection and semantic segmentation.
- We propose SegRPN, a framework for joint object detection and semantic segmentation that effectively exploits this observation.
- We show why leveraging the scale of objects is an integral part of SegRPN.

II. Related Work

Before we introduce our approach, we review in this section techniques for object detection, semantic segmentation and past attempts at jointly dealing with the two tasks.

A. Object Detection

The goal of object detection is to classify all objects and localize them via bounding boxes. The methods of object detection can be broadly summarized into two categories: one stage methods and two stage methods. Two stage method, as popularized in the R-CNN framework [22] and its variants [23], [1], is a proposal driven mechanism where the first stage generates a *sparse* set of candidate object locations and the second stage classifies each candidate location as one of the foreground classes or as background using a convolutional neural network. These methods have dominated the field of object detection and are the most representative frameworks among the two stage methods. Over the years, there have been some notable extensions to this framework, especially the ones based on multi-scale features with strong semantics [4], [18].

One-stage methods, on the other hand, are applied over a regular, *dense* sampling of object locations, scales and aspect ratios. Overfeat [24] was one of the first modern one-stage object detector based on deep networks. It was more recently followed by SSD [2], [25], YOLO [5], [6] and RetinaNet [3]. These detectors have been tuned for speed and so are much faster, but their accuracy still trails that of two-stage methods. In this paper, we adapt a two-stage method for the object detection pipeline since our focus is not on improving speed.

B. Semantic Segmentation

Semantic segmentation aims to predict the semantic label of each pixel in an image. It has achieved significant progress in the past few years [7], [8], [9], [10], [11]. The methods of semantic segmentation can also be broadly categorised into two categories: encoder-decoder methods and spatial pyramid methods. The encoder-decoder methods contain two subnetworks: an encoder subnetwork and a decoder subnetwork. The encoder subnetwork is usually based on the standard CNN models (e.g., VGG [26], ResNet [27], DenseNet [28]) pre-trained on ImageNet [29]. It extracts strong semantic features and reduces the spatial resolution of feature maps. The decoder subnetwork on the other hand, gradually upsamples these feature maps with reduced spatial resolution. While some methods [30], [31] directly upsample the feature maps using max-pooling indices of the encoder subnetwork, others [32], [33], [11] extract context information by adopting skip-layer connection between the feature maps from the encoder and decoder subnetworks.

Spatial pyramid methods rely on exploiting multi-scale information, which is extracted from the last output feature maps using the idea adopted from spatial pyramid pooling [34]. Specifically, this multi-scale information can also be utilised in different ways. PSPnet [9] propose pyramid pooling module, which downsamples and upsamples the feature maps in parallel. Some other prominent works [8], [35], [36], [37] propose to use multiple convolutional layers of different atrous rates in parallel (called ASPP) to extract multi-scale features. For our work, we opt a simple design for our semantic segmentation branch, appending a set of conolvutional layers on top of the FPN backbone in a multi-scale fashion.

C. Joint Object Detection and Semantic Segmentation

The task of jointly dealing with object detection and semantic segmentation has attracted a lot of attention in the past few years, where the goal is to simultaneously detect objects and predict pixel semantic labels by a single network. Recently, researchers have done some attempts. A graphical

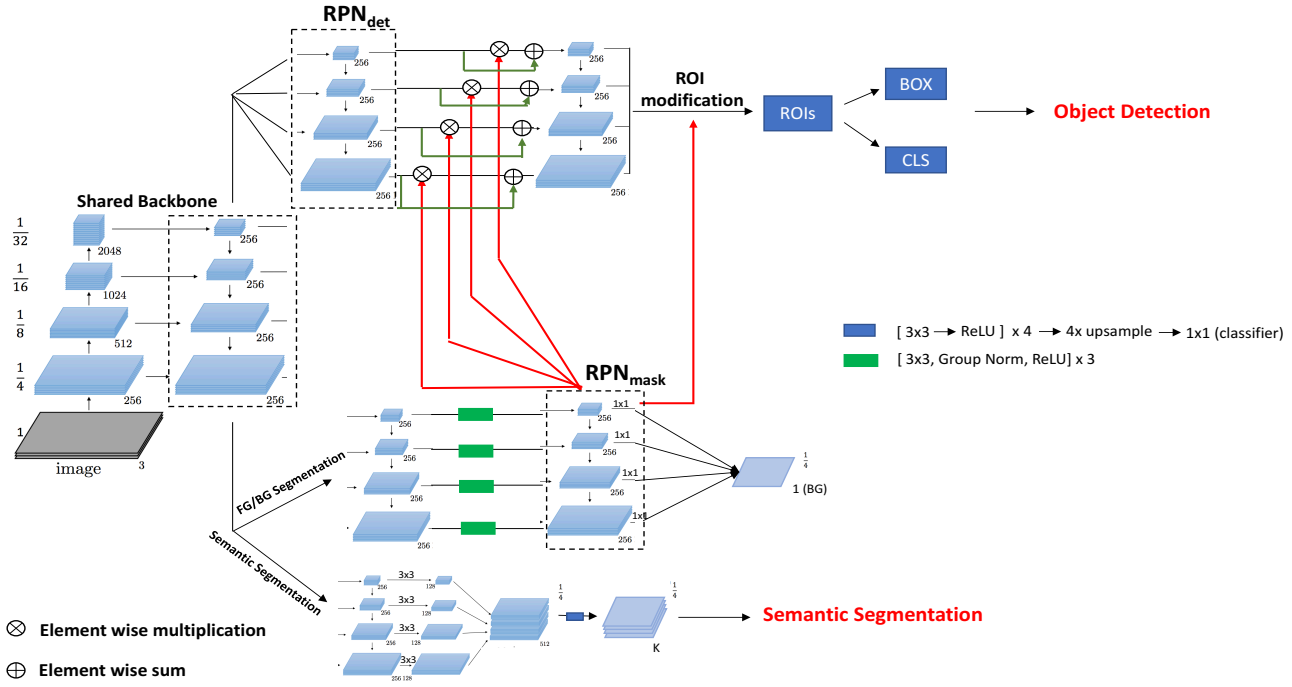

Fig. 2. Proposed Network Architecture. We use a Resnet+FPN backbone, which is shared by the object detection (top) and semantic segmentation (bottom) branches. We use the feature maps and probability maps from RPN_{mask} to modify the corresponding maps from RPN_{det} using element wise operations.

model to holistic scene understanding is proposed in [38]. [14] propose a joint object detection and semantic segmentation framework by sharing the encoder subnetwork. A real-time approach to jointly solve this task was also introduced in [12]. More recently, [13] propose an encoder-decoder network where each layer of the decoder is simultaneously used for object detection and semantic segmentation.

Almost all of the above works rely on sharing convolutional features in some or the other way, without giving a concrete explanation as to why these features are connected. Although a lot of progress has been made in this field, we argue that there is still room for further improvement. To support this claim, in this paper, we explore a novel connection between the two tasks of object detection and semantic segmentation. We subsequently provide a framework for exploiting this said connection and show some positive experimental results.

III. SegRPN

Different from the existing works dealing with joint object detection and semantic segmentation, we explore a new connection between the two tasks and build our model to exploit it. We observe that the feature maps used by RPN resemble a dense class-agnostic (i.e., foreground/background) segmentation map. The architecture of the proposed framework is shown in Figure 2. We endow the Faster-RCNN framework with a semantic and class-agnostic segmentation branch using a shared Feature Pyramid Network (FPN) backbone. We first discuss this observation in detail in Section III-A, and then move on to explaining the different components of the model.

A. RPN Feature Map as Class-Agnostic Segmentation Map

The RPN is the first stage of Faster R-CNN, and it's job is to generate regions of interest (i.e., ROIs) which potentially contain the object. When used on top of FPN, each level of the pyramid generates ROIs corresponding to a fixed anchor size. The lower the level of the FPN, the higher the resolution and the smaller the objects it focuses on. The RPN generates these ROIs using its own binary classifier (i.e. foreground vs background) and box regressor, which take as input a set of feature maps. Using these features maps, the RPN generates bounding box offsets and probability maps representing pixels whose anchors have a possibility of containing the objects. We visualize both the input RPN feature maps as well as the generated ROI probability map for an image from the Cityscapes dataset in Figure 3.

For our work, we stick with the standard RPN setting of 5 scales and 3 aspect ratios of $\{32, 64, 128, 256, 512\}$ and $\{.5, 1, 2\}$ respectively. This means that there are 5 pyramid levels (2-6), each generating a single RPN feature map and three ROI probability maps corresponding to each of the three aspect ratios. The RPN feature map along with its corresponding ROI probability maps are shown for each of the five pyramid levels in Figure 3 (a). There are two interesting observations here; First, the RPN feature maps have a lower activation region around the foreground objects in the image at all levels, where each level focuses on a certain scale of object. This is especially visible in the feature map of level 3, where the human and car contours are clearly

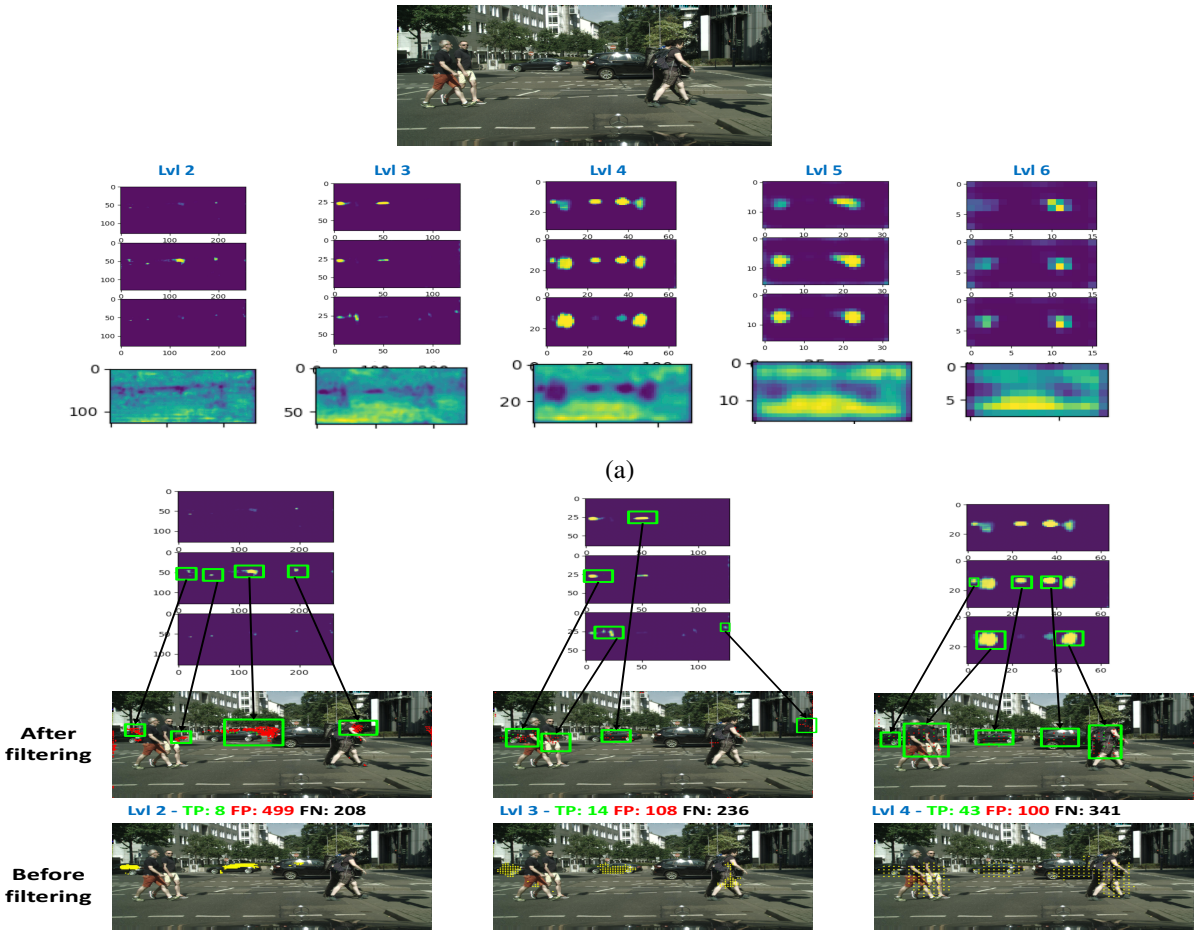

Fig. 3. In (a), visualization of the RPN feature maps and ROI probability maps for an image in the Cityscapes dataset is shown. Filtering of proposals by RPN based on ROI probability maps is shown in (b), where yellow points denote pixels with proposals having an overlap of greater than 0.5 with the ground truth bounding box before filtering. Red and green points are pixels denoting false positives (FPs) and true positives (TPs) respectively, after filtering.

visible. Second, the generated ROI probability map behaves in a complete opposite manner to the RPN feature map, having higher activation for the same region with lower activation in the RPN feature map. This suggests that the RPN classifier acts as a low-pass filter, trying to invert the activations of the RPN feature map. We also analyze the reason behind these two interesting observations in detail.

Each pixel in the RPN feature map generates 3 proposals corresponding to the three aspect ratios. The RPN filters these proposals using the ROI probability score and non-maximal suppression (NMS), to form a batch of positive and negative proposals with a ratio of 1:3. We show that this filtering done by the RPN based on the ROI probability score misses out on a lot of candidate regions, and we argue that this happens because these candidate regions were not segmented in the RPN feature map. In Figure 3 (b), we show the effect of the filtering done by the RPN as well as the importance of the ROI probability map. The pixels responsible for generating the proposals having an overlap of greater than 0.5 with the ground truth bounding box before filtering are shown in yellow at the bottom. After filtering, we show the true positives (TPs) and false positives (FPs) in green and red color respectively. These TPs and FPs are directly related to regions in the ROI probability map, as shown in Figure 3 (b). A lot of yellow pixels covering the object before filtering are missed once the filtering is done, which results in very few TPs and a lot of FNs and FPs at every level of the FPN. Note that we stick with the standard setting of using 2000 proposals after filtering by the RPN, and we show the exact number of proposals belonging to TPs, FPs and FNs for the three levels (2-4) individually in Figure 3 (b). We argue that this happens because these yellow pixels do not have a higher activation value in the corresponding ROI probability map, which in turn is caused by these regions not being segmented in the RPN feature map.

B. Class-Agnostic Segmentation Branch

As mentioned in Section III-A, the RPN feature maps rely on its segmentation characteristic for subsequent filtering of the generated proposals. However, these maps are not

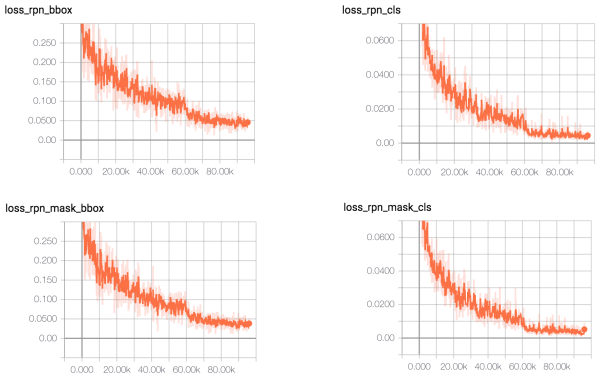

Fig. 4. Convergence of loss curves of both box regressor and class classifier corresponding to RPN_{det} (**top**) and RPN_{mask} (**bottom**).

entirely accurate. Additionally, semantic segmentation is class-aware, trying to discriminate between all the semantic classes at once. However, it does not specifically focus on the distinction between objects and the background.

So inspired by [13], we append a class agnostic module in parallel to our segmentation branch, which produces a binary (i.e., foreground/background) segmentation map. The estimated probability for the binary cross entropy (BCE) loss function of this branch for background class is denoted by $p \in [0, 1]$. This branch serves two purposes: (i) it provides a scale specific objectness prior for semantic segmentation and (ii) it supports RPN_{mask} which improves the functioning of RPN_{det} in the detection branch.

C. RPN in Class-Agnostic Segmentation Branch

Since the RPN feature maps resemble a class-agnostic segmentation map, a natural question to ask is: can we train RPN on top of the segmentation feature maps? We explore this question by adding a RPN called RPN_{mask} on top of the feature maps in the class-agnostic segmentation branch. We show the training loss converging for both box regressor and binary classifier corresponding to RPN_{det} and RPN_{mask} in Figure 4. This suggests that we're able to train the RPN_{mask} as well as RPN_{det}, further solidifying our claim that the RPN feature maps resemble a class-agnostic segmentation map.

The motivation behind adding another RPN in the class-agnostic segmentation branch and exploiting the segmentation modality to improve object detection is shown in Figure 5. We show an example of a challenging image from the Cityscapes validation set in Figure 5(a) and its corresponding predictions from our network in Figure 5(b) when RPN_{mask} is not attached. As can be gleaned from the object detection ground truth, the image contains a lot of objects, many of them being small in size. The small objects will naturally be tough to predict, but the network is still able to predict some of the small objects. However, it misses out on some objects which are reasonably large in size. As shown in Figure 3 and discussed in Section III-A, the RPN feature maps indirectly control the proposals which are picked after filtering by the RPN, and these filtered proposals eventually get picked as predictions. We argue here that the reason why objects are being missed in Figure 5(b) is because the RPN feature maps do not contain segmented regions corresponding to these objects. On the other hand, we show that the class-agnostic segmentation output, which also resembles an RPN feature maps, does a better job at segmenting these missed objects. Interestingly, the semantic segmentation output also recovers these missed objects, further facilitating the evidence that the segmentation modality can help the task of object detection.

Since we're able to successfully train the RPN_{mask}, we use the RPN_{mask} feature maps to modify the RPN_{det} feature maps. This modification is based on the intuition that the RPN_{mask} feature maps learn some relevant information missed out by the RPN_{det} feature maps, due to RPN_{mask} exploiting the segmentation modality. This modification is basically an element-wise multiplication followed by an element-wise sum of the RPN feature maps corresponding to each FPN level, as shown in Figure 2. We also modify the ROI probability maps based on the same intuition, by element-wise addition of the ROI probability maps generated by RPN_{mask} and RPN_{det} for each of the three aspect ratios, right before the sigmoid layer.

IV. Experiments

A. Datasets and Metric

Cityscapes. The Cityscapes dataset [39] is a collection of images with city street scenarios. It includes instance segmentation annotation which we transform into bounding boxes for our experiments. It contains 2,975 training images and 500 validation images.

Metric. For object detection, mean average precision (i.e., mAP) is used for performance evaluation. The mAP is calculated under the IoU threshold of 0.5. For semantic segmentation, mean intersection over union (i.e., mIoU) is used for performance evaluation.

B. Implementation Details

We use a standard FPN configuration with 256 output channels per scale. For the (pre-FPN) backbone, we use ResNet [27] models pre-trained on ImageNet [29] using batch norm (BN) [40]. All feature layers are jointly updated during training using Stochastic Gradient Descent (SGD). When used in fine-tuning, we replace BN with a fixed channel-wise affine transformation, as is typical [27]. The input images are rescaled to the size of 512×1024, and the size of mini-batch is 1. The total number of iteration in the training stage is 96k. We warm-start the learning rate from 0.0003 to 0.001 in 500 steps using linear annealing for stabilizing training, and then drop the learning rate by a factor of 10 at 60k and 80k iterations. We use a single Nvidia Titan XP, and implement the proposed method with Pytorch [41].

C. Results on Cityscapes

We show the results of our proposed framework in Table I. We first show the baseline results without our class-agnostic module. We note that the mIoU results are not comparable to

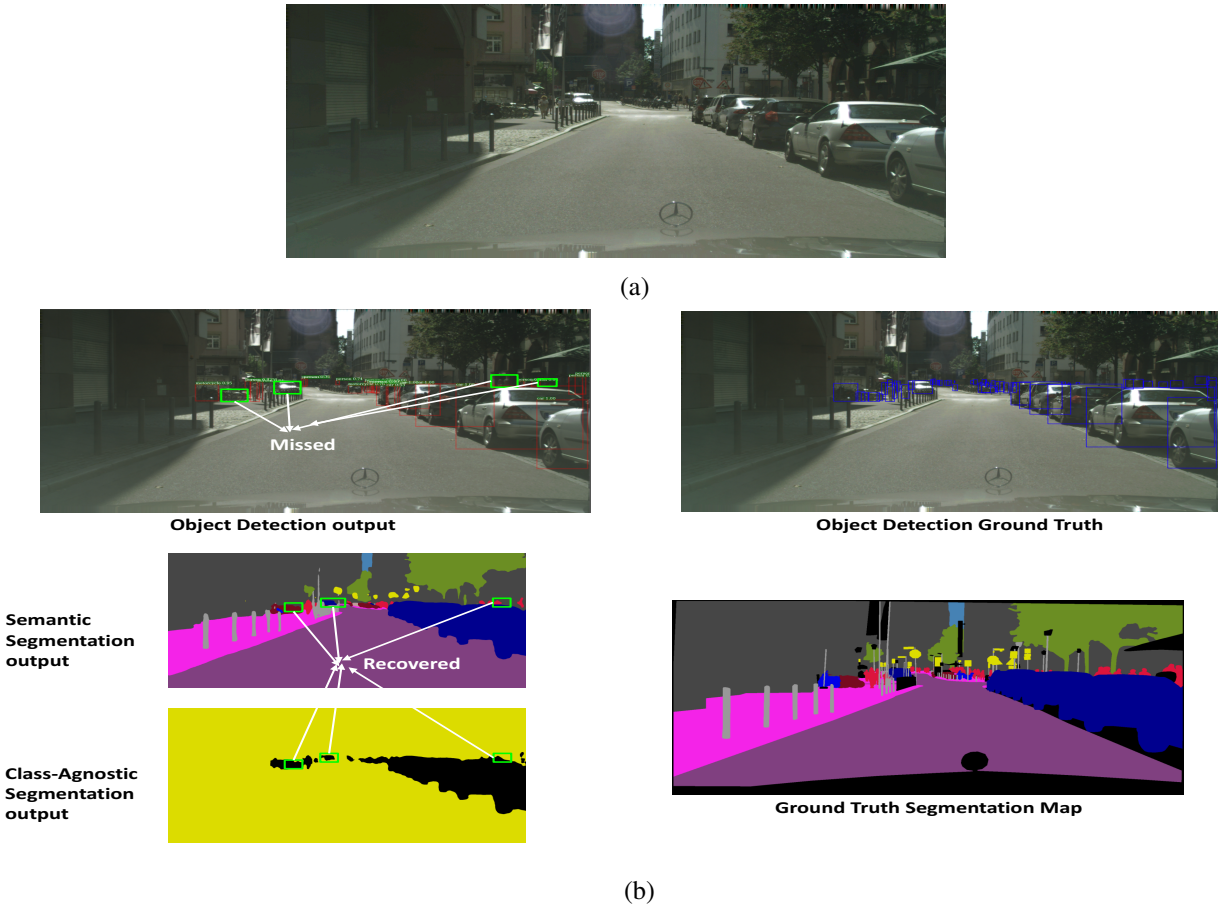

Fig. 5. Motivation behind RPN_{mask}. A challenging image from the Cityscapes validation set is shown in (a). In (b), we visualize the outputs from different parts of our proposed network when RPN_{mask} is not attached. Blue and red boxes represent ground truth and predicted bounding boxes. Green boxes represent large objects which are missed by the object detection branch but recovered by the semantic and class-agnostic segmentation branches.

the current state of the art [42], [43], primarily because of the training settings (batch size, no. of GPUs used, input image resolution, batch normalization etc). But since our focus is to explore the connection between object detection and semantic segmentation, we keep our training settings simple and don't compare with these methods. Also, since Cityscapes is not primarily used for object detection, we cannot compare the object detection results with other methods.

By modifying the ROI probability maps, we're able to improve the mAP by 1.5%. We also see an increase in the mIoU by .5%. The increase in the semantic segmentation performance can be attributed to the class-agnostic branch, which gives an objectness prior for semantic segmentation. Note that the mIoU* performance is extremely high since it's a binary classification task (i.e., foreground vs background). We also see an increase in the proposal average recall rate, suggesting that more objects are being covered by the RPN generated proposals. We also show the results without modifying ROI probability maps during inference to show the effect of the modification. Similar to the ROI probability map modification, the RPN feature map modification also improves the mAP by 1.1%. It's accompanied by a similar improvement in the mIoU and equally good mIoU* performance, along with an increase in the proposal average recall rate.

D. Additional Experiments and Analysis

We also conduct ablation studies to further explore the proposed hypothesis of treating RPN feature maps as a dense class-agnostic segmentation map. Since our framework is built on top of FPN, the scale of objects is an important aspect of the overall approach. In the original FPN paper [4], the authors adapt a heuristic to assign an ROI of width w and height h to a certain level P_k of the pyramid, which is given by:

$$k = \lfloor k_0 + \log_2(\sqrt{wh}/224) \rfloor \tag{1}$$

Here 224 is the canonical ImageNet pre-training size and $k_0 = 4$, as mentioned in [4]. This heuristic essentially suggests that each level of the FPN is designed to handle a certain scale of objects. Specifically,

$$k = \begin{cases} 5 & \sqrt{wh} \geq 448 \\ 4 & 448 > \sqrt{wh} \geq 224 \\ 3 & 224 > \sqrt{wh} \geq 112 \\ 2 & \text{otherwise} \end{cases} \tag{2}$$

TABLE I

RESULTS OF OBJECT DETECTION (MAP) AND SEMANTIC SEGMENTATION (MIOU) ON CITYSCAPES. WE ALSO SHOW CLASS-AGNOSTIC SEGMENTATION RESULTS (MIOU*) AS WELL AS PROPOSAL AVERAGE RECALL RATE (PROP. AR) AT 1000 PROPOSALS PER IMAGE.

Model	mIoU	mIoU*	mAP	Prop. AR
Baseline	67.34	NA	50.38	79.58
Baseline + ROI	67.80	94.41	51.87	80.22
Baseline + ROI (w/o ROI during inference)	67.80	94.41	51.65	79.77
Baseline + RPN	67.95	94.40	51.45	80.40
Baseline + RPN (w/o RPN during inference)	67.95	94.40	50.85	79.40

In this work, since we rely on the class-agnostic segmentation modality to improve object detection, we must divide the segmentation information based on scale in order for it to be applied to the FPN. We specifically use the ground truth class-agnostic segmentation map for this ablation study in order to verify the idea. Also, note that the canonical pre-training size (i.e., 224) is not really applicable for different input image sizes. So, we experiment with the above heuristic in three ways which we refer to as: i) Case 1 ii) Case 2, and iii) Case 3. The core idea behind these heuristics is to find a scale appropriate matching between the ground truth segmentation map and ROI assignment. The Cityscapes dataset has instance segmentation ground truth, which we use to find the scale of individual objects. We visualize these hueristics in Figure 7 and describe them in detail below.

Case 1. For this case, we use the same heuristic for ROI assignment as used in [4]. For dividing the ground truth segmentation map, we follow a different strategy. Each level of the FPN is designed to handle a certain scale of object, which is specified by the fixed anchor size used in the RPN. So we use these anchor sizes as a heuristic for dividing the segmentation map based on object/instance sizes. Specifically, let the FPN level to which the segmentation map will be divided and assigned to be represented by L. Then,

$$L = \begin{cases} 6 & \sqrt{a} \geq 256 \\ 5 & 256 > \sqrt{a} \geq 128 \\ 4 & 128 > \sqrt{a} \geq 64 \\ 3 & 64 > \sqrt{a} \geq 32 \\ 2 & \text{otherwise} \end{cases} \tag{3}$$

where a is the area of the bounding box of an object. Note that a is calculated keeping in mind the spatial stride of the network. The above heuristic essentially maps an object in the ground truth segmentation map to a level in the FPN. We visualize this heuristic in Figure 7a.

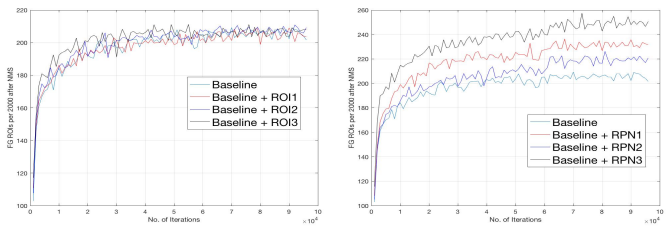

Fig. 6. Plot of Foreground ROIs per 2000 after filtering by NMS for all three heuristic cases over no. of iterations

Case 2. In the previous case, there was a difference in the heuristic used for ROI assignment [4] and the one we used for dividing the ground truth segmentation map. In order for them to be in sync, we keep the original heuristic for ROI assignment, but change our heuristic used for dividing the segmentation map. Specifically,

$$L = \begin{cases} 6,5 & \sqrt{a} \geq 448 \\ 4 & 448 > \sqrt{a} \geq 224 \\ 3 & 224 > \sqrt{a} \geq 112 \\ 2 & \text{otherwise} \end{cases} \tag{4}$$

We visualize this heuristic in Figure 7b.

Case 3. As mentioned earlier, the canonical pre-training size (i.e., 224) used in [4] is not really applicable for different input image size. This is because the canonical pre-training size affects the ROI assignment to the FPN levels, and the ROI sizes will be different for different input image size. Since we use an input size of $512x1024$, we empirically change the canonical pre-training size to 128 so that each level of the FPN gets some objects to handle. This can be visually seen in Figure 7c. Specifically,

$$L = \begin{cases} 6 & \sqrt{a} \geq 384 \\ 5 & 384 > \sqrt{a} \geq 192 \\ 4 & 192 > \sqrt{a} \geq 96 \\ 3 & 96 > \sqrt{a} \geq 48 \\ 2 & \text{otherwise} \end{cases} \tag{5}$$

Results. We show the results of the additional experiments in Table II. We also show effect of the modifications on the foreground ROIs generated by RPN after filtering by NMS in Figure 6. For Case 1, we observe that by modifying the ROI probability maps, we're able to improve the mAP by 1.1%. We also see an improvement in the semantic segmentation performance by .5%, which can be attributed to the improvement in the common FPN backbone features shared by the object detection and semantic segmentation branch. The improvement is also reflected in the proposal average recall rate, which suggests more objects being covered by the RPN after modification. We see a similar trend when modifying

TABLE II

RESULTS OF ADDITIONAL EXPERIMENTS ON CITYSCAPES. WE REFER TO THE THREE CASES BY THEIR CORRESPONDING SUBSCRIPT. WE SHOW OBJECT DETECTION (MAP) AND SEMANTIC SEGMENTATION (MIOU) RESULTS AS WELL AS PROPOSAL AVERAGE RECALL RATE (PROP. AR) AT 1000 PROPOSALS PER IMAGE.

Model	mIoU	mAP	Prop. AR
Baseline	67.34	50.38	79.58
Baseline + ROI_1	67.89	51.42	80.56
Baseline + ROI_1 (w/o ROI_1 during inference)	67.89	51.25	79.95
Baseline + RPN_1	68.52	51.57	82.59
Baseline + RPN_1 (w/o RPN_1 during inference)	67.95	50.84	78.86
Baseline + ROI_1 + RPN_1	68.18	51.40	83.34
Baseline + ROI_1 + RPN_1 (w/o ROI_1 and RPN_1 during inference)	68.18	49.93	79.02
Baseline + ROI_1 + RPN_1 (w/o ROI_1 during inference)	68.18	51.20	82.72
Baseline + ROI_1 + RPN_1 (w/o RPN_1 during inference)	68.18	50.30	79.90
Baseline + ROI_2	68.33	51.50	80.70
Baseline + ROI_2 (w/o ROI_2 during inference)	68.33	51.28	79.84
Baseline + RPN_2	68.23	51.71	82.54
Baseline + RPN_2 (w/o RPN_2 during inference)	68.23	50.17	79.84
Baseline + ROI_3	68.24	51.21	80.55
Baseline + ROI_3 (w/o ROI_3 during inference)	68.24	51.13	80.29
Baseline + RPN_3	68.43	51.56	82.84
Baseline + RPN_3 (w/o RPN_3 during inference)	68.43	50.43	79.64

the RPN feature maps, with an improvement of 1.2% for both mIoU and mAP. We also see an improvement in the proposal average recall rate by 3%. This improvement in the proposal average recall rate is also reflected in Figure 6 (**right**), where the FG ROIs generated are higher after modification when compared with the baseline. This however, is not very clearly reflected for the case of ROI modifications. In addition, we also provide results without incorporating the modifications at inference time, to further isolate the effect of the modification.

For Case 2 and Case 3, we observe a similar trend as in the previous case. By modifying the ROI probability maps, we're able to improve the mAP and and also the mIoU, which can again be attributed to the improvement in the common FPN backbone features shared by the object detection and semantic segmentation branch. The improvement is also reflected in the proposal average recall rate, which suggests more objects being covered by the RPN after modification. Figure 6 also shows the increasing trend in the FG ROIs for subsequent cases, which is also reflected in the higher values of proposal average recall rate. We see a similar trend when modifying the RPN feature maps, with an improvement for both mIoU and mAP.

V. CONCLUSION

In this paper, we propose an end-to-end learning framework called SegRPN for joint object detection and semantic segmentation. Our framework is based on the novel observation that the RPN feature maps in Faster-RCNN resemble a dense class-agnostic segmentation map. We treat this observation as a deeper connection between the tasks of object detection and semantic segmentation, and build a framework to exploit this observation. Experimental results on Cityscapes show the effectiveness of the proposed method. We also provide additional experiments and analysis to further explore this observation.

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] 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*, 2016, pp. 21–37.

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[4] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[6] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

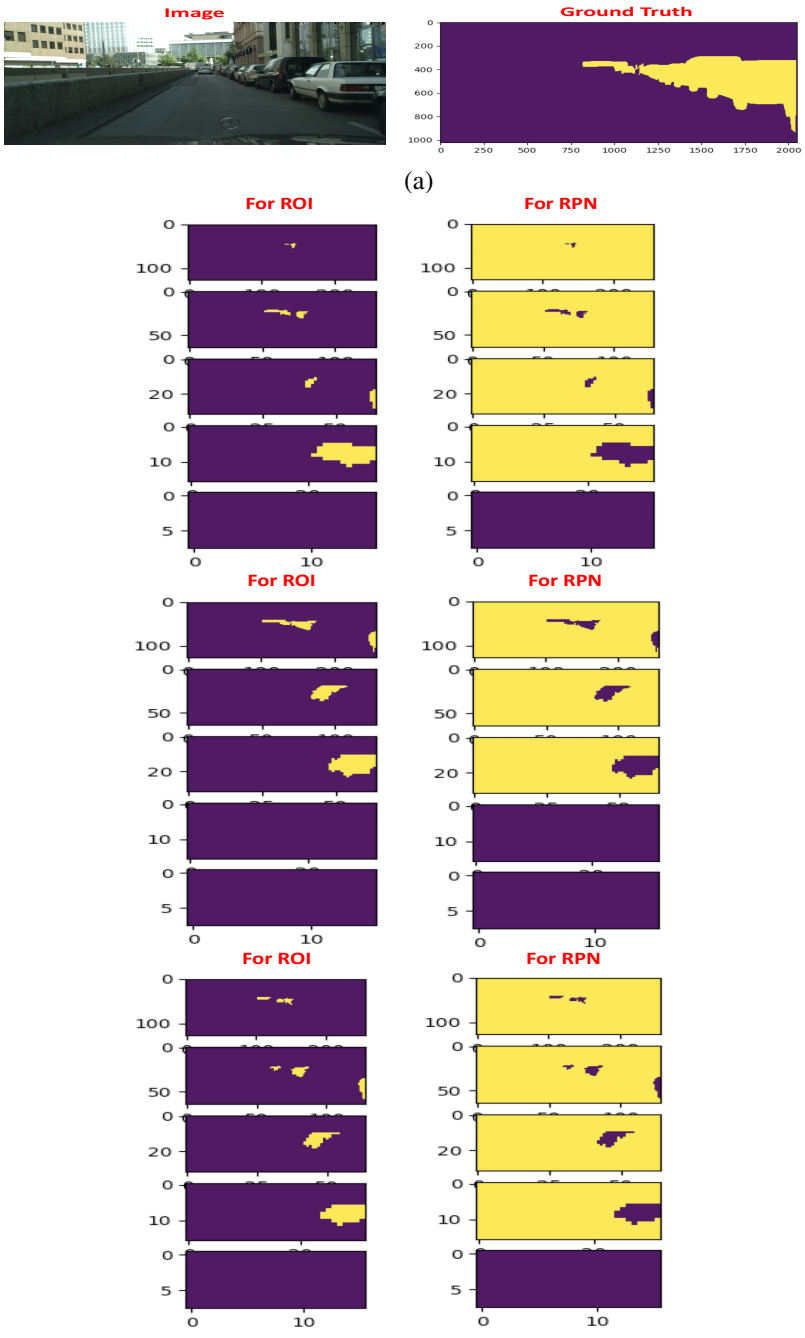

Fig. 7. Visualization of the heuristic cases. In (a) we show an example image (**left**) from Cityscapes validation set its class-agnostic segmentation ground truth (**right**). We visualize the segmentation map division for all 5 FPN levels (2-6) in (b) corresponding to heuristic 1 (**top**), heuristic 2 (**middle**) and heuristic 3 (**bottom**) for both ROI and RPN modifications.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890.

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conference on Computer Vision*, 2018, pp. 801–818.

[11] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1857–1866.

[12] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: A real-time deep network for scene understanding," in *IEEE International Conference on Computer Vision*, 2017, pp. 4154–4162.

[13] J. Cao, Y. Pang, and X. Li, "Triply supervised decoder networks for joint detection and segmentation," *arXiv preprint arXiv:1809.09299*, 2018.

[14] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *IEEE Intelligent Vehicles Symposium*, 2018, pp. 1013–1020.

[15] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6129–6138.

[16] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, "What can help pedestrian detection?" in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3127–3136.

[17] C. Lin, J. Lu, G. Wang, and J. Zhou, "Graininess-aware deep feature learning for pedestrian detection," in *European Conference on Computer Vision*, 2018, pp. 732–747.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[19] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*, 2016, pp. 75–91.

[20] G. Brazil, X. Yin, and X. Liu, "Illuminating pedestrians via simultaneous detection & segmentation," in *IEEE International Conference on Computer Vision*, 2017, pp. 4950–4959.

[21] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille, "Single-shot object detection with enriched semantics," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5813–5821.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[23] R. Girshick, "Fast r-cnn," in *IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

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

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[30] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[31] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[32] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4353–4361.

[33] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1925–1934.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[35] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[36] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *IEEE Winter Conference on Applications of Computer Vision*, 2018, pp. 1451–1460.

[37] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3684–3692.

[38] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: joint object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

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

[41] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[42] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6399–6408.

[43] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "Upsnet: A unified panoptic segmentation network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8818–8826.