2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

Efficient Instance and Semantic Segmentation for Automated Driving

Andra Petrovai and Sergiu Nedevschi

Abstract— **Environment perception for automated vehicles is achieved by fusing the outputs of different sensors such as cameras, LIDARs and RADARs. Images provide a semantic understanding of the environment at object level using instance segmentation, but also at background level using semantic segmentation. We propose a fully convolutional residual network based on Mask R-CNN to achieve both semantic and instance level recognition. We aim at developing an efficient network that could run in real-time for automated driving applications without compromising accuracy. Moreover, we compare and experiment with two different backbone architectures, a classification type of network and a faster segmentation type of network based on dilated convolutions. Experiments demonstrate top results on the publicly available Cityscapes dataset.**

I. INTRODUCTION

Automated vehicles must sense and understand their surroundings using a variety of sensors in order to be able to navigate in the complex traffic environment. A vehicle performs multiple tasks: it localizes itself on the map, it identifies all traffic participants, dynamic and static elements of the scene, and understands their movement and finally, it plans a route which is safe and obeys traffic rules. Automated vehicles usually perform sensor fusion based perception, based on cameras and LIDARs for example. A multi-sensor setting is more powerful than using only one sensor, providing richer and more accurate information due to the complementarity of sensors. In this work, we focus on image based recognition, but the results can be further fused with the outputs of other sensors for a complete understanding of the environment. 2D object detection and classification is robust due to the detailed appearance information that is present in the image. Moreover, it is easier to handle crowded scenarios and tackle difficult occlusion cases, often met in the complex traffic scene. We can approach sensor fusion based perception from a low-level perspective where fusion is performed at point/pixel level. Semantic image segmentation identifies the label of each pixel but cannot distinguish between instances of the same class. By fusing semantic image segmentation with the 3D point cloud we obtain a semantic point cloud where each 3D point has a semantic class. This provides a low level representation of the environment, where clustering methods could be used to detect and distinguish between different objects based on class and 3D position. Another approach to 3D object detection would be a high-level sensor fusion perception based on object detection and instance segmentation from image and 3D point cloud. By fusing the instance segmentation image with the 3D point cloud, each 3D object point has a semantic class and instance ID so we can directly detect and classify 3D objects.

In this work, we tackle the first task of the high-level sensor fusion based perception, which is semantic and instance image segmentation, using the Mask R-CNN framework. Lately, the research community has given attention to both tasks and proposed solutions using deep convolutional neural networks (CNN). Each task has its own architecture particularities. In the case of semantic segmentation, Fully Convolutional Neural Networks (FCN) [24][4][33][31] extract features using dilated residual blocks in order to preserve a higher output resolution. On the other hand, instance segmentation state-of-the-art results have been achieved by the Mask R-CNN framework [13] where a Feature Pyramid Network [22] provides a multi-scale feature representation for object detection and instance segmentation.

In this paper, we introduce and compare two types of convolutional neural networks for feature extraction in the Mask R-CNN framework. First, we explore a classification type of network with a $32\times$ downsampling factor. State-of-the-art solutions for semantic segmentation have shown that the results are greatly affected by this large downsampling factor due to the signal decimation, therefore dilated ResNet architectures have received much attention for solving this task. Therefore, we introduce and experiment with a much faster segmentation type of network that is based on 1D factorized dilated convolutions. Semantic segmentation results improve using this fast network and are comparable to a much deeper architecture of Mask R-CNN while running at least 4 times faster. Moreover, we make a comparative study of the two architectures. Experiments using the two types of architectures have been carried out on the challenging Cityscapes [7] dataset consisting of urban driving scenarios.

II. RELATED WORK

State-of-the-art semantic segmentation methods use deep learning for dense pixel prediction. Convolutional neural networks (CNN) have been extensively used for the classification task and Long et. al adapted them for semantic segmentation by introducing the Fully Convolutional Neural Network (FCN) [24]. One of the major benefits of the FCN is that it removes the fixed input size precondition by completely excluding the fully connected layers. Top performing methods have brought multiple improvements to the Fully Convolutional Neural Network architecture by using context modules [3], spatial pyramid pooling [33] or atrous convolutions [4]. Other solutions aim to improve

*Andra Petrovai and Sergiu Nedevschi are with the Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania , andra.petrovai@cs.utcluj.ro, sergiu.nedevschi@cs.utcluj.ro

semantic segmentation results with multi-task learning [16], where a shared representation is used to learn multiple tasks, for example: detection, semantic segmentation, instance segmentation and depth regression. A unified CNN architecture improves the performance of each separate task [2] since one task can leverage features learned by other tasks. In the following, we review two types of semantic segmentation solutions based on Fully Convolutional Neural Networks: single models specialized for semantic segmentation and unified architectures for multi-task learning.

A. Semantic segmentation architectures

Fully Convolutional Neural Networks (FCN) compute features by applying multiple convolutions. Since it is computationally unfeasible to learn features at original resolutions, strided convolutions and pooling layers are used to reduce feature maps size. Typically, a CNN trained for the task of classification learns features at 5 different scales, with the final layer having a $32\times$ lower resolution than the input. On one hand, downsampling feature maps brings the benefit of being able to better capture contextual information, on the other hand consecutive striding is harmful for the semantic segmentation task since the final segmentation map is obtained by upsampling the last layer of the CNN. In order to recover the resolution loss, several methods have been proposed.

Scale-controlling convolutions Atrous convolutions have been extensively used for controlling the resolution output [3], [4]. The authors modified the original ResNet [14] architecture by adopting dilated (atrous) convolutions with various rates in the last or last two residual blocks. Atrous convolutions enlarge the field of view of filters by capturing multi-scale context without decimating the resolution. Therefore, in a dilated FCN, the output resolution is typically $8\times$ or $16\times$ lower than the input size. Bilinear upsampling is applied on top of the last layer to obtain the final segmentation map.

An alternative to atrous convolutions represents scale-adaptive convolutions [32]. The authors overcome the problem of fixed-size receptive fields by introducing adaptive convolutions which are capable of learning dilation rates.

Spatial Pyramid Network These types of models are usually built on top of a dilated FCN and add a Spatial Pyramid Module. In PSPNet [33], this module encodes global information by applying various size average pooling kernels at the last layer.

DeepLabV2 [3] introduces Atrous Spatial Pyramid Pooling (ASPP) which performs parallel atrous convolutions with different rates. The resulting feature maps at different scales are concatenated and then bilinearly upsampled to the original resolution.

Spatial Pyramid Networks have shown outstanding results on multiple benchmarks [7], [10] by capturing context and multi-scale information.

Encoder-Decoder Networks using Atrous Convolutions and Spatial Pyramids have a large memory footprint due to the fact that features maps are generated at higher resolutions. Therefore, a simple bilinear interpolation operation is used to recover the original resolution. On the other hand, encoder-decoder networks usually use a deeper and narrower CNN for feature extraction and a more complex decoder replaces bilinear interpolation. The ENet [25] model has a lightweight encoder and deconvolution is used to learn the upsampling of low resolution features. The network runs in real time at the cost of reduced performance. ERFNet [27] achieves a better trade-off between high quality results and low computational costs by employing a residual network with factorized convolutions and deconvolutional layers. SegNet [1] has a symmetric encoder-decoder architecture and introduces the unpooling layer for upsampling, which transfers maxpooling indices from the encoder module to the decoder. The U-net model [28] uses shortcut connections from the encoder to decoder to help recover object details and spatial information. Encoders are usually based on the ResNet architecture as in RefineNet [20], [31] or on the DenseNet architecture [19] [15]. Encoder-decoder models achieve outstanding performance by learning the upsampling layers.

Image pyramid Another approach to capture multi-scale information is to resize the input samples at different resolutions and use a shared feature extractor [11]. The resulted feature maps at different scales are aggregated with concatenation [21] or attention models [5].

B. Multi-task learning

A perception system usually performs multiple tasks such as semantic segmentation, object detection, instance segmentation and others. Having separate models for each task implies a very high computational cost and memory footprint, which in most cases is unfeasible. A solution to this problem represents multi-task learning by having a shared CNN model that learns to optimize the tasks simultaneously. Since scene understanding can be achieved by perceiving both semantic and structure information, combining complementary tasks in a unified framework is beneficial for each particular task. In [16], the authors design a CNN that jointly learns semantic segmentation, instance segmentation and depth regression and propose a new multi-task loss which efficiently weights the loss of each task. UberNet [18] trains in an end-to-end manner a CNN that addresses several classification and regression tasks. Another solution in [9] predicts depth, surface normals and semantic labels using a single multi-scale architecture. MultiNet [30] enables real time applications such as autonomous driving with a very efficient network, solving vehicle detection and road segmentation. A top performing framework for object detection and instance segmentation is Mask R-CNN [13], which extends Faster R-CNN [26] with an instance segmentation branch. The unified architecture employs the Feature Pyramid Network [22] to learn multi-scale representations.

III. UNIFIED NETWORK ARCHITECTURE

In this section, we introduce our unified network architecture for object detection and classification, instance and semantic segmentation. It consists of a shared backbone

for feature extraction based on residual layers and multiple network heads for each task. The network can be trained end-to-end with a single optimization step.

A. Shared backbone for feature extraction

We experiment with two types of backbone architectures: one which is usually used in classification networks and one based on dilated convolutions used in semantic segmentation networks. Classification networks based on ResNet compute feature responses at multiple scales and usually use a $32\times$ downsampling factor. In [3][4] the authors have shown that using such a large factor degrades the performance of semantic segmentation, because of losing feature map details that could not be recovered. Therefore, they propose a network architecture with different dilation rates in the last residual blocks, which preserves important feature information and captures context.

The winning entry of the COCO Stuff Challenge 2017 [23] ResNeXt-FPN [17] proposed by team FAIR has shown that a backbone network used for detection and classification such as ResNet with a $32\times$ downsampling factor can be successfully used to achieve state-of-the-art results in semantic segmentation. For the classification type backbone feature extractor, we use as baseline the ResNet50-FPN network proposed in [8].

State-of-the-art semantic architectures use dilated residual blocks. Although dilated convolutions improve performance for semantic segmentation, it comes at the cost of higher memory usage and higher execution times. Computational resources are an important factor when developing intelligent vehicles applications, considering that the algorithms must run in real time on low power hardware with limited memory. Therefore, we shift our attention to efficient networks such as ERFNet [27] that achieves a good trade-off between accuracy and efficiency by leveraging residual blocks with 1D factorized convolutions.

1) Classification type of network: The first backbone network that we develop is based on the baseline ResNeXt-FPN FAIR architecture. 'ResNeXt-FPN' takes the Mask R-CNN framework for object detection and instance segmentation and extends it with a segmentation head. For the feature extraction stage, we employ ResNet50 [14], a 50 layer convolutional neural network with residual connections. A Feature Pyramid Network [22] is built in a top-down manner, starting with the last layer of the residual network by upsampling feature maps and merging them via lateral connections with corresponding features from the residual network. The process propagates coarser but semantically stronger features to the more finer feature maps, therefore each level of the pyramid will consist of more complex, richer features. On top of the shared backbone, we employ 3 networks heads. A Faster R-CNN detection and classification head consists of a Region Proposal Network (RPN) which generates Regions of Interest (ROIs) for each of the 5 levels of the Feature Pyramid Network. The ROIAlign layer extract features from ROI candidates having the largest objectness score and a bounding box classification and regression network outputs the final predictions. Mask R-CNN adds a mask prediction head by taking the top N predictions from the Faster-RCNN branch and learning a binary mask encoding the object for each bounding box.

For the segmentation head we adopt the network proposed in [8]. To capture multi-scale information, Atrous Spatial Pyramids with dilation rates 6, 12 and 18 are employed at the top layers of the FPN. The smaller resolution responses at 1/32 and 1/16 provide better localization and stronger classification while lower level layers capture scene details. Two 3×3 convolutions extract features at finer scales from 1/8 and 1/4. Then, at each level of the pyramid, we perform upsampling to reach the largest resolution. Feature maps are summed in a pyramidal manner and concatenated to obtain the final 512 feature maps. A 1×1 convolution will give the final logits.

2) Segmentation type of network: When designing the second backbone, we start from the state-of-the-art semantic segmentation networks architectures. Best performing architectures use dilated convolutions in the last residual blocks, which help in capturing context, while at the same time they keep the final output resolution 8 times or 16 times smaller than the input image. Keeping high resolution feature responses is important in order to preserve detail information which makes it easier to recover spatial dimension and object boundaries. We develop an efficient encoder-decoder architecture based on ERFNet [27] where the encoder network computes features at different scales and the decoder network combines the features to obtain a higher resolution representation. The building block of the ERFNet architecture represents the factorized residual layer. The authors redesign the non-bottleneck residual module by decomposing the 2D kernels into a linear combination of 1D kernels. A 2D non-bottleneck residual block sequentially stacks 3×3 convolutions and has a residual connection with the input of the module. The 1D non-bottleneck design transforms each 3×3 convolution into two 3×1 and 1×3 convolutions. This factorization results into fewer parameters with a reduction of 33% in the case of a kernel size of 3. The memory footprint is reduced and the computational efficiency is increased, while achieving high accuracy similar to more complex models. The encoder computes features maps at 3 resolutions by layering residual 1D non-bottleneck blocks with downsampling modules. On one hand, downsampling loses detail information from the image needed by the decoder to recover the semantic data but on the other hand, computation is more efficient on lower resolutions and deeper layers are important for capturing context. To achieve top accuracy, ERFNet's encoder avoids signal decimation by having an output resolution 8 times smaller than the image and captures long-range information especially important for classification of large objects by interleaving dilated 1D non-bottleneck blocks. We build a Feature Pyramid Network on top of the last layer by applying dilated convolutions with dilation 4 and stride 4 resulting in a $32\times$ smaller feature map and another dilated convolution with dilation 2 and stride 2 resulting in a $16\times$ smaller feature map. Since the dilation

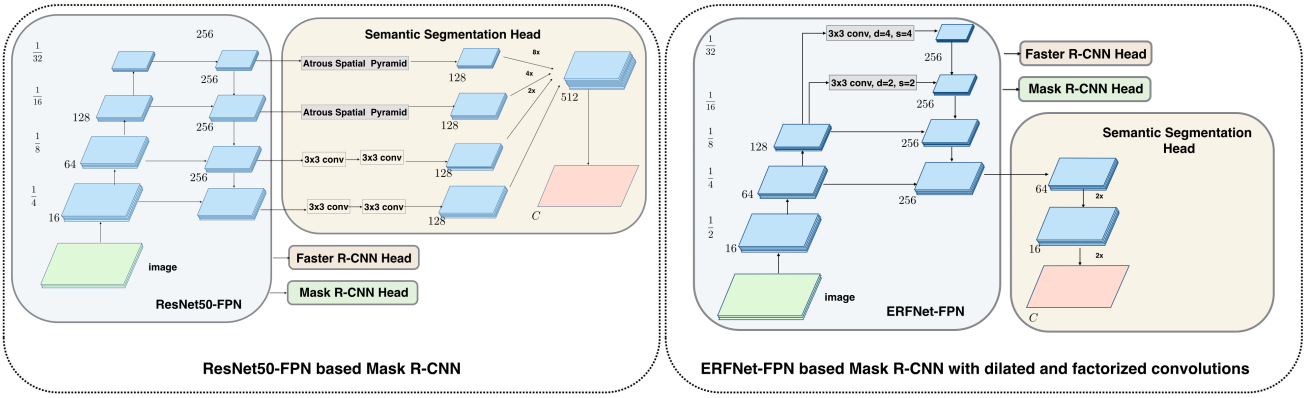

Fig. 1. Network architecture comparison between a ResNet50-based Mask R-CNN and ERFNet-based Mask R-CNN

rate is equal to the stride, this ensures subsampling features with equal rate.

We develop a lightweight decoder which upsamples feature responses to original image size by learning upsampling layers. The decoder actually represents the semantic segmentation head and is connected to the last layer of the Feature Pyramid Network at resolution 1/4. Two non-bottleneck blocks are interleaved between deconvolution layers which upsample the responses to the original image resolution.

B. Training and optimization

We provide training and optimization details for both architectures. Network training was carried out on a system with 2 NVIDIA GTX 1080Ti GPUs, each with 11 GB of memory.

1) Classification backbone: The Mask R-CNN solution based on ResNet50-FPN is initialized from pretrained weights on Microsoft COCO [23] dataset for object detection and instance segmentation. We use stochastic gradient descent (SGD) as optimizer with a polynomial learning rate decay starting from $5e-3$. We train with 2 resolutions: original 1024×2048 and 512×1024. We perform multi-scale training where the shorter edge of an image is scaled to a random choice from [800, 1024] for the original full resolution. For the smaller resolution, the set of scales is chosen from [416, 448, 480, 512]. We train for $32k$ iterations using a batch size of 1 per GPU due to memory constraints and a batch size of 3 per GPU for the smaller resolution. In the backbone, we do not use Batch Normalization due to small batch size and replace it with affine transformation from freezed batch normalization parameters from pretraining. In the segmentation head, we use Batch Normalization.

2) Segmentation backbone: The Mask R-CNN solution based on ERFNet-FPN is initialized from pretrained weights on ImageNet [29] for image classification. Only the ERFNet backbone is pretrained, the FPN and all heads are initialized from a normal distribution. Training this backbone is performed using Adam optimizer, with momentum 0.9, weight decay $2e-4$ and we start from a learning rate of $5e-4$. We employ a polynomial learning rate schedule. Since only the backbone is pretrained, this network needs longer training for $75k$ iterations. The ERFNet-FPN solution aims at a decreased processing time without compromising results. For faster runtime, the network was trained on a smaller resolution. We use multi-scale training, where the smaller edge of the image is resized from one of the following values [416, 448, 480, 512]. A batch size of 8 images (4 per GPU) is used in all our experiments. Batch normalization layers are trained in the entire network.

IV. EXPERIMENTAL RESULTS

We evaluate the two proposed models on the Cityscapes [7] dataset, which consists of 5000 high-resolution traffic images with pixel level semantic segmentation for 19 classes and instance segmentation for 8 object classes. The metrics used for semantic segmentation evaluation are Intersection Over Union (IoU) and for instance segmentation we use mean Average Precision (mAP@[.5:.05:.95], Average Precision over classes and 10 IoU levels from 0.5 to 0.95 with a step size of 0.05).

All experiments were performed in the PyTorch implementation of the Detectron [12] framework. We compare two unified architectures for semantic and instance segmentation, one is based on the classification type of network, ResNet50 and the other one is based on the segmentation type of network, ERFNet. First, we train a Mask-RCNN solution with ResNet50-FPN backbone that was pretrained on the Microsoft COCO [23] and ImageNet [29] dataset. After that, we train a Mask-RCNN solution with ResNet50-FPN backbone for the smaller 512×1024 resolution pretrained on ImageNet. Next, we train a Mask-RCNN solution with ERFNet-FPN backbone that was pretrained on ImageNet [29] dataset only.

First, we evaluate the semantic segmentation results obtained using the unified network using images of different resolutions. In Table I, we provide a comparison between

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bike	mIoU
ResNet50-FPN @ 1024x2048	97.7	82.3	91.2	48.6	51.3	56.9	66.9	73.1	91.5	61.8	93.1	80.1	59.8	93.1	63.0	77.5	64.1	59.7	75.3	72.9
ResNet50-FPN @ 512x1024	97.3	80.8	90.5	44.2	49.1	59.2	61.9	71.2	90	60	92.6	74.2	50.2	92.2	56	73.3	41.1	40.2	69.9	68.2
ERFNet-FPN @ 512x1024	97.3	80.4	90.4	47.8	50.7	60.3	61.9	72.3	90.9	59.7	92.4	75.5	54.1	92.4	60.6	77.4	57.9	40.4	70.1	70.1

TABLE I

SEMANTIC SEGMENTATION EVALUATION. CLASS MIOU ON CITYSCAPES VALIDATION DATASET, RESNET50-FPN TRAINED ON 1024×2048 AND 512×1024, ERFNET-FPN TRAINED ON 512×1024 AND EVALUATED ON 1024×2048 IMAGES.

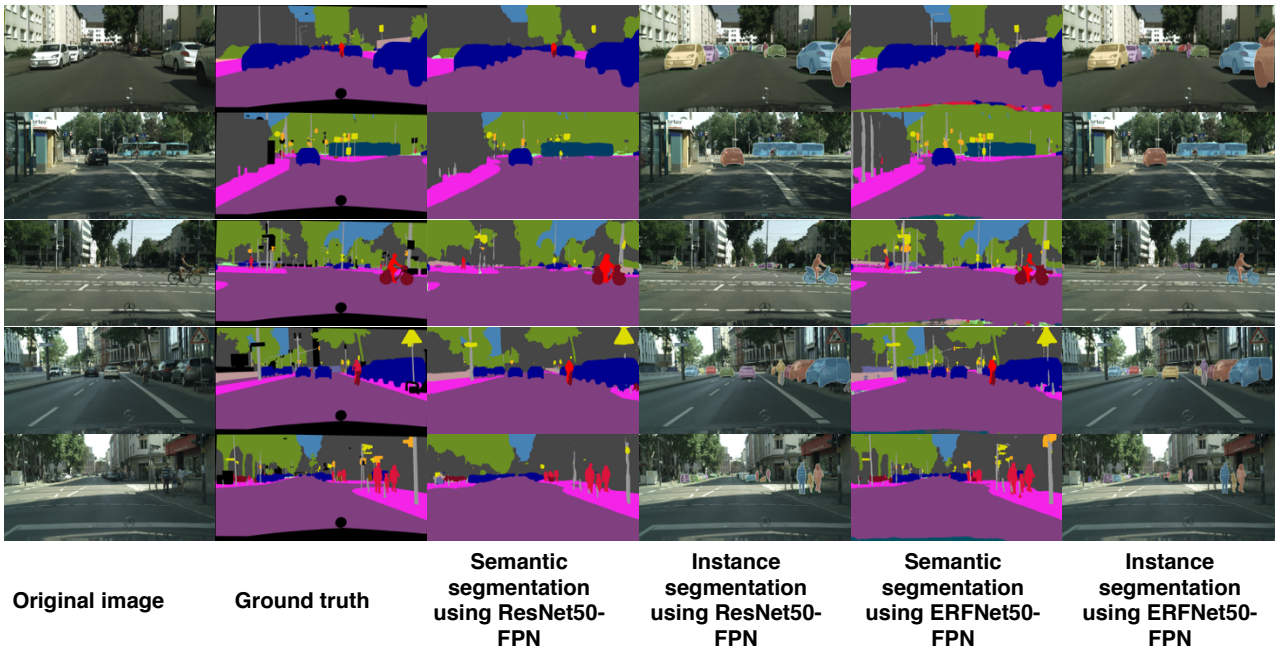

Fig. 2. Demo results for semantic and instance segmentation for the two types of networks

Method	mIoU class
DeepLabv2-CRF [3]	71.4
PSPNet [33]	78.4
DeepLabV3 [4]	79.3
DeepLabV3+ [6]	79.55
Ours - ResNet50-FPN@1024x2048	72.9
Ours - ERFNet-FPN@512x1024	70.1

TABLE II

CITYSCAPES RESULTS ON THE VALIDATION SET. MODELS TRAINED ON THE TRAIN FINE SET.

the two different networks based on ResNet50-FPN and ERFNet-50. Since we aim at obtaining lower execution time, we train the ResNet50-FPN based solution at two resolutions: the original 1024×2048 and the downsampled images at 512×1024. The reported results are evaluated at the full 1024×2048 resolution. Unless otherwise noted ResNet50-FPN refers to a network trained on the smaller resolution. Image resolution has a significant impact on the quality of the results as we can observe in Table I. Using full resolution for training, we obtain 72.9 IoU for semantic segmentation, while using 4 times smaller images, the IoU decreases with 4.7%. ERFNet, on the other hand is a network tailored for semantic segmentation by employing dilated convolutions in the last layers for increased receptive field and for information preservation. ERFNet-FPN performs better than ResNet50-FPN, having a 70.1 IoU.

In Table II we compare our solutions aimed at real-time applications with other state-of-the-art methods on the Cityscapes benchmark. Execution time is not specified for the other implementations. We consider only results for solutions trained on the train fine training set composed of 5000 semantic annotated images. In order to develop a fast network, we use more efficient building blocks such as factorized residual convolutions, but we also decrease the resolution of the input images. Table II shows that we can

still obtain competitive results with a unified network no matter the constraints against solutions that run in hundreds of milliseconds or seconds.

Backbone	AP det	AP mask	mIoU
ResNet50-FPN @ 1024x2048	37.2	36.4	72.9
ResNet50-FPN @ 512x1024	33.8	28.6	68.2
ERFNet-FPN @ 512x1024	31.1	27.7	70.1

TABLE III

OBJECT DETECTION, INSTANCE AND SEMANTIC SEGMENTATION RESULTS ON THE CITYSCAPES *VALIDATION* SET. OUR MODELS ARE TRAINED ONLY ON THE FINE CITYSCAPES TRAINING SET AND NO TEST-TIME AUGMENTATION WAS USED. THE RESOLUTION INDICATES TRAINING RESOLUTION.

In Table III, we evaluate all the 3 heads of our network. We can observe that decreasing the resolution also affects the object detection and instance segmentation results, since small objects will be harder to be detected in the smaller image. On the other hand, dilated convolutions prove to be an efficient mechanism for capturing long range information, improving the semantic segmentation. We can observe that the network based on ResNet50 still performs better at object detection and mask segmentation.

Backbone	Run time (ms)	AP mask	mIoU
ResNet50-FPN @ 1024x2048	150	36.4	72.9
ResNet50-FPN @ 512x1024	55	28.6	68.2
ERFNet-FPN @ 512x1024	44	27.7	70.1

TABLE IV

RUN-TIME (MS) MEASURED FOR MASK-RCNN PERFORMING OBJECT DETECTION, INSTANCE SEGMENTATION AND SEMANTIC SEGMENTATION ON NVIDIA RTX 2080TI GPU

In Table IV, we measure the inference time of the networks. ERFNet-FPN solution is faster than the ResNet50-FPN, while having significantly higher accuracy for semantic segmentation. The segmentation results of ERFNet-FPN trained on 512×1024 images are similar in accuracy with ResNet50-FPN trained on full resolution, while ERFNet-FPN network is more than 4 times faster. To be noted that we only measure the forward time of the network and do not consider pre-processing and post-processing operations such as image normalization, bounding box non-maxima suppresion or mask upsampling.

Component	ERFNet-FPN (ms)	ResNet50-FPN (ms)
Backbone	9	15
FPN	5	5
RPN	16	16
Box head	5	5
Segmentation head	5	10
Mask head	4	4

TABLE V

BREAK-DOWN OF RUNTIME IN MS FOR ERFNET-FPN BASED MASK-RCNN AND RESNET-50 BASED MASK-RCNN FOR A RESOLUTION OF 512X1024 ON NVIDIA RTX 2080TI GPU

We investigate the execution time of our proposed solutions and break down the inference time per module, in Table V. The Region Proposal Network is the most expensive module, being followed by the backbone. The mask and box head do not depend on the resolution of the input image since RoiAlign will sample the same number of points regardless of input resolution.

Moreover, we include some visual results for comparison in Figure 2. The ResNet50-FPN network and the ERFNet-FPN trained on 512×1024 input have visually similar results for the instance segmentation task. It can be seen that problems may appear for smaller objects for the instance segmentation task, while erroneous pixel level labeling for large objects for semantic segmentation may occur, resulting in objects having multiple labels.

V. CONCLUSION

In this work, we propose two types of unified end-to-end learnable deep neural network architectures for semantic, instance segmentation and object detection and classification based on the Mask R-CNN network. The contribution of the paper relies in the development of a fast and efficient network that can reach good accuracy, comparable to other state of the art solutions. Moreover, we study and compare two different backbone architectures suitable, one for classification and one for segmentation and present their benefits and drawbacks in the context of a unified framework for 3 different tasks.

ACKNOWLEDGMENT

This work was supported by the EU H2020 UP-Drive project under grant nr. 688652, SEPCA project PN III PCCF, no. 9/2018 and MULTISPECT project PN III PCE, no. 60/2017.

REFERENCES

[1] 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*, 39(12):2481–2495, 2017.

[2] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[3] 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*, 40(4):834–848, 2018.

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

[5] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*, 2018.

[7] 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 *CVPR*, 2016.

[8] A. D. Costea, A. Petrovai, and S. Nedevschi. Fusion scheme for semantic and instance-level segmentation. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3469–3475. IEEE, 2018.

[9] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[12] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[16] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] A. Kirillov, K. He, R. Girshick, and P. Dollár. A unified architecture for instance and semantic segmentation. `http://presentations.cocodataset.org/COCO17-Stuff-FAIR.pdf`, 2017.

[18] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *arXiv preprint arXiv:1609.02132*, 2, 2016.

[19] J. Krapac and I. K. S. egvic. Ladder-style densenets for semantic segmentation of large natural images. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 238–245, Oct 2017.

[20] 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 (CVPR)*, 2017.

[21] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[23] 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 *ECCV*, 2014.

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[25] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 91–99, Cambridge, MA, USA, 2015. MIT Press.

[27] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2018.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[30] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016.

[31] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *arXiv preprint arXiv:1702.08502*, 2017.

[32] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Scale-adaptive convolutions for scene parsing. In *Proc. 26th Int. Conf. Comput. Vis.*, pages 2031–2039, 2017.

[33] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017.