Pyramid-DRN: A Case Study on Category-wise Semantic Segmentation of Real Urban Scenes

December 14, 2019

ABSTRACT

Feature pyramids have proven to be useful in extracting feature maps at multiple scales. Similarly, dilated residual networks have the capability to produce high resolution feature maps with fewer parameters. In this work, we leverage the power of both feature pyramids, and dilated residual networks. To benchmark our model, we consider a simple yet highly practical problem - category-wise semantic segmentation. For this task, we build a light-weight architecture with just 10 M parameters. To further improve performance on under-represented classes with nuanced boundaries, we incorporate a boundary refinement module. This module acts on different levels of the architecture, that contain progressively increasing amounts of information, again reminiscent of a pyramid. On the highly diverse and complex dataset, Berkeley Deep Drive, our composite architecture performs better than each of the individual modules. The code is available here

1 Introduction

Right from the days of classical computer vision, building image pyramids and feature pyramids have proven to be extremely useful in solving problems that require the detection of objects of varying sizes. Even today, many deep learning papers use pyramids in their architectures. Another revolution in the field of deep learning has been spearheaded by the introduction of dilated residual networks Due to its ability to increase spatial resolution without increasing the number of parameters, it finds applications in object detection, segmentation among many other tasks. These observations lead us to the question: can the power of feature pyramids and dilated residual networks be combined to build a superior architecture? The answer is an astounding yes, which we explore in the context of category-wise semantic segmentation.

Semantic segmentation is an extremely important computer vision task that finds applications in diverse fields such as autonomous driving, medical imaging, robotics, satellite imaging, etc. Its goal is to assign a semantic label (car, tree, grass, sky, etc) to every pixel in the image. Deep learning for supervised semantic segmentation [1],[2],[3] has seen immense progress in the past few years. However, many of these architectures are very deep and have high memory requirements. Practical applications require the model to fit and evaluate on large images (as captured from the car) in limited memory.

Most current architectures perform semantic segmentation on the 19 classes benchmark as defined by Cityscapes [4]. Pragmatically speaking, it is not necessary for a vehicle to know whether there is a car or a bus at a particular location. Knowledge of the presence of a vehicle is sufficient. Similarly, vegetation and terrain can be clubbed under the category terrain; wall, fence and buildings can be thought of as construction. We follow Cityscapes' definition of categories to group similar classes. This leads us to a 7 class semantic segmentation problem. We postulate that this basis of classification has more relevance in autonomous driving systems, particularly for brake control and accident prevention applications.

Past work on class-wise semantic segmentation rely on a deep CNN backbone like ResNet-101 or ResNet-50. While these models perform well on the problem of category-wise semantic segmentation, we postulate that this task does not require architectures with backbones as heavy as ResNet-101. With the careful and intelligent design of a neural network pipeline, it is possible to solve this relatively simple task in a memory and speed efficient way. For improved performance on a diverse and complex dataset like Berkeley Deep Drive [5], we propose a light-weight architecture. In our work, we combine the power of feature pyramids [6] and dilated residual networks [7]. Following the recent success of UNet with lateral connections, we build a feature pyramid to extract feature maps at different scales. These

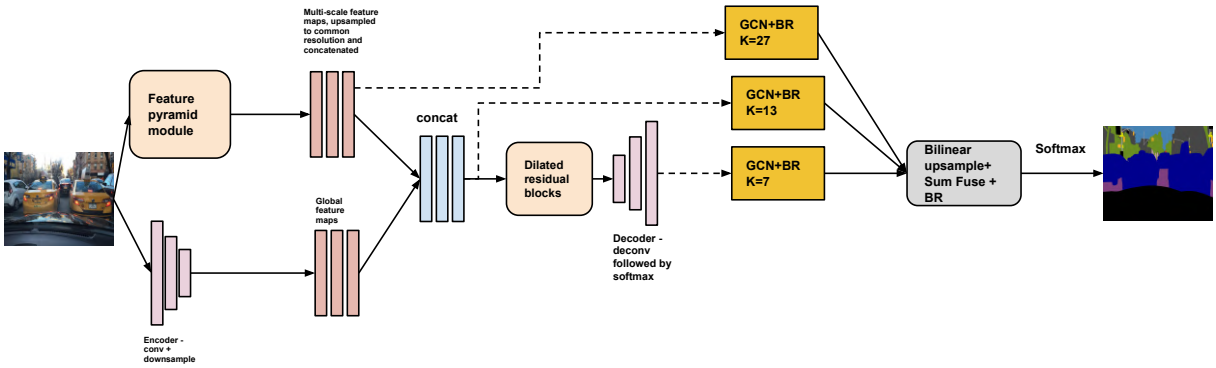

Figure 1: This schematic depicts our architecture that consists of a feature pyramid module and dilated residual module. The image is passed through a feature pyramid module to obtain multi-scale feature maps. These maps along with global feature maps from the encoder are refined using a dilated residual network module. A decoder with deconvolution layers is finally used to produce soft segmentation maps. Global convolution layers, as depicted in figure 3B, are applied over the outputs from the feature pyramid module, the output from the concatenation stage (input to DRN module) and the decoder. This is followed by the application of boundary refinement layers, as in, figure 3C. A final softmax operation produces the output soft segmentation maps, which along with the ground-truth labels are used to minimize the multi-class cross entropy loss function.

multi scale feature maps help in focusing on objects of varied sizes. To provide additional information for context, we learn global feature maps from the RGB image using residual blocks. These feature maps are concatenated with the multi-scale feature maps and passed through dilated residual blocks for refinement. A decoder with deconvolution layers is used to generate the soft segmentation map. To further improve performance of small objects and to enhance the boundaries, we incorporate a boundary refinement module. This module consists of global convolution [8] and boundary refinement layers. The global convolution layers are applied systematically at three levels of the architecture. The boundary refinement module thus capsulizes information from various components of the model, which by themselves form a pyramid with different amounts of knowledge at different levels.

Our experiments demonstrate that our method performs well on a very diverse and complex dataset, the Berkeley Deep Drive. In addition, we also conduct experiments on a comparatively small dataset, Cityscapes. We conduct extensive ablation studies to determine the performance of different components in our system. We believe that this idea of extracting multi-scale feature maps and global feature maps followed by refinement using a light-weight dilated residual networks to improve results can be extended to deeper versions of DRN like DRN-C-42, DRN-C-50 and also to DeepLab. In a nutshell, the contributions of this paper are as follows:

1. We propose an composite neural network architecture that combines the power of feature pyramids (multi scale feature maps) and dilated residual blocks. We benchmark our model on the task of category wise semantic segmentation.
2. We further incorporate a boundary refinement module with global convolution and boundary refinement layers to improve performance.
3. We demonstrate our results and analysis on Berkeley Deep Drive and prove that our composite architecture performs better than the individual components. We conduct extensive experiments and ablation studies in this regard. We further evaluate our model on a comparatively small dataset, Cityscapes.

2 Related work

Feature pyramids and image pyramids have proved to be useful for multiple computer vision tasks. The advent of convolutional neural networks has led to defining feature pyramids [6], [9] with learnable parameters. Further, deep learning has seen immense success in the form of ResNet and dilated ResNet [7].

The task of semantic segmentation has seen immense progress in the last few years. [3] uses a series of fully convolutional layers to generate segmentation maps. The UNet architecture uses skip connections and a symmetric shape to improve on [3]. [10] proposes a decoder with learnable deconvolution layers. [11] proposes an encoder decoder architecture built on object proposals for instance aware semantic segmentation. The method performs well on not just diffuse classes like road and sidewalk, but also on small objects with nuanced boundaries that may have a low

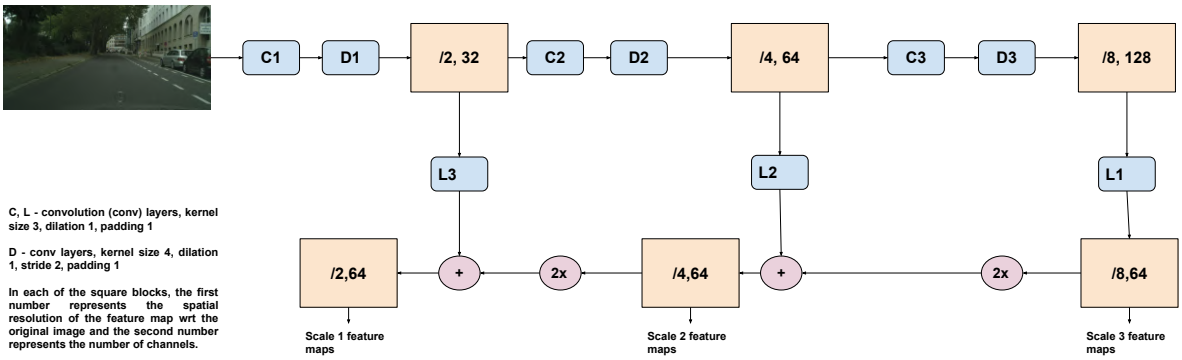

Figure 2: In this figure, we depict our feature pyramid moule. The feature pyramid module resembles a UNet with lateral connections. Various levels of the feature pyramid correspond to feature maps at different scales.

group	classes
flat	road, sidewalk
construction	building, wall, fence
object	pole, traffic light, traffic sign
nature	vegetation, terrain
sky	sky
human	person, rider
vehicle	car, truck, bus, train, motorcycle, bicycle

Table 1: In this table, we present the norm followed for grouping classes into categories.

representation in the dataset. Mask RCNN [12] combines Faster RCNN [13] and [3] in one architecture to detect predict masks. Weakly supervised methods such as [14], [15], [16] eliminate the need for large amounts of data. [2] uses pyramid pooling to construct segmentation maps while [7] uses dilated convolutions. [1], [17] use atrous convolution to generate state-of-the-art results.[18] uses object detection information from Faster RCNN to transfer the weights to the Mask RCNN network to learn segmentation maps. In [19], the authors design a fish-like network to preserve information at all resolutions. Recent papers like [20], [21], [22], [23], [24], [25] focus on improving the performance of classes that have objects with nuanced boundaries by incorporating a boundary refinement/ boundary perdiction branch. Light-weight architectures for semantic segmentation have been explored in [11], [26], [27], [28], [29]. We notice that all these architectures have been trained on the class wise segmentation problem, and the category mIoU scores reported are derived from the class wise segmentation results. In our work, we directly train using category-wise labels.

3 Our method

3.1 Description of the problem

The goal of this paper is to show how leveraging the power of feature pyramids and dilated residual blocks can lead to improved performance in computer vision tasks. In this context, we consider the problem of category-wise semantic segmentation. Given an RGB image, the goal is to learn a network to predict the category each pixel belongs to. Our problem is completely supervised, i.e. the labels are known. We follow Cityscapes' class description and define the seven classes for our category wise semantic segmentation problem as described in table 1.

3.2 Proposed model

We propose to combine the power of FPN and DRN for category wise semantic segmentation of complex and diverse datasets. The architecture consists of a feature pyramid module [6] to extract feature maps at various scales. This along with global feature maps (extracted using convolution layers from the original image) is passed through dilated residual blocks [7] for further refinement. Upsampling is done using a decoder that has deconvolution layers. A

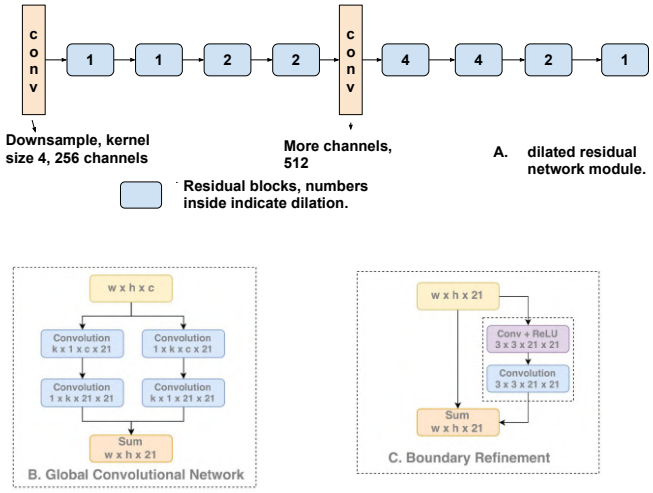

Figure 3: In this figure, we depict all other modules used in our architecture. Figure A depicts our dilated residual network module. Figures B and C [8] depict the global convolution and boundary refinement layers.

boundary refinement module with global convolution [8] and boundary refinement layers is further incorporate to refine boundaries and improve performance. Softmax is finally applied to produce soft segmentation maps.

Feature pyramid module: Figure 2 depicts our feature pyramid module [6]. It is a top-down architecture with lateral connections. It extracts high level semantic feature maps at all scales. The C blocks represent convolution layers, for which we use two convolution layers each, of kernel size 3. The D blocks represent downsampling layers, for which we use one convolution layer each, of kernel size 4. We prefer learnable downsampling over max pooling, or average pooling as pooling operations can lead to artifacts. For the lateral connections, we use one convolution layer each, of kernel size 3. We use one convolution layer each to refine the feature maps after the sum operation between the lateral feature maps and upsampled feature maps from the previous stage. This leads us to obtain multi-scale feature maps, all of which are semantically consistent. We now upscale these feature maps to a common resolution by bilinear upsampling. Further refinement is done using two residual blocks. In total, our feature pyramid module consists of 16 convolution layers of kernel size 3 and 3 convolution layers of kernel size 4. This module has 0.55 M parameters.

Encoder: We generate global feature maps from the original RGB image by passing it through a series of residual blocks. These feature maps are concatenated with the feature maps obtained from the feature pyramid in the previous stage. This module contributes to 9 convolution (kernel size - 3) layers and has 0.067 M parameters.

Dilated residual network module: The feature maps from the feature pyramid module and the encoder are concatenated and passed through a dilated residual network [7] module for further refinement. The dilated resnet block consists of dilated convolutions, which increases the receptive field and systematically aggregates multi-scale contextual information without loss of output resolution and increase in the number of parameters/ computation. The module consists of 8 residual blocks. The first two blocks have a dilation of 1, the next four have dilations of 2 and 4 respectively. This is followed by one residual block with dilations 2 and 1 each. The module consists of 18 convolution layers in total, each with a kernel size of 3. This is followed by a decoder with 3 deconvolution layers (configured to upsample by a factor of 2), each followed by a convolution layer with kernel size 3.

Decoder: The feature maps generated from the dilated residual network module have a resolution that is one eight of the resolution of the input image. These need to be upsampled to the size of the image. Architectures in the past use either bilinear upsampling or learnable deconvolution layers. The latter is more effective, but increases the number of parameters in the architecture. We conduct experiments with both.

Boundary refinement module: To further refine the boundaries, we propose to use a boundary refinement module. In this module, we use global convolution [8] and boundary refinement layers. The boundary refinement module, that comprises of global convolution (symmetric, separable large filters) and boundary refinement (modelled as a residual structure) layers. The global convolution layers, which serve as an equilibrium for the kernel size required for classification and localization tasks, act on various levels (multi-scale feature maps , the feature maps after concatenation (input to dilated residual network module) and the final decoder feature maps) of the feature pyramid-dilated residual network architecture. We consider the outputs of the different components of our architecture to perform form layers

Model	Data.	Flat	Con.	Obj.	Nat.	Sky	Hum.	Veh.	mIoU	Acc
FPN (Baseline)	BDD	91.04	71.6	20.39	78.25	92.47	10.95	74.52	62.75	89.71
DRN-deconv (Baseline)	BDD	92.27	72.23	30.1	75.25	92.4	22.51	79.85	66.37	90.04
FPN-DRN-Bi (Ours)	BDD	91.18	72.69	25.4	77.24	92.02	22.1	79.13	65.77	90.18
FPN-DRN-deconv (Ours)	BDD	93.74	77.62	37.04	80.62	94.32	35.85	85.85	72.15	92.41
FPN-DRN-deconv-BR (Ours)	BDD	94.16	79.32	39.93	81.56	93.97	37.09	87.73	73.46	92.96
FPN (Baseline)	city	96.9	82.56	33.85	86.56	88.11	50.29	78.95	73.89	93.17
DRN-deconv (Baseline)	city	75.9	77.97	25.82	82.19	86.69	37.23	75.19	68.71	91.32
FPN-DRN-Bi (Ours)	city	95.09	77.56	26.43	80.6	83.92	39.85	73.47	68.13	86.82
FPN-DRN-deconv (Ours)	city	96.77	81.79	34.76	85.19	88.5	47.8	80.56	73.62	92.88
FPN-DRN-deconv-BR (Ours)	city	97.23	83.55	40.05	85.61	89.31	53.84	83.96	76.22	93.61

Table 2: In this table, we present our results. The numbers for each category correspond to their respective IoU scores. The last column enlists the overall pixel accuracy.

of a pyramid, each level encapsulates different amounts of information about the model. Global convolution layers with receptive field sizes of 27, 13 and 7 are applied on these to generate multi-stage semantic score maps which are upsampled and sum-fused to generate new feature maps. The final semantic score map is generated after refinement using boundary refinement layers.

3.3 Datasets

We evaluate our model on two autonomous driving datasets: Berkeley Deep Drive (BDD) [5] and Cityscapes [4]. BDD train has 7000 images while Cityscapes train has far less number of images (2975). Cityscapes images span across cities in Europe while BDD is a much more diverse and complex dataset captured across the US under varying weather conditions, lighting conditions and times of the day. Both follow the 19 classes definition provided by Cityscapes. We use Cityscapes' definition to group the these classes into categories. Cityscapes has images of the size 1024 x 2048, which we resize to 512 x 1024. Image are resized by bilinear sampling, and the labels are resized by nearest neighbour sampling. Images in BDD are of the size 720 x 1280, which are used without any resizing. Resizing images in Cityscapes leads to loss of information. We believe that we can achieve better results in the case of Cityscapes if images of original resolution are used.

3.4 Training details

All our experiments are performed on a single NVIDIA GeForce GPU with 8 GB memory. We use a batch size of 1. We use the stochastic gradient descent optimizer with Nesterov acceleration with a momentum of 0.9 and weight decay of 1e-4. We use an initial learning rate of 0.001, which is decreased using the polynomial decay of 0.9. Cross entropy loss between the predicted soft segmentation maps and the ground truth is optimised. All our networks are randomly initialized and trained from scratch.

4 Experimental analysis

We quantify our results in table 2. With-GCN represents our proposed architecture consisting of the feature pyramid module, dilated residual network module, decoder with deconvolution layers and boundary refinement module as described in figure 1. FPN-DRN-deconv represents our model without the boundary refinement module. FPN-DRN-Bi represents the model obtained by replacing the final decoder block with bilinear interpolation. Here too, we don't use the boundary refinement module. We additionally use fewer layers at each level in the feature pyramid module. FPN represents just the feature pyramid module and DRN represents just the dilated residual network module.

Our proposed architecture (Row: With-GCN in the results table 2) performs the best on both Berkeley Deep Drive and on Cityscapes. This is followed by FPN-GCN, i.e. our proposed architecture minus the boundary refinement module, in the case of Berkeley Deep Drive. We observe that using a decoder with learnable deconvolution layers gives better results than bilinear upsampling for both datasets. Further, in the case of Berkeley Deep Drive, the FPN-DRN model performs better than both the individual modules, the feature pyramid module and the dilated residual network module. This proves that combining the power of multi-scale feature maps and high resolution outputs helps.

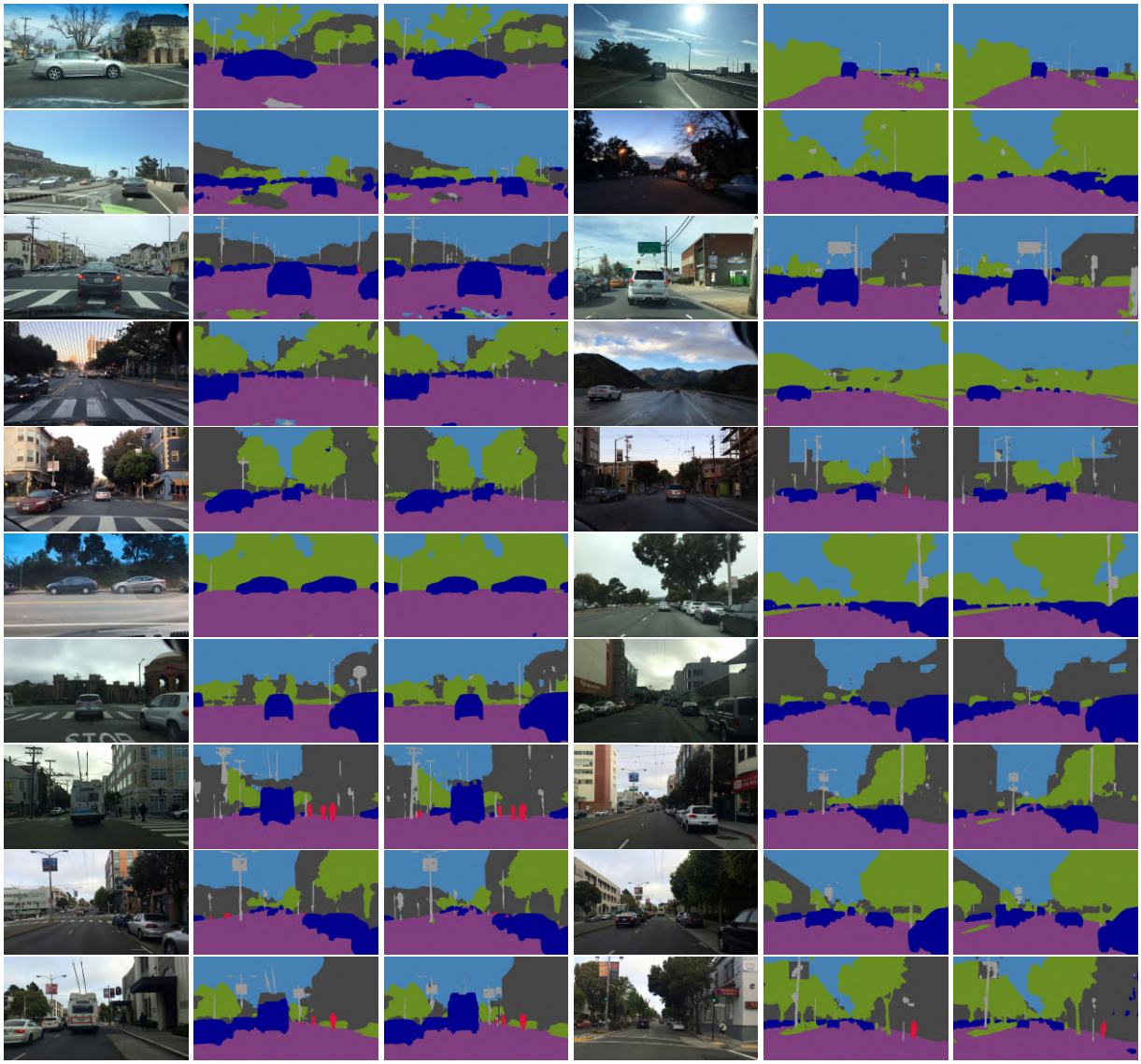

Figure 4: Berkeley Deep Drive Results: The first segmentation image depicts the results of 'FPN-DRN-deconv', the second corresponds to 'With GCN'

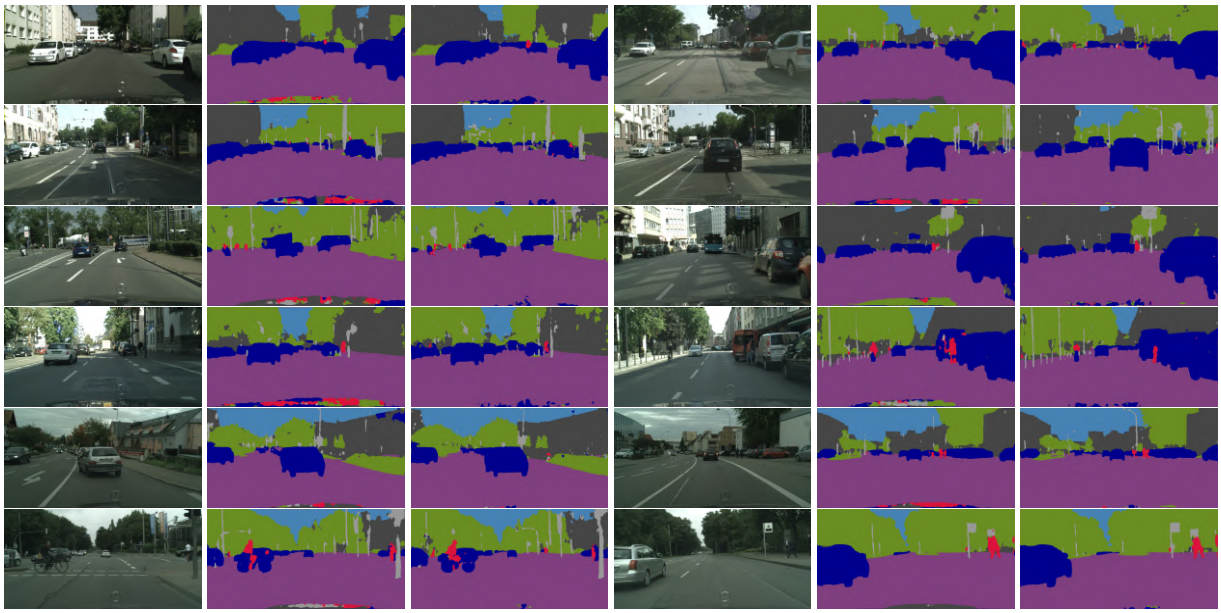

Figure 5: Cityscapes Results: The first segmentation image depicts the results of 'FPN-DRN-deconv', the second corresponds to 'With GCN'

In the case of Cityscapes, our model performs better than the dilated residual network module. We however notice an anomaly wherein the feature pyramid module outperforms our ensemble architecture. Cityscapes is a much smaller dataset. Further, there is loss of information as we resize the images. Additionally, this is a 7 class segmentation problem, which is much simpler than the 19 class problem. It is the 19 class problem which generally needs deep networks with large number of parameters may be needed to capture the complexities. We hypothesize that the much smaller feature pyramid module performs better due to better generalization.

5 Future work

Improving the performance of class "object" and class "human"

Through our experiments, we notice that our performance in the case of class 2 (objects) and class 5 (human) is particularly low. Two plausible reasons for this are as follows:

i) These classes have a particularly low representation (very few instances/ pixels) in the dataset.
ii) The boundaries of these classes are nuanced, making it hard for the light weight architecture to capture.

Methods like self-attention [30] and non-local neural networks [31] that are capable of capturing long range dependencies blow up the number of parameters. To resolve the uneven distribution of classes, we have tried experiments that incorporate class priors and use weighted cross entropy loss. This however led to a degradation in the performance of the model. We believe that the incorporation of category centroids or some other kind of object based class prior can help in further improvement.

6 Conclusion

In this paper, we leverage the power of feature pyramids and dilated residual blocks and show how combining the two can produce better results than the individual blocks. As a case study, we have considered the problem of category wise semantic segmentation. This is particularly useful in accident and brake control systems and even for autonomous driving to a large extent, where it is sufficient to know the type of object that each pixel in the scene belongs to. To customize the model to the problem at hand, we further incorporate a third module, a boundary refinement module to improve performance on categories that contain objects with nuanced boundaries. Our experiments on Berkeley Deep Drive confirms our hypothesis.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan 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, 2017.

[2] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[7] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[8] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2017.

[9] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016.

[10] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[11] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[14] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[15] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015.

[16] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017.

[17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[18] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2018.

[19] Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang. Fishnet: A versatile backbone for image, region, and pixel level prediction. In *Advances in Neural Information Processing Systems*, pages 754–764, 2018.

[20] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5229–5238, 2019.

[21] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.

[22] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6798–6807, 2019.

[23] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6819–6829, 2019.

[24] Cheng-Yang Fu, Tamara L Berg, and Alexander C Berg. Imp: Instance mask projection for high accuracy semantic segmentation of things. *arXiv preprint arXiv:1906.06597*, 2019.

[25] Jinming Su, Jia Li, Yu Zhang, Changqun Xia, and Yonghong Tian. Selectivity or invariance: Boundary-aware salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3799–3808, 2019.

[26] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[27] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 552–568, 2018.

[28] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[29] Tianyi Wu, Sheng Tang, Rui Zhang, and Yongdong Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *arXiv preprint arXiv:1811.08201*, 2018.

[30] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[31] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.