Demonstrating State-Of-The-Art Real-Time Semantic Segmentation

J. A. Lin (2416278) and Y. Klein (2981046)

Abstract. We propose to implement a recent CNN architecture for real-time semantic segmentation named DFANet [3] using PyTorch [4]. We then aim to visualize the output of the network using a real-time live stream demonstrator in the final presentation.

1 Motivation

We aim at reproducing the results of a convolutional neural network architecture for real-time semantic segmentation named DFANet [3]. The goal of semantic segmentation, a fundamental task in in computer vision, is to assign each pixel in an image a corresponding label (cf. Figure 1) [5, 3]. Popular applications include autonomous driving, video surveillance and robot sensing. For such tasks it is especially important to strike a good balance between computation time and accuracy, idealy maximizing both. On the one hand, the neural network has to be fast enough to do the pixel classification in real-time. On the other hand, the neural network should classify the image as good as possible. Consider autonomous driving for instance. When driving, the car must quickly react to changing environments. It would be devastating if the car recognizes a pedestrian too late because of a slow architecture. Then again, it would also be devastating if the car could not properly detect the pedestrian as such due to poor accuracy of the approach. Hence, it is important to perform semantic segmentation under resource constraints and process input data in real time.

We choose to implement DFANet because it promises to handle the trade-off between computational cost and accuracy better than other common approaches (cf. [3], Figure 1). More importantly, however, no working publicly available official or unofficial implementation currently exists. While there has been an attempt at implementing DFANet[1], it has not been successful. We therefore plan on making our implementation available to the public through GitHub[2].

2 Outline

We structure our proposed project into four major milestones and specify our reasons as well as our learning goals:

[1] https://github.com/huaifeng1993/DFANet
[2] https://github.com/j-a-lin/DFANet_PyTorch

Fig. 1. Semantic segmentation example in a cityscape environment (left) and ground truth (right). https://devblogs.nvidia.com/image-segmentation-using-digits-5/

1. Familiarize ourselves with DFANet [3], PyTorch [4] and dependent papers.
   - Get to know a relevant computer vision task.
   - Use popular, imperative machine learning framework.
   - Implement recent state-of-the-art algorithm.
   - Lightweight network architecture makes training feasible.
2. Implement backbone (Xception [1]) and pretrain on ImageNet [2].
   - Popular and proven image classification architecture.
   - Real-world data instead of toy problem.
3. Implement DFANet [3], transfer learning using pretrained backbone.
   - Hands-on experience with deep convolutional neural networks.
   - Utilize two-stage transfer learning.
4. Integrate into real-time live stream demonstrator.
   - Apply segmentation algorithm on image stream.
   - Interactive result visualization.

References

1. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1800–1807 (July 2017). https://doi.org/10.1109/CVPR.2017.195
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
3. Li, H., Xiong, P., Fan, H., Sun, J.: Dfanet: Deep feature aggregation for real-time semantic segmentation. CoRR **abs/1904.02216** (2019), http://arxiv.org/abs/1904.02216
4. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)
5. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)