Semantic Segmentation

Prakhar Agarwal, Cluster Innovation Centre
Dr. S.P. Mishra, DTRL, DRDO

Abstract—Convolutional networks due to their translation invariance, serve as powerful tools to detect key features from an image. This work shows the creation of a neural network using FCN + Deconvolution + Skip Connections. The network consisting of only convolution type layers can be both trained and evaluated using images of arbitrary size.

Index Terms—Semantic Segmentation, Deep Learning, Pixel-wise Segmentation

1 Introduction

The introduction of Convolutional networks has revolutionized the field of machine learning. They have shown applications not only in image-based tasks but also in other problems such as Natural Language Processing. They serve as the state-of-art for almost all image-based problems.

Fully Convolutional Networks, are an excellent way to extract prominent features from an image and are also fast. In order to maintain their speed, they use pooling layers at intervals to decrease image resolution. But, this also limits the networks' ability to resize the image back, as a great deal of information gets lost in the process. Hence, the output from an FCN needs to be upscaled using techniques such as Bilinear Upscaling. In this work, we use an advanced way of upscaling called Deconvolution. Deconvolutions are reverse convolutions and increase the size of the image. Like convolutions, they can be trained too.

This work uses FCNs and Deconvolution networks to decompose image scenes into separate meaningful identities. This task is also known as Semantic Segmentation.

1.1 Semantic Segmentation

Semantic refers to "meaning in language or logic", and when we talk about semantically segmenting an image we mean to divide the image into meaningful parts, such as physical objects.

Fig. 1: Semantic Segmentation

For example, Fig. 1 has been segmented into 4 different categories (Cat, Table, Chair, and Background). As it might be evident, this task is not achievable using unsupervised methods. We need to define the desired segmentation categories, as an image with infinite detail could be segmented into an infinite number of categories.

2 Network Architectures

2.1 Fully Convolutional Networks

Fully Convolutional Networks (FCNs) (Fig. 2) are what one gets on building a neural network using only convolutional-type layers. They keep the spatial information of an image intact.

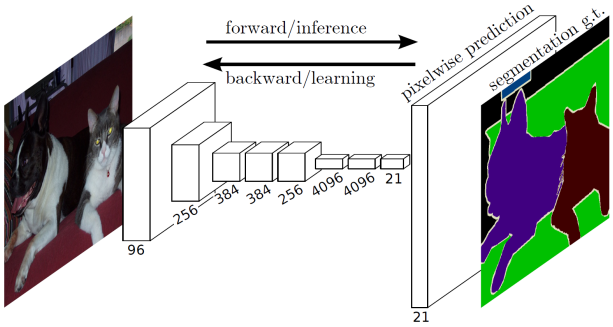

Fig. 2: Representation of an FCN (Source: J. Long et al, 2015 [1])

In the training process, they learn to highlight the key features describing a class in the image. With increasing network depth, receptive fields of the convolution layers also increase. Starting layers learn to recognize simple features such as a straight line, and curves, while the later layers might be learning geometries as complex as a face.

2.2 Deconvolution Networks

Pooling layers decrease the dimensions of an image. To scale it back up to its original size, instead of using Bilinear techniques, we have used Deconvolution layers. Deconvolution layers act as reverse convolution layers, and serve the purose of scaling up an image. Like there counterparts, they too are learnable.

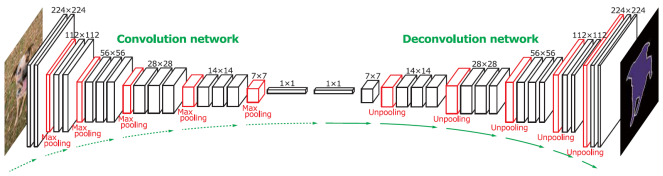

Fig. 3: Representation of a Deconvolution Network (Source: H Noh et al, 2015 [2])

3 Data Used

3.1 Cityscapes

The Cityscapes dataset [3] offers pixel-wise segmented images from stereo videos recorded in street scenes from 50 different

cities. It contains 5,000 fine quality and a larger set of 20,000 weakly annotated images. The images have a resolution of 2048 x 1024 pixels. For our purpose, we use the set of finely annotated images from the left camera.

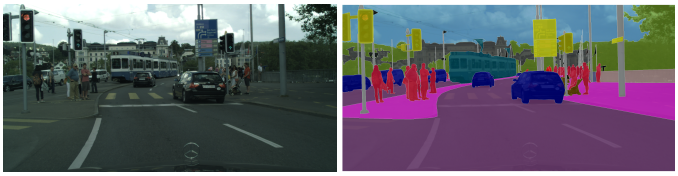

Fig. 4: Cityscapes Dataset

3.2 Stanford Scene Understanding Dataset

The Stanford Scene Understanding dataset [4], [5], [6] contains a total of 534 images (400 training and 134 test), with labels comprising of 9 classes (unknown, sky, tree/bush, road/path, grass, water, building, mountain, foreground object). Resolution of the images is 240 x 320 pixels. This served as a testing dataset to check for convergence of our models.

4 Segmentation Architecture

We use a hybrid architecture of a Fully Convolutional Network, a Deconvolution Network, and skip connections.

The training images and the corresponding segmentation labels are downsampled from 2048 x 1024 pixels to 512 x 256 pixels. This step was necessary to keep the size of the compiled model in control.

4.1 Skip Connections

With increasing network depths, vanishing gradients cause the accuracy to decrease rapidly. Deeper networks also have slower training, as gradient changes become minute.

Skip connections solve these problems by joining the output of an earlier layer to a later one, skipping all those in between. These shortcut connections help in keeping the effect of gradients intact during backpropagation.

When using pooling layers, information gets lost at each step. This limits the network's ability to resize the image back to its original size. Skip connections help pass the high-resolution information to later layers.

4.2 Architecture

The complete architecture can be found in the Appendix.

The network consists of 5 Convolution + Pooling layers, in the beginning, starting with a filter size of 128 and going up to 512. Convolution layers act as feature extractors. They detect key semantic features for each class. This process downscales the image too.

After training, the output of the last convolution layer represents a map of the detect features. This FCN is succeeded by a Deconvolution network, which upscales the image back to its original size.

The entire network consists of 3 skip connections, from the MaxPooling layers to Deconvolution layers of corresponding dimensions. These connections greatly increase accuracy.

5 Results

The trained model achieved a Mean IoU (Intersection over Union) accuracy of 78.6%, over the test data.

It was found that despite reducing the training image resolution by a factor of 16, the trained model was able to make good predictions even on larger images.

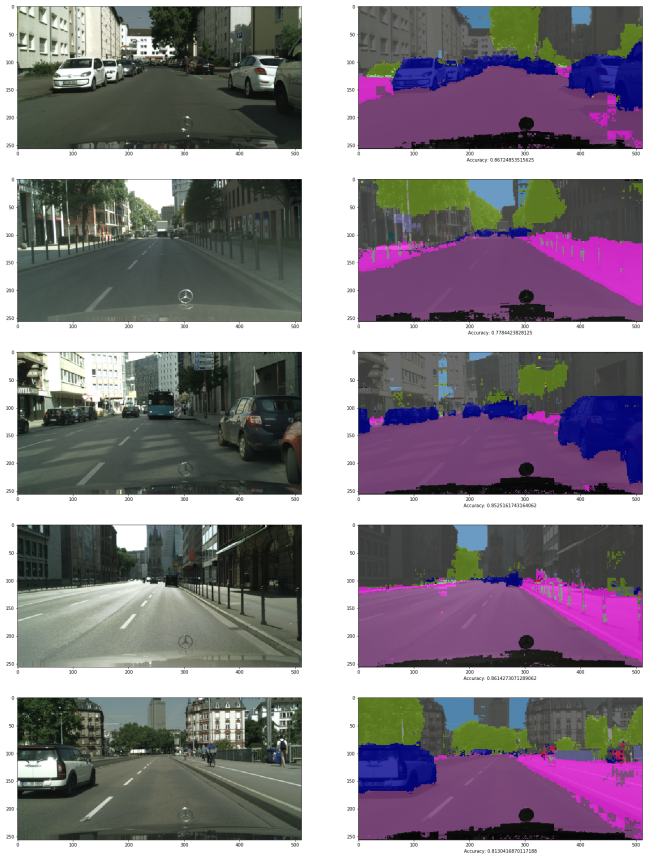

Fig. 5: Cityscapes Dataset - Results (FCN + Deconvolution + Skip Connections)

6 Conclusion

The network in this work can be further improved with the use of Conditional Random Field layers, which we plan to present in a future work.

The model was also tested with road scene images other than the dataset, and the results showed that the network was able to detect the classes properly, though the segmentation maps were not as good.

A shortcoming of the model was that it was not very good at predicting humans, and the segmentation map was jagged in most of the cases.

APPENDIX A
MODEL ARCHITECTURE VISUALISATION - FCN + DECONVOLUTION + SKIP CONNECTIONS - CITYSCAPES DATASET

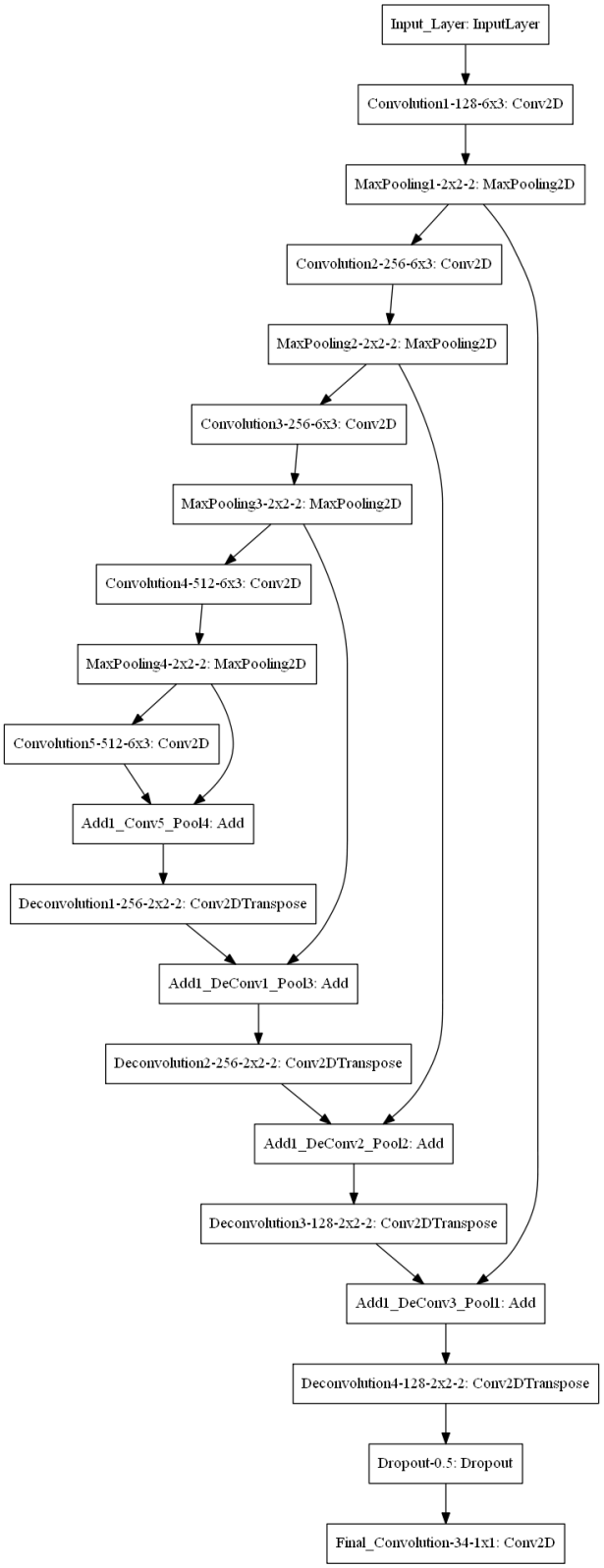

Fig. 6: Model Architecture - Cityscapes Dataset

ACKNOWLEDGMENTS

We would like to thank you Defence Terrain Research Laboratory, DRDO, Delhi for providing us with the opportunity to work on the project.

REFERENCES

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

[2] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[3] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1253–1260.

[5] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in neural information processing systems*, 2006, pp. 1161–1168.

[6] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2009.