A Comparative Study of Architectures for 2D Image Semantic Segmentation

Shasvat Desai
University of Massachusetts, Amherst
shasvatmukes@umass.edu

Debasmita Ghose
University of Massachusetts, Amherst
dghose@umass.edu

Abstract

Semantic Segmentation involves understanding the image on a pixel-by-pixel level i.e. to assign a class label to every pixel in the image. We experiment with different architectures to perform segmantic segmentation of images on the PASCAL VOC 2012 [3] dataset.

We implement the Fully Convolutional Networks (FCN) by Long et al.[7] as our baseline method for performing semantic segmentation. We perform various experiments with the number and position of skip connections and adding different layers to aggregate more context information.

We then implement an Improved Fully Convolutional Network (IFCN) architecture as suggested in the work of Shuai et al. [8] which introduces a context network that progressively expands the receptive fields of feature maps. In addition, dense skip connections are added so that the context network can be effectively optimized and fuses rich-scale context to make reliable predictions, which has proven to show significant improvements in segmentation on the PASCAL VOC 2012 [3] dataset.

We also modify the U-Net architecture for multi-class semantic segmentation with pre-trained weights from the VGG-16 architecture trained on the ImageNet dataset.

1. Introduction

An inherent issue in Semantic segmentation faces is the tension between semantics and location: global information resolves ”what” while local information resolves ”where”. Deep feature hierarchies encode location and semantics in a nonlinear local-to-global pyramid. Therefore skip connections between lower and higher layers are used to take advantage of this feature spectrum that combines deep, coarse, semantic information and shallow, fine, appearance information.

Apart from implementing the state of the art methods for image segmentation, our contributions have been:

- Implemented the Improved Fully Convolutional Network for semantic segmentation[8] on the PASCALVOC 2012 dataset.
- Modified the U-Net architecture[6] according to the VGG-16 architecture [9] pre-trained on weights from the ImageNet [2] dataset.

1.1. Dataset

PASCAL VOC 2012 dataset [3] has been used for performing segmentation. It contains 6,929 images with pixel-wise labels for 21 classes of objects (20 object classes + 1 background class).

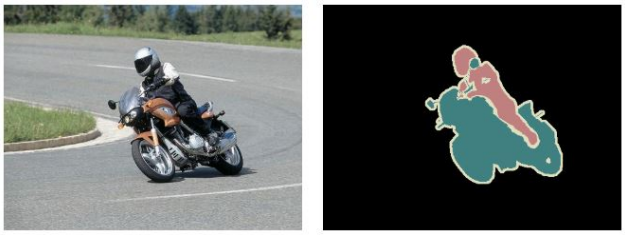

Figure 1. PASCAL VOC dataset sample

2. Related Work

Before the widespread application of deep learning for computer vision tasks, approaches like TextonForest and Random Forest based classifiers were used for semantic segmentation. After this, Convolutional Neural Networks (CNN) was found to be enormously successful for semantic segmentation tasks.

Fully Convolutional Networks (FCN) by Long et al.[7], popularized CNN architectures for dense predictions without any fully connected layers. This allowed segmentation maps to be generated for image of any size and was also much faster compared to the patch classification approach. Apart from fully connected layers, one of the main problems with using CNNs for segmentation is pooling layers. Pooling layers increase the field of view and are able to aggregate the context without localization of pixels. However, semantic segmentation requires the exact alignment of class maps and thus, needs the localization information of the pixels to be preserved. The encoder-decoder architecture is used to tackle the problem of localization.U-Net is a popular architecture from this class.

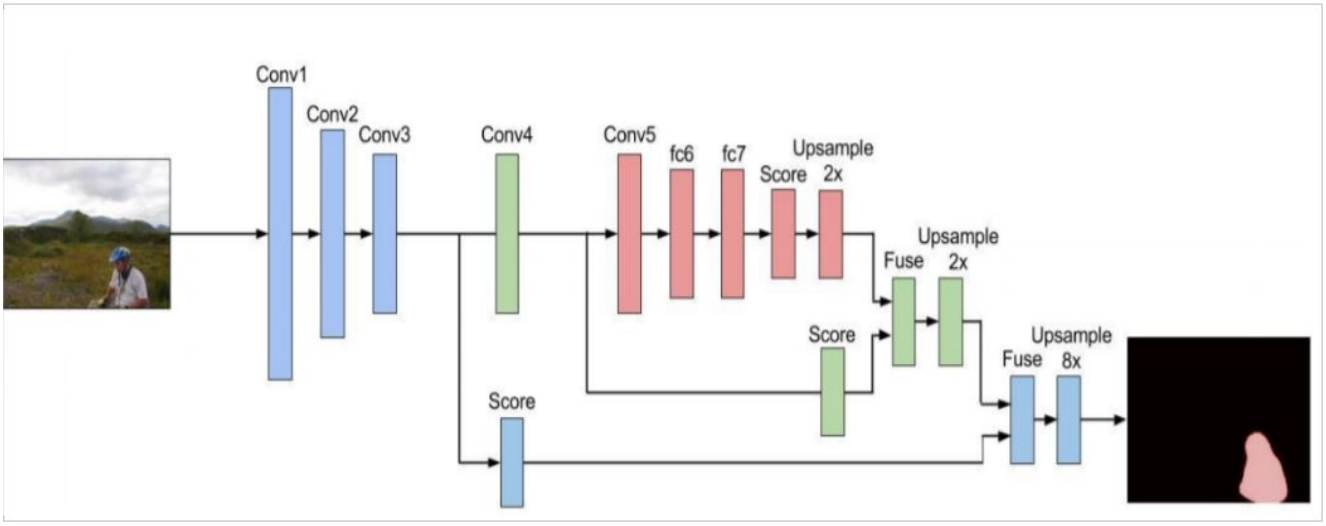

Figure 2. Fully Convolutional Network for Semantic Segmentation

3. Fully Convolutional Networks for Semantic Segmentation

3.1. Network Architecture

The architecture of Fully Convolutional Network [7] is shown in Figure 2. They are trained with pixel wise loss for performing dense predictions. The network was trained for segmentation by fine-tuning with pre-trained weights of VGG-16 [9] architecture trained on the ImageNet dataset [2]. Skip connections were added between layers to fuse coarse, semantic and local, appearance information. This skip architecture is learned end-to-end to refine the semantics and spatial precision of the output.

3.1.1 FCN-32s

Each net was decapitated by discarding the final classifier layer, and all fully connected layers converted to convolutions. 1 X 1 convolution was appended with channel dimension 21 to predict scores for each of the PASCAL classes (including background) at each of the coarse output locations, followed by a deconvolution layer to bilinearly upsample the coarse outputs to pixel-dense outputs. We experimented with other upsampling techniques like nearest neighbor interpolation and Transposed convolution but did not get good results.
The final prediction layer has a 32 pixel stride, so the network is called FCN-32s.The network has limited scale of detail in the upsampled output.

3.1.2 FCN-16s

The problem of localization was addressed by adding skip connections that combine the final prediction layer with lower layers with finer strides. Combining fine layers and coarse layers lets the model make local predictions that respect global structure.
The output stride is divided in half by predicting from a 16 pixel stride layer. A 1 X 1 convolution layer is added on top of pool4 to produce additional class predictions. FCN-16s is learned end-to-end, which is initialized with the weights of FCN-32s.

3.1.3 FCN-8s

As with FCN - 16s, predictions are fused from pool3 with a 2X upsampling of predictions fused from pool4 and conv7, building the net FCN-8s.

3.2. Limitations of Fully Convolutional Networks

It is really difficult to train a FCN from scratch, so it adapts the architecture of a pre-trained CNN VGG-16 [9] on the ImageNet [2] dataset. The pre-trained CNN is trained with low-resolution images (e.g. 224 X 224 pixels), whereas input segmentation images are usually in high resolution (e.g. 512 X 512 pixels). The simple adaptation techniques adopted in FCN cannot effectively address this domain gap, which leads to less optimized segmentation performance of FCN. This is because the feature maps that are used for classification in FCN have limited contextual fields.

4. Improved Fully Convolutional Neural Network for Semantic Segmentation

4.1. Network Architecture

The key modification to FCN is that IFCN adds a context network between the pre-trained CNN and upsampling

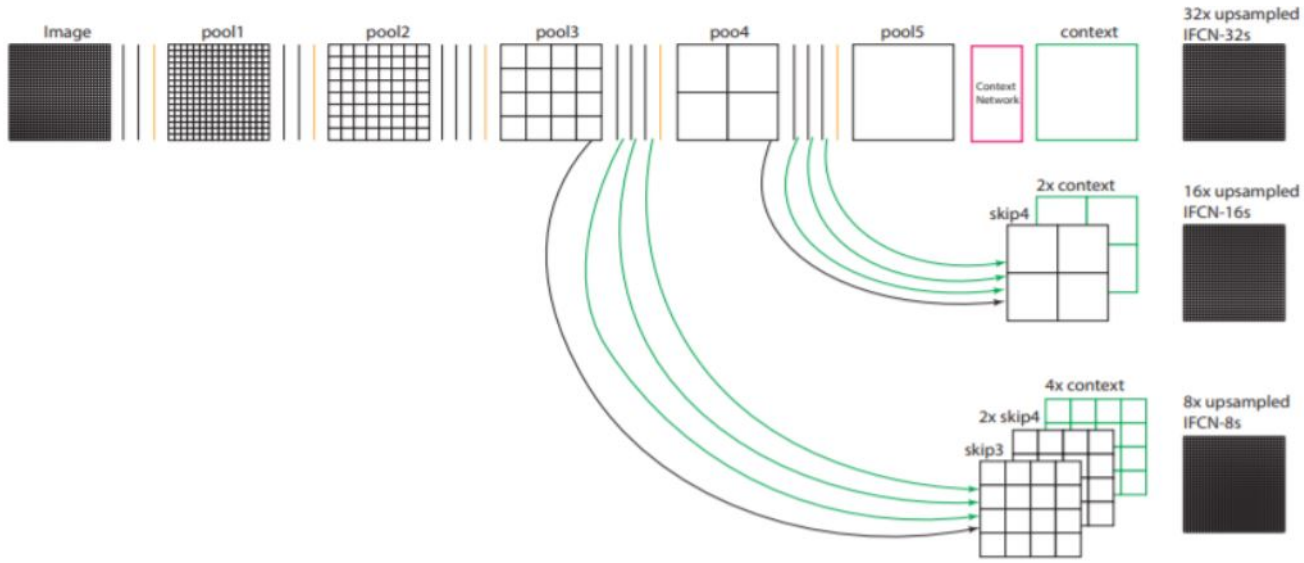

Figure 3. Improved Fully Convolutional Network for Semantic Segmentation

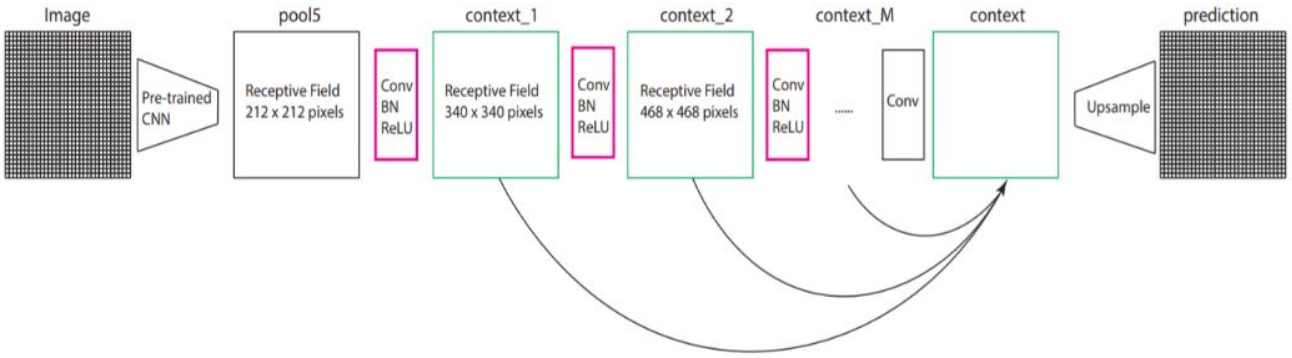

Figure 4. Context Network

layers. Thus, IFCN entails new parameters, in order to fill the domain gap during the network fine-tuning on high-resolution segmentation images. The context network is a densely branched convolution network. Specifically, it is stacked with multiple convolution blocks, and shortcut branches are further added from each intermediate feature map. Therefore, the context network is able to significantly expand the receptive fields of feature maps, which is essential to contextualize local semantic predictions. In addition, shortcut branches are also important to ease the optimization difficulty of context network, as they provide shortcut paths for the propagation of error signals and those shortcut branches enable IFCN to make predictions based on rich scale contexts.

As a consequence of this, IFCN is able to converge to a significantly better local optima than FCN on standard semantic segmentation images. Besides, IFCN discards the last two fully connected layers (fc6, fc7) which are specific to image classification in order to make the feature maps more compact and reduce the network size.

4.2. Context Network

The context network consists of the basic conv block (Conv + BN + ReLU). Using this architecture seems to be a good architecture because Conv can aggregate neighborhood Unfortunately, the resulting segmentation network doesnt work properly. There could be two possible reasons that lead to such undesirable behavior:

1. the network training is hard due to gradient vanishing problem when the context network is deep;

2. the final feature map generated by the context network is expected to have wide (global) range of contextual views, which makes the features less discriminative for local predictions

To address these issues, shortcut branches are introduced emanating from intermediate feature maps. The architecture of context network is shown in Figure 4. Specifically, those shortcut branches are responsible for locally predicting feature maps. Then these predictions are summed to

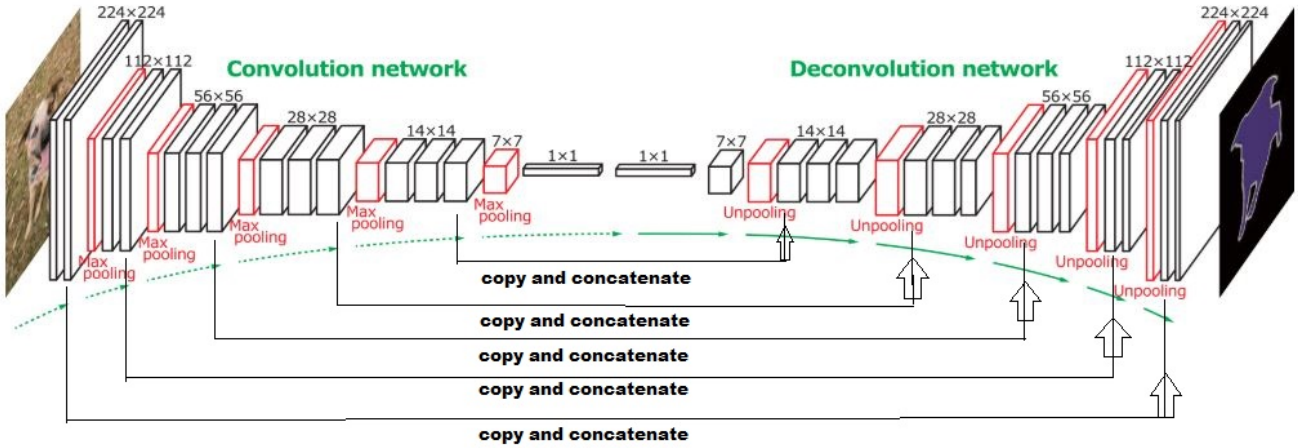

Figure 5. VGG-16 U-Net Architecture

generate the fused prediction map. They can be mathematically represented as

$$F = \sum_{i=1:M} S(\theta_i, f_i)$$

where, F is the fused prediction map, M is the number of conv blocks (i.e. depth of the context network), where S(',') is the shortcut function, θ_i represents the parameters of the shortcut function, f_i is the output feature map of i-th conv block. From the equation, we can see that f_i is directly connected to the error signals (exclude upsample layers), thus its preceding conv block receives strong supervision signals. From this perspective, the gradient vanishing problem is greatly mitigated, so the context network can be effectively trained. Moreover, the fused prediction map F combines multiple decisions from f_i . Knowing that each f_i preserves different scale context, their fused prediction map F is expected to be more robust.

5. U-Net for multi-class Semantic Segmentation

5.1. Network Architecture

U-Net consists of a contracting path the capture context information and a symmetric expanding path to enable precise localization. It was originally used on binary image segmentation of heavily augmented image datasets for biomedical applications. It is typically trained from scratch with randomly initialized weights.

5.2. U-Net with Pretrained weights

We demonstrate the application of U-Net to the Pascal VOC dataset to segment the images into 21 classes using a network based on VGG-16 pre-trained on the Image Net dataset since pre-trained networks substantially reduce training time and also help to prevent over-fitting.
VGG16 contains thirteen convolutional layers, each followed by a ReLU activation function, and five max polling operations, each reducing feature map by 2. All convolutional layers have 3X3 kernels. The first convolutional layer produces 64 channels and then, as the network deepens, the number of channels doubles after each max pooling operation until it reaches 512. On the following layers, the number of channels does not change. To construct an encoder, we remove the fully connected layers and replace them with a single convolutional layer of 512 channels that serves as a bottleneck central part of the network, separating encoder from the decoder. To construct the decoder we use transposed convolutions with learnable filters layers that doubles the size of a feature map while reducing the number of channels by half. And the output of a transposed convolution is then concatenated with an output of the corresponding part of the decoder. The resultant feature map is treated by convolution operation to keep the number of channels the same as in a symmetric encoder term. This upsampling procedure is repeated 5 times to pair up with 5 max poolings.
As per our knowledge, U-Net has never been pre-trained on Vgg-16 weights and used for making predictions on PASCAL VOC 2012 [3] dataset.

6. Experiments

6.1. Fully Convolutional Networks

1. **Upsampling Network:** We experimented with various approaches for upsampling method Nearest Neighbor implementation: This method failed due to its naive methodology of considering only one nearest neighbor Transposed Convolutions: Here we tried to use learn-

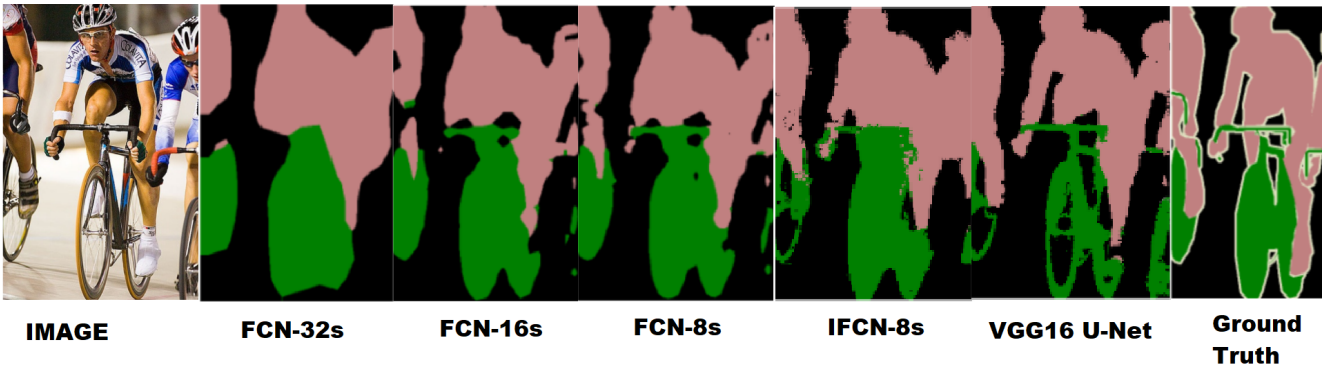

Figure 6. Results

able filters with Xavier initialization but failed to successfully obtain good prediction map.
Finally, we used Bilinear interpolation which seemed to perform well as shown in the result table

2. **Hyperparameters:** SGD optimizer with momentum of 0.9 and learning rate of 1e-4 on the pre-trained VGG-16 network architecture worked the best. We also tried to use adam optimizers but failed to achieve good performance

3. **FCN 4s:** Since FCN 8s performed well, we tried to add even finer strides from the pool 2 layer having a pixel stride of 4. But the results obtained degraded. We suppose this is due to the fact that the pool 2 layers has very little contextual information hampering it from making predictions with higher semantic information

6.2. Improved Fully Convolutional Networks

1. **Training using VGG-16 pretrained weights :** IFCN gave a good performance when training using vgg16 architecture as shown in the table

2. **Fine tuning on FCN-8s results :** We trained the IFCN network by using the pre-trained weights of the FCN-8s network, thus leveraging both-rich level semantic context of IFCN along with finer strides of FCN-8s network.This gave excellent results.

3. **Hyperparameters:** SGD optimizer with momentum of 0.9 and learning rate of 1e-7 on the pre-trained VGG-16 network architecture worked the best.
For the network trained on FCN-8s weights, we used SGD with momentum of 0.9 and a learning rate of 1e-4.

6.3. VGG-16 U-Net

1. **Training from scratch:** We tried to train the U-Net architecture from scratch but failed to do so. This may be because learning to predict 21 classes might be too difficult for it.

2. **VGG-16 pre-trained architecture:** We transferred the pre-trained weights on ImageNet to the U-Net architecture and achieved satisfactory results as shown in the table below.

3. **Hyperparameters:** Adam optimizer with a learning rate of 1e-3 was used. We failed to train this network with SGD optimizer.

7. Results

The methods were evaluated on the PASCAL VOC 2012 [3] dataset using the Keras/TensorFlow framework [1] and Mean Intersection over Union (IOU) and Pixel Accuracy were used as the evaluation metric. The table shows the Mean IOU and Pixel Accuracy for the validation set of the PASCAL VOC 2012[3] dataset and the results have been shown in Figure 6.

Method	Mean IOU	Pixel Accuracy
FCN-32s	56.8	85.4
FCN-16s	58.4	90.6
FCN-8s	60.6	91.2
VGG16 IFCN	62.8	93.5
IFCN-8s	63.6	93.7
VGG16 U-Net	**66.9**	**95.8**

8. Discussion

Beginning with FCN network, we moved onto implement of IFCN network which leverages the power of skip connections as shown by FCN to further obtain better predictions, The only drawback of IFCN is that it has longer training time due to addition of multiple skip connections along with context network.
Finally , our best performing model was U-Net pre-trained on the VGG-16 architecture with ImageNet dataset. The

motivation behind experimenting with the U-Net architecture was to show see if we leverage the power of U-net architecture on PASCAL-VOC dataset since it is one of the state-of -the art model for Biomedical segmentation. Initially we failed to train U-Net from scratch, but when training it using the VGG-16 weights, it was our best performing model.

9. Conclusion and Future Work

The models implemented by us have a huge scope of improvement, especially the U-Net architecture. In future, we would like to experiment by fine tuning U-Net on the trained weights of the FCN network.
Also, recently DenseNets have shown promising results for image segmentation. We attempted to experiment with this network, but due to resource and time constraints, we were not able to train it to achieve good results.
Another worthwhile experiment might encompass using CRF as a post-processing step.

References

[1] F. Chollet et al. Keras. `https://keras.io`, 2015.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[4] V. Iglovikov and A. Shvets. Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation. *CoRR*, abs/1801.05746, 2018.

[5] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, 2015.

[6] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[7] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017.

[8] B. Shuai, T. Liu, and G. Wang. Improving fully convolution network for semantic segmentation. *CoRR*, abs/1611.08986, 2016.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[7] [6] [3] [8] [4] [9] [2] [5] [1]