Semantic Segmentation Techniques – An Analogy between FCN and CRF-RNN

Vinita Boolchandani* Prof. Michael Ryoo

ABSTRACT

This paper presents an analogy of two important Semantic Segmentation techniques- using FCN and CRF as Recurrent Neural Networks. FCN uses Upsampling to produce segmented image maps of same size as input and with finer details using the skip architecture that fuses the local details from lower layers. While CRFs use FCN and added conditional random field based probabilistic inference models to output precisely segmented images. In spite of such strong architecture, the output produced by CRFs seem to perform differently when images from varied datasets are passed as inputs to it.

INTRODUCTION

Semantic Segmentation is an important problem in Computer Vision. It has been approached by a lot of researchers providing solutions of varying accuracies. The critical applications of Semantic Segmentation has made it quite important in Computer Vision field. Segmentation is being applied to images, videos and even 3D data. Some of the major applications where Semantic Segmentation is applied are autonomous driving [1] [2] [3], Human-Machine Interaction [4], Image search engines [6] and others. These problems are being solved using convolutional networks and deep networks these days. Whereas in the past they have been addressed using Machine Learning and traditional Computer Vision techniques.

In the field of computer vision semantic segmentation is essentially the classification of each pixel in an image. Each pixel is classified to a particular class label in order to understand the overall gist of an image or video. Real world images are often complex and contain a lot of information which our eyes and brain process in fractions of milliseconds. While it seems difficult for a computer to draw the same idea from an image as a human does, it has been smartly addressed by extracting critical features from an image. These critical features are those which neatly segment the objects in an image like for example, edges, corners, colors, spatial consistency. These critical features are extracted and fed to a model. This model learns the parameters from features of images in the training dataset and uses these parameters to predict the pixel classification of test dataset. The key problem here is extraction of features which decides the accuracy of final segmentation.

A couple of years back approaches using deep Convolutional Neural Networks (CNNs) were successful in performing high-level computer vision tasks such as object detection [13] and image recognition [31]. After this, there has been immense research in performing image segmentation using CNNs by applying them to pixel level labelling. Some of these researches like the ones using dense upsampling convolutions [7] or SegNet [9] have recently achieved huge success in precisely segmenting the pixels of input images.

In the journey of Semantic Segmentation, fully convolutional networks (FCN) [8] brought a significant breakthrough as it produced upsampled output that was same in size as input images and also retained the details of lower layers. This paper compares the network architecture and segmentation performance of FCN with Conditional Random Fields as recurrent Neural Network [5].

PREVIOUS WORK

Semantic Segmentation has been approached in a number of ways using different architectures. Most state-of-the-art semantic segmentation systems have three key components as stated in [7] :1) a fully convolutional network (FCN), where the last few fully connected layers are convolutional layers to make efficient end-to-end learning and the network can take arbitrary input size; 2) Conditional Random Fields (CRFs), to capture both local and long-range dependencies within an image to refine the prediction map; 3) dilated convolution which is used to increase the resolution of intermediate feature maps in order to generate more accurate predictions while maintaining the same computational cost.

Since the introduction of FCN in [8], improvements on fully-supervised semantic segmentation systems are generally focused on two perspectives - applying deeper FCN models and making CRFs more powerful. This includes applying fully connected pairwise CRFs [16] as a post-processing step [3], integrating CRFs into the network by approximating its mean-field inference steps [31, 20, 18] to enable end-to-end training, and incorporating additional information into CRFs such as edges [15] and object detections [1].

METHOD

Segmentation using fully convolutional network:

Fully Convolutional Networks for Semantic Segmentation[8] was published in 2016 and it brought a significant improvement in the results produced by the previous methods. It was successful as it could capture and propogate finer details of the image which was not done in previous methods. Architecture used by FCN can be seen in Figure1. An FCN can take an input of any size, and produces an output of corresponding (possibly resampled) spatial dimensions. It uses Upsampling and skip connections to improve the accuracy of its segmentation results.

The first layer is the image, with pixel size h × w, and d channels (Figure 1). In the higher layers each location corresponds to a receptive field in the image. The receptive field is made up of all the locations which are path connected to this location. These networks were called fully convolutional as they compute a nonlinear *filter* which we call a deep filter unlike general nets which compute a nonlinear function.

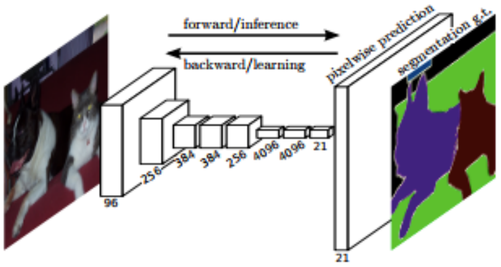


Fig. 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

The fully connected layers of these nets can also be viewed as convolutions with kernels that cover their entire input regions. Hence these networks become fully convolutional networks that take input of any size and produce a spatial output map.

Why Upsampling?

A way to connect this coarse output produced by these nets to dense pixels is interpolation. For instance, simple bilinear interpolation computes each output y_{ij} from the nearest four inputs by a linear map that depends only on the relative positions of the input and output cells.

$$y_{ij} = \sum_{\alpha,\beta=0}^{1} |1-\alpha-\{i/f\}|\,|1-\beta-\{i/j\}|\,x_{\lfloor i/f\rfloor+\alpha,\lfloor j/f\rfloor+\beta},$$

where f is the upsampling factor, and $\{\cdot\}$ denotes the fractional part. They have considered upsampling with factor f as a convolution with a fractional input stride of 1/f. Hence if f is integral, upsampling will be implemented through “backward convolution” by reversing the forward and backward passes of more typical input-strided convolution. Thus upsampling is performed in-network for end-to-end learning by backpropagation from the pixelwise loss. It is sometimes referred to as deconvolution layers.

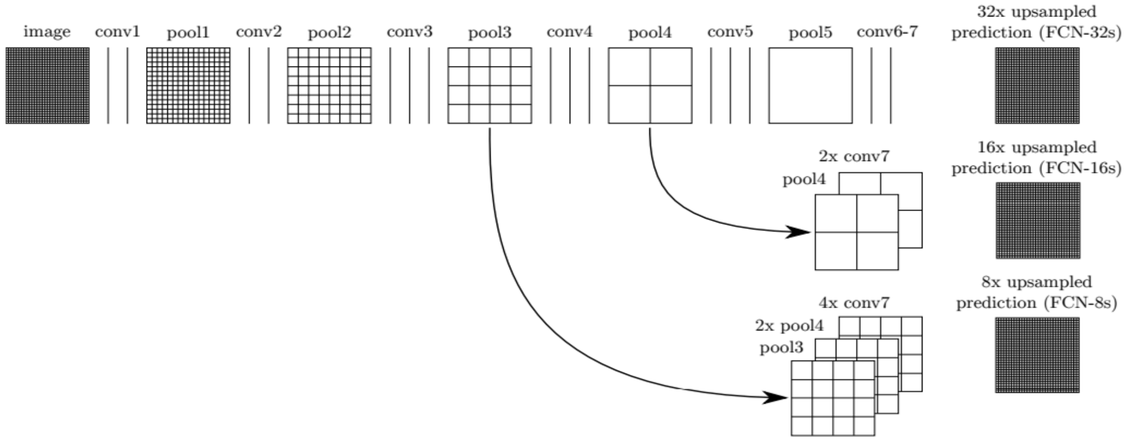


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Figure2. Skip Architecture

These networks can be improved to make direct use of shallower, more local features.

Why Skip Connections?

These networks scored highly on standard metrics, however their output is coarse. The intuition behind this is that the stride limits the scale of details included in the output. This was addressed by using skip connections (Figure 2), that fused the layer outputs. The aim behind this was to include shallower layers with more local predictions. Comparatively accurate predictions can be made from shallower layers as they have small receptive fields and see fewer pixels. This network was trained on scene parsing, exploring PASCAL VOC, NYUDv2, SIFT Flow, and PASCAL-Context and performed better than previous best SDS [12] as shown in the results (Figure 3).

Conditional Random Fields as Recurrent Neural Network

This is the ICCV 2015 paper, *Conditional Random Fields as Recurrent Neural Networks [5].* In this architecture an underlying FCN was used and on the top of that a CRF was used as a recurrent neural network. One central issue in segmentation techniques using deep networks is the network is unable to delineate all visual objects. Hence it is given that extra strength by combining a Convolutional Neural Network with Conditional Random Fields based probabilistic modelling. The pixels are modelled as random variables that form a Markov Random Field given a global observation (Image). It says that if X_i is assumed to be the random variable associated with label assigned to pixel i, let us assume it can take any value from predefined set of labels $L = \{l_1, l_2, \ldots, l_L\}$. Hence X is the vector formed by the random variables $X_1, X_2, \ldots, X_N$, where N is the number of pixels in the image. Given,

$$G = (V, E)$$

$$I = \text{the global image},$$

Where,

$$V = \{X_1, X_2, \ldots, X_N\} \text{ and G is a graph}$$

(I, X) can be modelled as a CRF with Gibbs distribution of the form,

$$P(X = x|I) = 1/Z(I) \exp(-E(x|I))$$

Here E(x) is called the energy of the configuration $x \in L^N$ and Z(I) is the partition function [14].

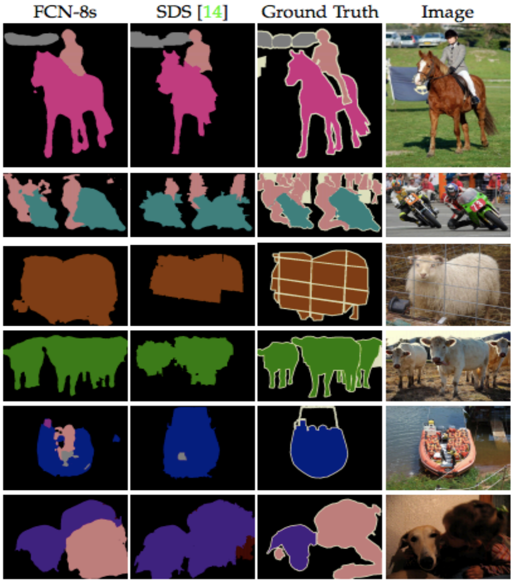


Figure 3: Performance of FCN

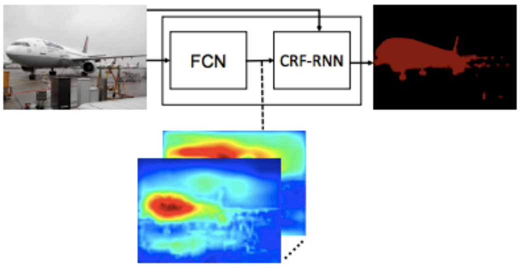


Figure 4: End-to-end trainable network for CRF

Comparing Segmentation results

These two methods were compared in terms of their segmentation results. The models were tested on MIT Scene Parsing benchmark (http://sceneparsing.csail.mit.edu/). This dataset comes from ADE20K Dataset which contains more than 20K scene-centric images exhaustively annotated with objects and object parts. For comparison the code for FCN was taken from the GitHub repository (http://sceneparsing.csail.mit.edu/). This code took approximately 20 hours in training and setting up the prerequisites. There was no post processing performed. Training was done for 9 epochs and the image size was taken as 256x256.

In case of CRF the code was taken from https://github.com/torrvision/crfasrnn.

In this case pre-trained model was used to test its performance on Google images in addition to MIT Scene Parsing. It was found that this end-to-end trained CRF model did better than FCN in precisely marking the boundaries including corners of the objects.

The results (Figure 4a and 4b) on this dataset were as expected and in Figure6. CRF as RNN with underlying FCN produced better and precise segmentation results as compared to FCN alone. This was because CRF considers the unary as well as pairwise energies while predicting a pixel label. It not only uses the strength of CRF but also integrates it with the underlying deep network hence forming an end-to-end trainable network (Figure 4) which has the strength of an FCN as well a CRF.

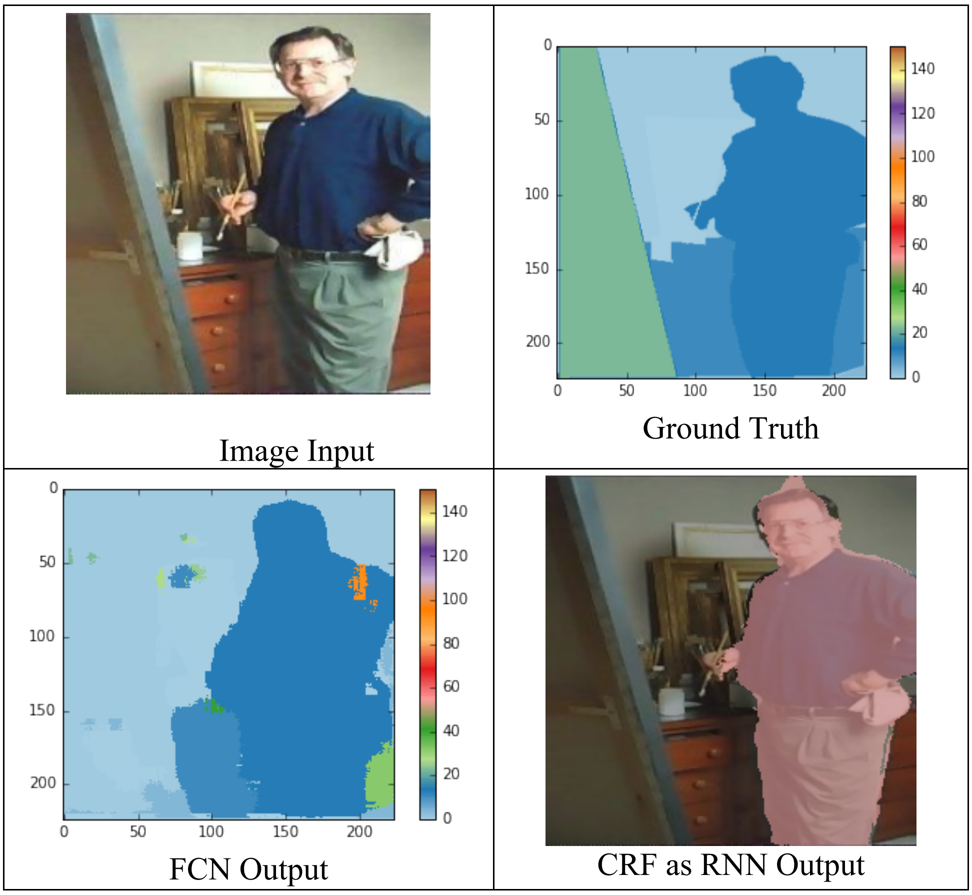
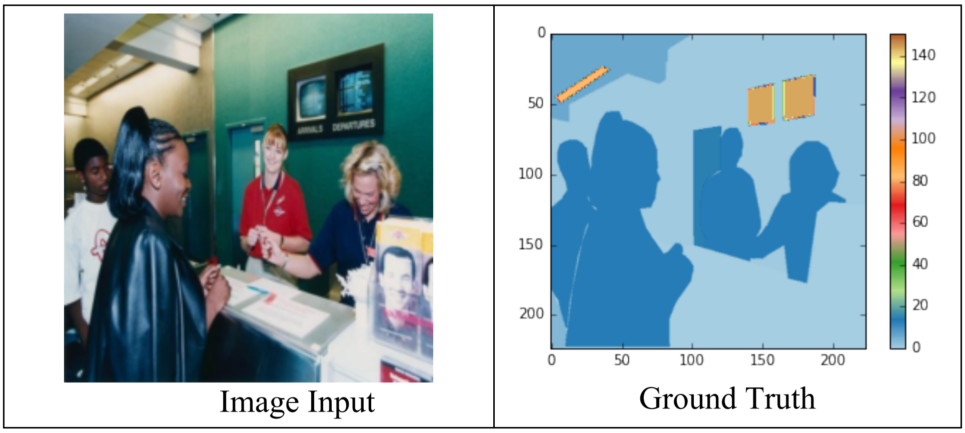


Figure4a:FCN and CRF on MIT scene parsing



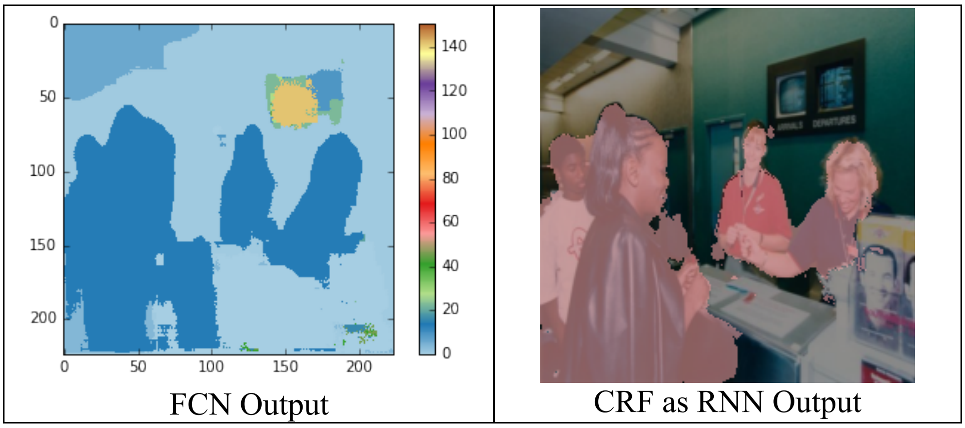


Figure 4b: FCN and CRF on MIT scene parsing

In addition to this, some failures were also observed and quite easily like in figure 4c. But major results were as expected with CRF performing better than FCN. Some results from CRF model were quite interesting. On testing the CRF as RNN on random Google images, there were some frequent misclassifications or classification errors. Some predicted classes were not actually present in the image (Figure 5a and 5b). While some other classes that were present in the image were not predicted (Figure 5c).

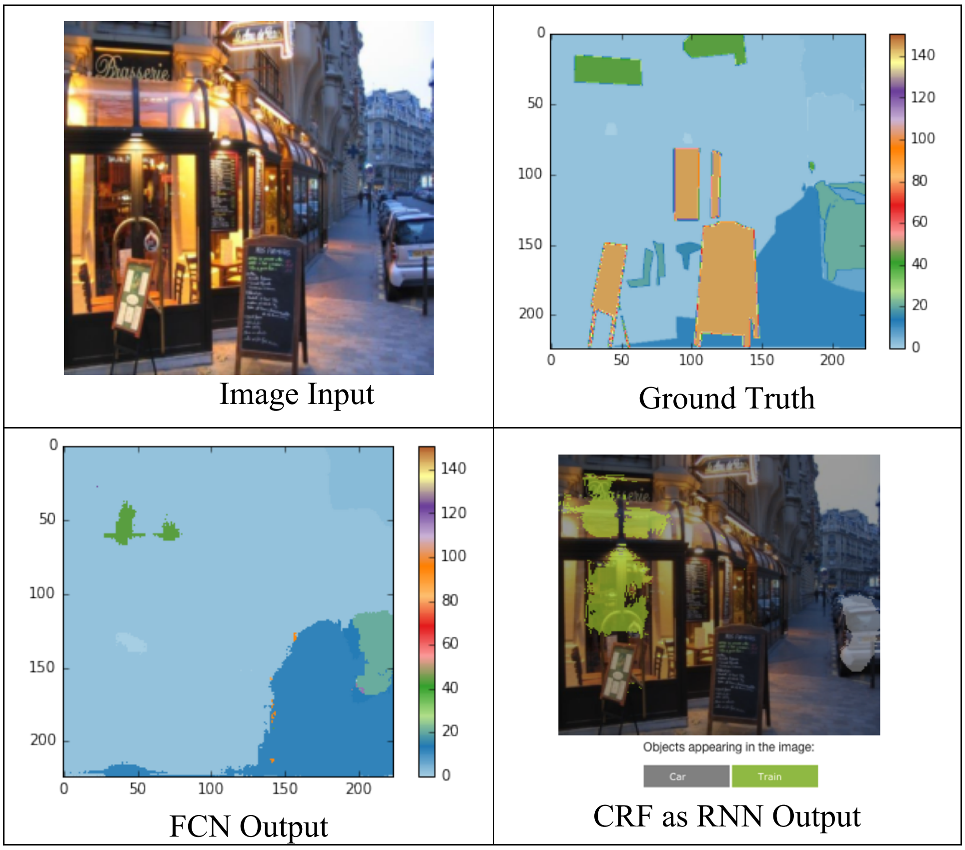


Figure 4c: FCN and CRF on MIT scene parsing

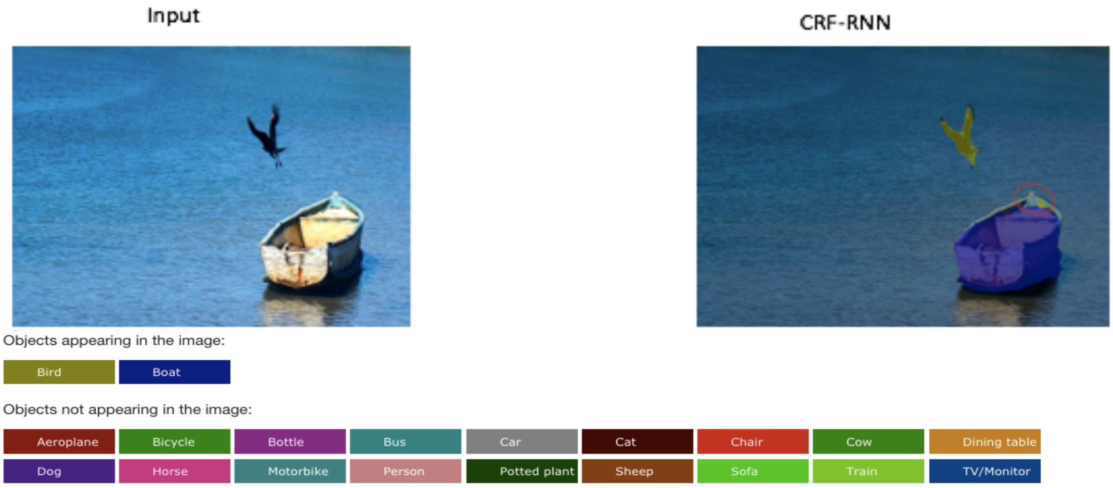


Figure 5a: A bird on boat was detected wrongly

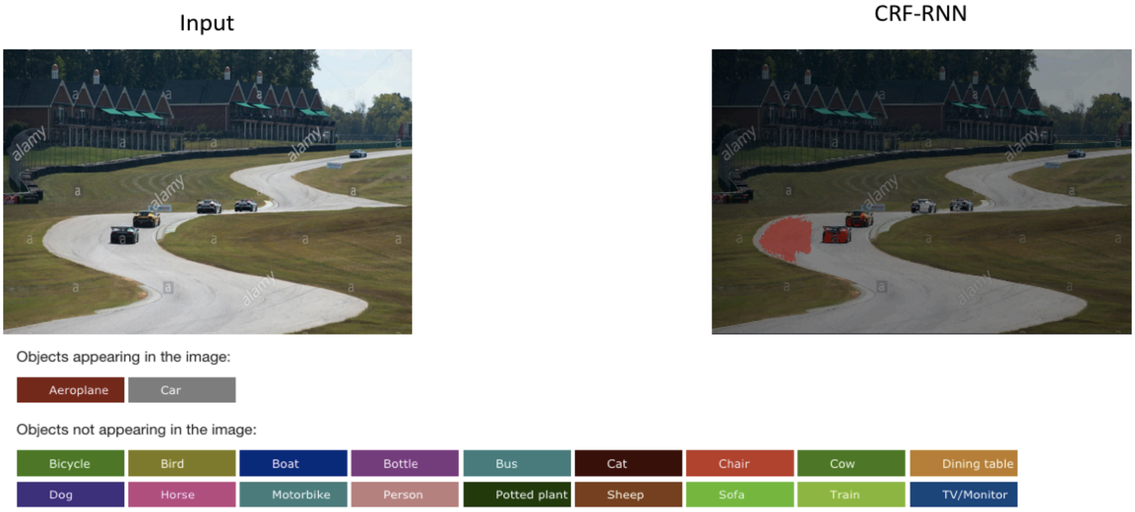


Figure 5b: The Aeroplane was detected wrongly

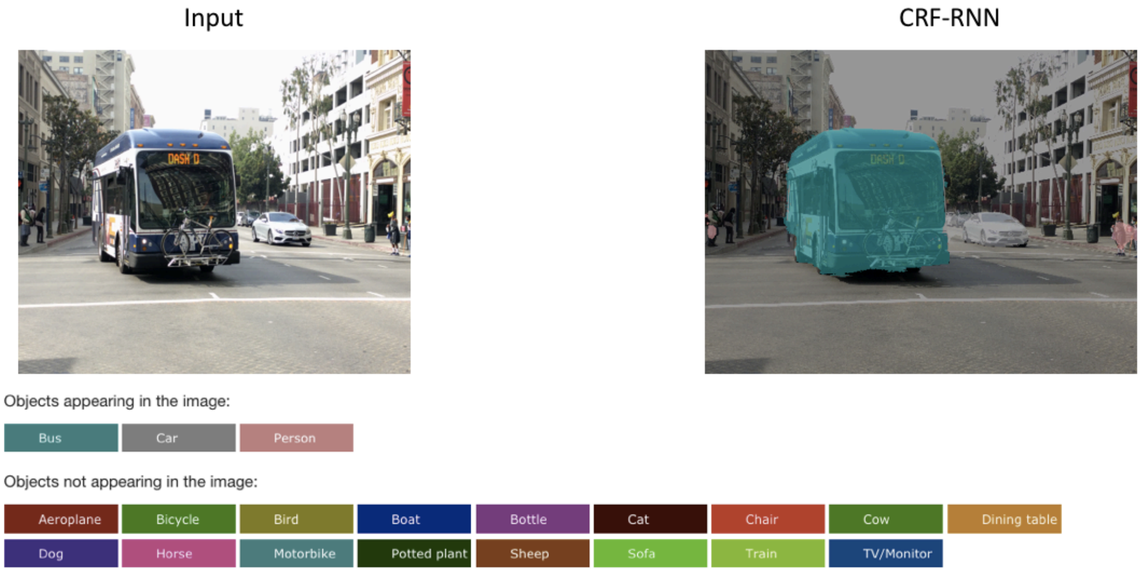
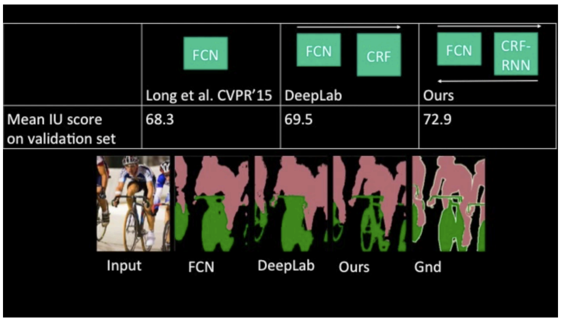


Figure 5c: The cycle and man on the left side were missed.



	FCN Long et al. CVPR'15	FCN CRF DeepLab	FCN CRF-RNN Ours
Mean IU score on validation set	68.3	69.5	72.9

Figure 6: The cycle and man on the left side were missed.

CONCLUSION:

From the two architectures, FCN and CRF as RNN, the later performs better segmentation in terms drawing precise boundaries. However, I feel it suffers from poor generalization to new images and doesn't seems to have parameters trained so as to perform segmentation of any image that is passed to it. This model is supposedly being used to design augmented reality glasses for the partially sighted . Hence, for such a critical application, the model should be able to segment generic images as well. As a part of the future work, this model will be tested on MS coco to further verify these shortcomings. Subsequently, the error in classification can be tracked at each step of the network to better understand the reason and propose a probable solution.

REFERENCES:

[1] A. Ess, T. M¨ uller, H. Grabner, and L. J. Van Gool, "Segmentationbased urban traffic scene understanding." in BMVC, vol. 1, 2009,p. 2.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012, pp. 3354–3361.

[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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[4] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deepin deep learning for hand pose estimation," arXiv preprint arXiv:1502.06807, 2015.

[5] "Conditional Random Fields as Recurrent Neural Networks" **arXiv:1502.03240** **[cs.CV]**

[6] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014, pp. 157–166.

[7] "Understanding Convolution for Semantic Segmentation" **arXiv:1702.08502** **[cs.CV]**

[8] 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.

[9] "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation" **arXiv:1511.00561** [cs.CV]

[10] http://warmspringwinds.github.io/tensorflow/tf-slim/2017/01/23/fully-convolutional-networks-(fcns)-for-image-segmentation/

[12] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Simultaneous ´ detection and segmentation," in ECCV, 2014. 1, 2, 4, 5, 7, 8, 9

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE CVPR, 2014

[14] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML, 2001