Semantic Segmentation using Fully Convolutional Networks with Skip Connections

1st Zaher, Eslam
Communications and Information Engineering
University of Science and Technology, Zewail City
Giza, Egypt
s-eslam.r.zaher@zewailcity.edu.eg

2nd Salah-Eldin, Wafaa
Communications and Information Engineering
University of Science and Technology, Zewail City
Giza, Egypt
s-wafaaashraf2015@zewailcity.edu.eg

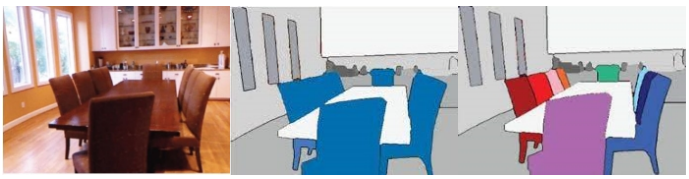
3rd Basheer, Riham
Communications and Information Engineering
University of Science and Technology, Zewail City
Giza, Egypt
s-rihamabdo@zewailcity.edu.eg

***Abstract*—Image segmentation is considered among the high-level tasks paving and enabling a complete understanding of the scene. The last decade has witnessed astonishing performance metrics achieved by CNNs in classification tasks in general and image segmentation in particular. FCNs build upon and utilize this performance by leveraging an arbitrary input, an edge that was not applicable in typical CNNs ending in dense layers. This work casts ILSVRC classifiers into FCNs and augments them for dense prediction through two novel skip-connection architectures. Not only have these architectures surpassed the performance of the previous solutions such as R-CNNs and ImageNet pioneering architectures, but they have achieved almost an identical perfomance of the original architectures presented by the authors of [1] in a significantly less computational time and resources. Indeed, our FCN-8s have reached a pixel accuracy of 89.9% and mean IOU of 57.8%. These results promotes the usage of FCN and suggests investigating simpler architectures and optimization techniques.**

***Index Terms*—segmentation, FCN, CNN, semantic, skip-connection**

I. Introduction

The incorporation of segmentation in computer vision systems has unlocked a wide range of commercially-valuable applications with simplified implementation. For instance, photography and videography have witnessed a recent addition of live view editings which include portrait modes, background changes, virtual try-ons, etc. In general, image segmentation is the task of classifying image pixels into regions or classes, which correspond to different objects or parts of objects. The segmentation task can be of two types as shown in [Fig. 1]. The first is Semantic Segmentation in which every pixel in an image is classified into one of a set of classes. The process involves classification which consists of making a prediction about the whole input image followed by localization, which provides additional information about the spatial location of those classes. Afterwards, semantic segmentation makes dense predictions by inferring categories (e.g., human, car, tree, sky) for each pixel in the image without identifying different occurrences/instances of the same class. The other type of segmentation is instance segmentation, in which every pixel is assigned to an object instance that belongs to it. In this work, we address the problem of semantic image segmentation by implementing the skip-connection-based Fully Convolutional Network (FCN) proposed by Long et al [1].



(a) original image (b) semantic segmentation (c) instance segmentation

Figure 1: Difference between semantic and instance segmentation : (a) original image; (b) semantic segmentation; (c) instance segmentation

II. Literature Review

It is common to adopt an encoder-decoder architecture for the purpose of semantic segmentation as unlike different classification tasks where the network's output is sufficient, semantic segmentation entails pixel-level discrimination and post-learning feature projection onto pixel-level. In this regard, ImageNet competition have produced a set of pioneering architectures that serve as a pre-trained encoder followed by a segmentation-specific decoder. As for the decoding process, it can be implemented with Weakly Supervised Semantic Segmentation or with Regions with CNN feature (R-CNN).

A. ImageNet Pioneering Architectures

In chronological order, AlexNet which won the competition in 2012 has achieved an accuracy of 84.6% with five convolutional layers and kernel filters ranging from 3x3 to 5x5 to 11x11, ReLU hidden activations, three Max-pooling layers, two fully-connected and two normalization layers and

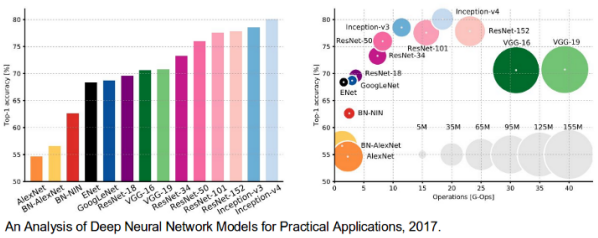


Figure 2: Summary of ImageNet pioneering architectures [4].

finally, a softmax layers. In the following year, VGG-16 has achieved an accuracy of 92.7% by replacing the large kernel filters/receptive fields with a stack of twelve 3x3 ones. A year later, GoogLeNet won with an accuracy of 93.3% using a local network topology (i.e., a network on top of another) with a total of twelve convolutional layers. The following significant contribution was in 2016 by ResNet, achieving an accuracy of 96.4% by first introducing the concept of skip-connections enabling an architecture of 152 layers! Ever since, the variation in the accuracy scores is insignificant.

B. Weakly Supervised Semantic Segmentation

Following the literal definition of semantic segmentation, this task can be tackled by a fully supervised manner. To elaborate, data can be inputted as images and their correlated pixels-level labels. Naturally, this task is computationally expensive. Accordingly for it to be more computationally efficient, it can constrained with image-level bounding boxes' annotations that are then mapped to pixel-level classifications. Hence, if left as is, this approach produces inaccurate boundaries and accordingly lower performance metrics. Nevertheless, this constraint can be addressed as a sub-problem following the weak supervised learning adaptation. For example, it was addressed as a denoising task in Simple Does It, achieving an accuracy of approximately 95%.

C. Regions with CNN feature (R-CNN)

To overcome the complexity in pixel-level computations, Girshick, et al., adopted a bottom-up approach where they employed a selective search algorithm to predict the most likely bounding boxes of objects. Afterwards, only these regions were inputted to AlexNet for computation of CNN foreground and full region features but replacing the softmax final layer with a SVM layer achieving a mAP of 53.3%. Indeed, R-CNN addresses object recognition before image segmentation. However, the obtained features are shown to not provide adequate spatial information needed for accurate boundary formation not to mention the non-negligible time complexity associated with the initial selective search when dedicated to semantic segmentation and not just object recognition.

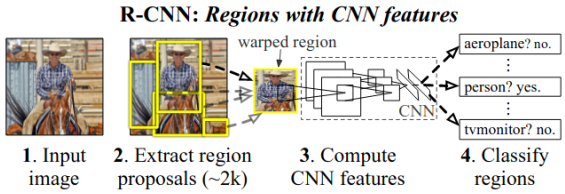


Figure 3: R-CNN framework [3].

III. Methodology

As per the procedure followed in the paper, we will be casting ILSVRC classifiers into FCNs and augment them for dense prediction. An upsampling in-network layer will, then, make predictions at the pixel level and learn these predictions by downsampling pooling (pixel-wise loss). We train for segmentation by fine-tuning. Next, we build a novel skip architecture that combines depth, rough semantic information and location. This architecture betters the precision by solving the inherit tension between semantics and location in semantic segmentation tasks. For this investigation, we train and validate on the Pascal VOC 2012 segmentation challenge [5]. We train with a per-pixel logistic loss and validate with the standard metric of mean pixel intersection over union (Mean IOU) and the mean is taken over all classes; including background.

A. From Classification to Semantic Segmentation

In classification problems, we use CNNs where the input goes through both convolutional layers and a fully connected (FC) layer. The output is, then, a predicted label for the entire image. In case of semantic segmentation, however, we need a label for every pixel in the image. The proposed architecture replaces the fully connected (FC) layers with a 1x1 convolutional layers as in Fig.4.

Then, if we do not down-sample the original image, we will not get a single label for the entire image. However, Alternatively, the output is only smaller than the image due to pooling as in Fig.5.

If we upsample the output, we get the pixel-wise output label map (heat map).

FCN has only locally connected layers such as convolutional layers and pooling layers making it capable of naturally operating on an input of any size, and produce an output of corresponding spatial dimensions. The output is, then, upsampled using deconvolution (backwards convolution).

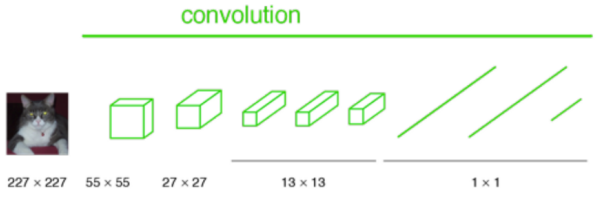


Figure 4: All Layers are convolutional layers [1].

Figure 5: All Layers are all convolutional layers [1].

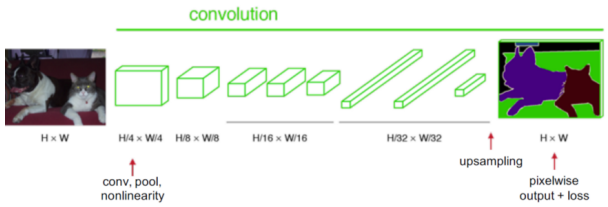


Figure 6: Upsampling [1].

B. Up-sampling Via Deconvolution

The process of upsampling could be performed in one of two ways:

- Resizing using interpolation techniques.
- Deconvolution, or better known as transposed convolution.

After multiple convolutions and pooling, the resulting image gets smaller and its resolution gets lower. We use deconvolution in order to get the output larger by upsampling it so, its pixels can be labeled.

C. Fusion: Combining What and Where

The output of the upsampling is dissatisfyingly coarse because even though deep features can be obtained, the spatial location information gets lost. The output of the 32x upsampling is a rough label map. And it is named **FCN-32s**,as shown in Fig.8.

However, the outputs from shallower layers have more location information. So, we use skip layer fusion to combine

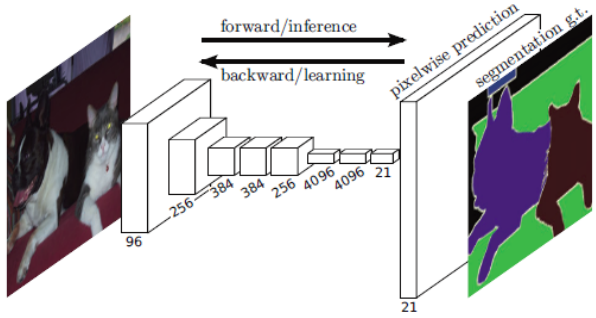


Figure 7: Feature Map [1].

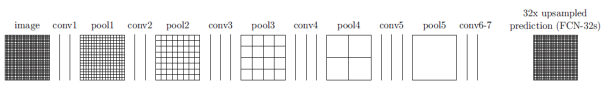


Figure 8: FCN-32s [1].

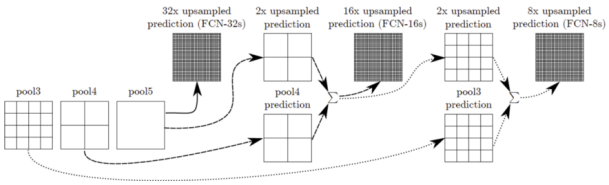


Figure 9: Combining of FCN-16s and FCN-8s [1].

the output with the outputs from shallower layer's output. This is guaranteed to enhance the results.

To do so, we fuse the output by element-wise addition of the deeper output upsampled x2 and the shallower layer. Then, upsampling x16 or x8, as presented shown in Fig.9.

In this work, we implement FCN-32s and FCN-8s with the help of pretrained VGG16 acting as our base encoder. We choose VGG16 from ILSVRC models for its convenience for layer fusion and commonness in benchmarking.

IV. Training

We trained our VGG16-based FCNs by fine tuning all the layers through the whole network. Adam optimizer with fixed learning rate of 10^{-4} was used with a minibatch size of 16 images. Multiple learning rates, momenta, and weigh decay values were tried but found to have no significant effect on our training, except for the learning rate. Over-fitting is likely to occur in FCNs, for this we added dropout layers with frequency of 0.2 and L2-norm kernel regularizers with a unified regularization factor of 10^{-6}. A customized Bilinear initializer was used for the upsampling layers.

A. Dataset

We augmented the Pascal VOC 2012 dataset [5] with the Berkeley Segmentation Boundaries Dataset (SBD) [6], which contains 11,355 labelled images (8,498 training, 2,857 validation). For training, the 676 unique images from the Pascal VOC dataset were augmented with both the training set and the last 1,657 images (out of 2,857 total) in the validation set of Berkeley SBD. The first 1,200 images of the SBD validation set for was for validating our models. Although our labels are unbalanced due to the dominance of background, FCN training can handle class unbalancing by weighting the loss, hence no class balancing was necessary.

V. Results

We evaluate both the fixed FCN-32s and FCN-8s skip architectures on the validation set using two common metrics for evaluating semantic segmentation and scene parsing, namely the pixel accuracy and mean intersection over union (mean IOU). Let c_{ij} be the numbers of pixels of class i misclassified to class j, where there is c_n total different classes, and let $n_i = \sum_i c_{ij}$ be the total number of pixels of class i. The computed metrics are formulated as follows:

- pixel accuracy: $\sum_i c_{ii} / \sum_i n_i$

- mean IOU: $(1/n_c)\sum_i c_{ii}/(n_i + \sum_j c_{ji} - c_{ii})$

Table I shows the performance of our FCN-32 and FCN-8s on the validation set of Pascal VOC 2012 compared to R-CNN[3], and SDS [7]. Our models outperform previous state-of-art architectures on mean IOU and inference time, with a significant relative margin.

Models	pixel accuracy	mean IOU	inference time
R-CNN [3]	-	47.9	-
SDS [7]	-	51.6	$\tilde{5}0$ s
FCN-32s	89.1%	55.5%	$\tilde{5}50$ ms
FCN-8s	89.9%	57.8%	$\tilde{8}65$ ms

Table I: Performance of our models on Pascal VOC 2012

Figure 10 shows the performance of our FCNs on samples of Pascal VOC 2012 dataset. The predicted segmentations show that FCN-8s produces finer segmentation maps than FCN-32s. This, as expected, is due to skip connections, allowing the model to capture more semantic, location, and boundary information from shallower layers.

(i) RGB image (j) Ground truth (k) FCN-32s (l) FCN-8s

Figure 10: Segmentation results for FCN-32s and FCN-8s compared to ground truth. The left column is RGB images from the validation set. The second is the ground truth after removing the boundaries. The third shows the segmentation produced by coarse FCN-32s, and the fourth is a finer segmentation map produced by FCN-8s with skip connections

The relative small margin ($\tilde{2}$-3%) in mean IOU between our models and the original ones by Long et al [1] is believed to be due to the differences in framework, optimization, and dataset augmentation. Our implementation is Keras-based while the original networks were built and trained using Caffe. The authors benefited from their augmented dataset, GPU resources, and the wrapping around network layers to assign different learning rates for different layers, resulting in better fine tuning, but at the cost of training time (3 days).

VI. Conclusion

In summary, FCNs is promoted due to its leveraging of an arbitrary input, an edge that was not applicable in typical CNNs ending in dense layers. This was verified as per the procedure followed in this work, casting ILSVRC classifiers into FCNs and augmenting them for dense predictions through a novel skip-connection architecture that combines depth, rough semantic information and location. Indeed, both the FCN-8 and FCN-32 architectures surpassed the performance of the previous solutions such as R-CNNs and ImageNet pioneering architectures while also achieveing almost identical perfomance of the original architectures presented by the authors of [1] in a significantly less computational time and resources. In fact, our FCN-8s have reached a pixel accuracy of 89.9% and mean IOU of 57.8%, results that promote the usage of FCN in segmentation tasks, in general, and suggests investigating simpler architectures and optimization techniques.

References

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

[2] Khoreva, Anna Benenson, Rodrigo Hosang, Jan Hein, Matthias Schiele, Bernt. (2017). Simple Does It: Weakly Supervised Instance and Semantic Segmentation. 1665-1674. 10.1109/CVPR.2017.181.

[3] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[4] Canziani, A., Paszke, A., Culurciello, E. (2016). An Analysis of Deep Neural Network Models for Practical Applications. ArXiv, abs/1605.07678.

[5] M. Everingham, L. Van-Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "Visual Object Classes Challenge 2012 (VOC2012)". [Online]. Available: http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2012/index.html. [Accessed: 03-Jul-2021].

[6] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in 2011 International Conference on Computer Vision, 2011, doi: 10.1109/iccv.2011.6126343 [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126343

[7] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous Detection and Segmentation," in Computer Vision – ECCV 2014, Springer International Publishing, 2014, pp. 297–312 [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10584-0$_2$0