IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 9 (September. 2018), ||V (III) || PP 62-68

Semantic Segmentation Using Fully Convolutional Net: A Review

Prisilla J [1], Iyyanki. V. Murali Krishna [2], Aruna. M [3]
[1] *Research Scholar, Secunderabad, India.*
[2] *Former Professor and Director R&D JNTUH, Hyderabad, India.*
[3] *Associate Professor, IBS, The ICFAI Foundation for Higher Education, Hyderabad, India.*

Abstract: - Semantic segmentation has paved its way in predicting the models using dense pixel-wise prediction method apart from classification. The presented models for semantic partitions the image into semantically meaningful chunks and classifies each chunk into one of the predetermined classes. The presented model reduces the parameters to be trained and helps in up-sampling; it describes quality and accuracy and efficient mechanism. Deeper the layers are, helps in capturing the high-level semantic features from the previous convolutional layers.

Keywords: - *Convolutional, conditional random fields, pixel, semantic segmentation*

Date of Submission: 08-09-2018 Date of acceptance: 24-09-2018

I. INTRODUCTION

Segmentation is indispensable for image analysis tasks; the study involves understanding the image whereas the technique of describing each pixel of an image with a class label such as buildings, bikes, person, road, sky, flower, medical images and so on is known as *semantic segmentation*. In computer vision, the well-known task is semantic segmentation**,** along with classification and object detection. Segmentation involves images as inputs and regions and structures as output. The image is processed with filters, gradient and color information; semantic segmentation or pixel classification is one of the pre-defined class labels for each pixel. The input image is segmented into the regions corresponds to the objects of any image scene. Semantic image segmentation, the task of assigning a semantic label such as road, sky, person, and so on to every pixel in an image facilitates several new applications. Exact localization accuracy requirements are assigned to the semantic labels which can locate the image outline, and thus imposes the visual entity recognition tasks such as image level classification or bounding box level detection. A semantic segmentation network classifies every pixel in an image, resulting in an image that is segmented by class. The study involves about object pixels. Semantic segmentation does not require classification rather it works with convolutional neural networks but deep learning requires classification.

The two solutions of recognition task are object class detection and semantic segmentation. Object detection addresses the problem of localization of objects of the classes. Minimum bounding rectangles (MBRs) of the objects are the ideal output. The approach involves the use of a sliding window of varying size and classifies sub-images defined by the window. Object detection reduces to semantic segmentation easily. The result of deep convolutional neural network may be blurred and inaccurate; adding conditional random fields (CRF) gives accurate results.

The applications of semantic segmentation include:
a. autonomous driving
b. industrial inspection
c. classification of terrain visible in satellite imagery
d. medical imaging analysis (cancer cell segmentation)
e. robot vision and understanding

II. PREVIOUS WORK

Convolutional Neural Network

The feedforward neural network is CNN which consists of two parts; the first part is the feature extractor consisting of convolutional and max-pooling layers. The second part consists of the fully connected layer which performs non-linear transformations of the extracted features and acts as the classifier.

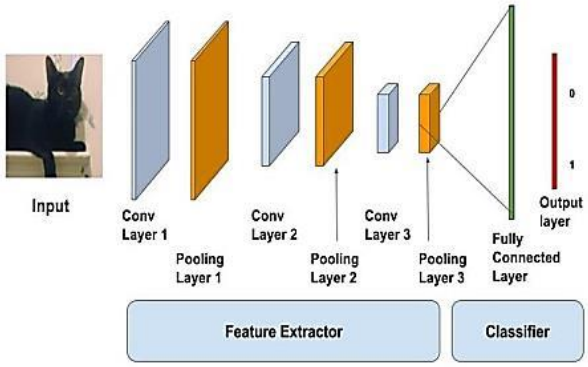


Fig 1: Convolutional neural network

The input is fed to the network of stacked conv layers, pooling and dense layers; softmax layer is the result layer that identifies whether it is a cat or some other image. The neurons in this layer search for specific features; if the neurons find the required features, they produce a high activation.

AlexNet

A. Krizhevsky et. a1 (2012) proposed AlexNet model containing eight layers; 1.2 million training images, 50,000 validation images, and 150,000 testing images. The first five are convolutional layers, and the last three are fully connected layers. It used the non-saturating ReLU activation function, which showed improved training performance over tanh and sigmoid. It reduced the TOP-5 error from 26% to 15.3% and achieved 15.3% TOP-5 test accuracy [1].

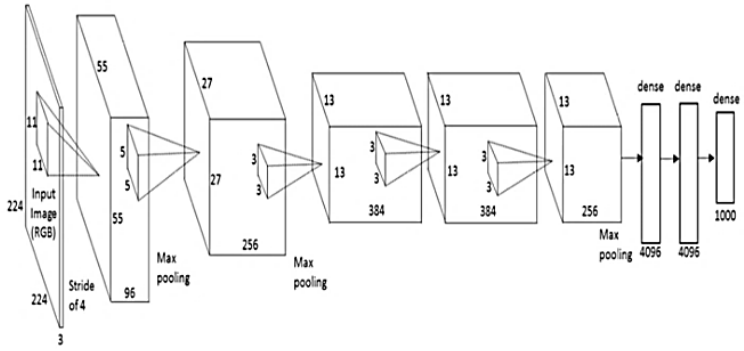


Figure 2: AlexNet Network

VGGNet-16

K. Simonyan and A. Zisserman (2014) proposed visual geometry group convolutional neural network model (VGG), the test accuracy was 92.7% TOP-5 in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It includes a preprocessing layer that takes the RGB image with pixels values in the range of 0-255 and subtracts the mean image values. It consists of 16 weight layers and 138 million parameters [2].

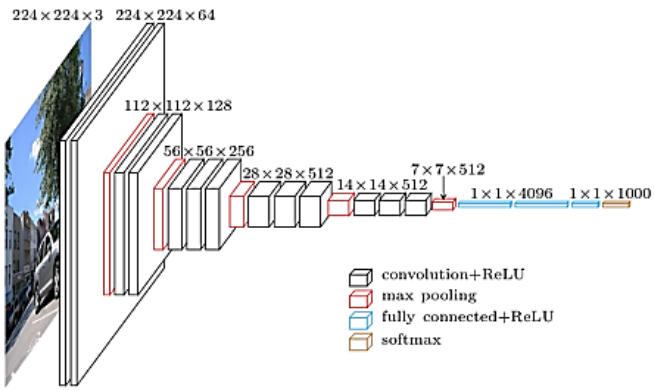


Figure 3: VGGNet -16

GoogLeNet

Szegedy et al. (2015) proposed GoogLeNet is a network which won the ILSVRC-2014 challenge with a TOP-5 test accuracy of 93.3%. The name is a tribute to Yann LeCuns pioneering LeNet 5 network. The network is 22 layers deep CNN, 1.2 million images for training, 50,000 for validation and 100,000 images for testing. GoogLeNet reduced parameters from 60 million to 4 million and it gained a TOP-5 error rate of 6.67% inception [3].

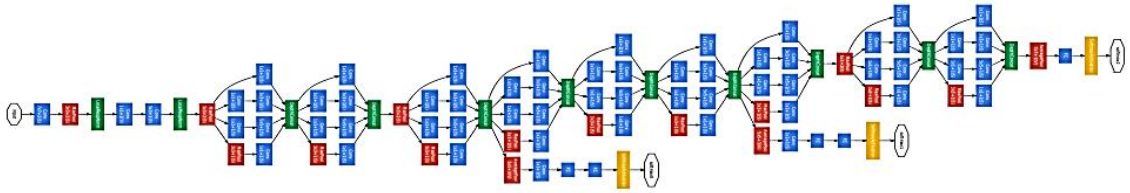


Figure 4: GoogLeNet

SegNet

Vijay Badrinarayanan et. al (2016) proposed model named SegNet, it has an encoder network and a corresponding decoder network. The encoder network consists of 13 convolutional layers which relate to the first 13 convolutional layers in the VGG-16 network designed for object classification. The decoder corresponding to the first encoder closest to the input image produces a multi-channel feature map, although its encoder input has 3 channels (RGB). SegNet, is designed to be an efficient architecture for pixel-wise semantic segmentation [4].

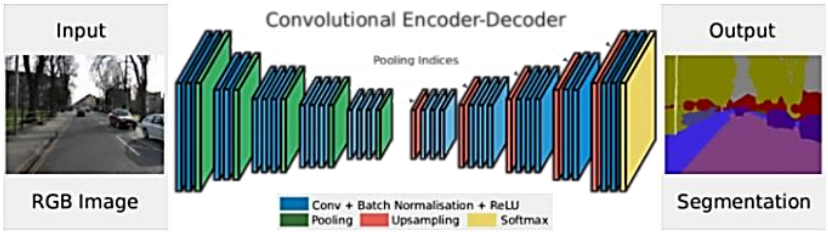


Figure 5: SegNet

III. DISCUSSION AND RESULTS

In this study, the implementation of FCN-8 and FCN-16 is carried out using tensorflow which is used as backend for keras, keras provides high- level building blocks for developing deep learning models, theano used for computation, does packing and unpacking of inputs and return values, scikit-image for image processing and transforming colors to gray and gray to color on the Jupyter notebook, which is an open source web application for running the python code on windows 7 64-bit Intel core i5. The sample images are acquired from ImageNet; own image dataset and the weights used are from pascal-fcn8s-dag.mat and pascal-fcn16s-dag.mat. The images in the ImageNet are preprocessed and labelled by hot encoding and are classified into the various categories. PASCAL VOC 2012 segmentation class images are used for training.

To train end-to-end FCNs involves two steps (1) for pixel-wise prediction and (2) from supervised pre-training. Fully convolutional of existing networks predict dense outputs from arbitrary-sized inputs. Dense feedforward computation and backpropagation are carried out at the same time on the whole image for learning and inference. In each network pixel-wise prediction is done by up-sampling layers and subsampled pooling lets learning in nets.

Transforming a classifier to dense fully connected network

The basic model is trained on the ImageNet prior for classification. The VGG is used to implement the task of semantic segmentation for dense prediction. The last fully connected layers of VGG is replaced with convolutions layer and addition of 1x1 convolution with channel dimension 21 to predict scores for each of the PASCAL classes including background at each of the coarse result locations, followed by a transpose convolution layer with the stride 16 to bilinear up-sampling the coarse results to pixel-dense outputs.

Combining the features from lower level layers to higher layers

The blend of layers of the hierarchy features by the new processing fully convolutional net (FCN) for segmentation refines the spatial precision of the outcome. While fully convolutional classifiers can be fine-tuned

to segmentation and using standard metric it can score high, their outcome is an dissatisfied coarse. In the final prediction layer, 32 pixel stride restricts the scale in the up-sample outcome.

The association of final prediction layer with lower layer with the finest strides by adding skips connections; thereby a line topology is curved into a direct acyclic graph with edges that skip ahead from lower layers to higher ones. The finest scale predictions need to have few layers, thus combining fine layers and coarse layers allows the model form local prediction that esteems the global structure.

The outcome stride is split into two halves by predicting from a 16 pixel stride layer. Then 1x1 convolution layers is added on the top of pool 4 to yield surplus class predictions. The predictions outcome computes on top of conv 7 (conv fc7) at stride 32 by adding a 2x up-sampling layer and summing both the predictions. Lastly, the stride 16 predictions are up-sampled back to the image; this net is known as FCN-16s [5].

Post processing

The discriminative model called a conditional random field is used for predicting the labels, these use contextual information from previous labels increases the amount of information to make a good prediction that correspond to inputs. By applying a post processing phase to refine the segmentation outcome and enhance its ability to capture fine-grained details is to use a conditional random field is the best method. Merging the image information at low level – such as the interactions between pixels with the outcome of multi-class inference systems generates per pixel class scores. When CNN fails to capture the long range dependencies, the merging method supports and fine local details are also taken into account [6].

When predicting FCN uses labels which are provided to each pixel independently including its surrounding labels, this may result in coarse segmentation. CRF takes two inputs one is the original image and the second is predicted probabilities for each pixel. The CRF uses a highly efficient inference algorithm for fully connected CRF models in which the pairwise edge potentials are defined by a linear combination of Gaussian kernels in an arbitrary feature space. Thereby it considers the surrounding pixels while assigning the class to particular pixel which results in better semantic segmentation results; whole image training is effective and efficient.

The result of the trained parameters using both FCN is shown in the table below, the number of conv layers are the same where as deconv size is 528 for FCN-16 given image size is 512 X 512 and deconv size is 2 for FCN-8 for the same image size.

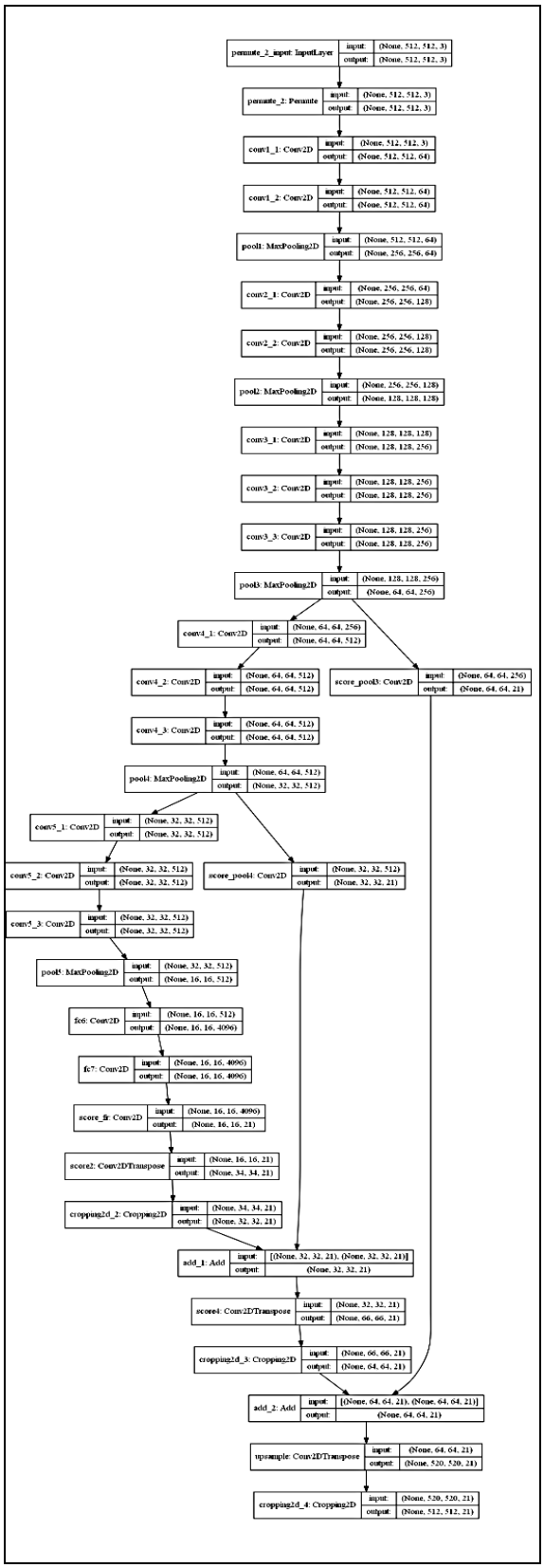
Table 1: comparison of FCN -8 and FCN-16

	FCN-8	FCN-16
Total Parameters	134,489,822	134,816,036
Trained Parameters	134,489,822	134,816,036
Conv. Layers	13	13
Max stride	16	32

The interpolation connects coarse results to dense pixels. From the nearest four inputs, simple bilinear interpolation computes each result using a linear map that depends only on the relative positions of the both input and output cells. The up-sampling with factor *f* is convolution with a fractional input stride of *1/f*. If *f* is integral, then to up-sample is backwards convolution (deconvolution) with an output stride of '*f*'. The operation reverses the forward and backward passes of convolution and are little important. Thus up-sampling can be performed in the network for end-to-end learning by backpropagation from the pixel-wise loss. Hence, deconvolution filter in a layer need not be fixed (bilinear up-sampling), but can be learned. A heap of deconvolution layers and activation functions can learn a nonlinear up-sampling.

In fully convolutional training, it is very usual practice of doing patch-wise training but it lacks efficiency. The class imbalance can be corrected by implementing sampling and mitigate the spatial correlation of dense patches. In fully convolutional training, class balance can also be achieved by weighting the loss, and loss sampling can be used to address spatial correlation [5].

Pooling helps in classifying networks but decreases the resolution with loss of information, hence the skip connections are implemented. In this study data augmentation is not used but it improves the training data size and gives the model more space for better performance.



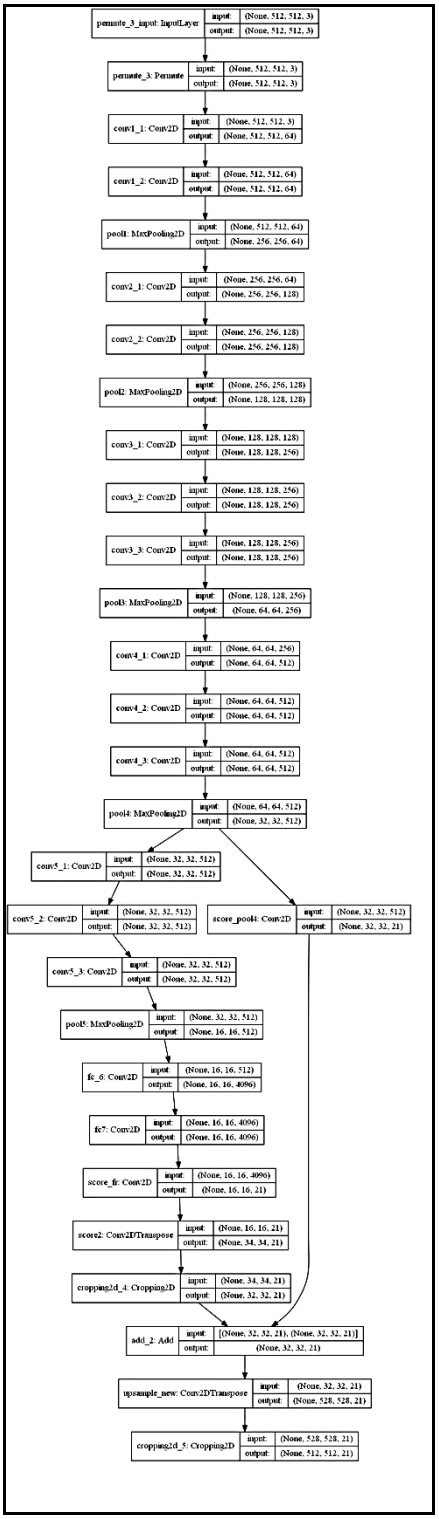
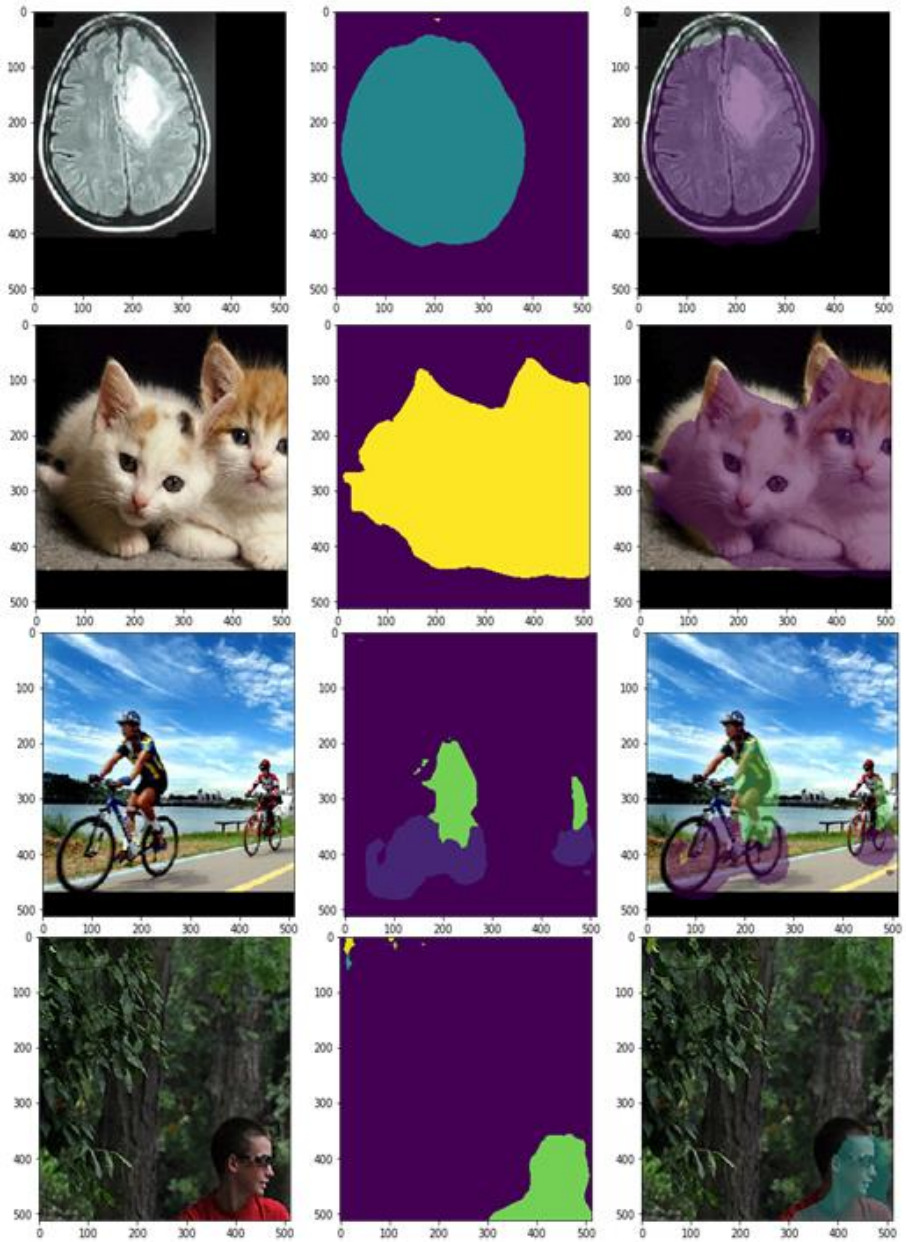


Figure 6: generation of a) FCN-8 model b) FCN-16 model



a) Test image b) ground truth c) prediction

Figure 7: output of FCN- 8 and FCN-16 with the implementation of CRF

IV. CONCLUSION

The presented model for semantics segmentation was able to effectively combine the high level and low level feature to produce high resolution segmentation on the own dataset. The transfer learning paradigm helps in saving lot of computational cost, time and increases the efficiency. Hence fully convolutional networks segments image at pixel level, this promotes the use of end-to-end convolution network for semantic segmentation. And moreover, the use of CRF improves the segmentation efficiently.

V. FUTURE WORK

Implementation of FCN-32 to be carried out on the image dataset as well as implement semantic segmentation of tumor detection in the biomedical areas.

REFERENCES

[1] Alex Krizhevsky et. al, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems, 2012, pp. 1097–1105.*

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv preprint arXiv: 1409.1556, 2014.*

[3] Szegedy et. al ,"Going deeper with convolutions," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.*
[4] Vijay Badrinarayanan et. al, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", *arXiv: 1511.00561v3, 2016.*
[5] Evan Shelhamer et. al, "Fully Convolutional Networks for Semantic Segmentation", *arXiv: 1605.06211, 2016.*
[6] A. Garcia-Garcia et.al, "A Review on Deep Learning Techniques Applied to Semantic Segmentation", *arXiv: 1704.06857, 2017.*

Prisilla J. et al. "Semantic Segmentation using Fully Convolutional Net: A Review" IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 9, 2018, pp. 62-68