Semantic Segmentation on Variable Sized Images of Red Blood Cells using Deconvolution Network

Kushagr Arora
BITS-Pilani, K.K.Birla Goa Campus

Taveesh Sharma
BITS-Pilani, K.K.Birla Goa Campus

Utkarsh Vaish
BITS-Pilani, K.K.Birla Goa Campus

***Abstract*—This paper focuses on the task of semantic segmentation on images of red blood cells. We are learning a deconvolution network for this task. The deconvolution network is composed of deconvolution and unpooling layers, which identify pixel-wise class labels and predict segmentation masks. We train the model on various images of red blood cells. We apply the trained model on the test dataset and construct the final semantic segmentation map. We have got 100% results with the intersection over mean (IoU) of the test images of more than 97%. This clearly signifies that the model that we have trained is highly efficient and can be used for any segmentation tasks with just few training iterations.**

Keywords—convolution; deconvolution; IoU; semantic segmentation.

I. Introduction

Semantic segmentation is the task of clustering parts of images together which belong to the same object class . This type of algorithm has several use cases such as detecting road signs , detecting medical instruments in operations, detecting tumors, colon crypts segmentation, land use and land cover classification. In contrast, non-semantic segmentation only clusters pixels together based on general characteristics of single objects. Hence, the task of non-semantic segmentation is not well defined, as many different segmentations might be acceptable. Figure 1 shows such task of semantic segmentation.

Object detection, in comparison to semantic segmentation, has to distinguish different instances of the same object. While having a semantic segmentation is certainly a big advantage when trying to get object instances, there are a couple of problems: neighboring pixels of the same class might belong to different object instances and regions which are not connected my belong to the same object instance. For example, a tree in front of a car which visually divides the car into two parts.

Convolutional neural networks (CNN) have shown excellent performance in various visual recognition problems such as image classification [1, 2, 3], object detection [4, 5], semantic segmentation and action recognition. The representation power of CNNs leads to successful results; a combination of feature descriptors extracted from CNNs and simple off-the-shelf classifiers works very well in practice. Encouraged by the success in classification problems, researchers start to apply CNNs to structured prediction problems, i.e., semantic segmentation, human pose estimation and so on.

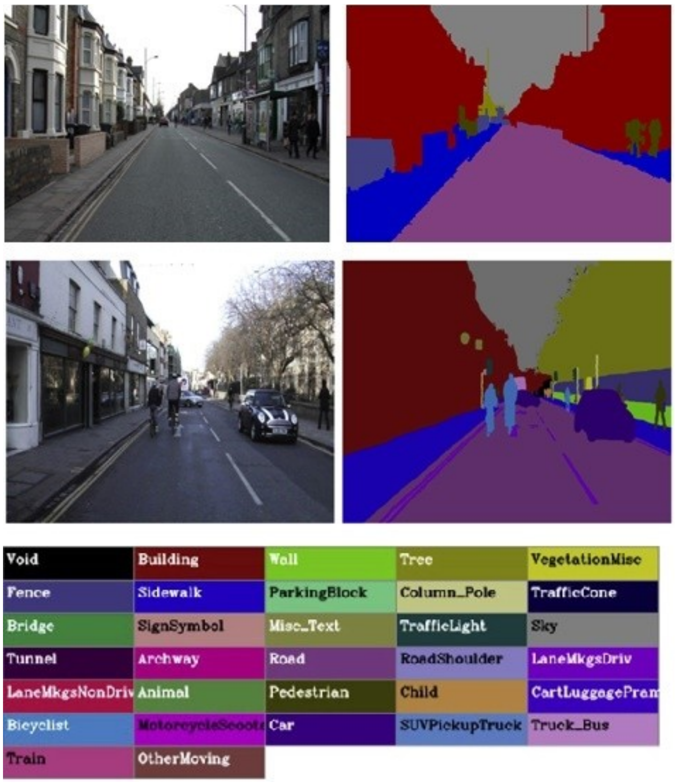

Figure 1: Task of semantic segmentation

We are using a deconvolution network for the training the images to perform semantic segmentation. We have chosen the deconvolution network instead of fully convolutional network because of some of limitations of FCNs. First, the FCN can only handle single scale semantics within image due to the fixed-size receptive field. Therefore, the object that is substantially larger or smaller than the receptive field may be fragmented or mislabeled. Also, small objects are often ignored and classified as background, which is very crucial in our case because the size of the RBCs in our image is very small. Thus, we require a high level accuracy. To overcome such limitations, we are using a deconvolution network.

II. Related Work

CNNs are very popular in many visual recognition problems and have also been applied to semantic segmentation actively. We first summarize the existing algorithms based on supervised learning for semantic segmentation.

There are several semantic segmentation methods based on classification. Some classify multi-scale superpixels into predefined categories and combine the classification results for pixel-wise labeling. Some algorithms [3] classify region proposals and refine the labels in the image-level segmentation map to obtain the final segmentation.

Fully convolutional network (FCN) [17] has driven recent breakthrough on deep learning based semantic segmentation. In this approach, fully connected layers in the standard CNNs are interpreted as convolutions with large receptive fields, and segmentation is achieved using coarse class score maps obtained by feedforwarding an input image. An interesting idea in this work is that a simple interpolation filter is employed for deconvolution and only the CNN part of the network is fine-tuned to learn deconvolution indirectly.

III. Architecture

This section discusses the architecture of our deconvolution network and describes the overall semantic segmentation algorithm.

A. Architecture

Figure 2 illustrates the detailed configuration of the entire deep network. Our trained network is composed of two parts—convolution and deconvolution networks. The convolution network corresponds to feature extractor that transforms the input image to multidimensional feature representation, whereas the deconvolution network is a shape generator that produces object segmentation from the feature extracted from the convolution network. The final output of the network is a probability map in the same size to input image, indicating probability of each pixel that belongs to one of the predefined classes.

B. Deconvolution Network for Segmentation

There are two major operations in any deconvolution network. These are:

- **Unpooling**

Pooling in convolution network is designed to filter noisy activations in a lower layer by abstracting activations in a receptive field with a single representative value. Although it helps classification by retaining only robust activations in upper layers, spatial information within a receptive field is lost during pooling, which may be critical for precise localization that is required for semantic segmentation[6]. To resolve such issue, we employ unpooling layers in deconvolution network, which perform the reverse operation of pooling and reconstruct the original size of activations as illustrated in Figure 3. To implement the unpooling operation, we follow the similar approach proposed in [7, 8]. It records the locations of maximum activations selected during pooling operation in switch variables, which are employed to place each activation back to its original pooled location. This unpooling strategy is particularly useful to reconstruct the structure of input object as described in [7].

- **Deconvolution**

The output of an unpooling layer is an enlarged, yet sparse activation map. The deconvolution layers densify the sparse activations obtained by unpooling through convolution-like operations with multiple learned filters. However, contrary to convolutional layers, which connect multiple input activations within a filter window to a single activation, deconvolutional layers associate a single input activation with multiple outputs, as illustrated in Figure 3 [6]. The output of the deconvolutional layer is an enlarged and dense activation map. We crop the boundary of the enlarged activation map to keep the size of the output map identical to the one from the preceding unpooling layer. The learned filters in deconvolutional layers correspond to bases to reconstruct shape of an input object. Therefore, similar to the convolution network, a hierarchical structure of deconvolutional layers are used to capture different level of shape details. The filters in lower layers tend to capture overall shape of an object while the class-specific fine-details are encoded in the filters in higher layers. In this

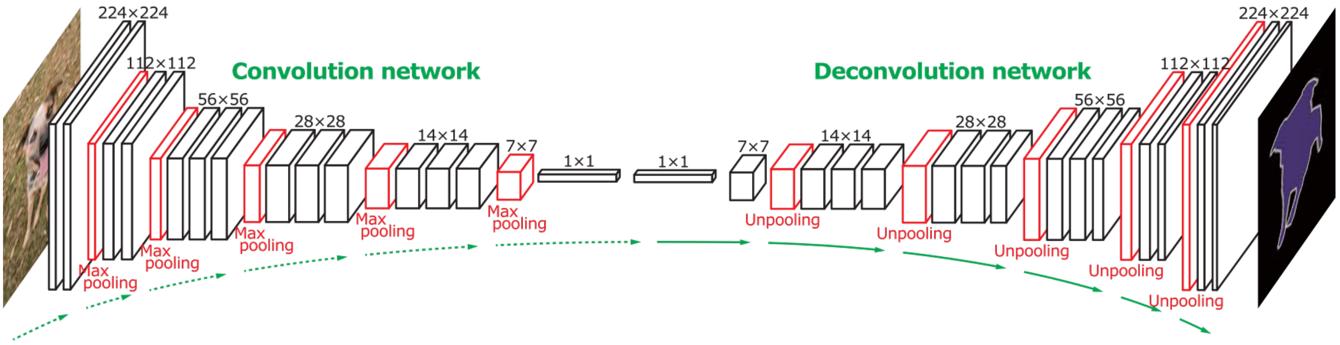

Figure 2: Overall architecture of the network

way, the network directly takes class-specific shape information into account for semantic segmentation, which is often ignored in other approaches based only on convolutional layers [1].

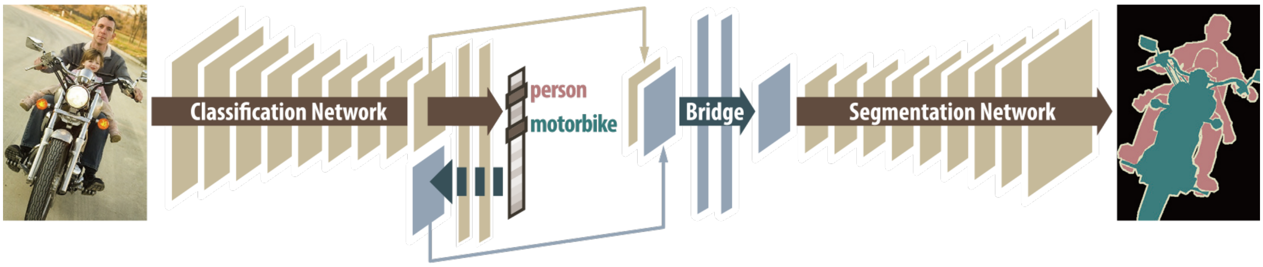

Figure 3: Working of the architecture for semantic segmentation

C. *Proposed Architecture*

Figure 3 shows the nature of proposed architecture. The architecture that we used for training the model is as follows:

- Each training image is 128 x 128 pixels image.
- We take the input as 128 x 128 x 3 where 3 denotes the channels of RGB image.
- This input is fed to a convolutional layer followed by ReLU layer.
- The output of the previous layer is again fed into a convolutional layer followed by ReLU layer.
- The next layer is max pooling layer with filter size of 2 x 2.
- The above set of layers is repeated three times, making in total of 6 convolutional layers each followed by a ReLU layer, with a max pooling layer at intervals of two.
- The output of the above network is fed into a dropout layer with keep probability of each neuron of 0.5.
- The next layer is an unpooling layer.
- The output of the layer is fed into a deconvolution layer.
- Similarly, the deconvolution network is just the reverse of the convolution network.
- The final output is scaled into height x width x 2 where 2 denotes the two class labels, namely class label and background label.
- We have used filters of size 3 x 3 with stride of 1 x 1 on both x and y axes.

IV. Experiment

The dataset comprises of 164 images of red blood cells each having dimensions of 128 x 128. The entire code is run using TensorFlow 1.0 and is trained on Tesla GPU, which took around 20 minutes for 5-fold cross validation. We have done a 5–fold cross validation on the training set using Scikit Learn. We have used cross entropy as our loss function. We have used Adam Optimizer fro minimizing the loss. The loss values after various iterations for different values of learning rate are shown in figure 4, 5 and 6. Since the loss is converging well with learning rate of 1e-4, we have chosen that as our learning rate.

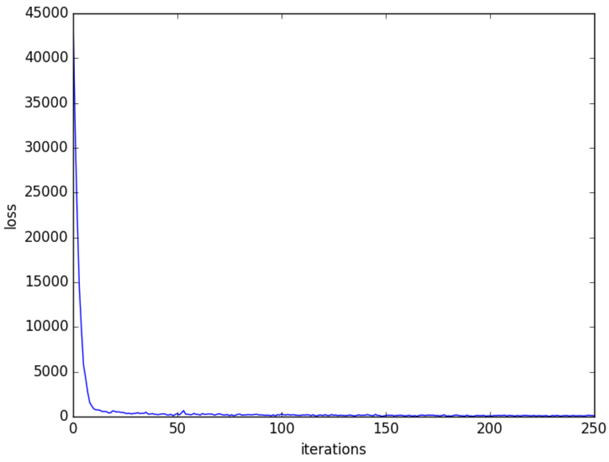

Figure 4: Plot of loss values versus iterations with learning rate 1e-3

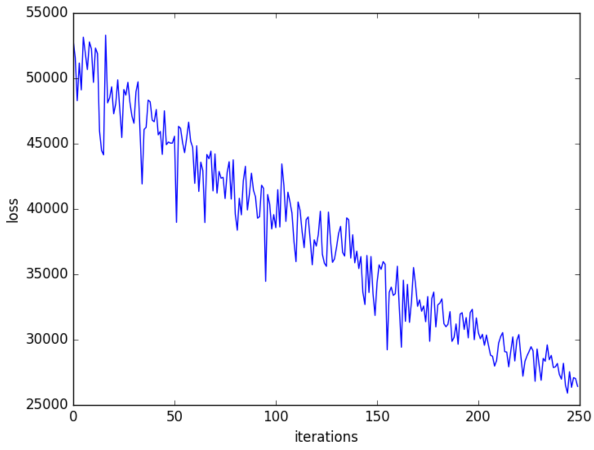

Figure 5: Plot of loss values versus iterations with learning rate 1e-5

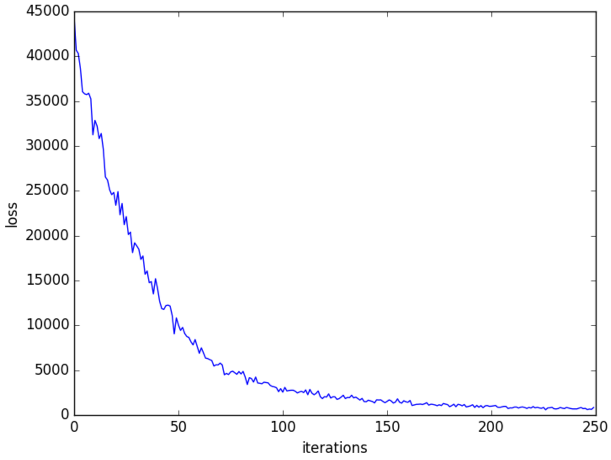

Figure 6: Plot of loss values versus iterations with learning rate 1e-4

The trained model is tested on the test dataset which comprises of 40 variable sized images of RBCs. The model shows very accurate results giving IoU of more than 0.97 for most of the images. This shows that our proposed model is highly accurate and efficient and can be used for segmenting any complex image within a few minutes. The measure used for testing the accuracy is IoU meaning intersection over union. Intersection over Union is an evaluation metric used to measure the accuracy of an object detector on a particular dataset. Intersection over Union is simply an *evaluation metric*. Any algorithm that provides predicted bounding boxes as output can be evaluated using IoU. More formally, in order to apply Intersection over Union to evaluate an (arbitrary) object detector we need:

1. The *ground-truth bounding boxes* (i.e., the hand labeled bounding boxes from the testing set that specify *where* in the image our object is).
2. The *predicted bounding boxes* from our model.

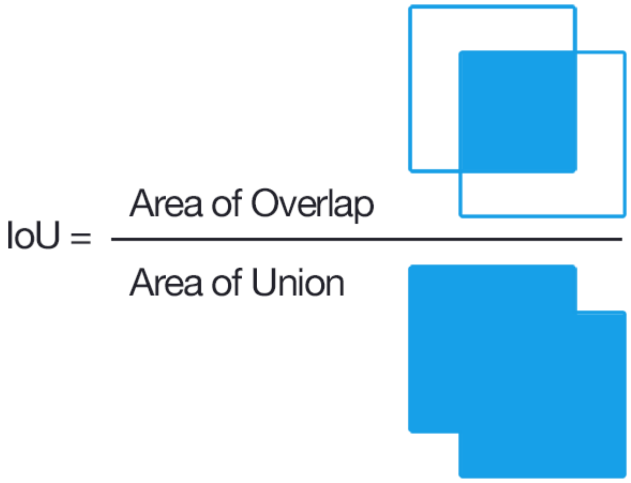

Figure 7: Visual meaning of intersection over union(IoU)

Figure 7 shows the visual meaning of the IoU for an image.

V. Inferences

The above results show that our proposed model is highly accurate and efficient. Any image can be properly segmented within a few minutes, but the computation is highly expensive. It requires high speed GPU with around 2 GB memory to train the weights. But the accuracy achieved is very high once the model is trained. We got around 0.97 IoU for each test image which was a variable sized RBC image with very minute class labels. Thus, a deconvolution network gives high accuracy for the task of semantic segmentation.

References

[1] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

[4] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. TPAMI, 35(8):1915–1929, 2013.

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

[6] Hyeonwoo Noh , Seunghoon Hong , Bohyung Han, Learning Deconvolution Network for Semantic Segmentation, Proceedings of the

2015 IEEE International Conference on Computer Vision (ICCV), p.1520-1528, December 07-13, 2015

[7] M. D. Zeiler and R. Fergus. Visualizing and understandingconvolutional networks. In ECCV, 2014.

[8] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In ICCV, 2011.