ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Semi-supervised image segmentation in a Bayesian framework using random forests and CNNs: a comparative study

Master Thesis

Prateek Purwar

Department of Electrical Engineering and Information Technology

Advisor: Dr. Gregory Paul

Supervisor: Prof. Dr. Orcun Goksel

Abstract

Image segmentation is a fundamental middle-level computer vision task, necessary to higher level image understanding, such as semantic image analysis, scene understanding, diseases diagnosis, etc. Recently, convolutional neural networks (CNN) have set new state-of-the-art standards in the field, and attract a lot of attention among both practitioners and experts. Beyond the accuracy they can achieve in many computer vision tasks, their attractivity lies in their versatility, their capability to be reused and transferred to similar problems, their layered architecture, and their capability to learn meaningful features. Nonetheless, in practice, the main obstacle is to obtain a sufficient number of annotated image data for the task at hand.

This poses a major problem for application scenarios where large annotated data-sets are not available or difficult to obtain. This thesis tackles such a case, and is motivated by the problems faced in an imaging facility: annotations can be difficult, even for experts, images are very diverse in nature, and in appearance.

In this thesis, we tackle the problem of applying CNNs in a semi-supervised and semi-interactive image segmentation scenario. We study how the segmentation accuracy of a segmentation pipeline evolves with the annotation effort of the user. Our base CNN is OSVOS, developed for video segmentation, where only a very small subset needs to be annotated (one to three frames fully labeled). This work focuses on partially annotated images with scribbles. We compare two strategies: a random forest (RF)-based and CNN-based. We derive a loss function for training CNNs from scribbles. Varying amounts and different types of scribbles are used to train either a RF or our modified OSVOS. We show that both the quantity and quality of the annotations are important for increasing the segmentation accuracy, and that the RF-based pipeline is better for the low-annotation regime.

We also compare different post-processing strategies of the predicted soft segmentation mask: thresholding and variational image segmentation. We show that the type of labeling cost used in the variational

model matters. The model we propose ensures that one can always benefit from post-processing the soft-mask with a variational method. This is not the case for the widespread cost function in the literature, that can degrade the segmentation accuracy, even when the RF or the CNN make good predictions.

Acknowledgements

I would like to acknowledge the support of my Master Thesis advisor Dr. Gregory Paul, Post-Doc. at Computer Vision Laboratory at ETH. I would like to thank Jordi Pont-Tuset for steering me in right direction whenever I got stuck. I would sincerely thank Denis Samuylov and Christoph Mayer, who helped with implementation and coding at each step of my thesis.

Contents

List of Figures

Chapter 1

Introduction

Image segmentation is a central task in biomedical image processing and image analysis. It can be used for detecting various diseases, shape diagnosis etc. The dataset which we are trying to segment is electron microscopic (EM) images of liver tissue. The dataset consists of a 3D stack of 2D slices of liver tissue as shown in figure 1.1. The image stack was generated at EMEZ (Electron Microscopy ETH Zrich) autonomously by iter- atively cutting off a slice of the sample and scanning the cut face. With very little interaction, the apparatus produced an output of 458 slices in 20 hours. The aim of this thesis is to compare and analyze different segmentaion methods to segment objects in histology images. In the following section, we describe the difficulties faced in segmentation of the histology images and give an overview of the methods used for segmentation.

1.1 Segmentation of Histology Images

The image segmentation is the process of partitioning a digital image into multiple segments. More precisely, image segmentation is the process of assigning a label to each pixel in an image such that pixels with the same label belong to same object. The main difficulty arises in assigning the labels to the pixels at the boundaries, which are difficult to discriminate from the surrounding pixels. The task of segmentaion in classical computer vision scenario, mainly consists of segmenting well defined objects such as chairs, faces. Due to this reason, the ground truth annotated masks for training can be generated with comparative ease. As opposed to classical scenario, the objects segmented in the histology images do not have known shapes

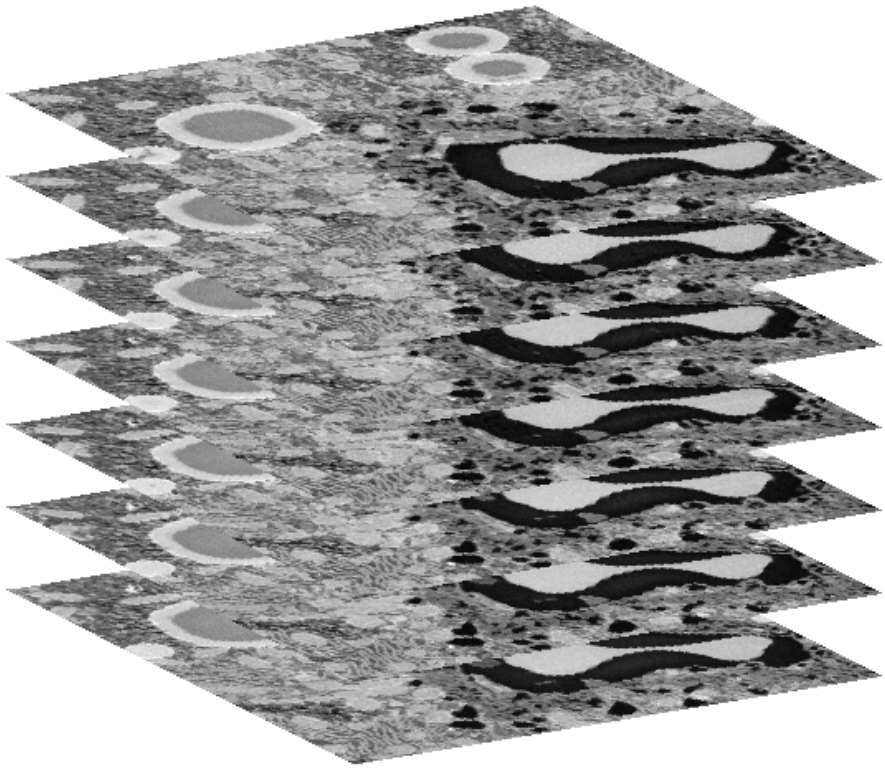


Figure 1.1: A 3D image stack of liver tissue, output from a scanning electron microscope. The stacks contains 458 2D images with a resolution of 1890x1952 pixel each.

and are not well defined. This difficulty can be obsereved in figure 1.2 where it can be seen that the shape of the nuclei is not well defined and even unknown to experts. In contrast to classical scenario, segmenting histology image data faces the following challenges:

- High variability between image data: The images to be segmented may be entirely different i.e. having fixed objects as in liver tissue or having layers to segments as in neuron tissue. The objects to be segmented may differ completely from being smooth (round vesicles) to branched (neurons). We can observe few objects of interest in figure 1.2.

- High variability between objects to segment: The object of interest, that the experts try to segment, may vary significantly in shape, size, and texture.

- High variability between goals: Even for a single image, the goal of the segmentation can be totally different. The images annotated for one object can not be used again for training purpose.

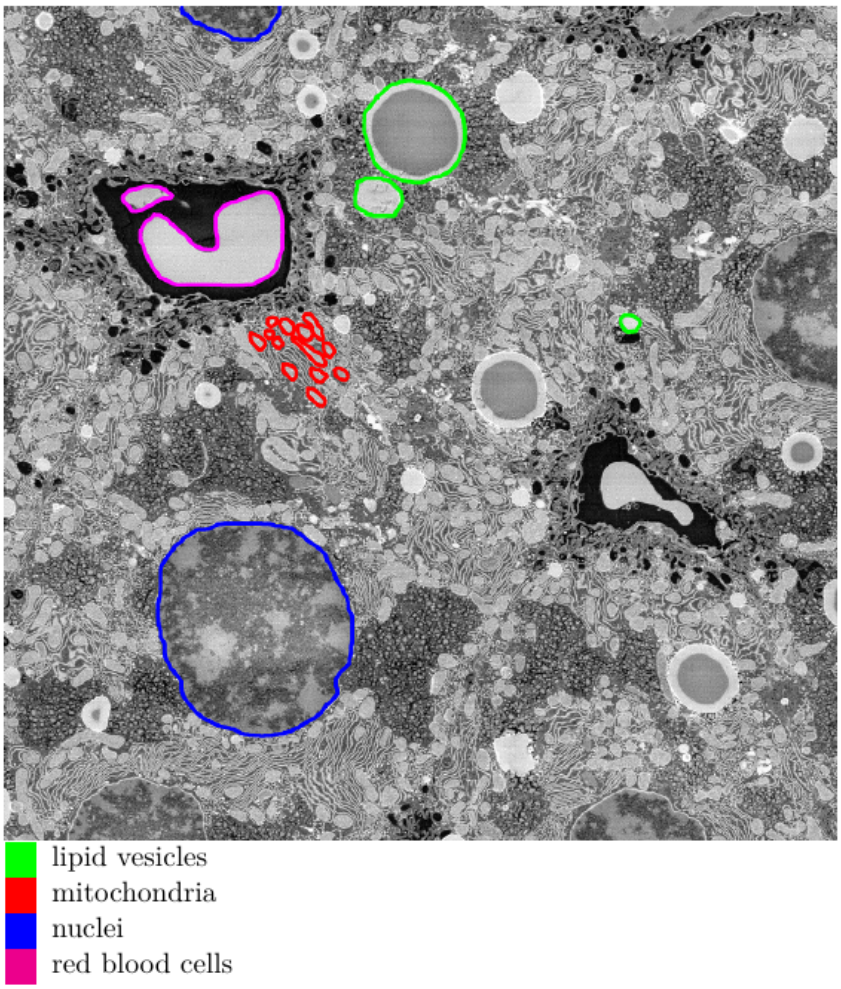


Figure 1.2: EM image of the liver tissue showing different objects of interest to experts.

These characteristics of the histology images make it extremely difficult to fully annotate the objects and, becomes an extremely time-consuming task for experts. In this thesis, the focus is to segment vesicles in EM images of liver tissue (objects shown with green boundary in figure 2). The figure 1.3 shows a segmentation mask containing annotations for all vesicles in the cropped part of a slice. We can realize the difficulty in annotating multiple vesicles of undefined shapes and sizes. Most of the vesicles in figure 1.3 are of round shape but a few can be observed of having irregular shape. We can also observe the huge variation in size of objects, making it difficult to recognize and even cumbersome to annotate. In addition to this intrinsic, the experts are uncertain about the existence of vesicles or about the shape, in certain parts

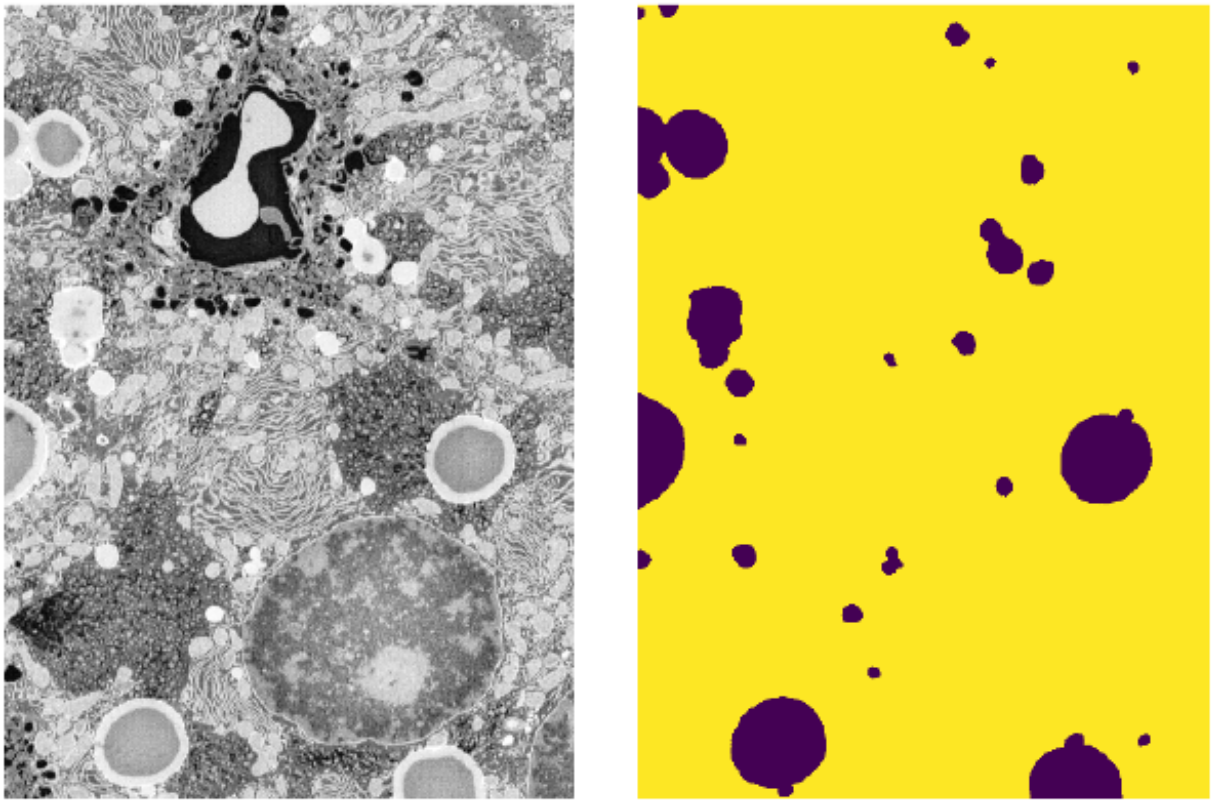


Figure 1.3: An example of the 2D slice and its segmentation mask showing vesicles as Foreground. Foreground: Purple, Background: Yellow

of images due to similarity in appearance or shape to other objtects. This uncertainty is shown in figure 1.4, where the differences (bounded by red rectangles) can be observed in manually annotated segmentation mask of the same image done by different experts. Moreover, sometimes there are differences in the masks for the same slice, annotated by the same expert at different times (shown in figure 1.5). The figure 1.5 shows difficulty in annotaion of vesicles due to not so well defined shapes and size of microscopic objects in microscopic images. This uncertainty has been analyzed a lot in literature and researchers have tried to come up with different methods to generate a ground truth mask from these multiple annotations by experts. The literature provide options of using either STAPLE [2] algorithm, or union, or majority voting to derive reference mask. The reference mask derived for one slice using the STAPLE and the union method is shown in figure 1.6. The choice of method differs for different tasks and varies according to the goal of the task. In this thesis, we make use of reference masks generated by the STAPLE algorithm and the union method to evaluate our approaches for the task of segmentation.

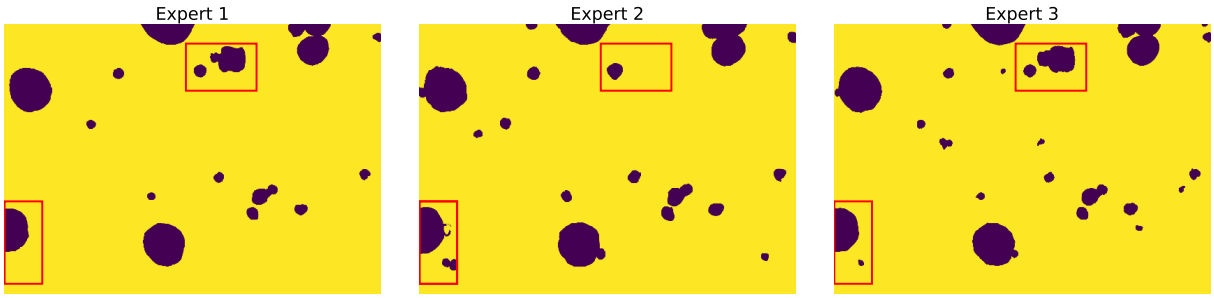


Figure 1.4: Segmentation mask for a slice annotated by different experts. The part of the mask bounded by red rectangles shows some of major differences.

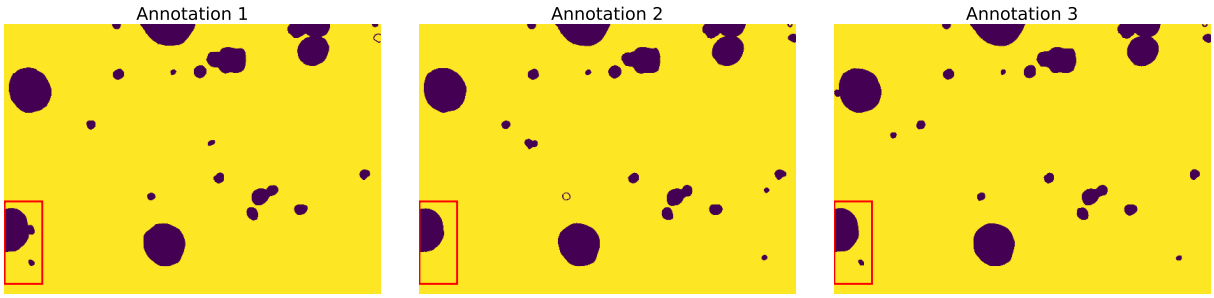


Figure 1.5: Segmentation mask for a slice annotated by same expert at different times.The part of the mask bounded by red rectangles shows some of major differences.

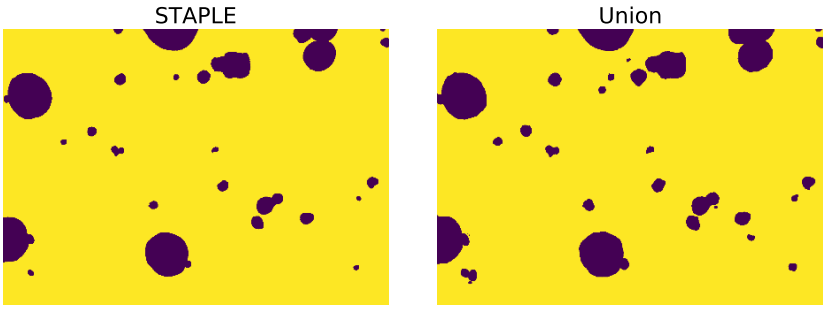


Figure 1.6: Grounth truth segmentation mask derived from multiple annotaions using 2 different methods (STAPLE and union).

1.2 Focus of this thesis

The focus of this thesis is to explore and analyze performance and limitations of two different methods, convolutional neural networks (CNN) and random forests (RF) to solve task of segmenting vesicles in 3D microscopic images of liver tissue. As we observed difficulty of obtaining significant amount of ground truth

segmentation masks in previous section, we wanted to analyse the performance of these two approaches with variation of amount of training data. Initially, we tried CNNs using fully annotated segmentaion mask to obtain a benchmark for performance. Nowadays, it is common to train deep convolutional neural networks (DNN) using transfer learning to compensate for the scarce training data. We decided to choose a network pre-trained for segmentation task and fine-tune it for our problem. We chose a pre-trained network explained in Caelles [1]. This paper describes a network, termed as **OSVOS**, which is designed for tackling the task of **semi-supervised** video object segmentation, i.e., segment an object in a video, given fully annotated mask of the object in the first frame. This task can be considered to be similar to segmenting objects in a 3D stack of slices. We tried to fine-tune the network using limited number of segmentation masks with fully annotated objects and obseved the variation in performance with increasing amount of training data (number of slices). The details of this approach and experiment are described in chapter 2.

The use of pre-trained networks makes it possible to use DNNs even with small amount of training data. But still to train the DNNs, we need to provide the segmentation masks with fully annotated objects of interest for all training images. This comes out to be a tedious and difficult task for our problem as explained in previous section. In addition, the presence of multiple objects of different shape and sizes makes it even more difficult and time-consuming. Imagine 1000 cells in a 2D slice and possibility to manually annotate all these cells of undefined shapes! This leaves us with the option of annotating few objects and train networks using either cropped images or treating rest of image as background. Or we can use semi-supervised learning using partial annotations. In literature, we can find various methods to use these partial annotations to classify each pixel as foreground or background. We made use of Random Forests as described by Santner [3] or Eugster [4] for segmenting objects in images using partial annotations. In this thesis, the main focus was to discover the effect of annotation budget i.e. the number of pixels to annotate on the accuracy achieved.

The RFs learn pixel level information and are uncertain for the maximum of pixels i.e. the probability of foreground learned is not binary but lies between 0 and 1. The simple approach is to use thresholding to generate binary segmentation mask. In literature, different approaches can be found to use prior information to compensate for training data and for the uncertainty of estimators. The most common is to use Conditional random fields (CRFs) to regularize the probability mask learned from RFs. We solved this problem using a prior in **Bayesian framework** using Total Variation (TV) similar to method described in thesis by Eugster

[4]. In thesis of Eugster [4], they tried to learn likelihood using Random forests and implemented prior as an isotropic total variation (TV). They used a non-linear cost function to formulate likelihood from probabilities learned from Random forests. Instead of using a non-linear function to use probabilities, Santner [3] uses a linear function to derive an energy minimization problem using probability mask obtained from RFs. This motivated us to analyze and compare these different cost functions and observe the advantage of using these cost functions in different scenarios.

In summary, we use a Bayesian approach with RF to parametrize likelihood and isotropic TV as prior to predict segmentation mask for a given image. This gives us chance to generate fully annotated segmentation masks and train CNN to obtain better accuracy. The common problem for use of prior is the choice of appropriate scaling to couple likelihood and prior costs. Ranftl [5] coupled the prior cost function with the likelihood cost function obtained from CNN. They optimized the final loss function to obtain optimal values for network parameters (weights and biases) and regularization parameter. Riegler [6] [7] proposed a method to implement TV as specialized layers in CNN and trained the complete model, CNN + TV, together. This motivated us to replace RF with CNN and try to fine-tune pre-trained fully convolutional network from partial annotations. We were able to restructure cross-entropy loss function of the pre-trained network to compute loss for partial annotations. We trained the CNNs using different annotation budgets and did a comparative analysis between results obtained for CNN and RF using different annotation budgets.

1.3 Thesis Organization

The thesis is divided mainly into two sections: segmentation using fully annotated objects and segmentation using partial annotations. The segmentation using full annotations is described in Section 2. The latter method is described in Section 3. In section 3.1, we describe the improvement in segmentation mask using RF for increased labeling effort. We introduce the use of prior and variational methods in section 3.2. In section 3.3, we introduce use of CNNs to learn from partial annotations. Finally, in the last section, we conclude this thesis and lay out future work that can be done.

Chapter 2

Training CNNs using fully annotated segmentation masks

Nowadays, the CNNs have evolved with great speed to solve major problems in classical computer vision scenario. Instead of using classical image processing filters, the CNNs are specialized to learn feature maps from the examples provided and, specific to the task at hand. The resereach in CNNs has evolved to provide an end-to-end solution without any requirement of pre/post-processing of images. Thus, the state-of-the-art approach to solve any task in computer vision is to use CNNs. In this chapter, we describe the diffculties faced in using the CNN for our task and the alternate approach we took to overcome the limitations and difficulties.

2.1 Overview

The classical way to use the CNNs is to design a neural network architecture and train it from scratch. One of the challenge in this classical approach is the choice of architecture of network and choice of training parameters such as initialization, loss function etc. In literature, we can find different architectures of neural networks specially designed for the task of segmentation, one of the popular architecture is U-Net [8]. The major prerequisite of this approach to perform well is huge amount of training data: images and ground truth labelled data. For classical problems such as segmenting faces etc., a huge amount of images available

on internet can used to generate a training dataset. This becomes a problem in the medical domain where it is very costly to generate images and, even more costly and time-consuming to prepare it for training. Although, due to advancement in technology, the microscopic images can be generated with an ease and is not a time-consuming and tedious process anymore. For example, the microscopic images of liver tissue were generated automatically with the help of a robotic arm. But, the diffuclty in using such CNNs lies with generating significant amount of annotated data for training. For our problem of image segmentation, we described the problem faced by experts and doctors in generating fully annotated segmented masks in introduction. This poses a limitation in using such CNNs for our task of segmentation of objects in microscopics images. The alternate approach to train CNNs in such cases has been by use of transfer learning.

2.2 Transfer Learning

The use of transfer learning provides a way to compensate for the scarce training data. Transfer learning tries to store the knowledge gained from solving one problem and applying it to a different but related problem. Thus, in practice, it is becoming rare to train an entire CNN from scratch, as it is relatively difficult to have a dataset (images and labels) of sufficient size for training. Instead, one of the common way to perform transfer learning is to use a pre-trained network, either to initialize a network or to extract required feature maps. Shelmar et al. [9] designed a ”fully convolutional” network that takes input of arbitrary size and produces segmented output for a complete image. They adapted contemporary classification networks (AlexNet, the VGG net, and GoogLeNet) into fully convolutional networks and transfer their learned representations by fine-tuning to the segmentation task. Similar to Shelmar, we could have chosen one of the contemporary classification networks and fine-tuned them for segmenting objects in liver tissue images. But, due to having very small set of slices for training (10 slices with fully annotated ground truth masks), we decided to choose a network pre-trained and modified for task of image segmentation. Then we could be able to fine-tune it for our task even with such small amount of images. Recently, Caelles et al. [1] described an architecture using a pre-trained model which performed well for the task of **semi-supervised** video object segmentation. This paper termed the network designed as **OSVOS** i.e. One shot video object segmentation.

2.3 One shot Video object segmentation (OSVOS)

Caelles et al. [1] implemented an architecture to segment an object in a video sequence using only one frame for training. The network is trained to learn object semantics from only one frame and generate segmentation mask for remaining all frames. The segmentation works well if the object remains in relatively similar shape and size. The example result of this architecture can be seen in figure 2.1.

Figure 2.1: Example result of OSVOS [1]: The segmentation of the first frame (red) is used to learn semantics of interested object, whicich is segmented in the rest of the frames independently (green).

Caelles et al. [1] modified a CNN network pre-trained for image recognition and fine-tuned it for object segementation, as shown in figure 2.2. This was achieved by training it on a set of videos with manually annotated objects. They designed OSVOS by adopting a fully convolutional network (FCN), suitable for dense predictions. The major drawback with use of FCN for task of segmentation is the coarse scale of the deeper layers due to downsampling of feature maps. This leads to inaccurately localized predictions. One way to overcome this drawback is by making use of skip connections of initial larger feature maps [8]. To avoid this inaccuracy, OSVOS combined information from all network layers to predict segmentation mask for the image.

The design of architecture of OSVOS network drew inspiration from the CNN architecture of [10], originally used for biomedical image segmentation of retinal vessles. The OSVOS network was based on the pre-trained VGG [11] network. They removed the last fully connected layers and the output from the deeper layers was interpolated to orginal image size. The VGG architecture can be divied into 5 stages, each consisting of groups of convolutional plus Rectified Linear Units (ReLU). Between these 5 stages, pooling operations downscale the feature maps as we go deeper into the network. The output feature maps (before pooling) from stage 2 to 5 are connected using convolutional layer to form separate skip paths. The output feature maps from each skip path are fused linearly and upscaled as required to generate a segmentation mask of same size of original image. They call these masks as ”side outputs”. In addition, the feature maps

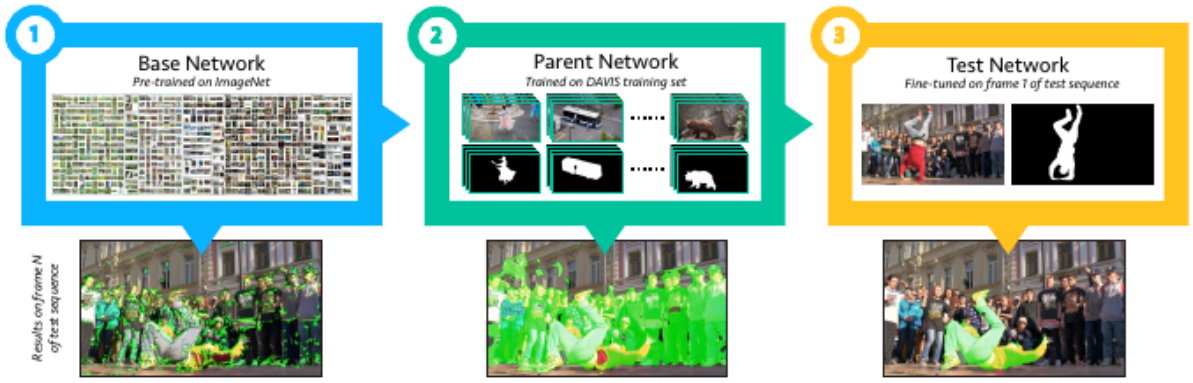


Figure 2.2: Overview of OSVOS [1]: (1) Pre-trained base CNN; its results in terms of segmen- tation. (2) Training of network for task of object segmentation, parent network. (3) By fine-tuning on a segmentation example for the specific object in first frame, the network rapidly adopts to focus on that target.

from separate paths are are concatenated to construct a volume with information from different levels of layers. Finally, they linearly fuse the volume of feature maps to a single output, called ”main output”, which has the same dimensions as the image. They computed cross entropy loss using both ”main” and ”side” outputs. The use of information from all layers enables the network not to loose finer details of the image.

2.4 Motivation for using OSVOS

Once the base network has been modified and trained to perform object segmentation, the parent network can be considered as a pre-trained network for image segmentation. The network is able to learn object semantics and identify various objects in the image. The test network, as shown in figure 2.2, only needs information about object of interest and starts considering rest of image as background. The test network learns about object from a single frame i.e. a single image and generates segmentation mask for all other frames independently. If we consider the 3D volume of slices of liver tissue similar to a video, each slice can be thought of as a frame. Motivated by this similarity, we decided to fine-tune the parent network using few slices and generate the segmentation mask for the whole 3D volume. The liver vesicles in different slices of 3D volume are approximately similar in size and shape, and thus, fine-tuning using few slices can enable the network to learn various features of vesicles. The use of OSVOS fits our problem due to limited availablity of training data.

2.5 Experiment and results

We annotated liver vesicles of different size and shapes in few randomly chosen slices. These slices were used to fine-tune the parent OSVOS network to target the object of interest. In literature, we can find that network tends to overfit on a relatively small dataset. The use of augmentation helps in generating more data and avoids network from overfitting. We generated more data using cropped and flipped slices for fine-tuning. The segmentation output from CNN trained using 2 slices is also shown in figure 2.1. It can be seen that the OSVOS is able to learn about the features of object and generates a good segmentation mask with accuracy (F-score) of 0.86.

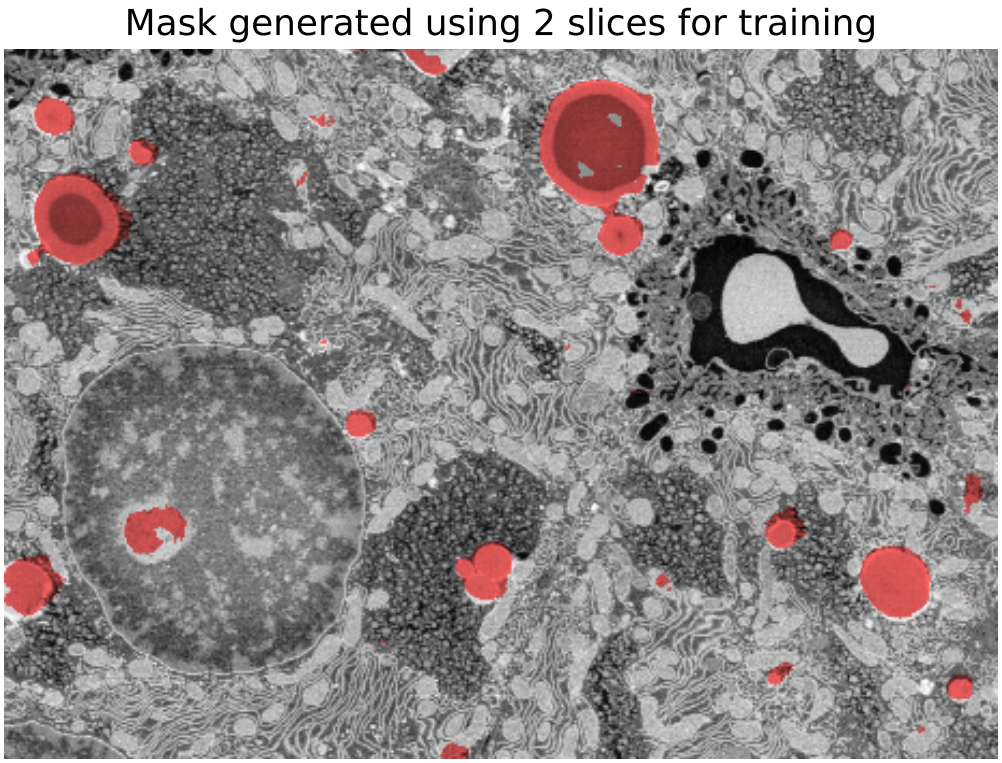


Figure 2.3: Predicted segmentaion mask for part of slice 45

In this thesis, we are more interested in observing variation in accuracy with the increase in amount of training data. Thus, we tested OSVOS by fine-tuning it using different number of slices, staring from 1 slice and increasing to 10 slices. For our experiment, set of slice with number {1,7,10,18,24,52,62,72,80,88}

were used for training. While training, the batch size was always 1. In case of multiple slices i.e. more than 1 slice for training, each slice was fed as separate input to train the network. The set of slices chosen was in ascending order of their number i.e. for training with 3 slices, slice no. {1,7,10} were used and for training with 4 slices, slice no. {1,7,10,18} were used. These slices were fed to network as an independent image in random order. Once, the parent network was fine-tuned, the test network predicted results for test set. We were not able to test the network on whole 3D volume due to lack of availability of ground truth segmentation mask for whole volume. The test set consisted of part of image from slice number 15, 30 and 45. We had multiple masks annotated by experts for these slices. Even for a single slice, for example 15, we had multiple mask differing from each other annotated by same expert. We generated a reference mask using STAPLE algorithm and computed segmentation accuracy for test slices using these refernce masks. The result of the change in traing data can be seen in figure 2.4.

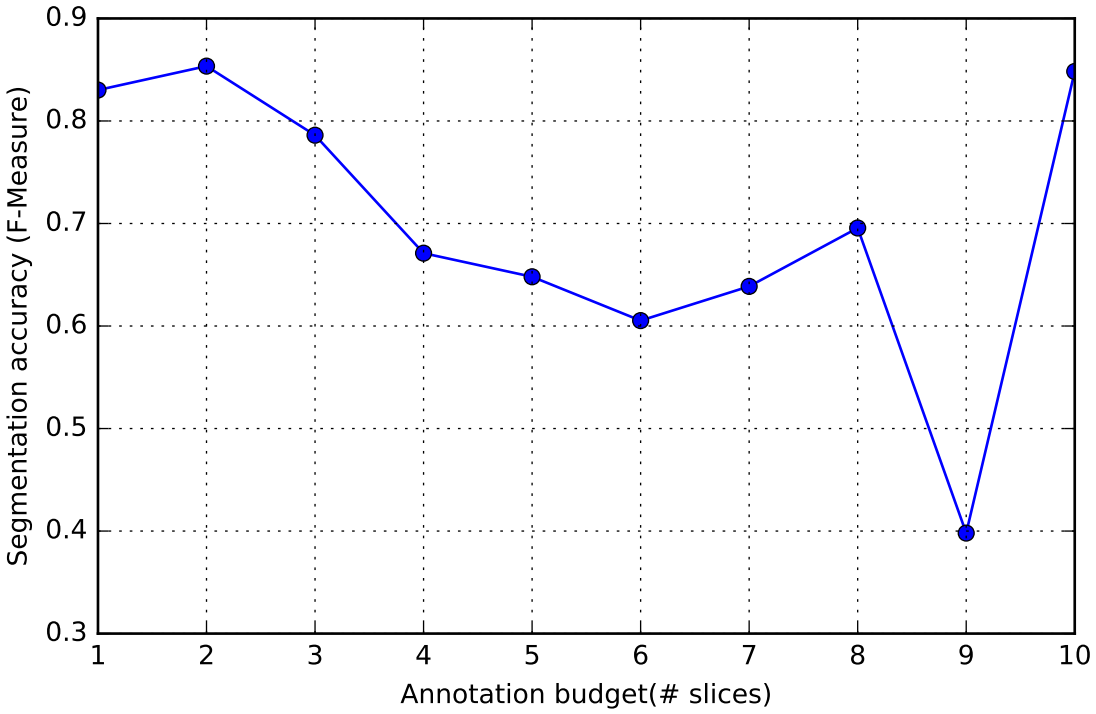


Figure 2.4: Segmentation accuracy for different amount of training data (Number of slices)

We expected an increase in performance with the increase in the amount of training data. This does not happen as CNN is not able to converge equivalently for all cases. We see a significant drop after addition of 9^{th} slice, i.e. slice number 80. On analysis, we realized that the training with each slice independently in

each forward pass is not allowing the network converge easily with increase in number of slices. Instead, the better way would be to use all available slices in a batch to train the network. Although, we were able to get a significant accuracy with such few annotated slices but it is important to remember here that annotating one slice is not same as annotating one object in a video. We can observe that a single slice contains multiple objects, as shown in figure 2.3. Also, if the experts make few changes in annotated masks used for training, this will force us to train the network again. These difficulties motivated us to try semi-supervised learning using RF for our task of segmentation and try to achieve similar accuracy.

Chapter 3

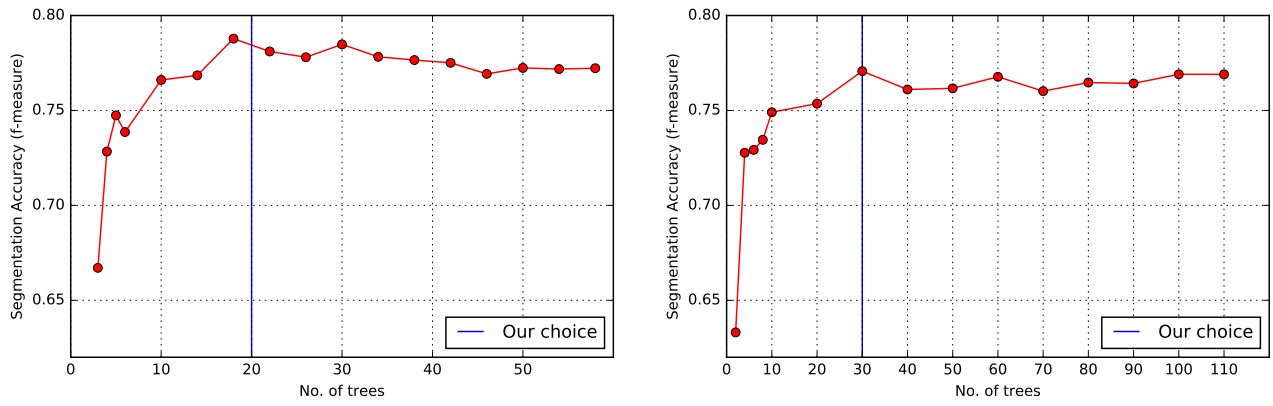
Semi-supervised image segmentation using RFs in a Bayesian framework

The philosophy behind semi-supervised learning is to propagate the label information from labeled to unlabeled data. Image segmentation can be seen as a classification problem which consists of assigning a class label to each pixel. For our task of binary segmentation, this means classifying each pixel as foreground or background. For our task of image segmentation, we make use of the partial annotations as *scribbles*. Scribbles are pixels in the image annotated by experts as foreground or background. In this chapter, we use the Random forests for our segmentation task. We train the Random forests using scribbles and try to achieve similar accuracy as we obtained with CNNs in previous chapter. One major contribution of this chapter is analysis of effect of scribbles on accuracy. We designed few experiments to train RF with varying quality and quantity of scribbles, and observed the effect of these variations on segmentation accuracy. In addition, we improve the accuracy of RF by use of variational methods.

3.1 Random Forest

We can find a lot of research using RF for the binary image segmentation. Eugster [4] explains the use of RF for the segmentation of microscopic images. This chapter is an extension of methods used by Eugster. For training RF, we compute set of features in Python. We compute different features ranging from simple

Sobel edge detectors to higher level Gabor filters. The choice of features was made according to WEKA [12] toolset of FIJI [13] plugin. These are set of 2D features and perform well for microscopic images. We compute different type of features for a range of sigmas, which gives 69 feature maps for a single image. The use of all 69 feature maps increases computational cost significantly, and also increases the training and testing time of RF. In the thesis by Eugster [4], we can find details of feature selection to choose N best features and use these N featrues to obtain an optimal balance between computational cost and accuracy. Similar to Eugster, we choose N best features and significant number of trees such that increasing the features or trees does not change the results significantly. This can be seen in figure 3.1. We can observe that for a given annotation budget i.e. fixed amount of scribbles, the segmentation measure does not change significantly for more than 30 trees and for more than 20 features. Therefore, for all experiments using RF, we use 20 best features and 30 trees for training. In following sections, we focus on how to get best results for given annotation budget and thus, use the fixed number of features and trees to generate segmentation masks.



(a) (b)

Figure 3.1: Plot of segmentation measure vs changing complexity of RF: (a) with features, (b) with trees

3.2 Best use of annotaion budget

Earlier research for segmenting images in classical computer vision problems has mostly been focused on improving the segmentation accuracy or improving the performance speed. This can be understood as the annotating the ground truth is not as difficult and time consuming as it is in microscopic images. For our

scenario, the use of annotation budget to produce best results is of utter most importance. Due to these reasons, we try to observe the effect of annotaion budget on segmentaion accuracy. We try to observe this by varying quality and quantity of scibbles. The quality of scribbles is related to the position of scribbles or where to scribble i.e. the scribble can be near to boundary of object or in the inner body of the object. The quantity of scribbles correspond to amount of pixels annotated. Thus, we try to obtain optimal quality and quantity of scribble needed in order to obtain a significant level performance.

3.2.1 Where and how much to scribble?

In general, we believe that more the training data we provide, more we can improve the results. Does this hold for partial annotation such as scribbles? If we go on increasing the pixels annotated arbitrarily, will it improve the segmentation mask or we have to use our labeling effort intelligently to improve results?

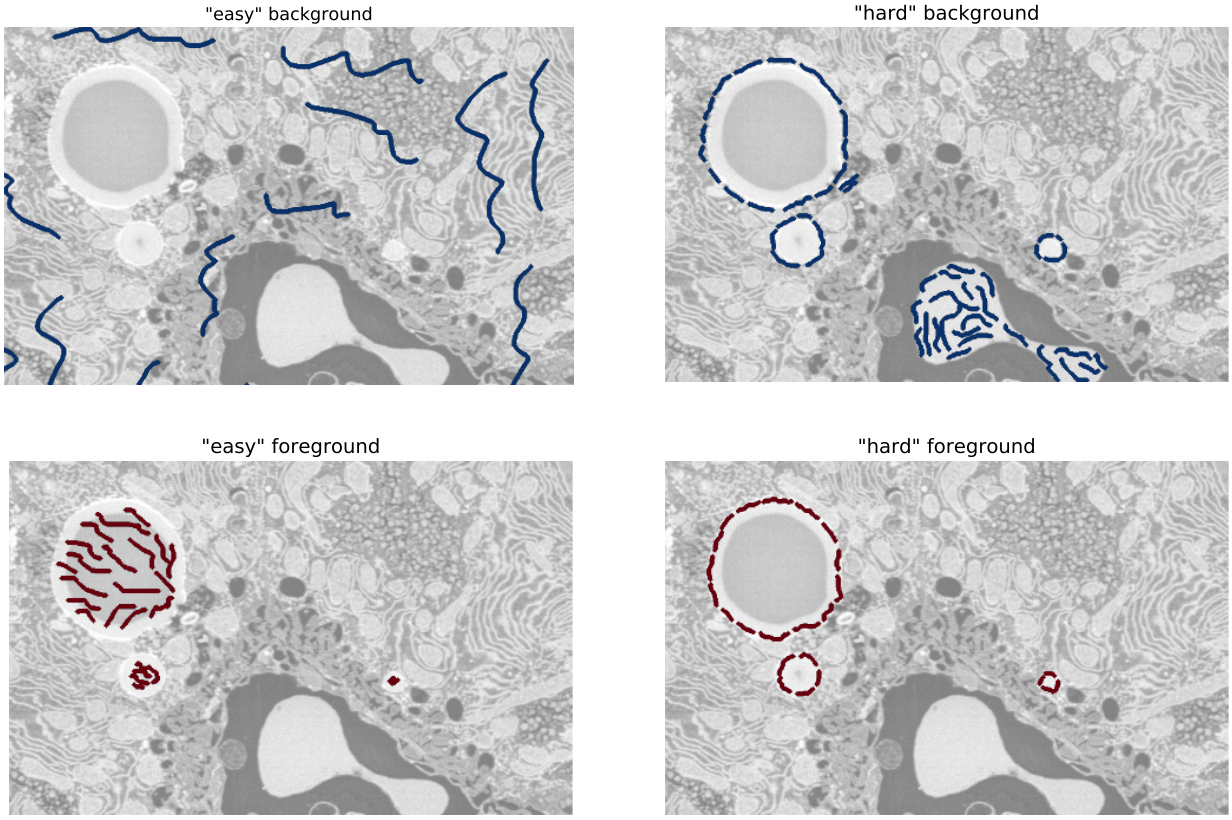


Figure 3.2: Manual scribbles in ”easy” and ”hard” annotation classes

We designed an experiment by dividing our set of foreground and background scribbles into 2 annotation

classes: easy and hard scribbles. We classified the scribbles as ”easy” and ”hard” depending on the effort required to annotate these pixels. For example, the pixels are difficult to annotate near the boundary of foreground and background, and we classify these pixels as ”hard”, as shown in figure 3.2. We manually scribbled image for both ”easy” and ”hard” annotation subclasses. Then, we trained and tested RF on one image by increasing percentage of scribbles belonging to ”easy” foreground and background class. After we have used all scribbles belonging to ”easy” annotation class, we added scribbles from ”hard” annotaion class for both foreground and background. The increment was made in percentage w.r.t. the total amount of scribbles we have. At each step of increment, only new scribbles were chosen randomly on the top of already chosen pixels. Also, as we increased the amount of scribbles we tried to maintain a ratio between foreground and background pixels. The ratio was choosen proportional to ratio of total foreground and background pixels. To measure the segmented accuracy, we comapre the segmentation mask for the same slice, on which the pixels were annotated for training RF. For measuring segmentation accuracy, the output probability mask from RF was thresholded using a value of 0.5.

The result can be observed in figure 3.3. In the figure, we can observe that after a total of 3000 pixels (10% of ”easy” scribbles) selected from ”easy” foreground and background, the segmentation accuracy does not change significantly. The segmentation accuracy varies between 0.73 to 0.76. This implies that further addition of ”easy” scribbles do not provide any extra information to RF. A boost in accuracy can be observed, once we add ”hard” scribbles after all available ”easy” scribbles were used. We can observe that we are able to achieve accuracy of 0.84 by a small annotation budget of around 25000 pixels in an image of size This shows that the best results can be obtained by adding ”hard” scribbles over a certain fixed percentage of ”easy” scribbles. Looking at the plot and keeping the insignificant variation in accuracy after 3000 pixels, it feels best to add the pixels from ”hard” scribbles after initial amount of 3000 ”easy” scribbles. We except a similar boost in accuracy and makes best use of our annotaion budget. We tried this and results can be observed in figure 3.4.

In figure 3.4, the addition of ”hard” scribbles on top of 0.15% (3000 pixels) of initial ”easy” scribbles did not produce a boost as we expected. Instead of observing a continuous boost with addition of ”hard” scribbles, we observed a fall in performance (dark blue plot in figure 3.4). This may be due to lack of enough ”easy” scribbles and RF starts training its trees to focus more on ”hard” scribbles. We repeated

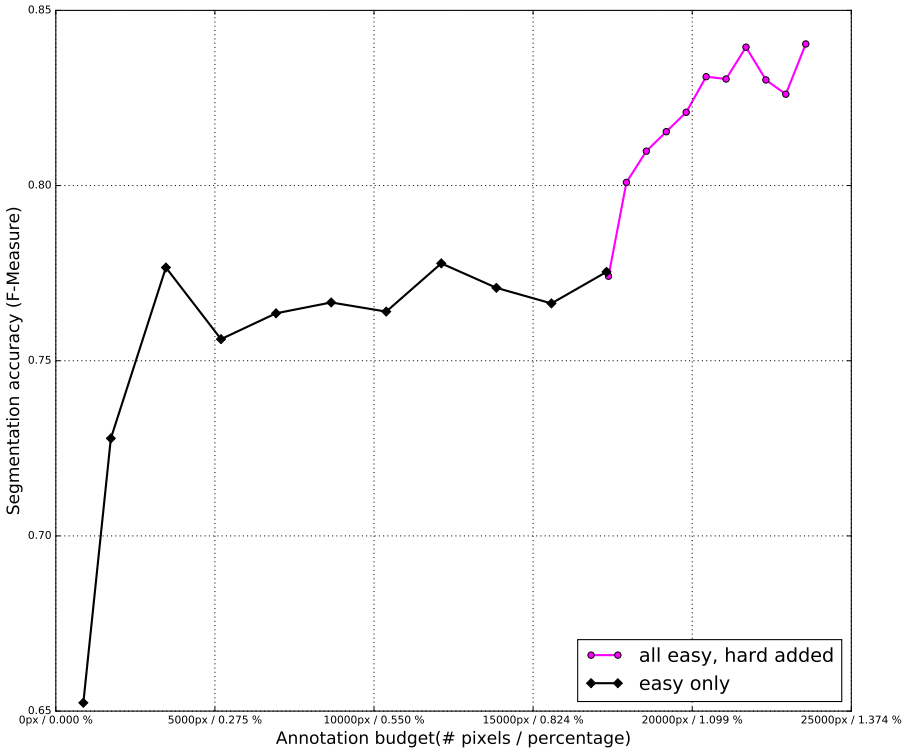


Figure 3.3: Plot of segmentation measure vs annotation budget. The black curve shows the increment of scribbles from the ”easy” annotaion class. The pink curve shows the addition of scribbles from the ”hard” annotaion class. The percentage in x-axix corresponds to amount of pixels w.r.t. to total pixels in image.

the addition of ”hard” scribbles for different fixed amount of initial ”easy” scribbles. We started seeing a significant improvement similar to earlier boost, when we utilized with 70% of all ”easy” scribbles to train RF (green plot in figure 3.4). We were able to achieve best f-measure score of 0.83 in comparison of 0.84 achieved with 100% usage of ”easy” scribbles (See green and pink plot in figure 3.4). The accuracy of 0.83 was achieved with use of only 1% of total pixels in image. This was possible only because of quality of scribbles used. We can not achieve this accuracy if we annotate same amount of scribbles arbitatrily. Thus, the question arises how to decide the point of addition of ”hard” scribbles and amount of ”hard” scribbles to be added. The other observation is that continuous addition of ”hard” scribbles do not always improve the results. We require an optimal balance between ”easy” and ”hard” scribbles. It is difficult to decide the position and quantity of scribbles prior to training RFs.

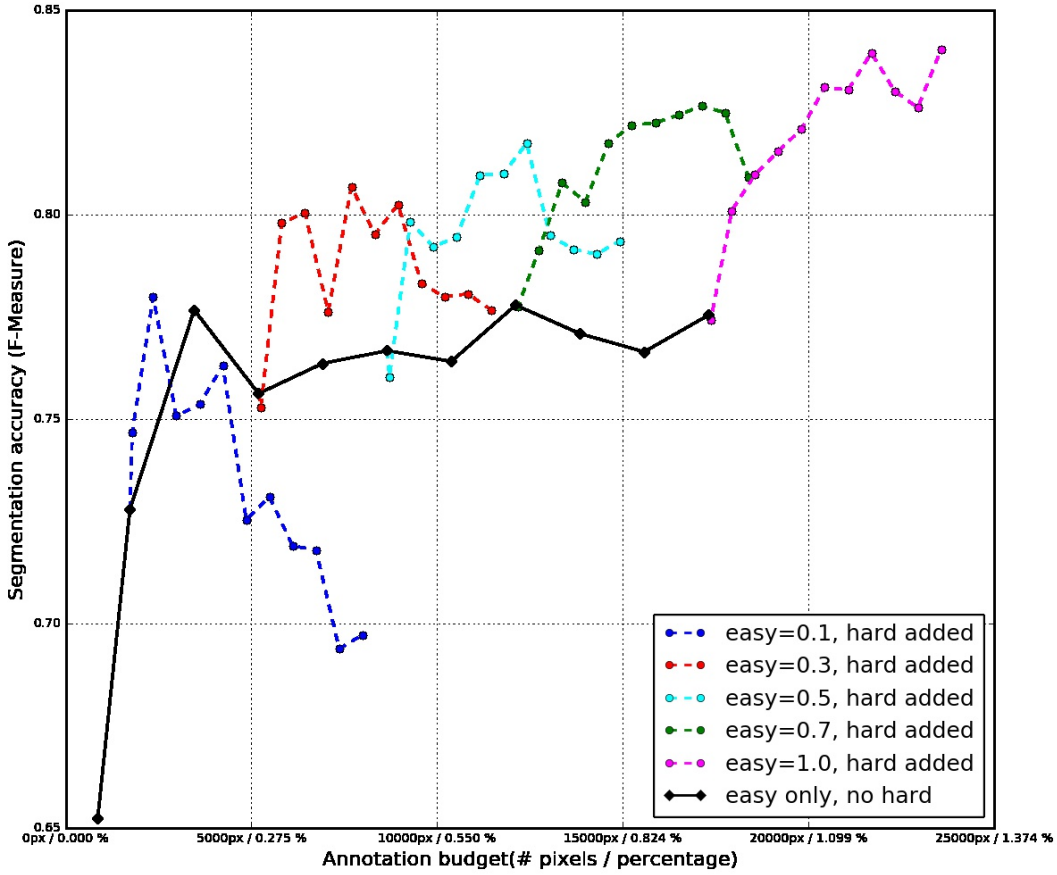


Figure 3.4: Plot of segmentation measure vs annotation budget. Different curves show addition of scribbles from ”hard” annotaion class, starting with different fixed amount of ”easy” annotation class.

3.2.2 Iterative semi-interactive approach

We observed the need of using our annotation budget intelligently to get the best performance out of it. But, we observed the problem of deciding on how many ”easy” and ”hard” scribbles are needed to achieve best results. For our problem, we divided the scribbles as ”easy” and ”hard” according to labeling effort, but this division for scribbles may not be same from point of view of the RFs. Apriori, we don't know which pixels will be difficult for the Random forest to classify correctly. The above mentioned two problems can be solved by annotating pixels iteratively to improve results. We can iterate the annotation at least once, to understand which pixels are difficult for RF to classify. We show the improvement in the result by doing one iteration in figure 3.5. We can observe that the scribbles are very few to produce a good result. Still, we can observe improvement in f-measure from 0.76 to 0.80 for increasing the annotation budget from 7500

pixels to 10900. Although the increment looks small, but we can observe the improvement at boundary in the large vesicle in left-top corner of the image. Thus, we can say that we can get a significant improvement with few iterations, even if we add small amount of pixels every time.

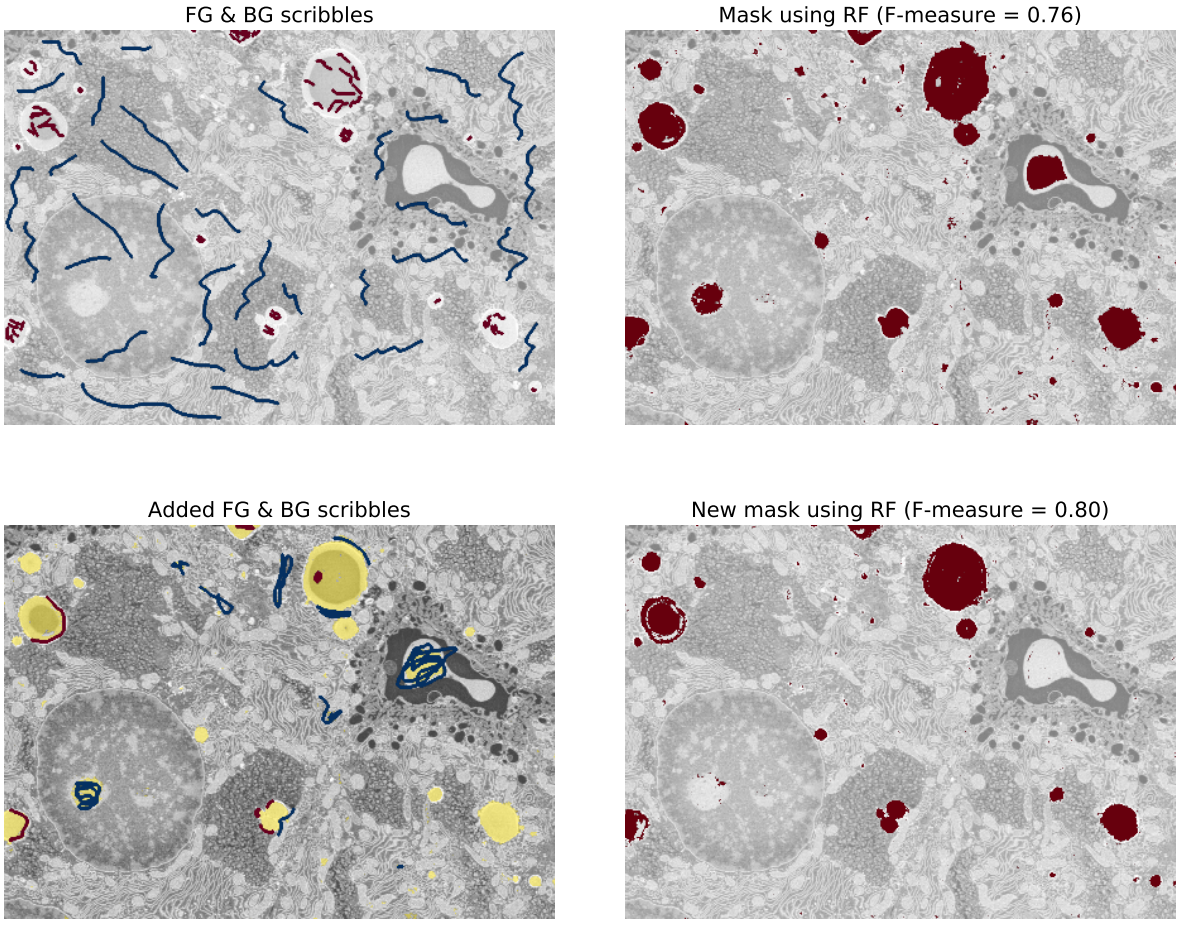
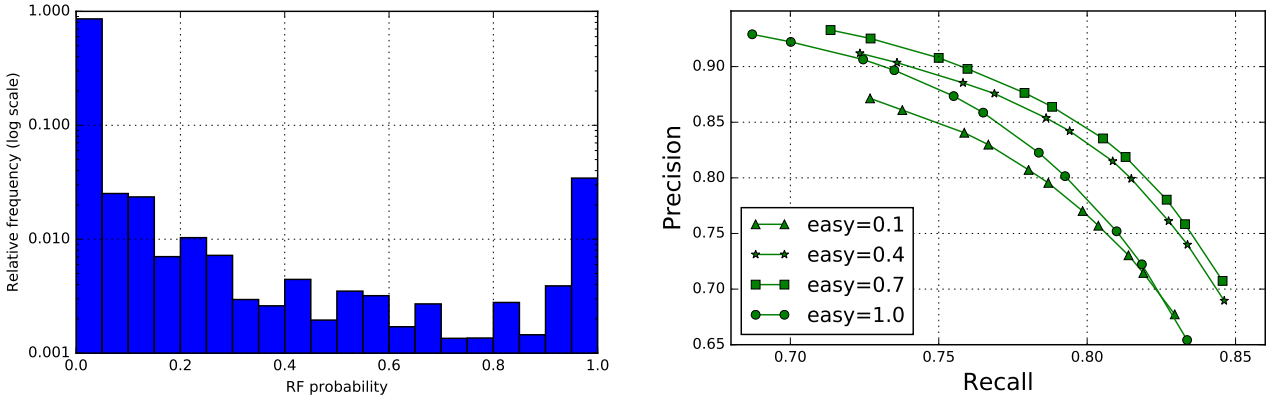


Figure 3.5: Semi-interactive segmentaion with one interaction. The bottom-left image shows scribbles added after first interaction using previously generated mask. Red: foregroud, Blue: background

3.2.3 Limitation of RF

The use of iterative semi-interactivity gives the best result for given annotation budget, but there is a limit on accuracy that can be achieved by using only RFs. The output mask of RF is noisy and uncertain. The uncertainty lies in the inability to classify maximum of pixels as foreground and background, as shown in figure 3.6(a). The histogram shows the distribution of probability of pixel to be foreground values for the

complete image. It can be seen that a large number of pixels are not given a probability of 0 (background) or 1 (foreground). In figure 3.6(b), we can observe varying results for different threshold applied on probability mask obtained from RF. RF acts as a classifier and classifies each pixel but we need to group these pixels into objects for segmentation.



(a) (b)

Figure 3.6: (a) Histogram of predicted foreground probabilities using RF. (b) Precision-recall curve for varying thresholded RF mask. Different curves correspond to different annotation budget

3.3 Bayesian Formulation of segmentation problem

In this thesis, we make use of prior information to compensate for the lack of sufficient training data and for the uncertainty of RF classifier. To make use of prior, we model our image segmentation problem as a Bayesian inference problem. Let us cosider an observed image, $\boldsymbol{I}$ and labeled or segmented ground truth, $\boldsymbol{M}$, the joint probabilty can be defined as:

$$\boldsymbol{p}(\boldsymbol{I}, \boldsymbol{M}) = \boldsymbol{p}(\boldsymbol{M})\boldsymbol{p}(\boldsymbol{I}|\boldsymbol{M}) \quad ,$$

and applying Bayes theorem,

$$\begin{aligned} \boldsymbol{p}(\boldsymbol{M}|\boldsymbol{I}) &= \frac{\boldsymbol{p}(\boldsymbol{M})\boldsymbol{p}(\boldsymbol{I}|\boldsymbol{M})}{\boldsymbol{p}(\boldsymbol{I})} \\ &\propto \boldsymbol{p}(\boldsymbol{M})\boldsymbol{p}(\boldsymbol{I}|\boldsymbol{M}) \end{aligned}$$

The left hand side is the probability of obtaining segmentation mask, $\boldsymbol{M}$ given the image $\boldsymbol{I}$, is called the posterior probability. $\boldsymbol{p}(\boldsymbol{M})$ is the prior probability of mask, $\boldsymbol{M}$. The Maximum a posteriori (MAP) estimate, $\boldsymbol{M^*}$ can be calculated as follow:

$$\boldsymbol{M^*} = \arg\max_{\boldsymbol{M}} \boldsymbol{p}(\boldsymbol{M})\boldsymbol{p}(\boldsymbol{I}|\boldsymbol{M}) \quad . \tag{3.1}$$

The above problem can as well be stated as an energy minimization problem by writing Equation 3.1 in terms of energy by taking negative log-likelihood:

$$\begin{aligned} E(\boldsymbol{M}) = & -\log(\boldsymbol{p}(\boldsymbol{I}, \boldsymbol{M})) \\ & = -\log(\boldsymbol{p}(\boldsymbol{I}|\boldsymbol{M})) - \log(\boldsymbol{p}(\boldsymbol{M})) \\ & = E_d(\boldsymbol{I}, \boldsymbol{M}) + E_r(\boldsymbol{M}) \end{aligned}$$

The total energy, E, that we want to minimize can be considered as linear combination of data or likelihood term, E_d, and prior term or regularization, E_r. This modifies calculating MAP estimate to:

$$\boldsymbol{M^*} = \arg\min_{\boldsymbol{M}} E_d(\boldsymbol{I}, \boldsymbol{M}) + E_r(\boldsymbol{M}) \quad .$$

To obtain MAP estimate, we need to formulate likelihood term and prior term. We formulate the prior using Total variation(TV). We can find use of different TV priors such as Wulff shapes etc. In our thesis, as the objects we need to segment are smooth and shaped like a circle, we make use of isotropic total variation, TV. Also, we can try to use isotropic total variation in 2D or 3D as the data we are trying to segment is a 3D stack. For likelihood term, G. Paul et al.[14] proposed an energy formulation which is not derived from a statistical model but learnt from training set. This gives the advantage of combining example-based and model-based approaches. Similar to Eugster [4], we formulate the likelihood term using a cost function, C. The likelihood term is formulated as a product of cost function and mask. The cost function assigns a cost depending on predicted foreground probabilities obtained from RF. Let p be the probability of pixel being foreground (learnt from RF), and $\boldsymbol{M}$ be the optimal mask to be estimated, the energy minimization problems becomes:

$$\begin{aligned} E(\boldsymbol{M}) &= -\log(p(\boldsymbol{I},\boldsymbol{M})) = -\log(p(\boldsymbol{I}|\boldsymbol{M})) - \log(p(\boldsymbol{M})) \\ &= E_d(\boldsymbol{I},\boldsymbol{M}) + E_r(\boldsymbol{M}) \\ &= \langle \mathcal{C}(p), \boldsymbol{M} \rangle + \lambda\, TV(\boldsymbol{M}) + \iota_{[0,1]}(\boldsymbol{M}) \end{aligned} \tag{3.2}$$

where $\mathrm{i}_{[0,1]}(\boldsymbol{M})$ is an indicator function to ensure values of $\boldsymbol{M}$ remain in [0,1]. In addition to use of cost function as explained in Eugster [4], we enforce a constraint in our energy minimization problem to obtain correct mask values for the pixels annotated by experts as foreground and background. This constraint was not enforced by Eugster [4]. We used an indicator function, $\mathrm{i}_{\mathcal{FG}}(\boldsymbol{M})$, to ensure pixels in foreground scribbles have value of 1 in mask, and indicator function, $\mathrm{i}_{\mathcal{BG}}(\boldsymbol{M})$, to ensure pixels in background scribbles have value of 0 in mask. Using discrete implementaion of TV, the final energy minimization problem becomes:

$$E(\boldsymbol{M}) = \langle \mathcal{C}(p), \boldsymbol{M} \rangle + \lambda\, TV(\boldsymbol{M}) + \iota_{[0,1]}(\boldsymbol{M}) + \iota_{\mathcal{FG}}(\boldsymbol{M}) + \iota_{\mathcal{BG}}(\boldsymbol{M}) \tag{3.3}$$

The optimization problem is solved using Alternating Split Bregman method (ASB), as described in Eugster [4]. The advantage of using ASB is that it splits the above problem into subproblems. Each subproblem is easy to solve and can be solved independently. Due to this reason, the new constraints for scribbles can be easily coupled with one of the subproblem or can be solved separately. For our formulation, we are able to couple the new constraints with constraint on mask values. The final solution to the problem is obtained by iterating updates. The details of implementation and solution can be obtained from Eugster [4].

3.3.1 Anti-log likelihood cost function

The cost function has to be defined such that it assigns a positive cost to probabilities less than 0.5 and negative cost to probabilities greater than 0.5. This ensures that the pixels with low probability get a mask value of 0 and reverse for pixels with high probability. We make use of use of *anti-log likelihood* cost function explained by Eugster [4] and conduct different experiments. The *anti-log likelihood* cost function

is defined as:

$$\mathcal{C}(\widehat{p}(x_k)) = \begin{cases} 0, & \text{if } x_k \in \mathcal{S} \\ -\log \frac{\widehat{p}(x_k)}{1-\widehat{p}(x_k)}, & \text{else} \end{cases},$$

where $\mathcal{S}$ corresponds to set of annotated pixels and $\widehat{p}(x_k)$ is the probability learnt from RF at pixel, x_k. The plot of *anti-log likelihood* cost function can be seen in figure 3.7. We can observe that cost tends to positive infinity for probablity values close to 0 and, to negative infinity for probablity close to 1. The pixels with values exactly equal to 0 and 1 are given cost of zero and added to $\iota_{\mathcal{FG}}(\boldsymbol{M})$ and $\iota_{\mathcal{BG}}(\boldsymbol{M})$ respectively for implementation purpose.

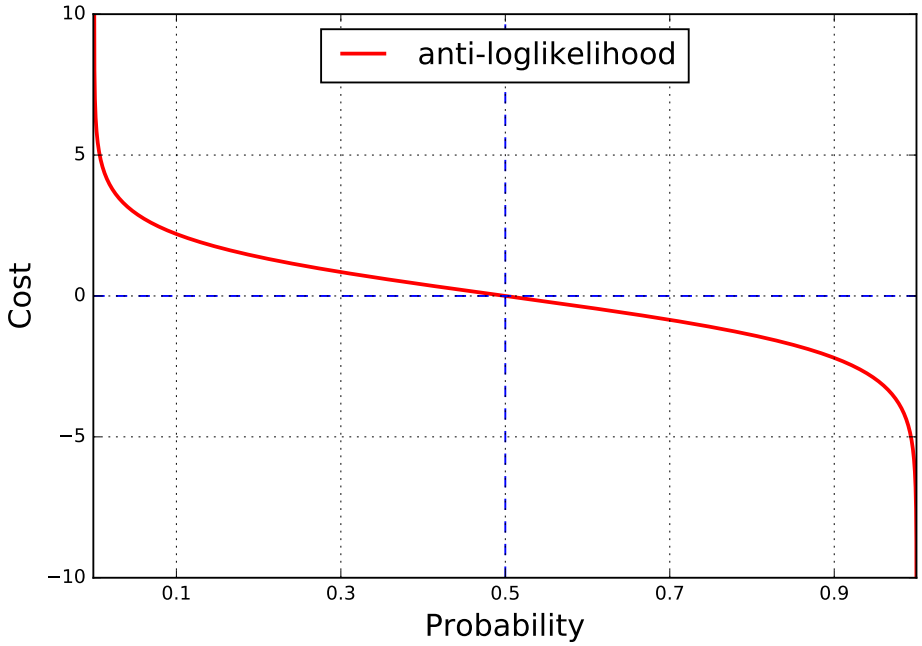


Figure 3.7: Anti-log likelihood cost function

To observe the effect of using prior, we repeated the previous experiment of addition of "easy" scribbles and "hard" scribbles in steps. But instead of measuring segmentation accuracy on thresholded RF output mask, we used the predicted RF probabilities to obtain a cost function and obtain the MAP estimate solving equation 3.2. The experiment was conducted for a range of regularization parameter (λ).

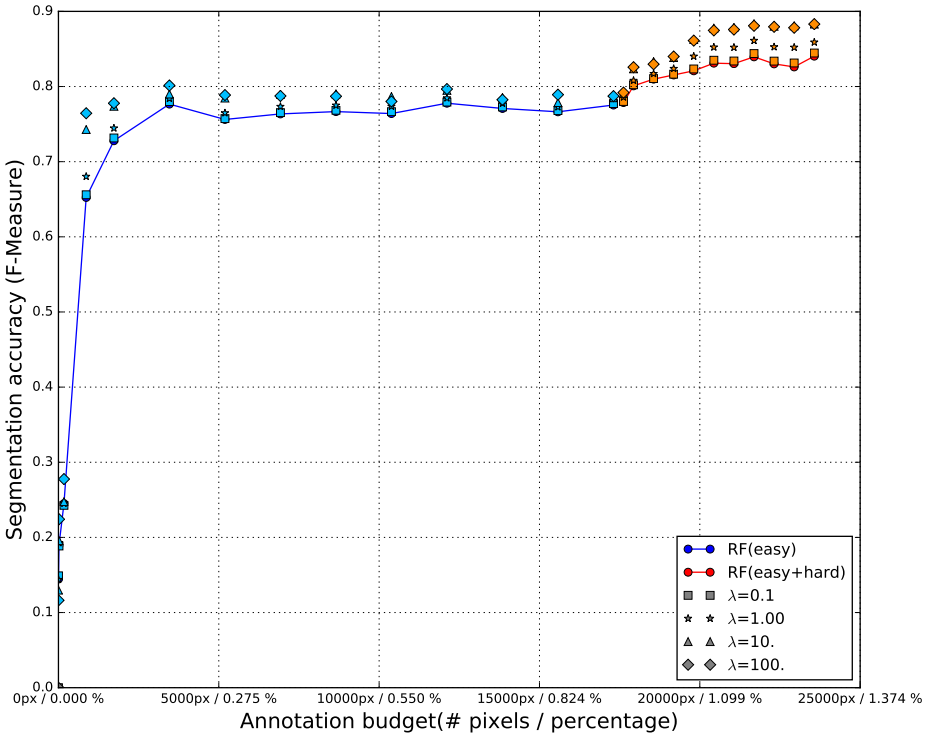


Figure 3.8: Segmentation score with RF and prior(TV) for different annotation budget

The results for RF with variational image processing (VIP) can be seen in figure 3.8. We can observe an improvement due to the use of the variational method. Even for very small annotation budget as 0.1%, we observe an improvement with the use of VIP. Figure 3.9 shows the improvement more clearly for different values of λ. We can observe that the value of λ giving maximum improvement is different for different annotation budgets. The use of prior information boosts up the performance but we get different boost for different values of λ. The regularisation parameter (λ) decides the weight of TV prior term. Thus, it becomes important to choose an appropriate value of λ to obtain best results. We generate results for 2 extreme values of λ (0.001 and 100) and try to visualize their effect in figure 3.10. For λ=0.001, the value of λ is too small and the prior term in getting MAP estimate, tend to become insignificant. We can observe that for such a small value of λ, the VIP is unable to remove noi se from mask obtained from RF and does not smoothen the boundaries (bottom-left image in figure 3.10). While, choosing a large value of λ (100) generates smooth boundaries and removes noise, but also deteriorates the mask for small vesicles (bottom-right image in figure 3.10). The small vesicles around the large vesicle tend to diminish with large value of

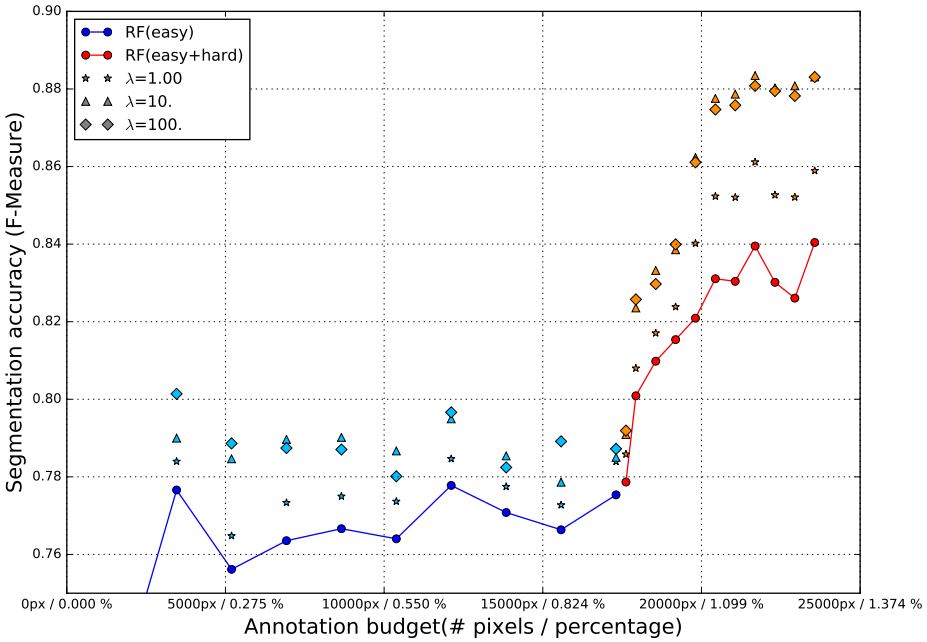


Figure 3.9: Segmentation score with RF and prior(TV) for different annotation budget (removing very annotation budgets)

λ. This points to the fact that there may not be a single appropriate value of λ for the whole image. Instead, it will be ideal to choose different λ for different part of images or to combine results for different λ.

3.3.2 Comparing different cost functions

In literature, people have used different cost functions to formulate likelihood term. Santner [3] makes use a linear cost function to formulate likelihood. The exact functional form in terms of probability is not given in [3]. Therefore, for our purpose of comparing results, we choose the *linear* cost function as given below:

$$C_l(\boldsymbol{P}) = -4\,(\boldsymbol{p} - 0.5) \quad .$$

The above function is chosen to show a behaviour similar to *anti-log likelihood* cost function at probablity of 0.5. The slope of 4 is selected to match the slope of *anti-log likelihood* cost function at 0.5. In addition to the cost function, Santner [3] mentioned use of hard constraint for pixels in foreground or background scribbles i.e. using cost of $-\infty$ for foreground scribbles and ∞ for background scribbles. They didn't show a way to enforce this constraint with *linear* cost function. We enforce this constraint with use of indicator

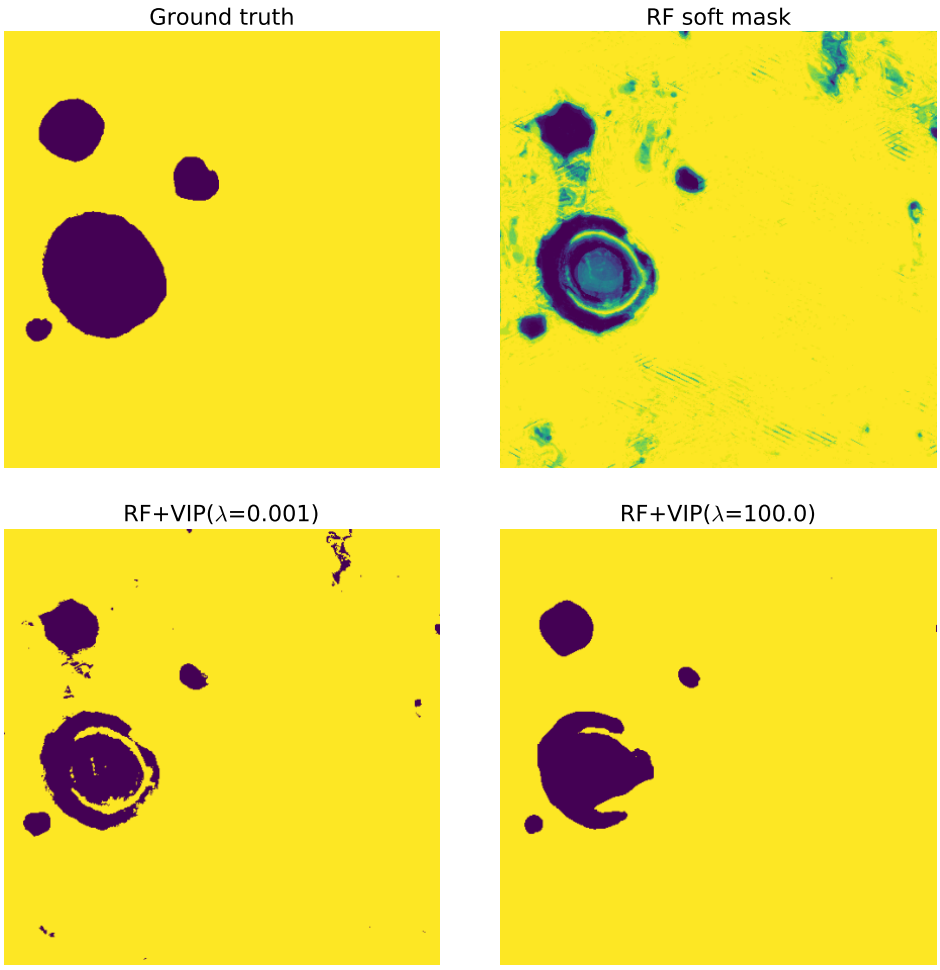


Figure 3.10: Segmentation mask for $\lambda = 0.001, 100$ for a small part image

functions, $\iota_{\mathcal{FG}}(\boldsymbol{M})$ and $\iota_{\mathcal{BG}}(\boldsymbol{M})$.

To compare results of these different cost functions, we designed a similar experiment to observe effect of annotation budget on their usage. Instead of choosing pixels from scribbles, we select the annotated pixels randomly from fully annotated ground truth segmentation mask. We increased the annotation budget from using few pixels to using all pixels for training RF. We generate results to compare 3 cost functions: *anti-log likelihood*, *linear* and *linear* (with constraints). The results for these different functions can be observed in figure 3.11, 3.12 and 3.13. The blue dotted line in all these figures shows the accuracy obtained for thresholded RF output mask.

Figure 3.11 shows that for all values of λ, the *anti-log likelihood* cost function does not decrease the

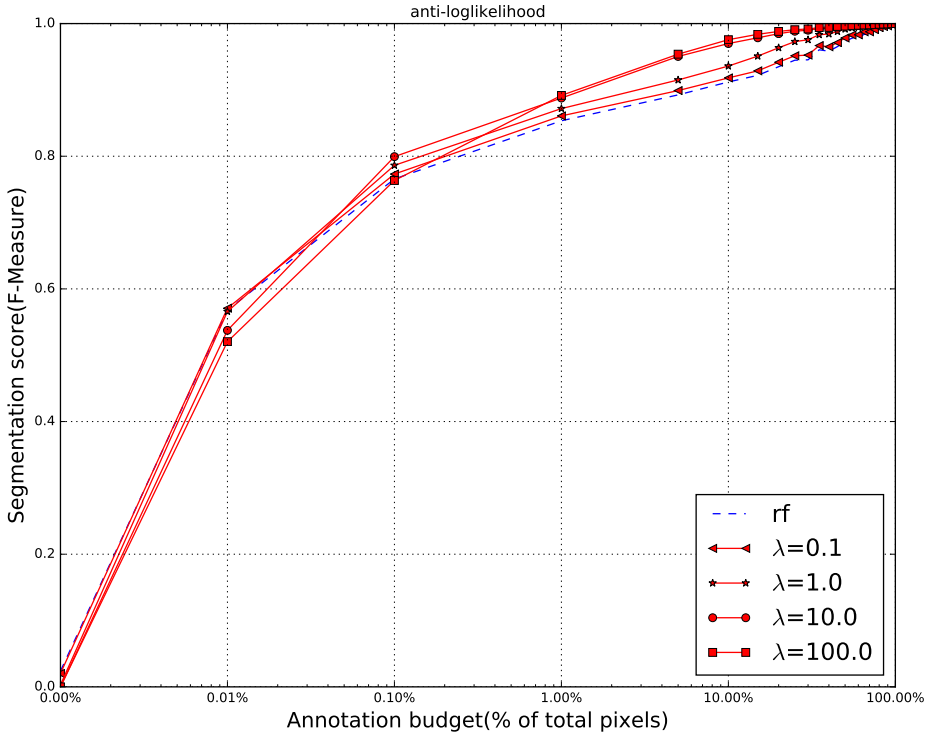


Figure 3.11: Segmentation accuracy vs annotation budget for *anti-log likelihood* cost function.

segmentation accuracy. Only for very low annotation budget (¡ 0.1%), the accuracy decreases by a small amount. Else, the use of *anti-log likelihood* cost function ensures that the use of VIP does not decrease the accuracy and thus, always improving the output mask obtained from RF. Especially for the higher annotation budgets (¿ 50%), the use of constraints does not allow VIP to change mask value for annotated pixels. This shows the robustness of cost function and allows the user to choose values of λ freely in a reasonable range.

From figure 3.12, we can observe that *linear* cost function does not work well with large values of λ and also significanty deteriorates the mask obtained from RF. Even for λ equal to 10, the accuracy drop significantly in comparison to accuracy obtained from thresholded RF mask. This shows that while using *linear* cost function, the value of λ is to be chosen carefuly.

Contrary to results for *linear* cost function, the results for the *linear* cost function with constraints in figure 3.13 is almost similar to the results for *anti-log likelihood* cost function. The use of constraints makes it robust and avoids corruption of the mask obtained from RF. In figure 3.14, we foucus on higher values of annotation budgets. It can be seen that the segmentation accuracy for all cost functions converge to 1 with

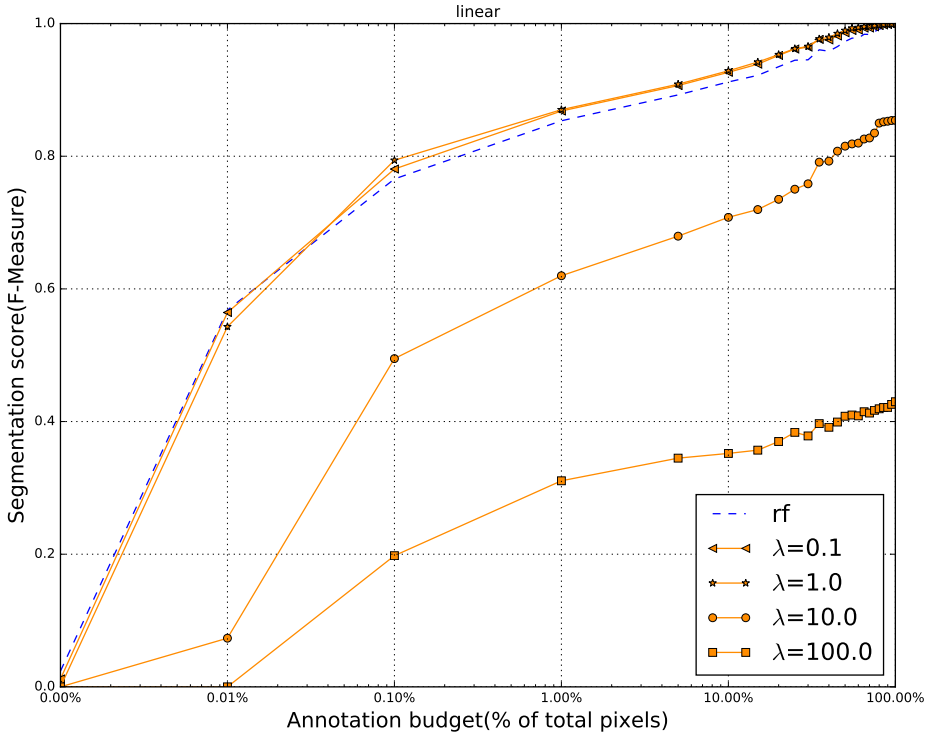


Figure 3.12: Segmentation accuracy vs annotation budget for *linear* cost function.

the increment of annotation budget to 100% i.e. using all pixels in the image for training. The difference can be seen in convergence for the 3 different cost functions. These cost functions shows different amount of improvement for same annotation budget. The effect of the same value of λ is also different for these cost functions. It can be seen that *anti-log likelihood* and *linear* (with constraints) show maximum and almost same improvement for λ equal to 100.

3.3.3 Semi-interactive segmentation using VIP

We realized the need for iterative semi-interactive segmentation in section 3.2.2. With 1 iteration of interactive annotation, we were able to improve f-measure results significantly. Now, we combine semi-interactive annotation with the use of variational segmentation. We take segmentation mask obtained from VIP and add scribbles manually to image where the RF and VIP together are unable to predict correctly. Then, we retrain Random forest and generate new mask using variational segmentation. The image, mask, and scribbles are shown in figure 3.15. The f-measure improves from 0.79 to 0.86 with an addition of only 3000 pixels (less

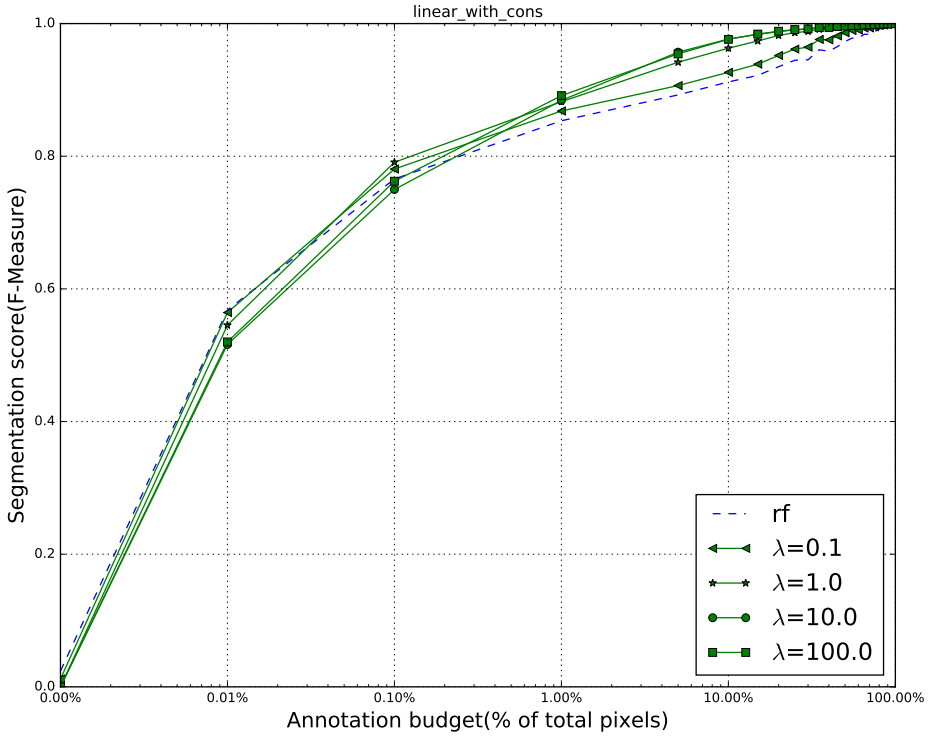


Figure 3.13: Segmentation accuracy vs annotation budget for *linear* cost function with constraints.

than 0.17% of pixels in the image). In comparison to figure 3.5, where we add scribbles on thresholded output from RF, we can observe that the noise is completely removed by VIP. Also, the use of VIP smoothens the boundary of vesicles. This enables us to use our annotation budget more effectively and trying to correct boundaries and other pixels which are difficult for RF to classify.

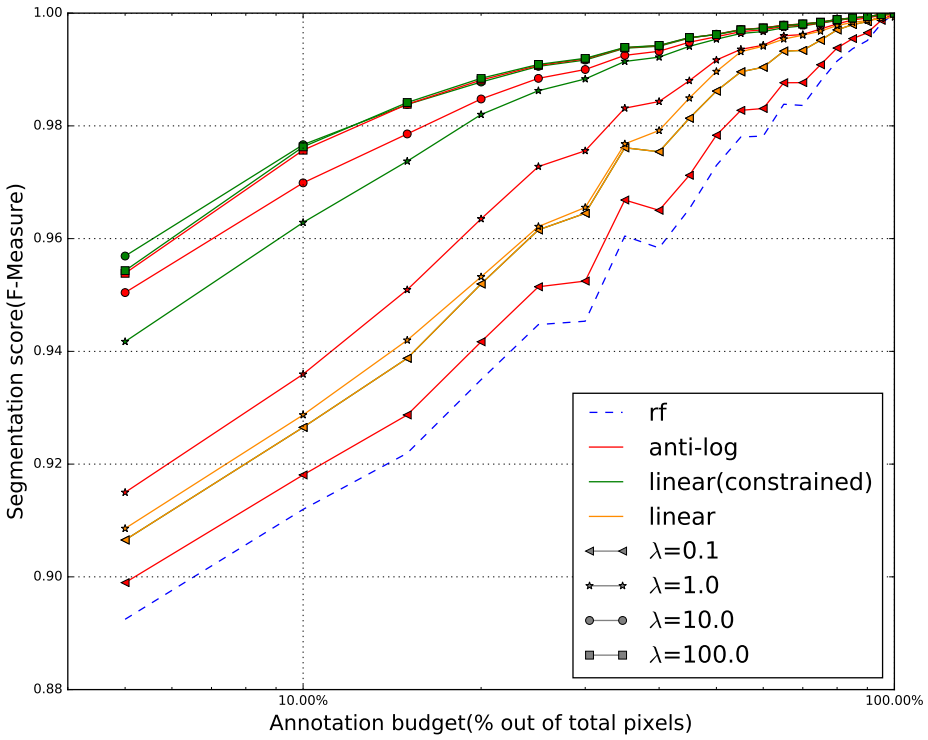


Figure 3.14: Comparison of different cost functions for higher annotation budgets.

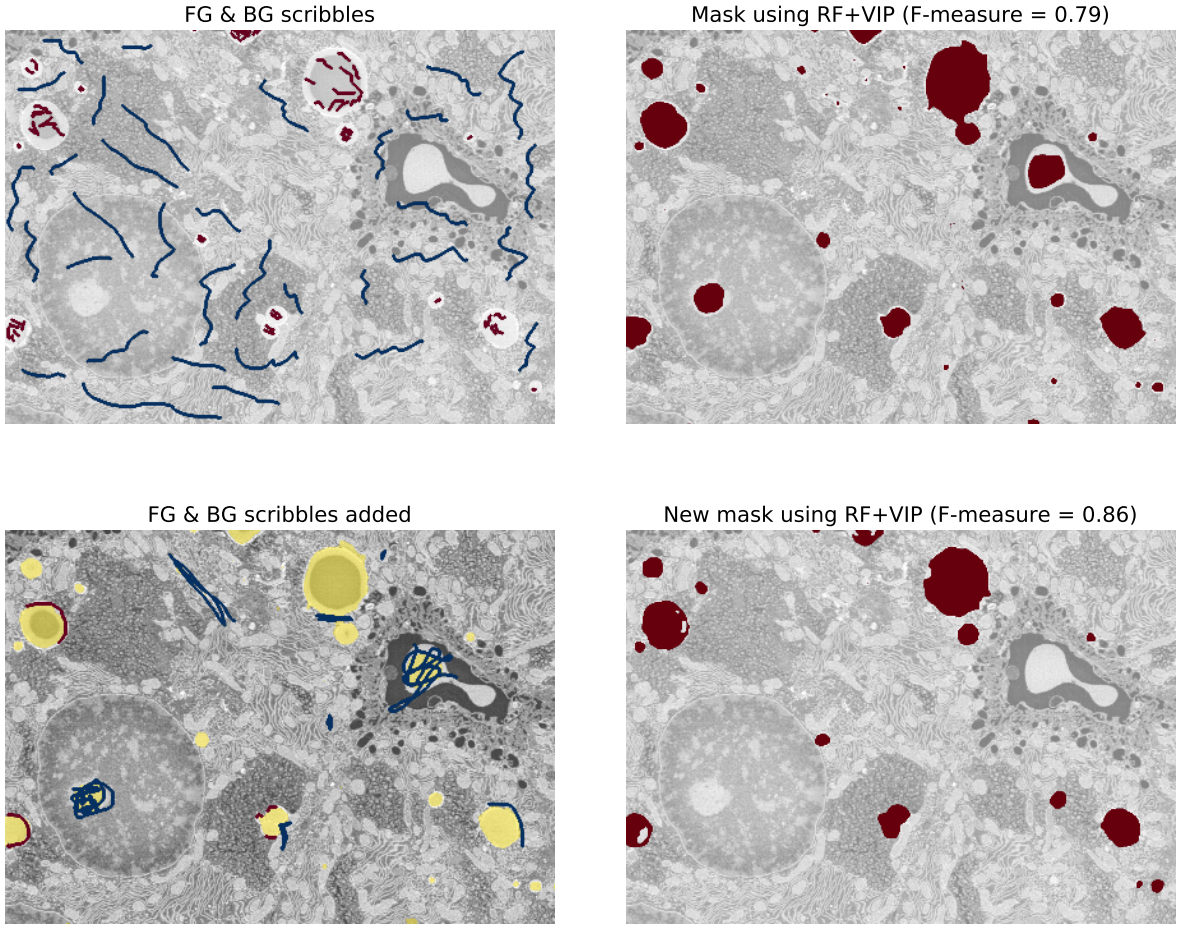


Figure 3.15: Semi-interactive segmentation with one interaction (RF+VIP). The bottom-left image shows scribbles added after first interaction using previously generated mask. Red: foregroud, Blue: background

Chapter 4

Semi-supervised image segmentation using CNNs in a Bayesian framework

In this chapter, we learn probability mask using for CNNs in place of RFs and use the output mask in Bayesian framework. Instead of RFs, we train the CNNs using scribbles and try to achieve similar accuracy as we obtained with CNNs using full annotations in chapter 2. Instead of using pixels as training samples for CNN, we modify a pre-trained network, OSVOS (chapter 2), to train it using a whole image with partial annotations. One major contribution of this chapter is comparison of performance of RFs and CNNs for different annotation budgets.

4.1 Motivation for replacing RFs

The use of RF and VIP together is able to achieve a good segmentation accuracy in a semi-supervised scenario. In previous chapter, we explained that with proper use of our annotation budget, an accuracy can be achieved which is comparable to accuracy achieved by using fully annotated masks for training. Then, why do we need to replace RFs with CNNs?

The training procedure for RFs is simple and computationally less expensive in comparison to CNNs. But the performance of RFs heavily depends on the quality of features used for training. For our segmentation task, the RF was trained with features described in WEKA toolset. These features are expected to work

well for medical images. These featured performed well for segmentation of vesicles in liver tissue but may not work well for other medical images. Therefore, the use of CNNs proves beneficial as it learns different filters from the training samples and according to the task at hand. It is known that the initial layers of a CNN learn basic image features while the final layers try to learn features specific to the problem. In addition to this, since we make use of VIP to use prior to improve our results, it has always been a challenge to find appropriate λ. In last few years, we can find a lot of research in direction of training CNN and VIP together to get best results. Ranftl [5] uses CNN with VIP together and modifies the loss function to learn optimal values of λ along with CNN parameters. The paper describes a method of combining CNN(5 layers) with a final variational/inference layer where the inference layer has activation function in form of Total variation. Similarly, Taylor et al. [15] implemented CNN as a scalable ADMM approach. They split the objective function into subproblems (as we did using ASB) and trained CNN without gradients. This approach can help us to couple CNN with VIP to gain from both methods.

4.2 Training CNN from scribbles

This motivated us to replace RF with CNN to parametrize the likelihood cost function. The usual approach to train CNN from scribbles is to use the pixels as training samples for CNN. Each pixel is feeded to network one by one and the CNN acts as a classifier to classify it as foreground or background pixel. Gonda et al.[16] uses an interactive approach to train deep neural networks for segmentation of neuronal structures using scribbles. Lai et al. [17] uses patch-based 3D image segmentation. They make use of patches around pixels annotated to train the neural network. A similar approach has been used by Havaei et al.[18] for brain tumor segmentation using deep neural networks. This approach appears to be disadvantageous for segmentation task where we are unable to use contextual information provided by full image. These approahes remove one major property of CNN i.e. to adapt its final layers according to full image for segmentation task. Due to these reasons, we wanted to make best use of research in deep learning to produce best results. Instead of designing a new network and training it from scratch, we decide to modify OSVOS (Chapter 2) to train it using scribbles. As explained in Chapter 2, OSVOS is fully-convolutional network, which predicts segmentation mask for the complete image in one forward pass. Then, it computes cross entropy loss for whole image and back-propogates this loss. This is the step which restricts us from using OSVOS as we

scribbles as partial annotations in the image.

4.2.1 Cross entropy scribble loss function

In general, when cross entropy loss is computed in CNNs over complete image, it is computed for all pixels separately and finally accumulated over all pixels. The loss is computed as following:

$$\mathcal{L} = \sum_{p \in \mathcal{P}} \ell_p \quad ,$$

where, ℓ_p is loss computed at each pixel p and $\mathcal{P}$ is set all pixels. Since, we have labels for pixels belonging to scribbles only, we compute loss for these pixels only and ignore rest of pixels. This idea is motivated by formulation of loss function in *inpainting*. In inpainting, we formulate a loss which is non-zero for only missing pixels in the image. We defined the new loss as, **cross entropy scribble loss** ($\mathcal{L}_{\text{scribble}}$), which can be computed as given below:

$$\mathcal{L}_{\text{scribble}} = \sum_{p \in \mathcal{P}} w_p \, \ell_p \quad \text{where,} \quad w_p = \begin{cases} 1, & \text{if } \; p \in \mathcal{S} \\ 0, & \text{else.} \end{cases}$$

4.3 Experiments and results

We train OSVOS using scribbles on a single image and $\mathcal{L}_{\text{scribble}}$. Once, OSVOS is fine-tuned, we test the network on the same image. It can be observed that this is different from usual test setup where the training and test images are different sets. For our task, although we use the whole image, but only few pixels are annotated and used for loss computation. Thus, it is valid to test the network on the same image. The results can be seen in figure 4.1.

We can observe that using significant amout of scribbles CNN is able to learn the shape of object. As we manually annotated some pixels in each vesicle for the slice in figure 4.1, the CNNs are able to segment all vesicles. The accuracy gets limited by the amount of false positives. We can observe that for a lot of background pixels which are not annotated as background scribbles for training, the CNN predicts them as foreground pixels in output mask. This happens mainly for the pixels which are similar to foreground pixels

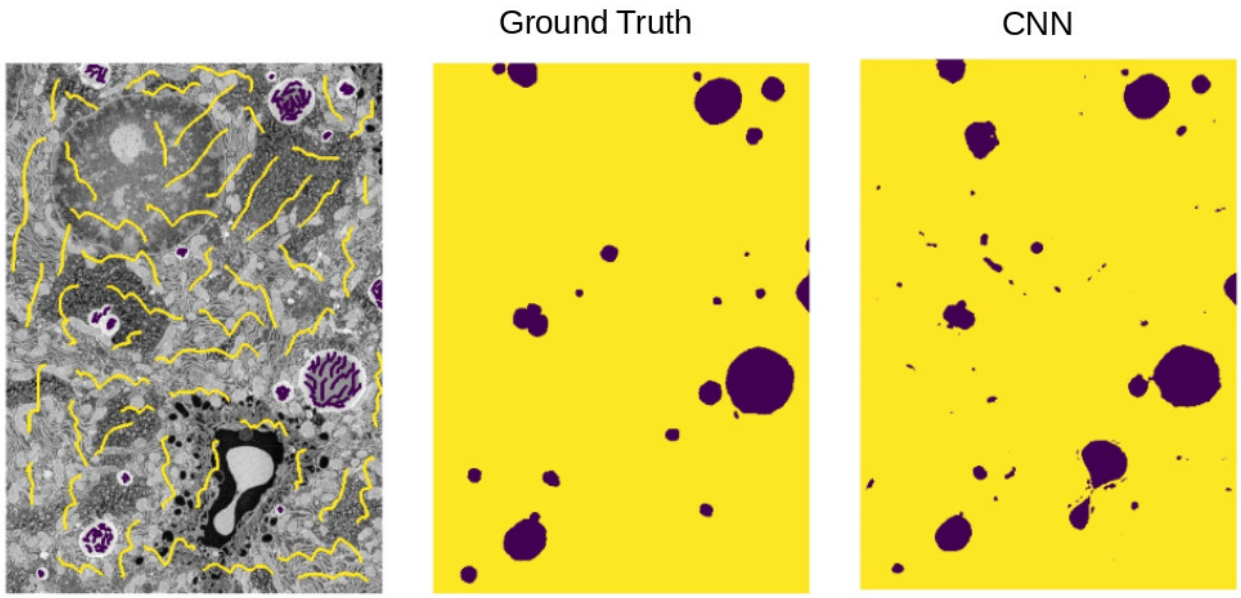


Figure 4.1: Segmentation mask generated using OSVOS and $\mathcal{L}_{\text{scribble}}$. Left-most image shows foreground (violet) and background (yellow) scribbles

in terms of pixel intensity. In addition to these pixels, we can see some noisy pixels predicted as foreground. This motivated us to make use of prior and clear this mask using variational image processing.

To compare the results and accuracy of CNN and VIP for different annotation budgets, we repeated the experiment done for RF using ”easy” and ”hard” scribbles, explained in section 3.2.1. Similar to experiment in RF, we trained the CNN by first adding annotations from ”easy” scribbles and then, from ”hard” scribbles. We take the original pre-trained parent OSVOS network and fine-tune it using scribbles. Each time we add new scribbles, we fine-tune the parent OSVOS network and test the network on the same image. Similar to section 3.3, we use probability mask predicted from CNN to compute anti-loglikelihood cost function and estimate a mask using equation 3.3. For results (shown in figure 4.2) using CNN, we make use of only anti-loglikelihood cost function.

Using around only 0.35% of total pixels, we are able to achieve an accuracy of 0.72. Although, when CNN is trained using full annotations, it is expected to produce highly accurate results and do not improve with use of VIP. But, here, the accuracy imporves with the use of VIP for all annotation budgets. We are able to achieve this improvement without significant addition to computational cost. Similar to RF, the addition of ”hard” scribbles (red curve in figure 4.2), the accuracy starts improving with a jump. Next, we compare

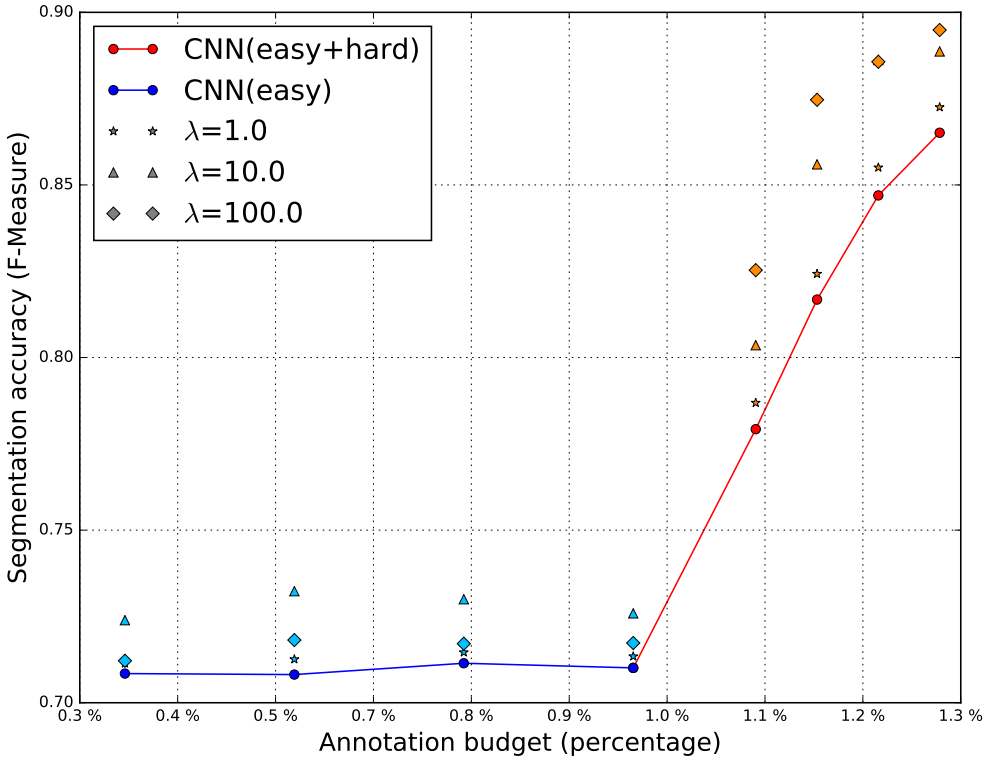


Figure 4.2: Segmentation accuracy for CNN with VIP (different λ)

the results from RF and CNN. We use the same amount of scribbles for ”easy” and ”hard” annotation class and make a comarison of results in figure 4.3.

For lower annotation budgets in our segmentation task, RF performs significantly better than CNN. We can expect this for CNN due to significantly less training data. The weights and biases in multiple layers of OSVOS are not able to adapt much and converge properly. But as the annotation budget becomer more than 1.2%, the CNN outperforms RF. With significant amount of scribbles, it also tries to learn the shape and the size of vesicles. The CNN is able to achieve a segmentation accuracy of 0.86 with only 1.3% of pixels used for training. Also, the intersection (red and blue curve in figure 4.3) can be seen as a boundary to choose RF or CNN for a given annotation budget.

Finally, we compare the results of RF and CNN in a Bayesian framework. Using the masks obtained from RF and CNN for a certain annotation budget, we use VIP to obtain an optimal mask. The results is figure 4.4 show improvement for different λ. For annotation budget of around 1.3% and use of VIP, RF is able to achieve accuracy close to that obtained from CNN and VIP.

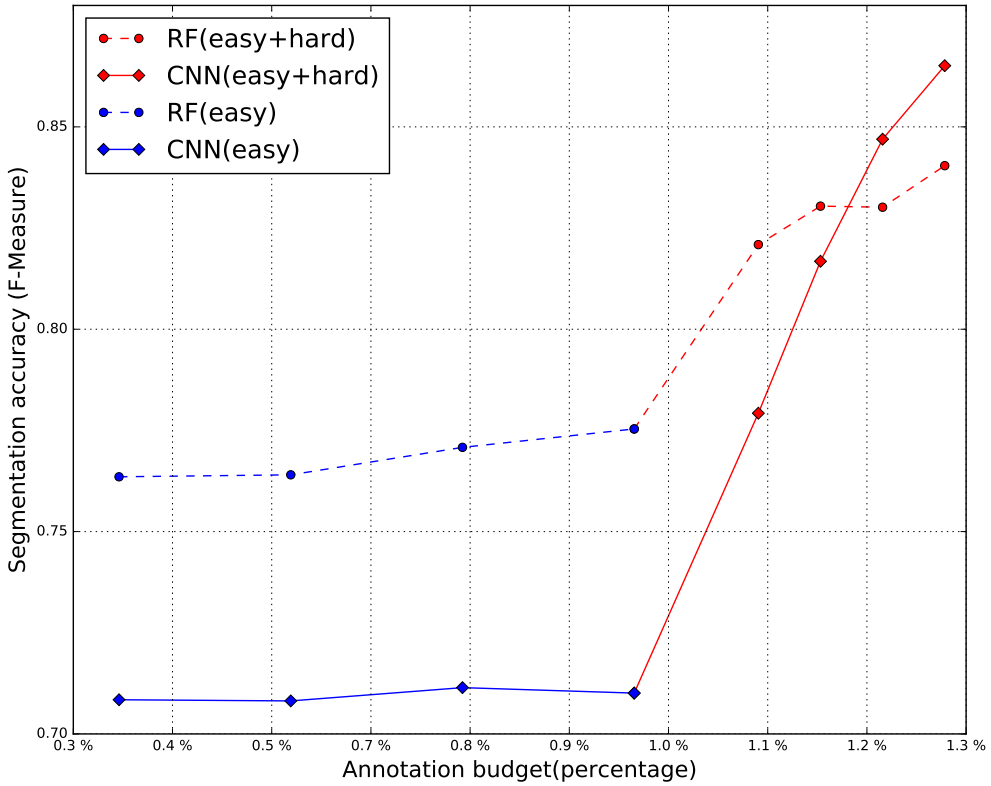


Figure 4.3: Segmentation accuracy for mask obtained from RF and CNN, after thresholding

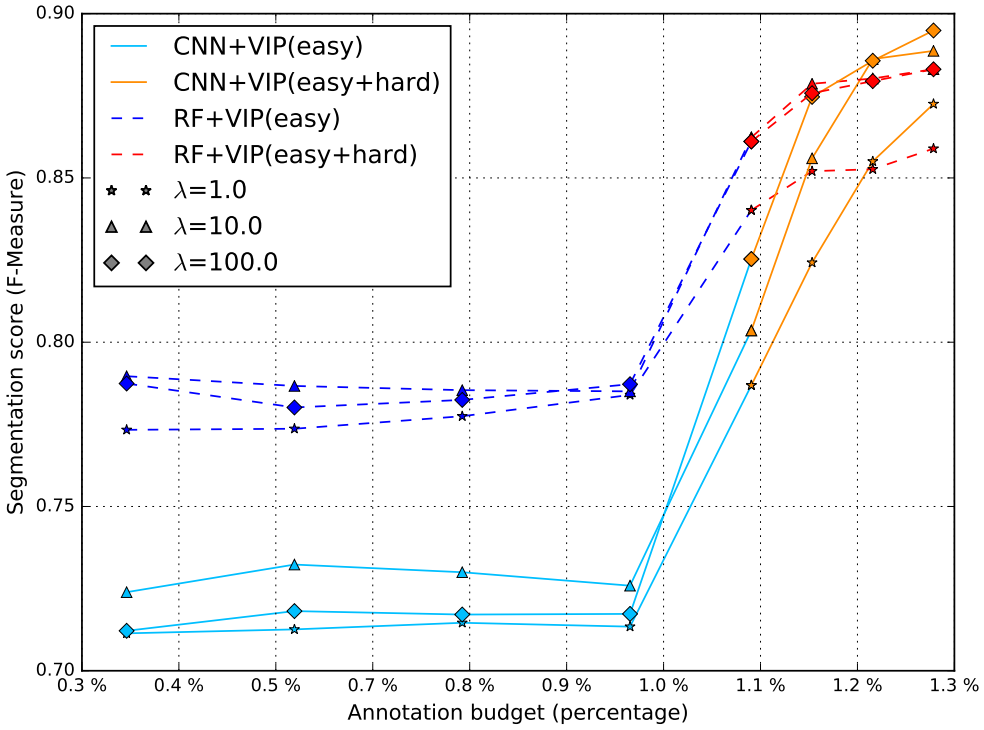


Figure 4.4: Segmentation accuracy for mask obtained from RF and CNN, both with VIP

Chapter 5

Conclusion

The comparison of results for RF and CNN shows that both methods have their advantages and disadvantages. RF can outperform CNN under certain conditions especially for low amount of training data. The amount of training data significantly effects the results for both methods. The aim of this thesis was to analyze the relation between segmentation accuracy and annotation effort. We can conclude that the results do not always improve with increment of training data. We observed that with arbitarily addition of scribbles, the accuracy did not improve in many cases. We need to increase not only the quantity of training data, but also the quality of it. Similarly, for training CNN with full annotations in chapter 2, we observed a fall in accuracy with increment of training data. In case of semi-supervised learning, we understood the requirement of iterative semi-interactive annotation to obtain best results for given annotation budget. The semi-interactive process can direct us to obtain best results instead of using our effort arbitrarily in semi-supervised learning. Another thing we tried, were different methods to compensate for training data in our task of segmentation. We were able to use a pre-trained network using non-microscopic images and fine-tune it to produce good results for microscopic images. In addition to that, we used the Bayesian framework to use variational methods to compensate for training data. The comparison of different cost functions showed precautions to be taken while using different cost functions.

Another major contribution of this thesis was the success of fine-tuning a pre-trained network using partial annotations. The simple modification of cross entropy loss for scribbles made it possible to use pre-trained fully convolutional networks. This opens a path to use networks pre-trained on million of images

in semi-supervised learning. Currently, the use of total variation may not be common because of its implementation. It is mainly considered as a post-processing step to improve results. The VIP can be used more, if it can be easily implemented and used. Recently, we can find a lot research trying to combine CNN and variational methods to obtain better results. Some papers show that it can also simplify the complexity of CNN. Ranftl [5] did this by adding total variation as an inference layer and solved the final problem using bilevel optimization [19]. This approach has a benefit to learn appropriate regularization parameter (λ) for total variation. The other approach related to implementation is to split the iterations used to optimize final problem (mentioned in section 3.2) as separate layers in the neural network. This is termed as Primal-dual network and described in Riegler et al. [6]. Each iteration can be implemented as one additional layer in networks.

Due to lack of sufficient time, we were able to theorotically couple these two approaches and came up with the idea of having variational neurons similar to convolution filters. These variational neurons will take the image as input and produce a TV regularized output. Instead of learning λ, we can use multiple values of λ and combine then using a CNN. This also opens a way to remove drawback of finding appropriate λ, while using variational methods. Due to time restriction, we were not able to complete this, but this idea can provide an option to get best out of multiple methods in a simple unified way.

Bibliography

[1] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. *CoRR*, abs/1611.05198, 2016.

[2] Simon K. Warfield, Kelly H. Zou, and William M. Wells. Simultaneous truth and performance level estimation (staple): An algorithm for the validation of image segmentation. *IEEE TRANS. MED. IMAG*, 23:903–921, 2004.

[3] Jakob Santner, Markus Unger, Thomas Pock, Christian Leistner, Amir Saffari, and Horst Bischof. Interactive texture segmentation using random forests and total variation. In *BMVC*, pages 1–12, 2009.

[4] Dominic Eugster. Semi-automated 3d object recognition in electron microscopy image data. Master's thesis, 2013.

[5] René Ranftl and Thomas Pock. *A Deep Variational Model for Image Segmentation*, pages 107–118. Springer International Publishing, 2014.

[6] Gernot Riegler, David Ferstl, Matthias Rüther, and Horst Bischof. A deep primal-dual network for guided depth super-resolution. *CoRR*, abs/1607.08569, 2016.

[7] Gernot Riegler, Matthias Rüther, and Horst Bischof. Atgv-net: Accurate depth super-resolution. *CoRR*, abs/1607.07988, 2016.

[8] 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]).

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[10] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Andrés Arbeláez, and Luc J. Van Gool. Deep retinal image understanding. *CoRR*, abs/1609.01103, 2016.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[13] J. Schindelin, I. Arganda-Carreras, and E. Frise. Fiji: an open-source platform for biological-image analysis. 9(7):676–682, 2012.

[14] Grégory Paul, Janick Cardinale, and Ivo F. Sbalzarini. Coupling image restoration and segmentation: A generalized linear model/bregman perspective. *International Journal of Computer Vision*, 104(1):69–93, 2013.

[15] Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: A scalable ADMM approach. *CoRR*, abs/1605.02026, 2016.

[16] Felix Gonda, Ray Thouis Verena Kaynig, Toufiq Parag Daniel Haehn, Jeff Lichtman, and Hanspeter Pfister. Icon: An interactive approach to train deep neural networks for segmentation of neuronal structures. *CoRR*, abs/1610.09032, 2016.

[17] Matthew Lai. Deep learning for medical image segmentation. 2015.

[18] Mohammad Havaei, David Warde-Farley Axel Davy, Aaron C. Courville Antoine Biard, Chris Pal Yoshua Bengio, and Hugo Larochelle Pierre-Marc Jodoin. Brain tumor segmentation with deep neural networks. *CoRR*, abs/1505.03540, 2015.

[19] Peter Ochs, René Ranftl, Thomas Brox, and Thomas Pock. Bilevel optimization with nonsmooth lower level problems. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 654–665. Springer, 2015.