Generalizability of Semantic Segmentation Techniques

A Comparative Case Study using LiTS Data Sets

Keshav Bhandari
Texas State University, San Marcos, TX
k_b459@txstate.edu

Abstract

In the recent years, popularity of semantic segmentation in computer vision has massively increased. Proposed deep learning architectures has their own pros and cons. Some architectures require huge amount of training data while others rely on the heavy use of data augmentation to address this. These models are however trained and experimented on baseline datasets and there are enough rooms to discuss about their generalizability in rare data sets. In this work we try show how naïve architectures in this domain fails and requires modification. We will also discuss an intuition behind a modification and experiment our own custom architecture based on the work [7]. The major objective of this work is to diagnose the weak spot of the generalizability of the previous work [2, 7] and define a new research goal for future work.

1. Introduction

Semantic Segmentation or scene labeling task in computer vision is to identify class for each pixel. In contrast with object detection, semantic segmentation is more precise and more complex work. Semantic segmentations tasks open huge applications opportunity in different domains. However, with decades of research it is still a tough task.

1.1. Summary

Convolution neural networks provides weight sharing, an important scheme for localization task that we see often in semantic segmentation. However, naïve implementation of convolution neural network for such tasks doesn't work well. As we can see that the work of [7] out performs [2].

Semantic segmentation has another major challenge. It requires a lots of training data sets. So, data augmentation is an equally important aspects, as we can see this in work [7]. However, as these experiments are subjected to baseline data sets and not discussed largely in adverse conditions where data sets are challenging, there is an opportunity to doubt the generalizability of these algorithms.

In this paper, we implement a naïve convnets and custom Unet based convnets for semantic segmentations. The baseline datasets we are using is LiTS data set, from a challenge organised in conjunction with ISBI 2017 and MICCAI 2017. The objective is to segment the liver lesions in contrast-enhanced abdominal CT scans. Due to their heterogeneous and diffusive shape, automatic segmentation of tumor lesions is very challenging. Until now, only interactive methods achieved acceptable results segmenting liver lesions. Moreover, the nature of these scans is sequential. For the majority of scans lesions appear gradually in the sequence so there are no semantic labels in such cases. So, it opens another research opportunity to consider sequence modeling as well. We will discuss this in future work.

This work tries to investigate why naïve convnets failed in this task and how simple modification on the naïve implementation performs well.

1.2. Previous Work

The ISBI challenge launched in the context of the ISBI 2012 conference (Barcelona, Spain, 2-5th May 2012) open a new contribution in the field of medical image segmentation. The best method (a sliding-window convolutional network) at that time was outperformed by the work of [7] in 2015 and won the ISBI cell tracking challenge. These works create a huge attention to the researcher working in computer vision. Work like [5] uses semi supervised approaches , moreover other works like [9] tries to combine the ideas and addressed the need of sequential modeling from work [7]. There are many other impressive works on this field.

FRRN (Full-Resolution Residual Networks) [6, 8] is one of the state-of-the art model. It uses multi-scale processing techniques by using two separate streams, the residual and pooling stream. This helps to process semantic features for higher classification accuracy. It progressively downsamples the features maps in the pooling stream, meanwhile processing the feature maps at full resolution in the residual stream. So these two streams accounts for high classification accuracy and low-level pixel information for high localization accuracy.

FRRN did excellent job but with heavy processing overhead at every scale. PSPNet [12, 8] is another state-of-the-art to get around this overhead. It uses ResNet and DenseNet like architecture to extract feature. This architecture combined multi-scale feature maps without applying many convolutions.

The One Hundred Layers (FCDenseNet) [4, 8] is another kind which uses U-Net architecture. The main contribution of this architecture is the clever use of dense connections similar to that of the DenseNet classification model.

The state-of-the-art model we discussed so far has huge amount of parameters overhead. DeepLabV3 [1, 8] tries to address the parameters overhead by using feature extraction frontend. This is a very ligthweight model. It downsamples the input images to 16 times smaller input, and there are high odds of getting good localization and can leads to poor pixel accuracy. The main contribution of this architecture is the clever use of its state-of-the-art Atrous convolutions. However, it still uses the same upscaling techniques as in PSPNet.

One of the attempts to review these techniques were done in the work of [3], they reviewed existing methods, highlighting contributions and significance of those methods in the field. The recent work of [10] is more closely related work, their contribution on developing and evaluating recent advances in uncertainty estimation and model interpretability in the context of semantic segmentation using several enhanced architectures of Fully Convolutional Networks is one of the amazing work.

1.3. Methods and Results

In this paper, we implement two different architecture, a naïve – symmetric convolutional neural network-based model and modified our naïve implementation. This model is inspired by the work of [7].

The first method is symmetric convolutional neural network which has same down-sampling and up-sampling architecture. 512 x 512 x 1 pixels are down-sampled to 16 x 16 x 1024, and up-sampled to 512 x 512 x 1. We modified the network to store the indices of max-value during down-sampling and these indices were used in corresponding stages of up-sampling to obtain an approximate inverse by recording the locations of the maxima within each pooling region. This helps preserving the structure of the activations [11].

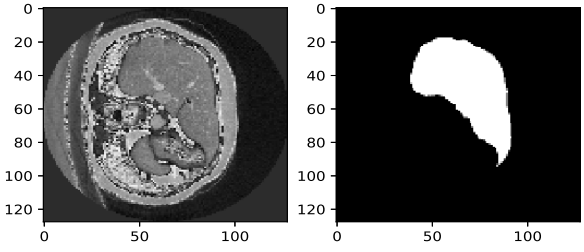

Figure 1: Sample Image scans(left) and segmentation(right)

The second method is a modification on the first method. This approach is inspired by U-Net architecture [7], however this method is tailored in such a way that it has different skip connections schemes. We exploit the nature of our symmetric model to pass the weights to corresponding stages from down-sampling part to up-sampling part and add these weights together. Moreover, as we discussed we also preserve the structure of the activations by approximating the true inverse during max-Unpooling.

The former method tries to predict the pixel values to find the segmentation region corresponding to specific scans, as opposed to later, which tries to predict a class per every pixel within the mask. These two tasks differ each other from their orientation of objectives. Former being regression and later being classification.

From the experiment results we can see that later method works better than the former one. We will discuss more these on results section.

2. Problem Description

Manually identifying liver lesions is cumbersome task, computer aided segmentation will alleviate the efficiency and save valuable time in medical industry and in medical research where researchers have to deal with thousands of such task in their usual work routine. Creating a deep learning model to develop automatic segmentation is a challenging task. Here we will discuss two methods that earned their popularity few years back with few modifications. However, the major focus here is to demonstrate failure and generalizability of the deep learning models and to identify why few modifications are needed for advancements, along which opens more research understanding for future.

2.1. Methodology

We will demonstrate how naïve implementations of convolution neural network is not enough to do the segmentation tasks, and how simple modification of this implementa-

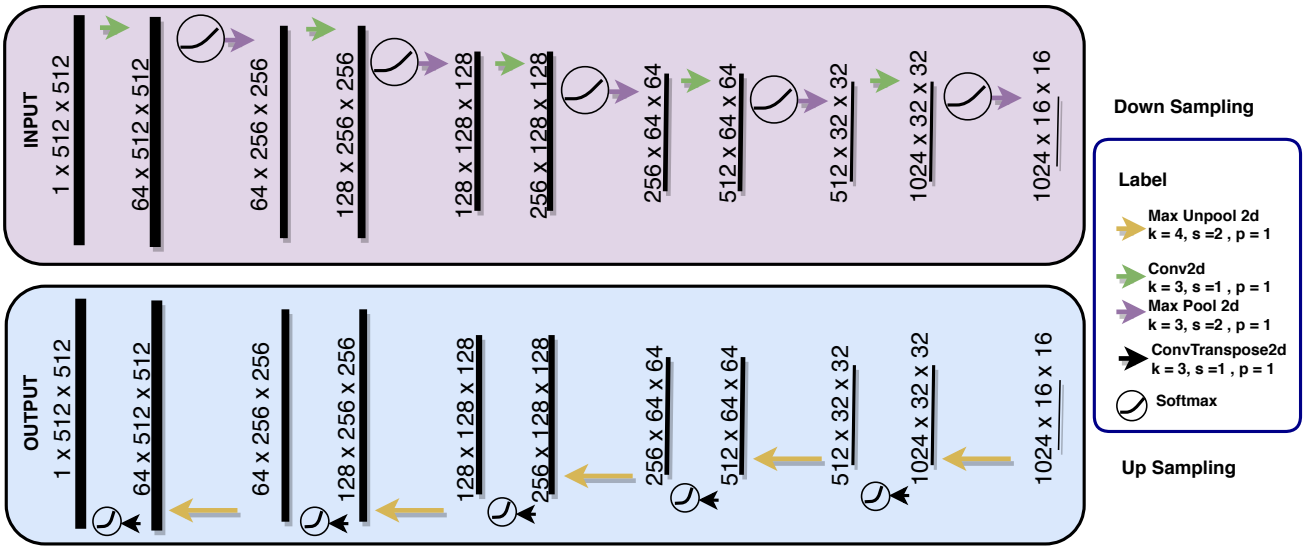

Figure 2: Convnet based model Corresponding layers from downsampling and upsampling are stacked together

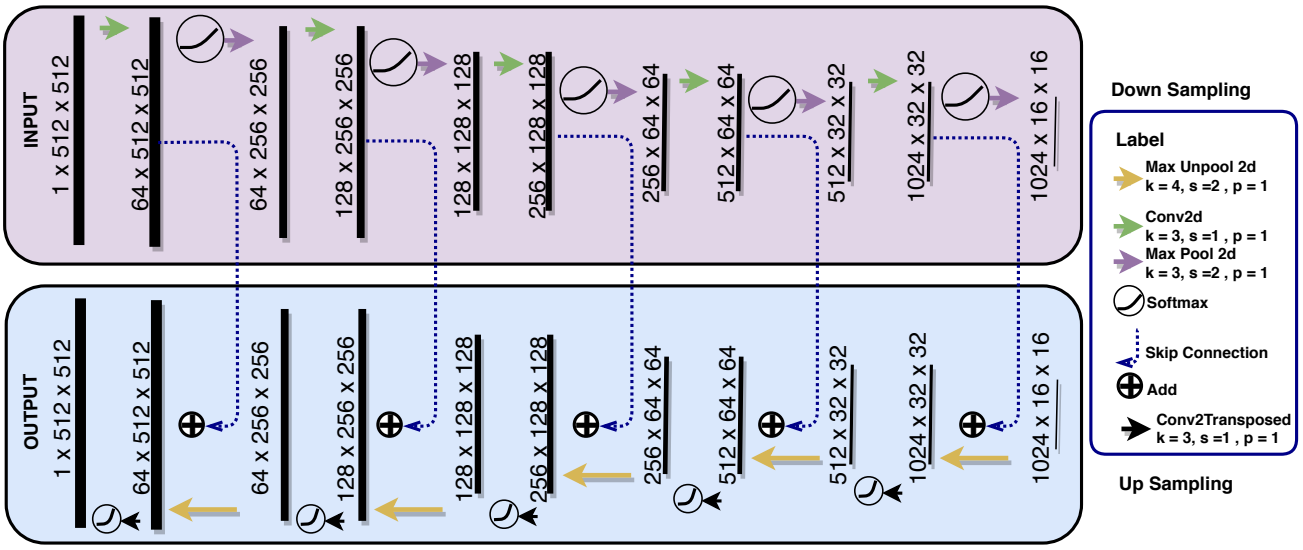

Figure 3: U-net based model Corresponding layers from downsampling and upsampling are stacked together

tion drastically changes the results and learning speed. We perform our experiment on LiTS data sets. These data sets contain 125 CT scans files. These scans contain varied number of sequential images. Detail description of the data sets is given below

Sample datasets with label segmentation is shown in figure 1. Our first model architecture is defined in figure 2 .This convolution neural network based architecture has two main parts downsampling and upsampling. Dimensions are

Info	**Description**
Source	MICCAI 2017
Number of Scans	125 Scans, 108890 images, ~871 images per scan in average
Train Scans	117 Scans, 101966 images,
Test Scans	8 Scans, 6924 images
Data Format	NIFTI File Format

equal within levels in corresponding parts. Most important things to notice about this architecture is the way how Max-Unpooling is done, we pass indices of max value at corresponding stages so that we can construct an approximate inverse. Corresponding stages are the steps from max-pool and max-unpool with same dimensions as shown in figure 2 and 3. Suppose we have following examples, assume max-pool with kernel=2, and stride = 1, maxUnpool with kernel = 3, and stride = 1 and A as a latent feature matrix

$$A = \begin{bmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ a7 & a8 & a9 \end{bmatrix}; A' = maxpool(A) = \begin{bmatrix} a5 & a6 \\ a8 & a9 \end{bmatrix}$$

Max-Unpooling in general from A', when we don't store the indices

$$A'' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a5 & 0 & a6 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & a8 & 0 & a9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

;

$$A^{-1} = maxUnpool(A') = maxPool(A'')$$

$$= \begin{bmatrix} a5 & a6 & a6 \\ a8 & a9 & a9 \\ a8 & a9 & a9 \end{bmatrix}$$

Max-Unpooling that we used, from A', when we store the indices

$$A^{-1} = maxUnpool(A') = maxPool(A'')$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & a5 & a6 \\ 0 & a8 & a9 \end{bmatrix}$$

We can see that the inverse calculated later preservers original matrix better.

We define first model as pixel prediction problem, minimizing mean square error. We trained the models to 100 epochs.

Our second model as shown in figure 3 is introduced as an improvement from the first one. We believe our model suffers huge feature loss during up-sampling. Inspired from U-net architecture [2]. We implement a unet-like architecture with minor difference in our model. Here we pass the output from corresponding down-sampling layers to the up-sampling layers and add them together whereas original work of Unet architecture does cropping and concatenation. With image processing on train data, we create a mask from the given ground truth and run classification problem for each pixel if it is within the mask, minimizing cross entropy loss. We trained the models to 100 epochs.

Following are the summary of the two models

Info/Model	**Conv-based**	**Unet-based**
Loss	MSE	Binary Cross Entropy
Optimizer	Adam	Adam
#epochs	100	100
Tr.Time	8	6
GPU	NVIDIA 1080Ti, 16GB	NVIDIA 1080 Ti, 16GB
Library	Pytorch	Pytorch

Code can be found at `https://github.com/keshavsbhandari/Image_Segmentation`

3. Results

Both models show significant improvement in validation loss over training loss as we can see in figure 5 and 6. However, the convolution-based models did worse job from generalization perspective. We analyze our results by simulating real-time scanning and segmentation task in test scans. Convnets based model averaged out all the features and gave same segmentation results throughout the simulation, as we can see in figure 4.

To perform some qunatitative analysis we used a metric called as Jaccard index.

Jaccard index, also known as Intersection over Union (IOU) is a statistic used for comparing the similarity and dissimilarity of sample sets.

The Jaccard coefficients is defined as the size of the intersection divided by the size of the union of sample sets:

$$IOU = J(A,B) = \frac{|\, A \cap B \,|}{|\, A \cup B \,|} = \frac{|\, A \cap B \,|}{|\, A \,| + |\, B \,| - |\, A \cup B \,|}$$

(if A and B are both empty, we define J(A,B) = 1.)

$$0 \leq J(A,B) \leq 1$$

The Jaccard distance, is a measure of dissimilarity between samples. Jaccard similarity is for comparing two binary vectors so we can compute this easily for unet-based

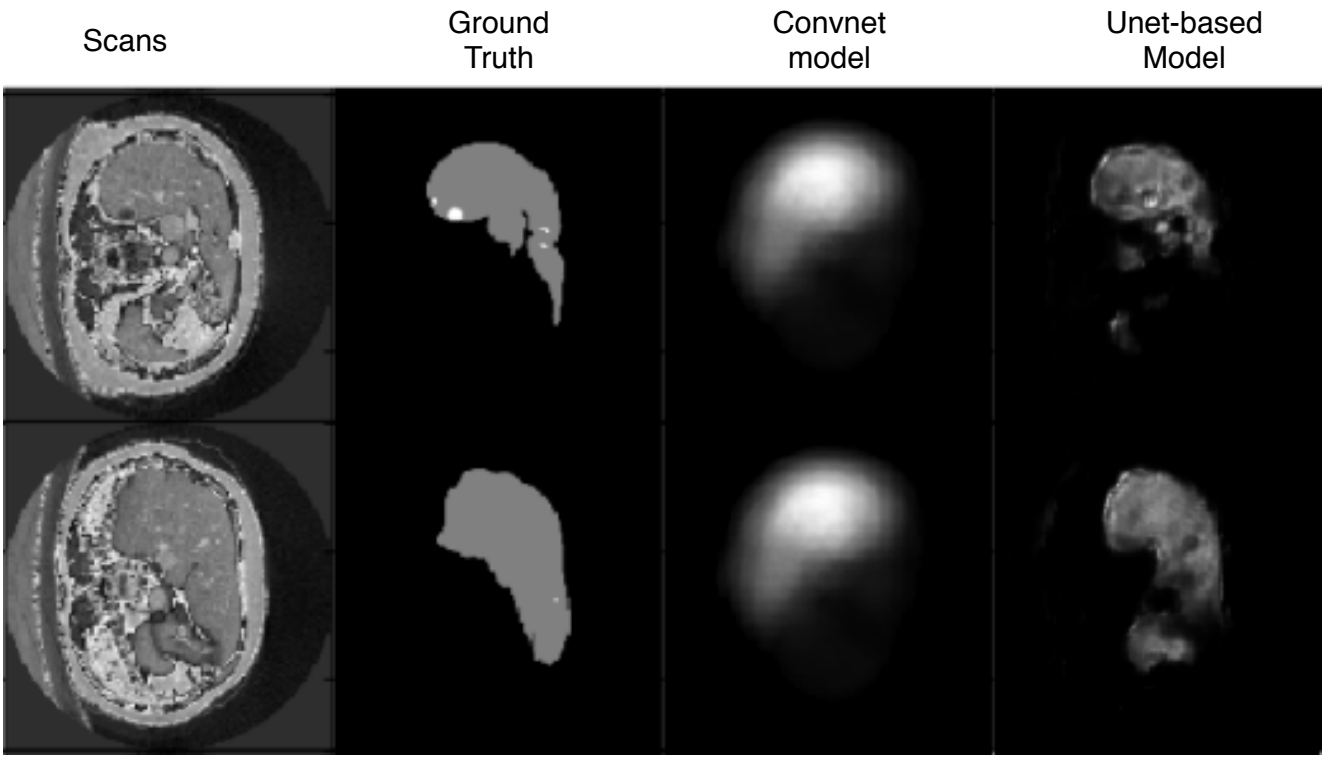

Figure 4: Experimented Result from both Convnet based and Unet based model Convnet based model doesn't show any generalizaiton to localization accuracy Unet-based model perform well comparaitiviely

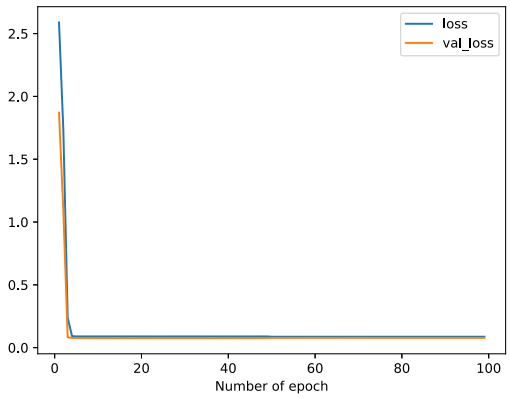

Figure 5: Convnet based model training and validation loss over number of epochs

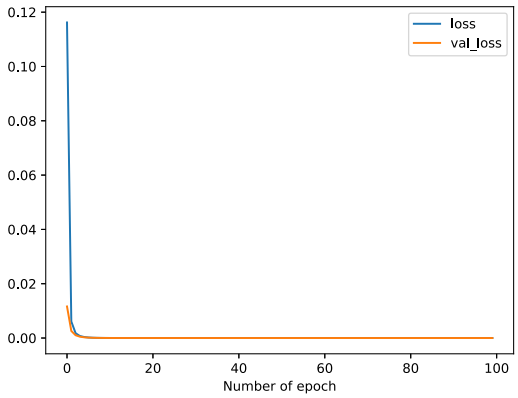

Figure 6: U-net based model training and validation loss over number of epochs

model but for the sake of our conv-net based model we can use generalized version of the Jaccard index as given by

$$IOU = J(L, E) = \frac{\sum_i min(L_i, E_i)}{\sum_i max(L_i, E_i)}$$

Moreover, we also calculate the pixel accuracy for unet-based model. The pixel accuracy is defined as

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where,
TP : True Positive
TN ; True Negative
FP : False Positive
FN : False Negative

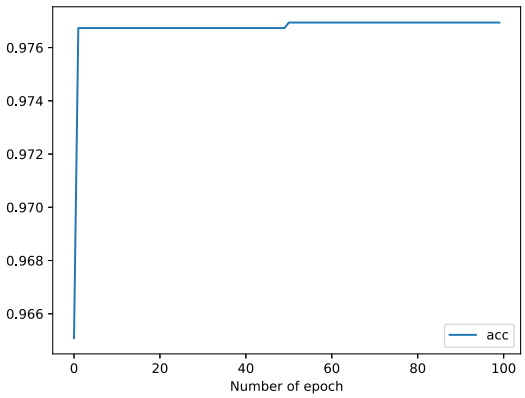

Figure 7: Unet Based Model : Pixel accuracy calculate per

Since, our class representation is too small this metric can be misleading as well, as the measure will be biased in mainly reporting how well model identified negative case.

Calculated metrics for both model is given below.

Info/Model	**Conv-based**	**Unet-based**
IOU	0.19	0.53
Pixel Accuracy	NA	~100

Above results clearly shows performance of U-net based model over convnet based model qunatitatively.

4. Conclusion and Future Work

The datasets we used are rarely mention in literature. We showed that how semantic segmentation models can be questioned to its generalizability. The unet-based model achieves very good performance compared to naïve convnet based model. With little modification on the convnet based model we are able to show how we can significantly improve the model.

We established an understanding of skip connections we saw in Unet-based model and importance of data augmentation. These techniques turned out to be important for the model to learn more features.

We did our best in this work to compare and contrast between naive model with unet-based model. However, we did not consider two important aspects here, first smaller datasets and second sequential nature of data. We would like to continue our future research on how we can tackle these problems with other data augmentation techniques and combine sequence modelling aspects as well.

References

[1] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *ArXiv e-prints*, page arXiv:1706.05587, June 2017.

[2] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[3] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *ArXiv e-prints*, page arXiv:1704.06857, Apr. 2017.

[4] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *ArXiv e-prints*, page arXiv:1611.09326, Nov. 2016.

[5] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic Segmentation using Adversarial Networks. *ArXiv e-prints*, page arXiv:1611.08408, Nov. 2016.

[6] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes. *ArXiv e-prints*, page arXiv:1611.08323, Nov. 2016.

[7] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv e-prints*, page arXiv:1505.04597, May 2015.

[8] G. Seif. Semantic segmentation with deep learning – towards data science, Sep 2018.

[9] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville. ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation. *ArXiv e-prints*, page arXiv:1511.07053, Nov. 2015.

[10] K. Wickstrøm, M. Kampffmeyer, and R. Jenssen. Uncertainty and Interpretability in Convolutional Neural Networks for Semantic Segmentation of Colorectal Polyps. *ArXiv e-prints*, page arXiv:1807.10584, July 2018.

[11] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. *ArXiv e-prints*, page arXiv:1311.2901, Nov. 2013.

[12] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. *ArXiv e-prints*, page arXiv:1612.01105, Dec. 2016.