Automated Detection of Colon Polyps

Udit Adhikari
Department of Computer Science
UMASS - Lowell Lowell, MA
Udit_Adhikari@student.uml.edu

I. Introduction

This research hypothesizes that a vanilla U-NET can be modified to fit dimensions of input image and provide a better model for segmenting the image as background versus colon polyp. The modification will be to simply modify the width and height dimensions of the filter so resizing to the input dataset does not need to be performed. The ultimate goal of this project is to build a model that produces good precision, but more importantly, better recall than previous works in validation dataset. I also compared by model with the models researched and implemented in the Y-NET paper [9].

II. Motivations

Colonic polyps have large variation in terms of shape, texture, size, and color and because of this automatically detecting polyps becomes quite hard. In this paper, I apply a Fully Convolutional Network-Based Semantic Segmentation (FCN) approach for the automatic detection of polyps in the images obtained from colonoscopy examinations. I also apply image augmentation strategies to overcome the limited amount of polyp images for training the FCN networks.

III. Related work/Literature Review

This experiment is very heavily based on U-NET architecture. However, the experiment modifies the U-NET architecture slightly to fit the image dimensions of the dataset.

U-NET : Convolution Neural Network for Image Segmentation

U-NET network architecture is illustrated in Figure 1. The U-NET consists of two blocks. The block on the left is the encoder block while the block on the right is the decoder block. The encoder block is a normal convolutional network. This block consists of two 3x3 convolutions each with rectified linear unit (ReLu) activation and followed by 2x2 max pooling. The 3x3 convolutions have a stride of 2. This convolution down samples but doubles the number of channel features.

The decoder block consists of upsampling of the feature map followed by a 2x2 up-convolution. The up-convolution halves the number of channel features. This step also concatenates the corresponding cropped feature map from the encoder block. Then this is followed by two 3x3 convolutions each with a ReLU activation. The cropping is necessary because convolution reduces the border pixels. The final layer is 1x1 convolution and it used to map each of the 32-component feature vector to the desired

number of classes. In this case there are only two classes i.e. background and polyp.

Y-NET : A deep Convolutional Neural Network for Polyp Detection

I also researched a modification of U-NET, called Y-NET [9]. The Y-NET network described in the Y-NET paper [9] is very similar to the encoder decoder model of vanilla U-NET. However, it adds another encoder block that follows the VGG19 network architecture and uses the pre-trained weights of VGG-19 trained on ImageNet dataset. This encoder block is concatenated with the normal U-NET encoded block.

Activation Function

Activation functions allow weights of a neural network to be incrementally updated. The derivative of the activation functions allow for a gradual update of weights that can either maximize or minimize the output [7]. I used a *ReLu* activation function in the last layer of the neural network architecture. The following is the formula of ReLu activation function:

$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{if } x_i < 0. \end{cases}$$

- [8]

Dice Coefficient and Dice Loss

The network processes image through a soft-max layer which outputs the probability of each pixel to belong to background or polyp. It is not uncommon that the area of interest, i.e. the polyp region, occupies only a very small region of the image. This can cause the learning process to get trapped in local minima of the loss function and produce predictions that are strongly biased towards background [3]. In previous approaches, the loss functions were based on sample re-weighting where polyp regions are given more importance than background ones during learning. This balances out the overweighting of the background pixel against the polyp pixel.

In this paper, I use a loss function based on the dice coefficient, which is a quantity ranging between 0 and 1 and which I will maximise. The dice coefficient removes the need to re-weight to samples of different classes and establish the right balance between background and polyp pixels.
The dice coefficient D between two binary values can be written as

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

where we sum over the N pixels, of the predicted binary segmentation volume pi ∈ P and the ground truth binary volume gi ∈ G.

Cross Entropy

In order for any neural network to test its model it needs to know how well it performed against expected output.
Cross Entropy is one of the many loss functions that allows us to measure the loss during training of the neural network.

The cross entropy is defined as:

$$-\sum_J y^{(j)} log\ \sigma(o)^{(j)}$$

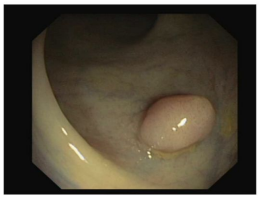
----- [5]

Where ‘y’ is true label as one-hot encoding and ‘ o’ is the output vector and σ(·) denotes probability estimate [6].

IV. Proposed Approach

The Data

The referenced dataset is from the CVC-ClinicDB and it can be accessed from https://polyp.grand-challenge.org/CVCClinicDB/. The dataset contains 612 images of the colon with polyps and the corresponding mask image of where the polyps lie. The images come from 29 different videos of the colonoscopy procedure. For example below is the original image and its corresponding mask -




The image on the left is the original image and the image on the right is the mask highlighting the polyp region in white.

Pre-processing

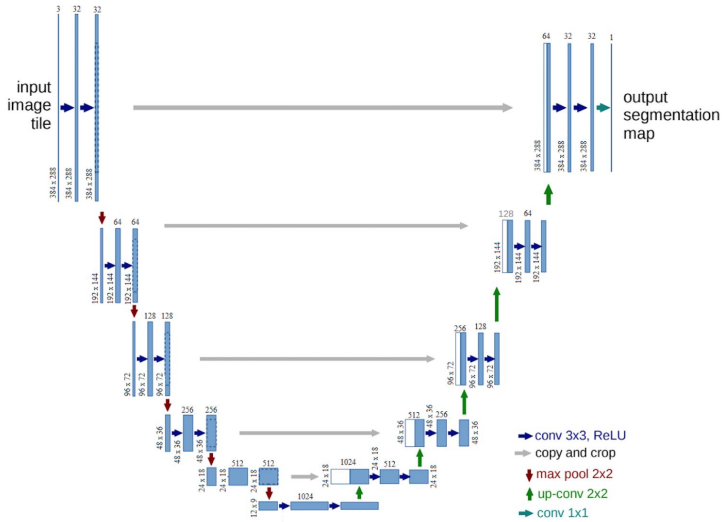
The image was not resized in this experiment. The reason was so the model during training would not lose any information while training the U-Net. The dataset was shuffled to make sure that the model wasn't getting trained on only certain video sequences. The shuffling of the dataset allowed diversity in the shapes and sizes of poly images during the training process. This experiment did not use any transfer learning unlike the Y-NET paper. I hypothesize that using transfer learning from networks trained on very different datasets would not valuable information to the model.

Data Augmentation

Data augmentation was carried out to add variation to the dataset and avoid overfitting. The image pixel values were normalized between 0 and 1 by dividing them by 255. A hue delta of 0.1 was added. Horizontal flip was also performed to both the original image and its mask. Width shift and height shift of 0.1 was done to both the original image and its mask.

Model and Experiment

I used the Tensorflow framework for this deep learning experiment. U-NET network architecture was used for this image segmentation task. I only had to classify two classes - a background class and a polyp class. Batch normalization was added after every convolution layer and before ReLu activation was used. The U-NET architecture was kept intact but the dimensions of each layer was modified to keep the input image resolution constant and not resize it. Each convolution was done with the same padding. The result of the convolution did not change the dimensions of the image after the convolution, but the convolution did change the channel depth. Following is the diagram of my U-NET diagram with updated dimensions of image as processes every layer -



Blue box corresponds to a multi-channel feature map. The width and height of image are located at the lower left edge of the box. The number of

channels is mentioned on top of the box. White boxes represent copied feature maps. The arrows denote the different operations and are described in the bottom right corner of the above image.

Configuring the Model

The model was configured for the learning process. Following is how I configured rest of the training process -

1. Epochs - The model ran for 200 epochs.
2. Weight Initializing - the weights were initialized with Xavier uniform initializer. In tensorflow this done using 'Glorot uniform initializer'.
3. Optimizer: It is the optimization algorithm to use to train the network. I used the 'Adam' optimizer. This optimizer has a momentum built into it, and unlike SGD is not prone to getting stuck in local minima.
4. Loss function: It is the objective that the model will try to minimize. II used a custom loss function that combined 'dice loss' and 'binary cross entropy'. Binary cross entropy works great when you have balanced data set. However, in the imbalance data set binary cross entropy becomes biased towards the class with more population. In the colon polyp case, the background class heavily outnumbers the polyp class. Thus, I had to combine binary cross entropy with dice loss which performs intersection over union.
5. A list of metrics: A metric function is used to evaluate the performance of the model. I used dice loss, dice loss + binary cross entropy, precision, recall, F1, and F2 as metrics. In the medical imaging domain, more weight is given to achieving higher recall. Recall is defined as :

$$Recall = \frac{true\ positives}{(true\ positives + false\ negatives)}$$

It is better to predict a false positive of colon polyp and draw attention of the doctor to verify than to miss out on predicting a colon polyp. This means less false negatives which implies a higher recall. It would also be a good model if it can precisely identify pixels that have colon polyp. I also measured precision Precision is defined as :

$$Precision = \frac{true\ positives}{(true\ positives + false\ positives)}$$

I will also calculated F1 and F2 scores to compare it to others. The F1 and F2 score are calculated as :

$$F1 = \frac{2 * Precision * Recall}{(Precision + Recall)}$$
$$F2 = \frac{5 * Precision * Recall}{4 * Precision + Recall}$$

V. Experimental Results and Discussions

The U-NET setup with no resizing of input images did yield promising results. The training and validation loss of my model during the 200 epochs is as follows: -

The best validation loss was 0.7353
The loss curve demonstrates that I am not overfitting. Validation loss decreases after every epoch and is close to training loss.

Comparison with the state-of-the-art

The performance of the model is superior in recall, F1, and F2 compared to other previous work. In the table below, I measure the precision recall, F1, and F2 from my model to that from the table 1 of Y-NET paper [9].

Method	Precision (%)	Recall (%)	F1	F2
PLS	13.6	36.9	19.9	27.5
CVC-CLINIC	31.3	36.6	33.8	35.4
OUS	90.6	51.5	65.7	56.4
ASU	**93.5**	61.1	73.9	65.7
CUMED	80.0	71.4	75.5	73.0
Fusion	88.1	71.0	78.6	73.9
Y-Net	87.4	84.4	85.9	85.0
Vanilla U-NET (Mine)	91.37	**85.13**	**88.14**	**86.31**

The model supports my proposed approach of building a vanilla U-NET without performing any resizing of the image, creating a better model for automated colon polyp detection.

VI. Conclusions and Future Work

I addressed the colon polyp detection problem by using a vanilla U-NET architecture and only modifying the dimensions of the network. The approach of using pre-trained network like in the Y-NET paper [9] was completely discarded. This produced better results because pre-trained networks like VGG-19 on ImageNet dataset are trained on very different images compared to medical images.

I did notice in my training and validation loss curve that there were slight variances towards the end of 200 epochs. I think my model can use more training perhaps for 400-500 epochs. Training for more epochs might only slightly improve the precision and recall, and result in a better loss curve.

In conclusion, this experiment shows that a vanilla U-NET with modifications to dimensions of the weight matrix suffices to create a better automated polyp detector that can segment the image into background and colon polyp.

References

[1] Olaf Ronneberger, Philipp Fischer, Thomas Brox. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation, https://arxiv.org/pdf/1505.04597.pdf

[2] Long, J., Shelhamer, E., Darrell. T. (2014).: Fully convolutional networks for semantic segmentation, arXiv:1411.4038 [cs.CV]

[3] Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi, (2016), V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, http://campar.in.tum.de/pub/milletari2016Vnet/milletari2016Vnet.pdf

[4] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, M. Jorge Cardoso, (2017), Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations, https://arxiv.org/pdf/1707.03237.pdf

[5] Zachary C. Lipton, John Berkowitz, “A Critical Review of Recurrent Neural Networks for Sequence Learning”, 2015

[6] Katarzyna Janocha, Wojciech Marian Czarnecki, “On Loss Functions for Deep Neural Networks in Classification”, 2017

[7] Joao Carlos Canto de Almeida, Delson Faria DaSilva, Jonathan Mwaura, “Classifying EEG Data Set Using Neural Networks”

[8] Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li“Empirical Evaluation of Rectified Activations in Convolution Network”, 2015

[9] Ahmed Mohammed, Sule Yildirim, Ivar Farup, Marius Pedersen, Øistein Hovde, Y-NET: A deep Convolutional Neural Network for Polyp Detection, 2018