Tissue recognition during Third-Space Endoscopy using a Deep Learning algorithm

Pradeep Mundlik
ai21btech11022@iith.ac.in

Naman Chhibhar
ma21btech11011@iith.ac.in

Abstract

In this project we aim to implement semantic segmentation model to help with segmentation task during Endoscopic procedure. The literature proposed a DeepLabV3+ model with Resenest101 backbone. We propose a DeepLabV3 model with MobileNetV3-Large as backbone layer which has less number of parameters and layers than Resenest101 hence improving efficiency. We have extracted images from POEM videos and annotated using Segment Anything Model [5]. Our proposed model aims to streamline and improve the accuracy of segmentation tasks in endoscopic procedures, potentially enhancing surgical outcomes and efficiency.

1. Introduction

Endoscopy is non-surgical examination of internal organs using a flexible tube with a light and camera called an Endoscope. It is an established resection technique to diagnose, treat, or monitor conditions in the digestive or respiratory tract. There are two major endoscopic procedures, **Endoscopic submucosal dissection (ESD)** and **Peroral Endoscopic Myotomy (POEM)**. ESD and POEM are complex procedures with elevated risk of operator dependent adverse events. There are high chances of intraprocedural bleeding and perforation which lead to significant risks to patient safety and procedural success. In this project we aim to develop an Artificial Intelligence Clinical Decision Support Solution (AI-CDSS) for on-time detection of Muscle layer, Submucosal layer and Electrode; making it helpful and assisting for operator during surgery.

2. Methodology

2.1. Database

We have collected 8 POEM videos of length around 40-45 mins each. Total 8000 snapshots are extracted from these videos. The snapshots are cropped to 720 x 930 pixels. Out of these 24 snapshots are annotated with the help of Segment Anything Model (SAM) with 4 classes: *Background, Muscle layer, Mucosal layer, and Electrode*. The annotated snapshots are used to train the model.

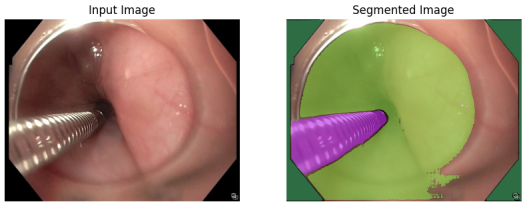


Figure 1. Annotated image
Background: Green, Muscle layer: Yellow, Electrode: Purple. There is no Mucosal layer in this image.

2.2. Architecture

A DeepLabv3 model with MobileNetV3 backbone is used. We explain these in detail below.

2.2.1 DeepLabv3

DeepLabv3 is a semantic segmentation model that uses Atrous convolution to capture multi-scale context by using different dilation rates. It uses a MobileNetV3 backbone with atrous convolution. The model uses atrous spatial pyramid pooling (ASPP) to capture multi-scale context by using different dilation rates. The model also uses a fully connected conditional random field (CRF) to refine the segmentation results.
It takes N feature maps as input from the backbone layer and outputs N*4*H*W tensor, where N is the batch size, 4 is the number of classes, H is the height, and W is the width of the image.The model outputs a probability distribution over the classes for each pixel in the input image.

2.2.2 MobileNetV3-Large Backbone

MobileNetV3-Large is a convolutional neural network designed for feature extraction. MobileNetV3-Large architecture uses 16 initial filters in the first convolution layer and then reduces the number of filters in the later layers by a

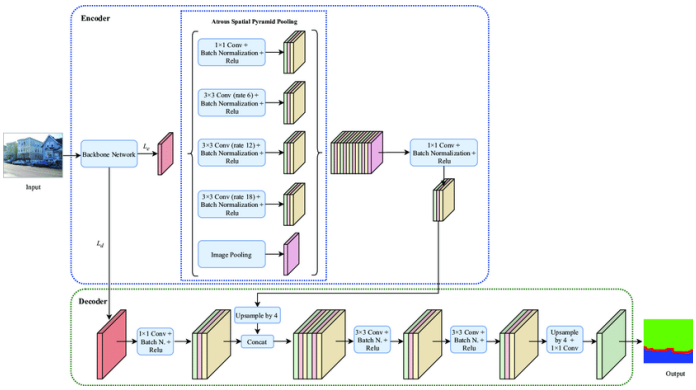
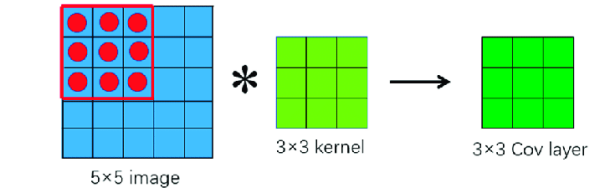
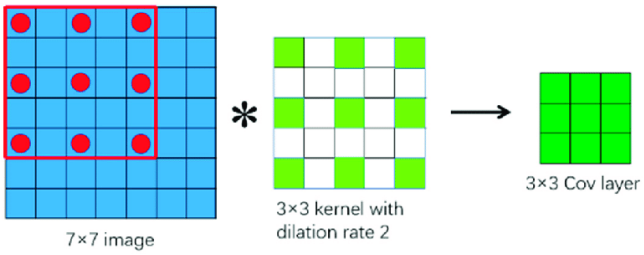


Figure 2. DeepLabv3 architecture [1]



(a)



(b)

Figure 3. Atrous Spatial Pyramid Pooling (ASPP) in DeepLabv3
(a) Conv 3x3, rate=1 (b) Conv 3x3, rate=2
[2]

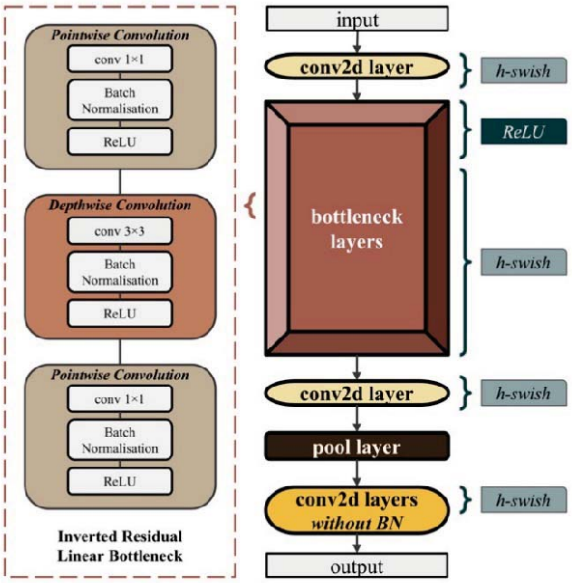


Figure 1 MobileNetV3 architecture. The general architectures are the same for both MobileNetV3-Large and MobileNetV3-Small.

Figure 4. [3]

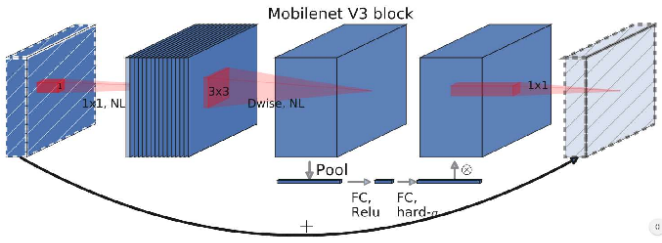


Fig. 1. MobileNet V3 Block [1].

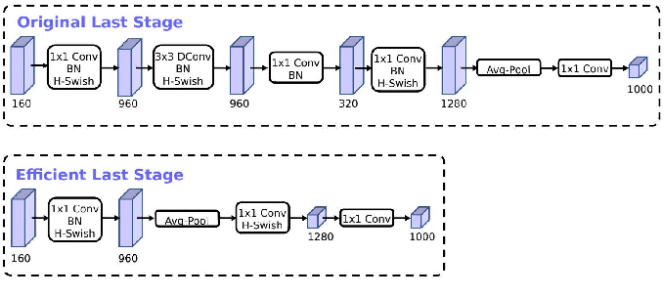


Fig. 2. Comparison of original and efficient last stage in MobileNet V3 [1].

Figure 5. [4]

factor of 2. It has typically 60 layers. Its building block involves an Inverted Residual Block, which includes a depthwise convolution, a squeeze-and-excitation module, and a point-wise convolution. It takes normalized RGB images as input and outputs feature maps that are used by DeepLabv3. Input to the layer is a 4-dimensional tensor with normalized RGB channels, and the output is N*32*32, where N is the batch size. It involves downsampling layers.

2.3. Training and Evaluation Setup

While training on data, the input is a 4-dimensional tensor with normalized RGB channels, and the labels are 2D tensors with integer values representing the class labels for each pixel. The output from the PyTorch model is a 4-dimensional tensor with the shape (N, 4, H, W), where N is the batch size, 4 is the number of classes, H is the height, and W is the width of the output. The output is a probability distribution over the classes for each pixel in the input image.

The model is trained with the cross-entropy loss function. We use the Adam optimizer with exponential learning rate schedule with step size of 1 epoch and gamma of 0.8 (Initial

Lerning rate = 0.001). Model is trained for 40 epochs with a batch size of 8. The model is trained on 70 images. The model is evaluated on 14 images. The evaluation metrics used are precision, mean intersection over union (mIoU), and pixel accuracy.

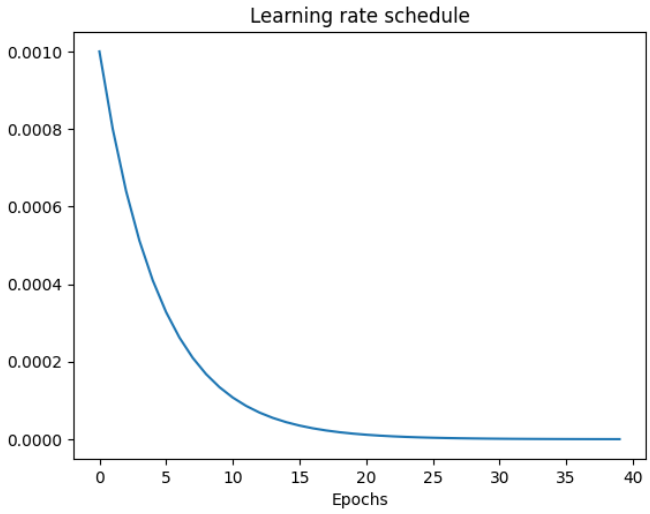


Figure 6. Exponential learning rate schedule

3. Results

We trained our model for 40 epochs and the loss greatly reduced from 1.0 to 0.3. The reduction in loss after the 40th epoch was neglegible. The performance of model is tested using the following metrics: epoch loss, pixel accuracy, IoU score, Dice score, and F1 score. The model achieved 88.52% pixel accuracy (cumulative for all classes) after training on train set. IoU score, Dice score, and F1 score values for train set can be found in Table 1. The pixel accuracy on test set is 81.85%. IoU score, Dice score, and F1 score values fort test set can be found in Table 2. We can see, from the plots, the model performs best on the background class, followed by the muscle layer class, electrode class and mucosal layer class. The high difference between the metric values of classes indicated high bias in our data.

Table 1. Train set performance metrics

	Background	**Muscle layer**
IoU score	0.8312	0.6159
Dice score	0.4528	0.3423
F1 score	0.9056	0.6846

	Mucosal layer	**Electrode**
IoU score	0.2282	0.4474
Dice score	0.1339	0.2656
F1 score	0.2678	0.5312

Table 2. Test set performance metrics

	Background	**Muscle layer**
IoU score	0.7733	0.4744
Dice score	0.4338	0.2735
F1 score	0.8677	0.5470

	Mucosal layer	**Electrode**
IoU score	0.2081	0.3466
Dice score	0.1314	0.2167
F1 score	0.2628	0.4334

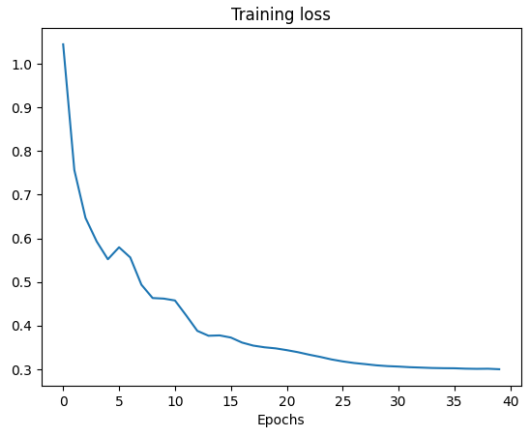


Figure 7. MobileNetV3 training loss

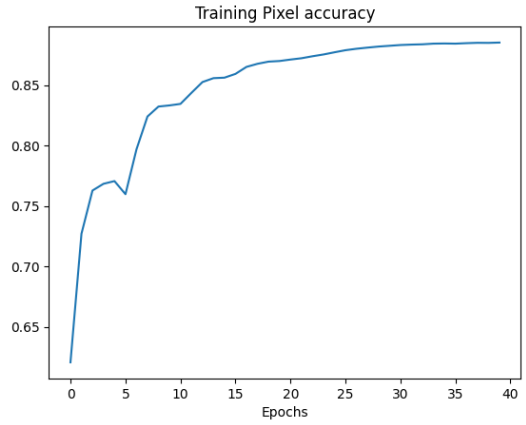


Figure 8. MobileNetV3 pixel accuracy

4. Conclusion

The pixel accuracy mentioned in the paper is 80.99% while the proposed model has achieved 88.52% on train set and 81.85% on test set, outperforming the model stated in the original paper in terms of performance and efficiency (since the proposed backbone has lesser weights). Although our model has better performance metrics, we have to consider fact that our model is trained on a very small dataset which

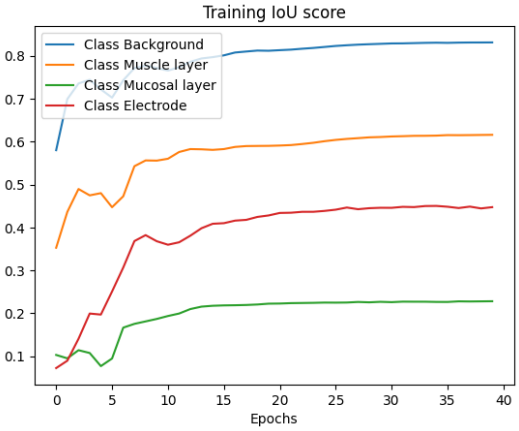


Figure 9. MobileNetV3 IoU score

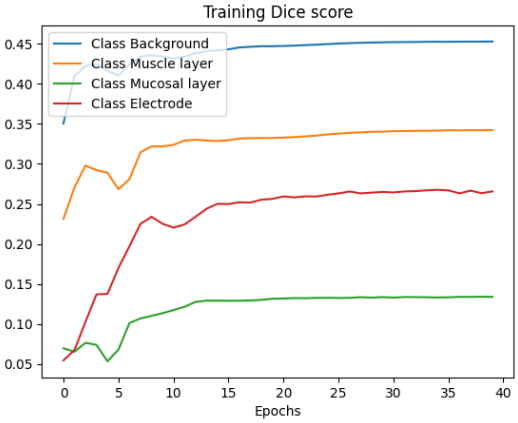


Figure 10. MobileNetV3 dice score

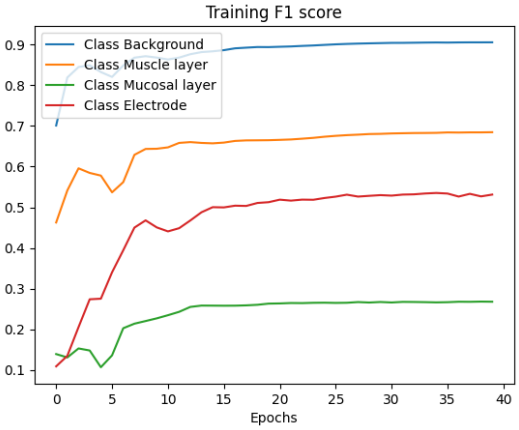


Figure 11. MobileNetV3 F1 score

significantly increases the risk of overfitting.

5. References

References

[1] DeepLabv3-architecture 2
[2] ASPP-images 2
[3] MobileNetV3-architecture 2
[4] MobileNetV3-comparison 2
[5] https://segment-anything.com/ 1