Liver Tumour Segmentation using UNet

Ravi Theja Gunti

2020

Abstract

This report summarises the work done so far for Liver tumour segmentation.

1 Introduction

2 Literature Research

About Liver tumor Detection : Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks.

Working with 3D Data : V-Net - Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation

Working with patch data : From Patch to Image Segmentation using Fully Convolutional Networks

Transfer Learning : Transfusion - Understanding transfer learning for Medical Imaging

Detection and Localization :

3 Data Set:

The Dataset consists of 54 anonymised Liver MRI scans, with liver and lesions segmented by radiologists. The ground truth is marked as 0 for background, 1 for liver, 10 and above for lesions, each integer corresponding to a unique lesion. Voxel spacing is 1x1x2.5mm. Data is split into train(44) and validation(10).

3.1 Opportunities

Following are the observations that model can take advantage of:

- All records are with same voxel spacing.
- Clear boundaries marked for liver and lesions.

3.2 Challenges

- Limited number of patient records : 54
- For a few records, histogram of intensity distribution plots of liver and lesions overlap to a large extent, which makes it difficult for model to discriminate based on just intensities.
- A few lesions are very small in size (<150 voxels)

3.3 Data Preprocessing

Each record is re-sampled from 1x1x2.5mm to 1x1x1mm. Non liver voxels of each record are set to 0 using the available ground truth(mask). The mask is now altered to have value 1 for lesion, 0 for live and background.

4 Implementation and Design Considerations

Two cascaded CNNs are used, first one to extract the liver and second one to segment lesions from Liver ROI obtained from first model. As a baseline for second model, UNet and VNet architectures are considered.

BatchGenerators is used for augmenting data as given below. Due to GPU resource constraints, patchsize of 128x128x64 is chosen. Running on Google Colab(https://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-PCIe-datasheet.pdf)

4.1 Data Augmentation

Data augmentation is done on the fly using BatchGenerators. Following Augmentations are done on each patch.

- Random Crop
- Elastic Deform (0, 0.25)
- Scale : (0.8, 1.2)
- Rotation along axes : (-5 to 5)
- Mirror transform along axes
- Gamma Transform
- Gaussian Noise

5 Initial Results

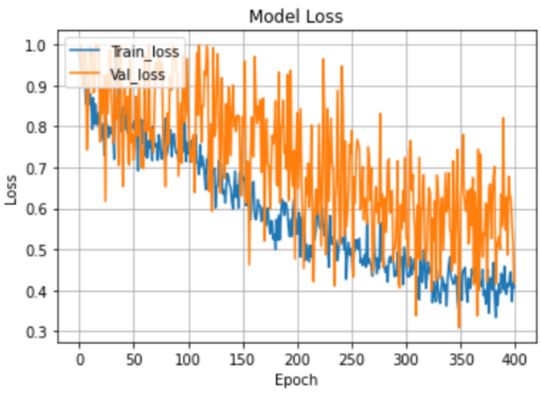

Figure 1: Dice Loss

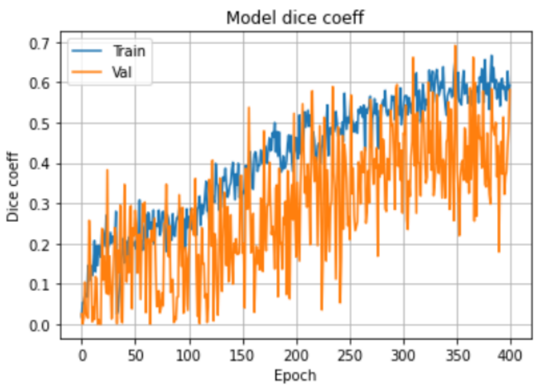

Figure 2: Dice Score

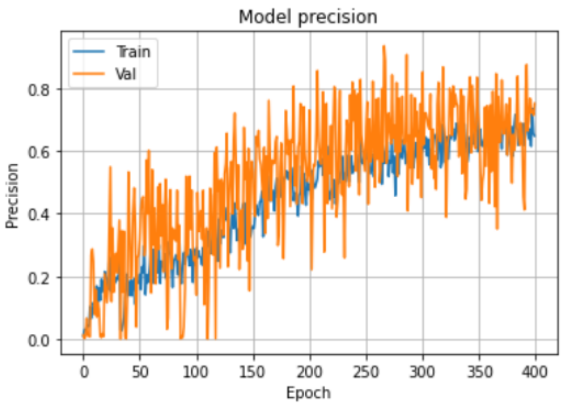

Figure 3: Precision

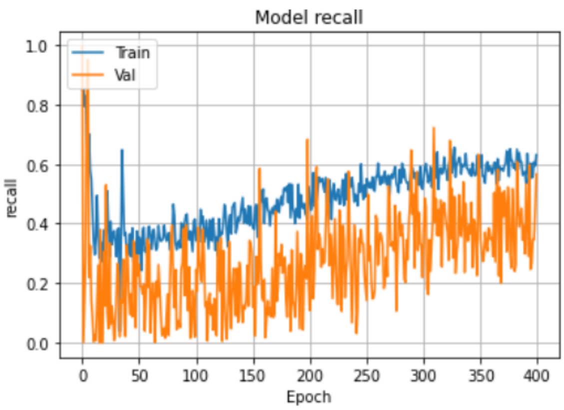

Figure 4: Recall

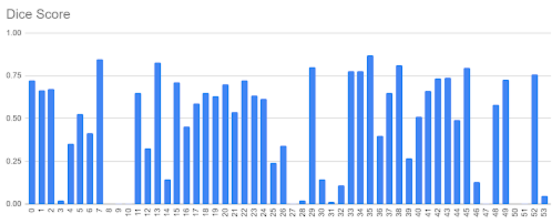

Figure 5: Per patient Dice Score

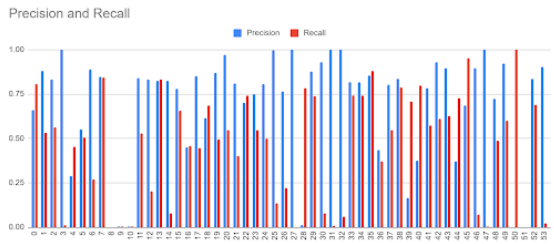

Figure 6: Per patient precision recall

6 Observations :

1. A few records have very less dice score. This is due to the lesions that are comparatively very small (<5%). To address this issue, one approach could be to add Inception level modules in the backbone.

7 Tasks in Queue :

1. Compare the performance with object detection (using the library MedicalDetection Toolkit). The library provides implementations for RetinaNet and a few other single stage detectors. Adding YOLO to the library in progress.