

# Tissue Segmentation and Quantification

Victor Morland u1803998

March 2022

## Introduction

The aim of the project is to train a semantic segmentation model using Unet [1] on the ORCA [2] dataset. The ORCA dataset contains digital tissue images (HPF) of TCGA head and neck cancer cases and their corresponding masks. The aim is for our model to segment tissue into tumorous tissue, non-tumorous tissue and background (corresponding to classes 2,1 and 0).

The closer our segmentation model to the true values (the masks) the better. An example of such an HPF is shown in figure 1. The white regions are tumorous tissue, the grey regions are non-tumorous tissue and black is background.

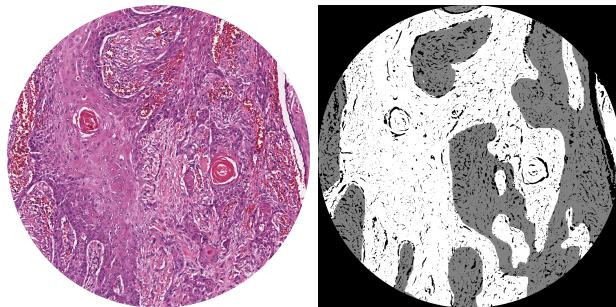


Figure 1: Example of a HPF and its corresponding mask from ORCA dataset

We will then apply this model to HPFs from 10 whole slide images (WSIs) and comment on the results. Five of the WSIs are from HPV positive cases and five are from HPV negative cases. Much of the code for this project was taken and adapted from Lab 3 Part 2 [3].

# 1 Training the Segmentation Model

## 1.1 Separating images into patches

Firstly, the images and masks from the ORCA dataset were downloaded and separated into training and validation data. This was done by using the validation dataset from the ORCA dataset as our training data and the testing dataset from the ORCA dataset as our validation data. Then the images and masks were separated.

Before just using our images into our Unet model for training we first split the HPFs (both the training and validation images and masks) into patches this makes them much smaller which allows us to input them into our Unet model which we cannot do with the larger HPFs due to memory constraints.

Before converting our images/masks into patches they will become 3d array where each 3rd dimensional array represents the RGB value for each pixel. Our masks however will be 2d arrays were each value will be 0,1 or 2. Representing our three classes black - background, grey - non-tumorous, white - tumorous. This was done by finding the value of grey in the grey scale masks images representing non-tumorous tissue as 129 (this was found by plotting a frequency count of the different values). While white and black were 255 and 0 respectively.

We then pad our images/masks to standardize the size of all the images/masks. Most of the images/masks from the ORCA dataset are 4500 x 4500 pixels but some are 2250 x 2250 and 1000 x 1000. We standardize all the images/masks to 5000 x 5000 by using dynamic padding. Padding has the added benefit of allowing the edges of the images, which are not normally analyzed as well to be more centred.

We then separate the images and masks into patches of size 500 x 500 with a stride length of 250 so there are no overlapping patches.

We now have two sets of data stored in two different pytables (the training and validation tables) which allows us to now ignore the images/masks and just work on the patches stored in the pytables.

## 1.2 Training Unet Model

We now feed the patches into our Unet model (with depth=5) for 10 epochs. Higher epochs can to achieve better results but in the interest of time 10 epochs were used. Additionally, we reduce our patch size to 64 x 64 pixels, this does reduce the quality of the images but again allows us to run our model significantly faster (halving the resolution of the patches lead to a 2x reduction in the running time of the Unet model). We also feed in our patches in batches of 64, this means that 64 patches are fed into the model before the weights and filters

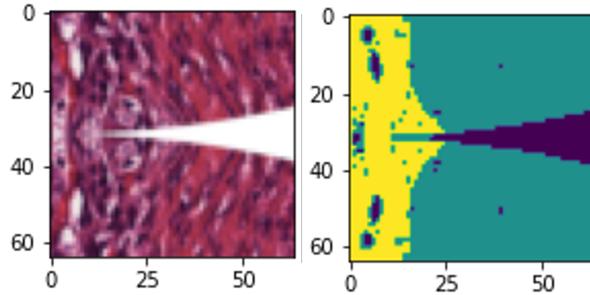


Figure 2: Example patch image and its corresponding mask

are updated using the averaged results from all 64 patches.

During every epoch we calculate the loss of the current model against a validation batch. If the current model beats the previous best model’s performance than the current model’s state is saved and becomes the best model. After all ten epochs we pick the model with the best performance as our model to be used in the next stages.

### 1.2.1 Evaluating Performance

We use the dice coefficient to measure the performance of our model (the dice coefficient ranges from 0 - worst to 1 - best). The dice coefficient takes is  $2 \cdot \frac{\text{overlap}}{\text{totalpixels}}$ . Where the overlap is the area where for a specific class the predicted mask and true mask agree. This score is then averaged for all three classes to get a multi class dice coefficient. For our best model averaged over the validation dataset is:

Mean Dice Coefficient: 0.8660269694010416 +- (0.1483506353447467)

Figure 3: Mean and St dev of Dice Coefficient on Validation dataset

We also plot the box plot of the dice coefficient scores. The box plot is comparatively tall which suggests that are model performs very well on certain images and very poorly on others. To explore this further we found the images with the best and worst dice coefficient scores. *Note Yellow is tumorous tissue, green is normal tissue and purple is background.*

The best classified image (figure 4) is a very simply image with only 2 classes in it background and tumorous areas. With the background taking the large majority of the image. This makes it very easy to classify (an untrained human could probably classify this).

While the worst classified image (figure 5) has all 3 classes with tumour and non-tumorous areas overlapping, which makes it much harder to classify.

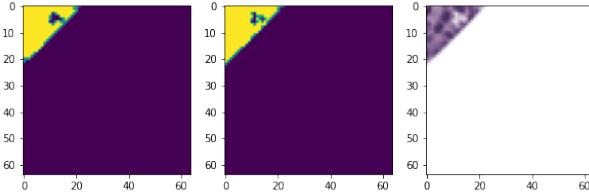


Figure 4: Best classified image with dice coefficient of 1.0

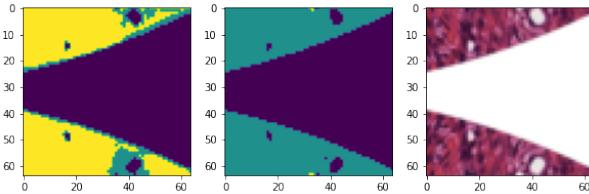


Figure 5: Worst classified image with dice coefficient of 0.333

## 2 Segmentation of HPFs in WSIs

We now take our WSIs and using a heuristic get 5 HPFs from each WSI to input into our best model. The HPFs are 512x512 pixel regions at 20x zoom. Some WSIs are at 40x zoom so we go to level 1 to get the 20x zoom.

The heuristic used is to check if the number of unique pixel values (in an image) is greater than a certain threshold (a threshold found after experimentation was 90). This gives us confidence that we have not just picked background for our region. Figure 6 is an example of such an image.

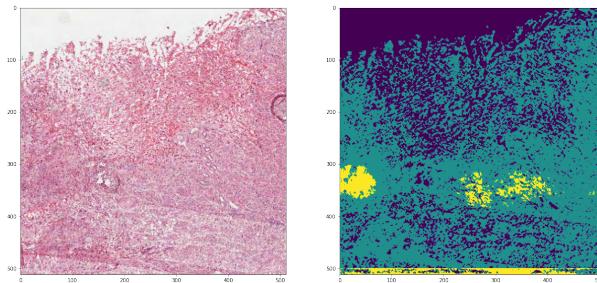


Figure 6: Example of a 512x512 region at 20x zoom from WSI with predicted mask

Using the tiatoolbox [4] and the already trained model from part 1, we get an output. An example is shown in figure 6. Just from a visual inspection (disclaimer not a doctor), it seems tumorous tissue is being detected on on the

edges too strongly, but otherwise the background/tissue detection seems to be pretty good.

### 3 Tissue Stats

To compute the areas of tumour and non-tumour regions is easy, it is simply required to count the number of pixels of each type in our predicted mask for the whole WSI. However, for the connected tumour and non tumour components the graph is too large and a recursion error occurs while running DFS to find the connected components. Therefore, we will manually count the connected components for each WSI (ignoring tiny components).

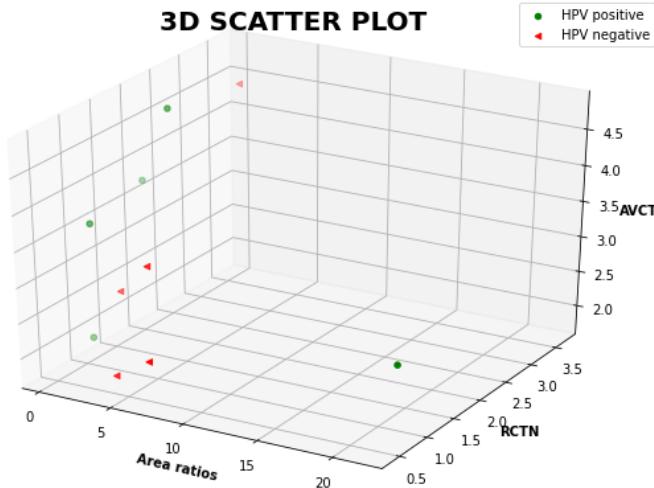


Figure 7: Scatter plot of RTNA, RCTN and AVCT

From this data it is very hard to see if any of the features are relevant to HPV classification. This is restricted by the number of HPFs taken from each WSI. But from the scatter graph in figure 7 it seems that these features do not help in classification.

### 4 Conclusion

Our predicted mask is definitely not at an acceptable quality to be used in a clinical setting but the quality can be easily improved by increasing the resolution of the patches fed into the Unet model (currently they are 64 x 64 which is very low). Additionally, we can run the model for many more than 10 epochs as the loss did not seem to converge after 10. However, this is beyond the scope

of this report as it would take far too long to train.

Additionally, the heuristic for picking areas of high cellularity in a WSI can be greatly improved as we often pick very similar regions and can still pick areas which are mainly background.

There are more difficult issues to deal with as well such as taking each HPF individually and ignoring it's location comparative to others and combining that information to make better predictions. This is a limitation of splitting the WSI into HPFs and then splitting those into patches. Perhaps a different representation of the images such as in a graph would allow us to maintain relative locational data.

## References

- [1] jvanvugt. *Unet*. URL: <https://github.com/jvanvugt/pytorch-unet>. (accessed: 16.03.2022).
- [2] F. Martino et al. *Orca*. URL: <https://sites.google.com/unibas.it/orca>. (accessed: 16.03.2022).
- [3] Nasir Rajpoot. *Lab 3 Part 2*. URL: [https://colab.research.google.com/drive/1f3Ta11CE26GE20KLFqVmTMUltBkhTHyk?usp=sharing#scrollTo=uj\\_HwJHJ3M6g](https://colab.research.google.com/drive/1f3Ta11CE26GE20KLFqVmTMUltBkhTHyk?usp=sharing#scrollTo=uj_HwJHJ3M6g). (accessed: 16.03.2022).
- [4] shaneahmed. *TiaToolbox*. URL: <https://github.com/TissueImageAnalytics/tiatoolbox/tree/develop/examples>. (accessed: 16.03.2022).