CS424 Coursework 3

Matthew Wight (2008398)

March 12, 2024

Overview

All code for this assignment can be found in `CS424CW3.ipynb`, inside the submitted zip file. The notebook was created in Google Colab. Model weights for the two models are stored inside `patch_cnn.pt` and `wsi_rfr.pkl`; the corresponding paths in the notebook must be set correctly in order for the program to run properly. Results are given in this report as well as in `Results.pdf`.

Task 1

I used the `WSIReader.tissue_mask()` function in TIAToolbox to complete this task. Below are the WSIs and corresponding tissue masks for the first two images in the `wsitils/` folder.

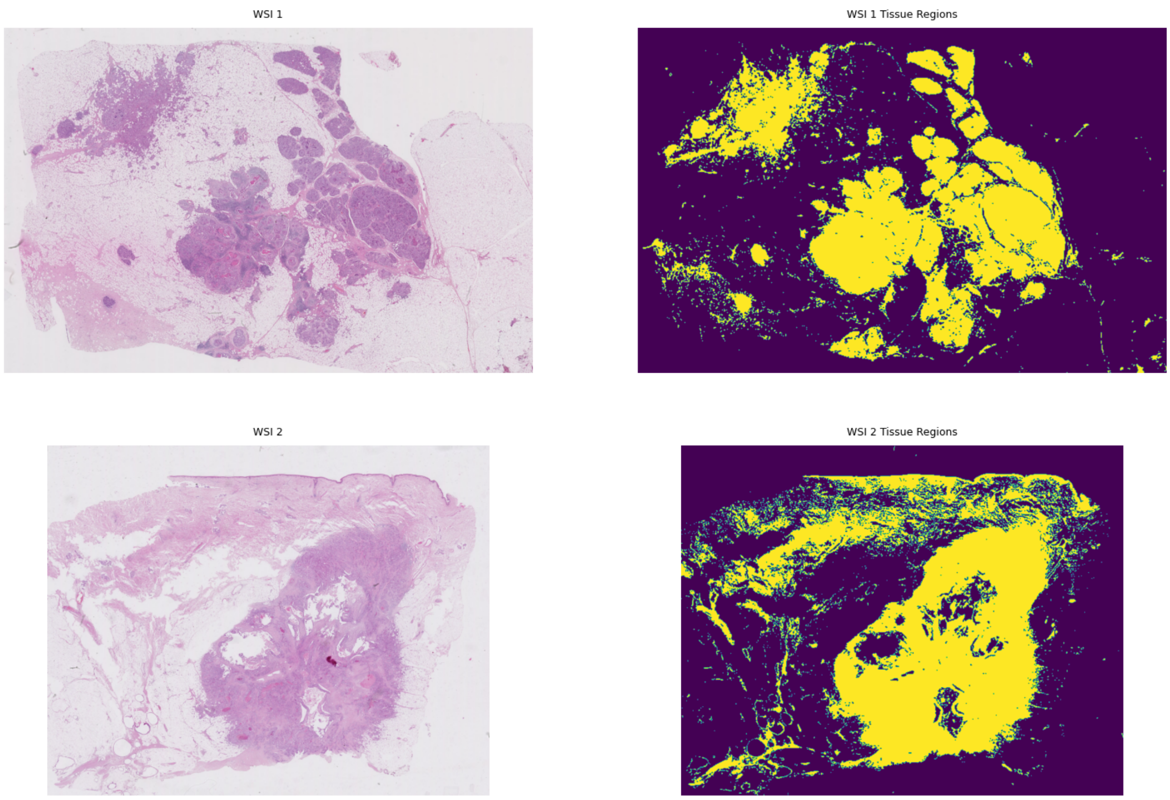

Figure 1: Two WSIs and segmented images.

Task 2

I used a patch-based learning approach to complete this task. I began by extracting patches correlating to regions of interest on the whole slide images. This was achieved by locating square regions on the tissue mask which contained a high proportion of foreground (over 70%). The corresponding patch locations on the WSI were then extracted and stored, along with the TILs score for the WSI. Most images contained around 100 usable 1024×1024 patch regions, which I then downsampled by a factor of 4.

Once the region-of-interest patches were obtained, I tried several different approaches. Initially I experimented with using the annotated patches in the `tissue_segmentation` and `cell_detection` folders as training data, but I was unable to achieve better than random performance with this method (potentially because each folder only contains 50 patches). I then opted to use the automatically extracted patches as training data, with the TILs scores as labels.

I experimented with a multi-head attention transformer neural network model to make predictions on the patches. I constructed and trained a model inspired by one featured in CS429: Data Mining and Machine Learning (Minhas, 2024). However, the performance of the model on the collected data was poor, achieving a Spearman rank coefficient of 0.240 when plotting the patch prediction scores against their true labels. Instead, I trained a simpler convolutional network and found it to perform marginally better.

The final CNN model features two convolutional layers and a single hidden fully connected layer, all using a RelU activation function. The output layer has a single channel, using an MSE loss function to predict the continuous TILs scores for the patches. The patches are downsampled from 256×256 to 64×64 to reduce training time; patches are also normalised to have 0.5 mean and standard deviation intensity values. The learning rate used in training was 0.01 and the model was trained for 5 epochs.

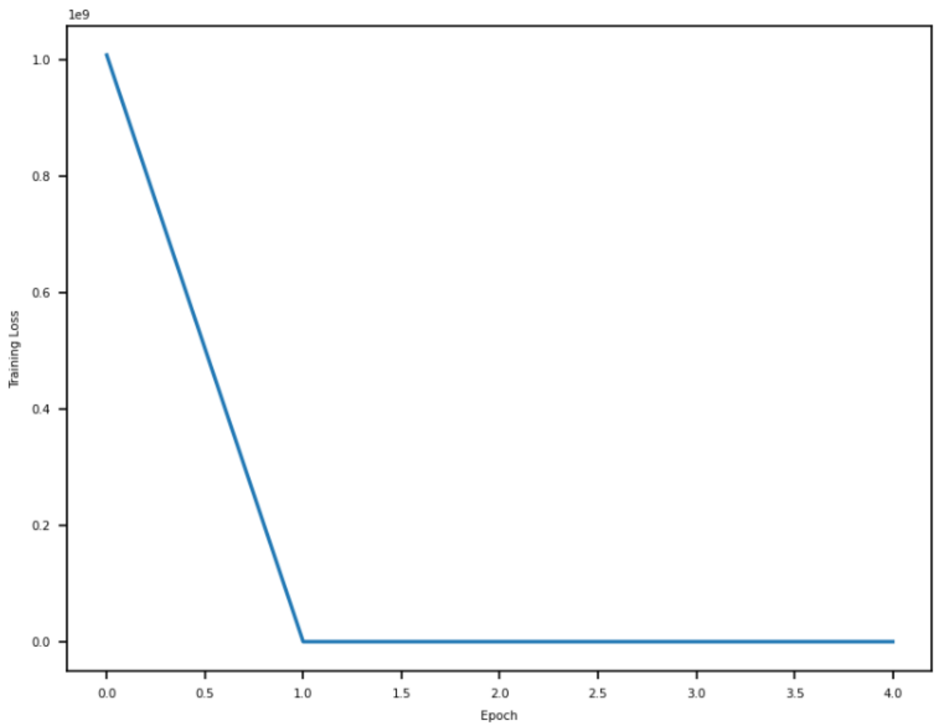

Figure 2: Convergence plot of the CNN for patch prediction.

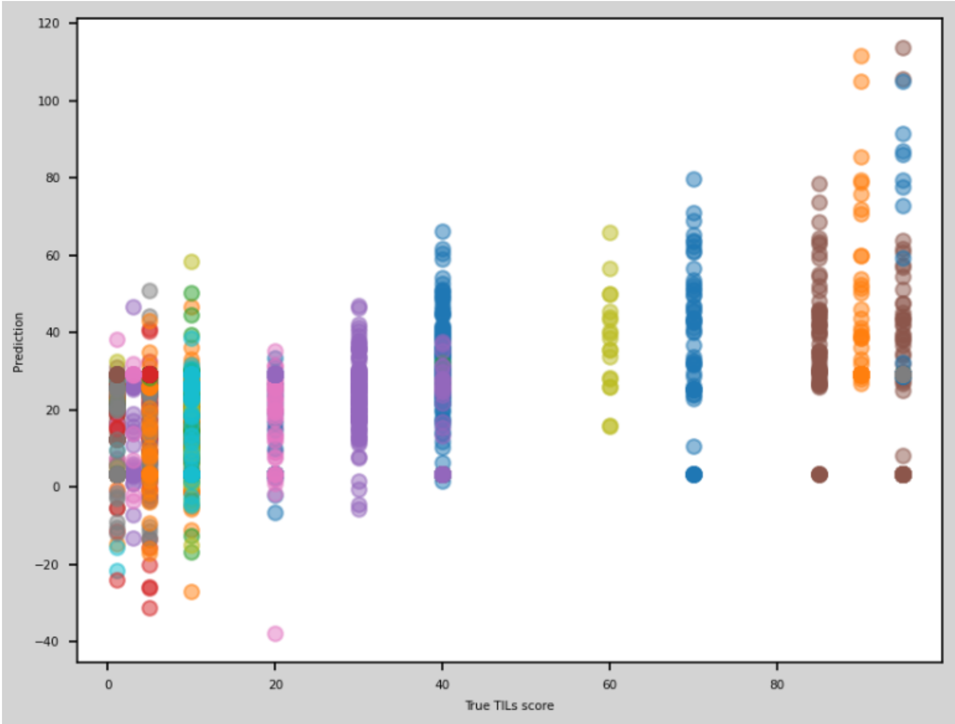

Figure 3: Predicted and actual TILs scores for each patch.

The above figure shows the scatter plot of predicted values for each patch against the actual TILs score for its WSI. The trained model achieved a Spearman rank coefficient of 0.416. While this is somewhat low, I believe that it is a reasonable score given the nature of the training data (since many patches will have a lymphocyte count/area different to what is suggested by its label).

Once the patch predictions have been calculated, there are several potential methods of aggregating the TILs scores for an entire WSI. A simple idea is to take the mean TILs score prediction across all patches for a WSI. This makes logical sense because the TILs score is based on the area of the tissue covered by the lymphocytes (not the number of lymphocytes). However, I found this method to give poor quality results. Instead, for each WSI, we can take the k highest patch scores, order them, then train a machine learning model on the patch scores, aiming to predict the WSI TILs score.

After testing different ML models, I found that a random forest regressor performed the best, achieving a Spearman rank coefficient of 0.774 when given the 8 highest TILs score predictions for each WSI.

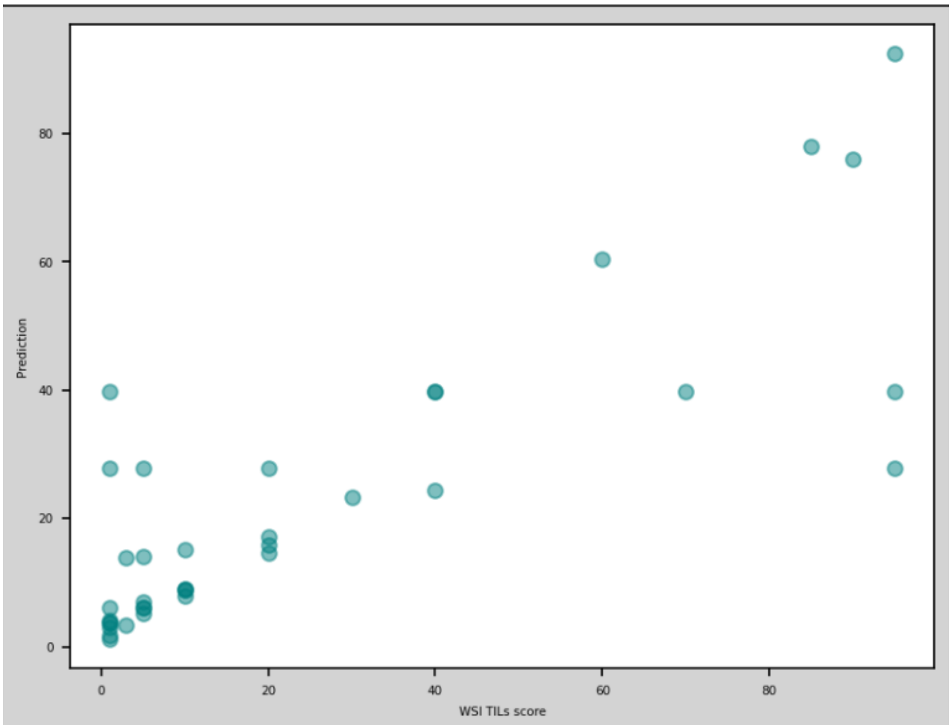

Figure 4: Predicted and actual TILs score for each labelled WSI.

The final predicted TILs scores for the unlabelled WSIs are given in the table below.

Image ID	TILs score prediction
TC_S01_P000055_C0001_B101	69.9
TC_S01_P000058_C0001_B101	27.8
TC_S01_P000061_C0001_B101	27.8
TC_S01_P000063_C0001_B101	27.8
TC_S01_P000064_C0001_B101	14.3
159S	13.0
160B	39.8
161S	12.3
162S	84.5
165B	39.8

Table 1: Final TILs score predictions.

Task 3

I have included visualisations for three WSIs in this report. I struggled working with the TIA-Toolbox visualisation features, so used Pyplot to render them instead. More overlays can be generated using the notebook.

The first figure shows a WSI with a TILs score of 10. The annotation shows that most of the ROI patches were given a low TILs score prediction, in line with the true score.

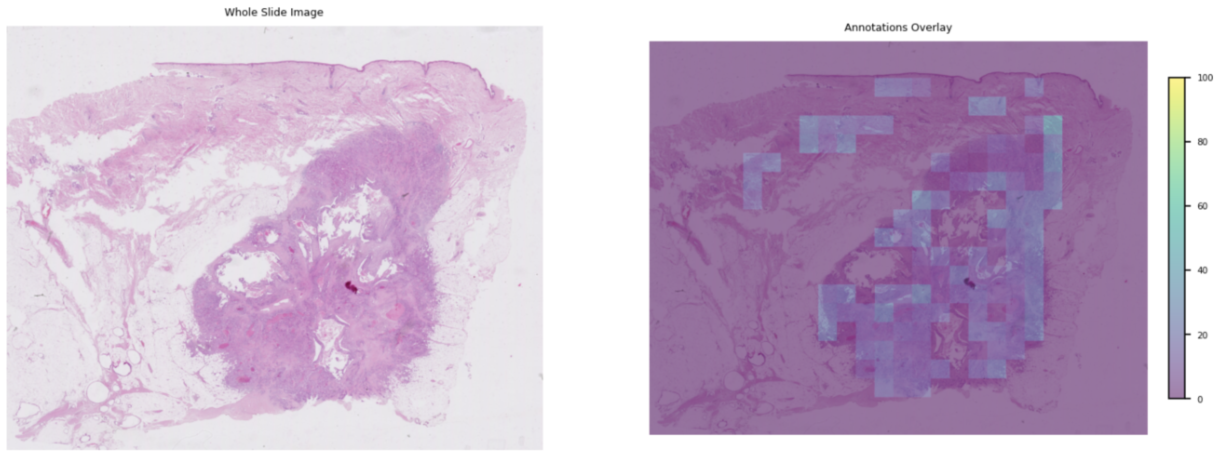

Figure 5: WSI and annotated WSI with manually determined TILs score 10/100.

The figure below shows a WSI with manually determined TILs score 30. The annotated image shows a large area of tissue with relatively low TILs scores given per patch, indicating an accurate set of predictions.

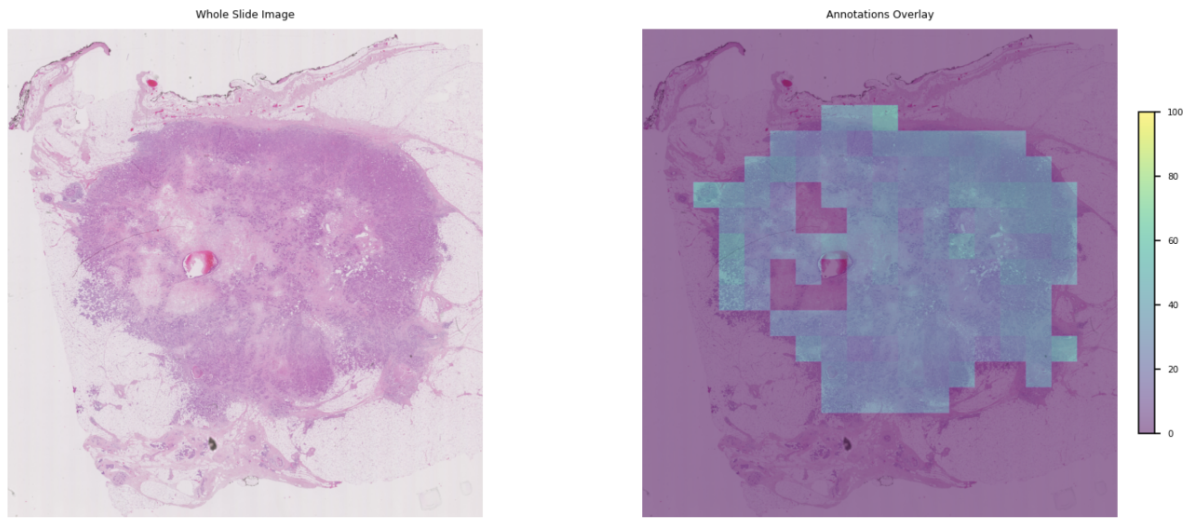

Figure 6: WSI and annotated WSI with manually determined TILs score 30/100.

The WSI in the next figure was manually given a TILs score of 60. The CNN model also seems to have detected a higher proportion of patches which it believes to correspond to higher TILs scores—this can be seen in the annotated image.

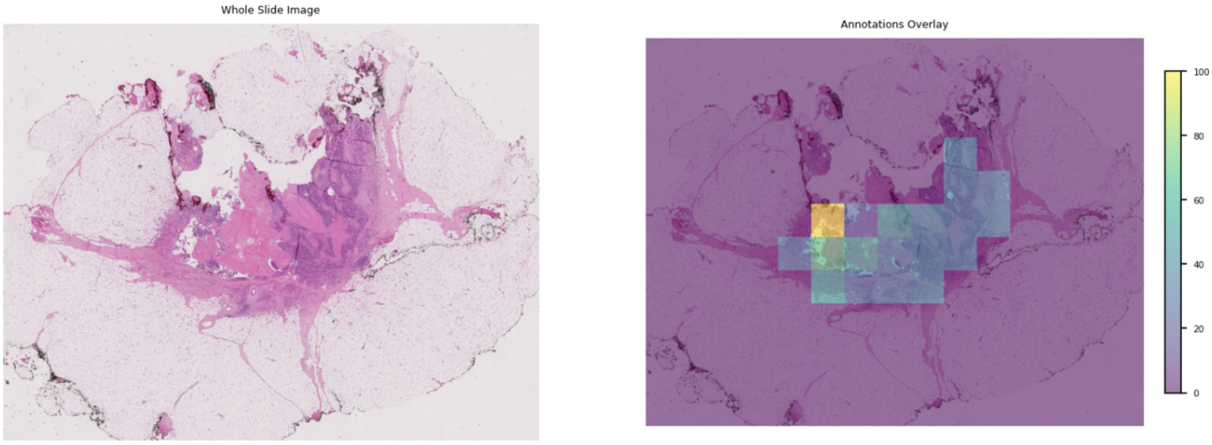

Figure 7: WSI and annotated WSI with manually determined TILs score 60/100.

References

[1] J. Brownlee. How to choose loss functions when training deep learning neural networks, 08 2019. URL https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learnin

[2] F. Minhas. Transformers for mnist: A foundational tutorial, 2024. URL https://github.com/foxtrotmike/CS909/blob/master/mnist$_t$*ransformer.ipynb*.

[3] M. R. Segal. Machine learning benchmarks and random forest regression, 2004. URL https://escholarship.org/uc/item/35x3v9t4.