**Machine Learning: Assignment 2**


- Akshita Agrawal – s3933878
- Penusha Udapamunuwa – s3925899


## Introduction

Our tasks for this report were to build two machine learning models. One to classify images according to whether the given cell image is cancerous or not and one to classify images according to cell type (fibroblast, inflammatory, epithelial and others).

Initially we performed an exploratory data analysis on the given dataset for both tasks. From this we observed that the number of non-cancerous cell images is higher than the number of cancerous cell images and the cell type images were not evenly distributed among fibroblast, inflammatory, epithelial and others. As a result, we decided that the most suitable evaluation metric for our model would be the F1 score as it provides the mean between recall and precision where precision measures the proportion of true positives over the total number of predicted positives, while recall measures the proportion of true positives over the total number of actual positives.

We used a data split of 60/20/20 for train, validation, and test data.

For our baseline model, we decided to implement a Convolutional Neural Network as it uses sparse connectivity between layers, which reduces the number of parameters required to train the model and reduces overfitting. It also learns hierarchical features from raw pixels, and they can be used for a wide variety of image-related tasks, such as object recognition, image segmentation, and image generation. This is ideal for us as we are using cell images for classification in the two tasks.

## Classify images according to whether the given cell image is cancerous or not

To begin this task, we created data generators for train and validation with a batch size of 32. Then we implemented a CNN with 3 convolutional layers on those generators with 25 epochs. As shown in Figure 1, the validation loss was increasing as the epochs increased and the validation F1 score was decreasing. This meant that the model was over-fitting on the data.

To minimize the effect of over-fitting, we applied regularisation. We used L2 regularisation as it has a lambda value that allows us to control the amount of regularisation applied which in turn reduces the model complexity and training error. Initially we used a lambda value of 0.01 but it had an adverse effect on the model. It had a higher loss and had a very low F1 score. We then decreased the lambda value to 0.001. We observed a loss of around 0.25 and a F1 score of around 0.9 as shown in Figure 2. This had minimized over-fitting and had improved the baseline model. To check for further improvements, we added a dropout layer to this to make the model less complex and minimize any more over-fitting but with this model the loss increased comparatively.

To see if the model could be improved further, we tuned certain parameters. As the model was giving us up to a 90% F1 score so far with 25 epochs we decided to keep it constant when improving our model. We added another layer to the above regularised model. This was not improving the model as the loss increased up to 0.68 and the F1 score was around 0. Next, we tried data generators with a batch size of 64 with the Figure 2 model. Although this had a F1 score of around 0.9, the loss was around 0.3. This meant that the model from Figure 2 was our best model in comparison with loss values and F1 scores from the other models.

## Classify images according to cell type

To classify images according to cell type we first created data generators for the training and validation data sets with a batch size of 32. We implemented the baseline model with 3 convolutional

layers with 10 epochs. As shown in Figure 3, the model did well for the first few epochs but then showed signs of over-fitting towards the end. The loss for the training and validation datasets also began to differ towards the end.

Regularisation is a method we applied to minimise the effect of over-fitting. We used L2 regularisation it can control model complexity because it adds a penalty term to the loss function of the neural network and avoids overfitting by limiting the capacity of the network to fit the training data too closely. The strength of the L2 regularisation penalty is controlled by a hyperparameter lambda. The initial lambda value we used was 0.001 on 15 epochs. This model performed well because it did not over-fit the data. We also tried using different lambda values (0.0001 & 0.00001) to see if the model's performance improved, but they ended up over-fitting the data.

Data Augmentation is a useful technique to increase the size and diversity of the training data, which can then improve the performance of the model. By applying data augmentation methods such as rotation, flipping and zooming, new training samples can be created that is similar to the original data but with some variations. To improve our model's performance, we used rotation to generate new variations of the training data. We used the same lambda value we found before (0.001). The first model we tried had 15 epochs. This model didn't improve the performance much, so we applied Hyper-Parameter Tuning to fine tune the model and increased the epochs to 50 and then 150. Both those models had better performances, but it was overfitting the data. For the model with 150 epochs, we decided to change the lambda value from 0.001 to 0.0001 and see if that would solve the overfitting and it did as we can see in Figure 4. We had a loss of about 0.3 with an F1 score of about 0.74.

Since this was best performance we got without over-fitting, we decided to use this model as our final model and test it on our unseen data. The model performed well with a 0.33 loss and 0.70 F1 score.

In addition to this, we were also given another dataset which did not contain labels for the cell type. We were required to explore how this dataset could improve our original final model. Using unlabelled data to improve a neural network model trained on labelled data can be done using a method called Semi-Supervised Learning. The technique we used to apply semi-supervised learning is by having our model generate pseudo-labels for the unlabelled data and using them as if they were true labels.

We predicted the pseudo-labels for the unlabelled data and then combined the labelled and unlabelled data to generate a new dataset. We then trained our model in this new dataset. Since we had more data to train the model on now, in theory the model should improve even though we are using pseudo labels.

When we ran our model on the new dataset, we saw that the F1 scores for both the training and validation datasets decreased from 0.70 (original final model) to about 0.53 (this model). We also observed over-fitting in this model, so we used regularisation to fix that. We used a lambda value of 0.00001 which minimised the over-fitting. In Figure 5, we can see a F1 score of 0.55 and a loss of 0.46.

In total we tested 9 models to determine which one was best to classify the cell types.

## Ultimate Judgement

Using the best model for detecting cancerous cells we obtained from classifying whether images are cancerous or not, we tested it on the unseen test data from the initial splitting. This gave us a loss of 0.27 and a F1 score of 0.905. This indicates that the model is performing well on unseen data. Though the F1 score is high there is still some room for improvement on the loss values. This can potentially be done by using a larger dataset to train the model and adding or removing layers according to how the model is performing.

To classify cell types, we have two final models from two datasets. Upon testing the test (unseen) data with our first final model (the one trained with only labelled dataset, Figure 4), we reported a F1 score of 0.706 and a loss of 0.33. When the second final model (the one trained on the combination of labelled and pseudo-labelled dataset, Figure 5) was tested, the F1 score was 0.553 and the loss was 0.44. Based on these results, we would recommend using the first final model (the one trained on just labelled data, Figure 4).
There are ways to improve this model. For starters, a larger labelled dataset could be beneficial for the model to have more data to train on; using different model architectures other than convolutional neural networks might also yield different results; etc.
While in theory, the second final model should have done better, there are a number of reasons why it didn't. We generated pseudo labels on a model with a F1 score of 0.74. If the F1would have been higher, the pseudo labels would have been more accurate. More labelled data could have been used to train the model before generating pseudo labels etc. Due to limitations of resources, we were unable to improve the semi-supervised model for the purposes of this assignment.

**Analysis & Comparison between the two categories**

Upon analysing our two final models, we observed that the isCancerous model achieved a higher F1 score compared to the cell type classification model. The isCancerous model, which had 2 classes, reached this higher score quicker than the 4-class cell type classification model. The increased complexity of the multi-class problem and the smaller sample size for each class in the cell type model could have been contributors to its lower performance. Other factors may also have influenced the better performance of the isCancerous model. In conclusion, the isCancerous model outperformed the cell type classification model in terms of F1 score.

**Independent Evaluation**

In the publication "*Handcrafted features with convolutional neural networks for detection of tumor cells in histology images"*, a SC-CNN was implemented with MatConvNet toolbox to detect cancerous cells using image classification by adding handcrafted features to the raw data. Similar to our model, they used precision and recall, and combined them to calculate the F1 score as their evaluation metric. In their final proposed model, they achieved a F1 score of 0.75. This is less than our F1 score of 0.90. This could be because we are only feeding the raw data to model instead of using extra handcrafted features. Using extra features would improve accuracy when using larger datasets with more unseen data.

In another publication, "*Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images,*" they used a lot of different models, including a SC-CNN model, for nuclear classification & cell-type classification on two datasets; CoNSeP and CRCHisto. We are only concerned with the results for the CRCHisto dataset, since that is the one we are using. Unlike us, they found the F1 score for each cell type individually. We can get the average F1 score for their HoVer-Net model since that is their proposed model. The average F1 score they reported is 0.52625. This is less than our F1 score of 0.706. This could be because they are using a different model which perhaps is less suitable to classify cell types; and could also be because their model uses 50 layers and have removed the subsequent max-pooling operation unlike our model.

The publication, "*Detection and Classification of Calls in Colorectal Histology Images Utilizing U-Net Machine Learning Architecture*", also utilises the dataset given to us to detect the different cell types using the U-Net neural network architecture. Similar to us they are also using F1 Scores to evaluate their model. Their model achieved a F1 score of 0.745. This is slightly better than our score of 0.706. A reason their model did better can be because they used a difference model architecture, and therefore the model was trained differently.

## Appendix

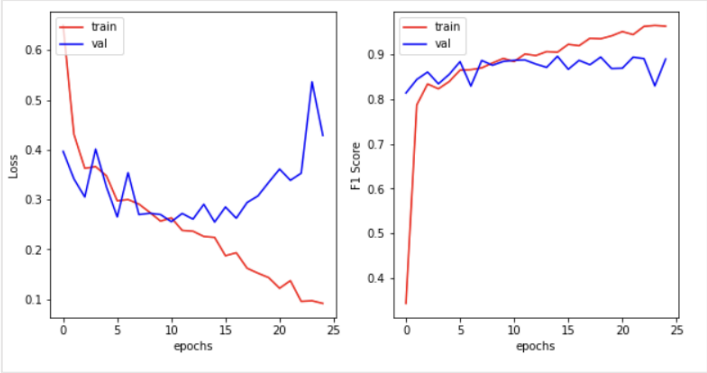


*Figure 1: Baseline Model for isCancerous classification*

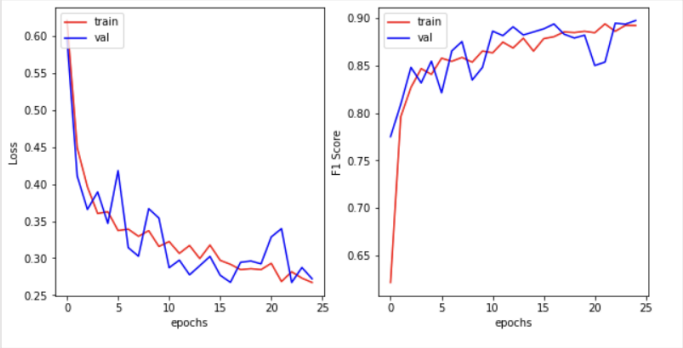


*Figure 2: Final Model for isCancerous classification*

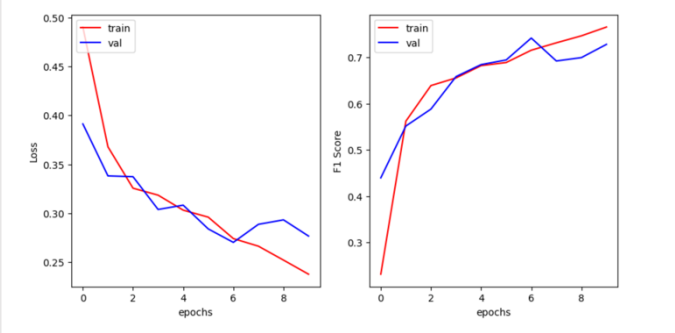


*Figure 3: Baseline Model for cell type classification*

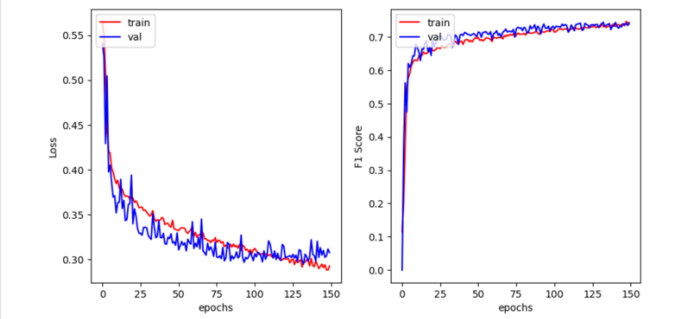


*Figure 4: Final Model 1 for cell type classification on labelled dataset*

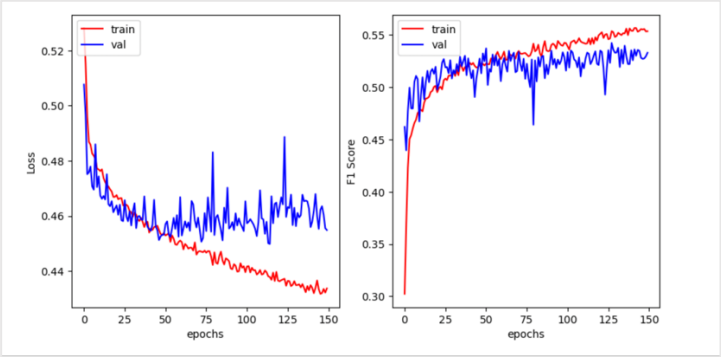


*Figure 5: Final Model 2 for cell type classification on combined dataset*

## References


- Gates, Q. D., Allali, M. and Martin-King, C. (2021) "Detection and classification of cells in colorectal histology images utilizing U-net machine learning architecture," *SSRN Electronic Journal*. doi: 10.2139/ssrn.3981501. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3981501
- Graham, S. *et al.* (2019) "Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Medical image analysis*, 58(101563), p. 101563. doi: 10.1016/j.media.2019.101563. Available at: https://www.sciencedirect.com/science/article/pii/S1361841519301045
- Kashif, M. N. *et al.* (2016) "Handcrafted features with convolutional neural networks for detection of tumor cells in histology images," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, Prague, Czech Republic, 2016, pp. 1029-1032, doi: 10.1109/ISBI.2016.7493441. Available at: https://ieeexplore.ieee.org/document/7493441