Jainam Doshi (s3825891)
Adhiraj Jain (s3821245)

# Classify Images of Cancer

The use of machine learning technology in the biotech industry has rapidly increased over the past decade. The ability to have powerful predictive models that can identify potential issues in the human body has been a breakthrough and has helped save countless lives. This report will be focusing on the approach taken to solve the problem of identifying if the cell is cancerous or not-cancerous (Task 1), and the problem of classifying cells into cell types (Task 2).

In essence, this is an image classification problem and there are many ways to tackle the issues. The first thing comes to mind when approaching a model to train on images is neural network (NN) models (Sanghvi, 2020). The ideology of NN machine learning (ML) models is to simulate the way of our nervous system operates. There are many different types of supervised NN algorithms for image classification, such as feed forward neural network, Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Modular Neural Networks (MNN), etc.

The goal for task 1 is to construct a NN model, where the model's loss function is less than 0.2 and is able to correctly classify at least 90% ($\geq$ 0.9 of ROC) on unseen data. On the other hand, the goal for task 2 is to build a NN model, where the model's loss function is less than 0.3 and can correctly classify at least 80% ($\geq$ 0.8 of F1 score) on unseen data.

The performance metrics being used for the evaluation of models is the loss value and the Area under the curve (AUC) of the Receiver operating characteristic (ROC) curve. The loss values are derived through the utilization of the loss function, and these values play a crucial role in assessing the overall performance of the model. Additionally, the AUC-ROC values can indicate whether the model is exhibiting under-fitting or over-fitting tendencies. ROC curve is very useful in medical related fields and a great performance metric for a binary classification problem because ROC values show the accuracy of how good the model predicts for *True Positives* and *True Negatives* (Nellore, 2015). This reflects to be industry standards, as it very vital that patients with cancerous cells are classified as cancerous and patients without cancerous cells are classified as non-cancerous.

For task 1, a good initial model for classifying cells as cancerous and non-cancerous can be an MLP model. One of the main reasons to choose MLP as a baseline model (*model_1*), is that the dimension of the images is relatively small (27x27), so that the number of neurons created will be manageable without expending heavy computation energy (Dinesh, 2019). Furthermore, every neural network requires an optimizer that adapts the network's attributes, such as learning rate and weights (Vadapalli, 2022). However, choosing the right optimizer is sometimes very challenging, but one of the simplest and most common optimizer function used in binary classification is Stochastic Gradient Descent (SGD).

All NN models require to have a loss function to calculate distance between the actual and the predicted values. A lower loss value typically indicates that the model, on average, predicts values that are closer to the actual values, suggesting that the model is robust. As task 1 is a binary classification problem, the most suitable loss function to use in all the models is binary cross-entropy.

Based on the graphical depiction of *model_1(see figure 1)*, it can be inferred that the model is demonstrating decent performance. This is evident from the low loss values and high ROC values observed in both the training and validation datasets. While the loss values decrease and the ROC values increase for the training dataset, there is no significant disparity observed for the validation dataset beyond a certain *epoch*. These findings indicate that the MLP model might be slightly overfitting and there is room for enhancement.

According to researchers Jiudong Yang and Jianping Li, CNN models are very powerful in tasks related to image recognition and is an enhancement from MLP models. CNN is a deep learning algorithm and is designed to adaptively update the weights through backpropagation. Furthermore, CNN model excels at accurately extracting and analysing the topological features (colour, texture, shape, etc) of an image even when there are invariances within the images (Yang & Li, 2017). Hence, the next model will be a CNN model.

The CNN model (*model_2*) will have multiple hidden layers which includes: 3 convolution layers, 3 pooling layers and a full connected layer consisting of 2 dense layers. Having multiple convolution layers enables the CNN model to have a hierarchical decomposition of the input dimensions (Brownlee, 2019). Furthermore, for every convolution layer, the CNN model will implement a padding strategy to prevent any information loss on the border of the input dimensions as the convolution layers perform down sampling (Li, et al., 2022). Likewise, every convolution layer will also use *ReLU* function to introduce complex non-linearity to the network's decision-making process. *ReLU* is a common activation function, it selectively activates or deactivates neurons, hence, decreasing the computation power compared to other functions (AnalyticsVidhya, 2020). Finally, *model_2* will only be trained for 25 *epochs* because in *model_1*, the performance metrics did not have any noticeable changes after *epoch-25* and is showing signs of over-fitting.

The performance metrics of the *model_2* (see figure 2) shows that this CNN is just-right, and no longer over-fitting as the difference of the ROC values between the train and validation dataset is insignificant while still retaining high ROC values. Additionally, there is an improvement in the loss value from *model_1* to *model_2*. Fortunately, as this model is not over-fitting, any regularization techniques are not required to implement. However, the loss value for CNN model does not satisfy the goal. Therefore, further investigation is required on the tuning of hyper-parameters to improve the loss value.

A new CNN model (*model_3)* will be constructed carefully by tunning various hyper-parameters of *model_2*. One major change in the new CNN model is to use *Adam* as an optimizer. The advantage of using *Adam* is that it converges much faster on a multi-layer NN models than any other optimizer algorithms (Bock, et al., 2018). *Adam* also introduces a learning rate that can be tuned to improve the performance. Another tunning parameter that is applicable is the number of filters used at each convolution layers. Researchers Wafaa Ahmed and Abdul Karim claim that the number of filters have major effects on the correctness of the model and as per them, having three convolution layers with number of layers anywhere from 64 to 512 should have prominent improvement of the model (Ahmed & Karim, 2020).

Fortunately, KerasTuner library provides a random search tuner to find a model with best possible tuned hyper-parameter. The tunned hyper-parameters of CNN *model_3* is 0.000792 for the learning rate of *Adam* and 32, 64, and 128 for the number of kernels in first, second, and third convolutional layers, respectively. Additionally, data augmentation is good practice to implement to increase the data size by having different variants of original cell images.

After train and testing *model*_3 on the datasets, the performance metrics seems to be improved, especially the loss values have dropped down as compared *model_2 (see figure 4)*. Likewise, the ROC values have resulted to be more polished.

The methodology taken for task 2 is a combination of transfer learning and semi-supervised learning. Transfer learning allows to take leverage from existing pre-trained model rather than building a new model from scratch on the same dataset. Initially, the starting model (*model_4)* for this task will be a modified version of *model_3*. The main differences are the output classes and loss function is changed to categorical cross-entropy, resulting in a multiclass classification model. Therefore, the new CNN model is fitted with all the layers frozen, except for the last dense layer in the fully connected layer to include match the correct output classes with SoftMax normalization (Sultan, et al., 2019). Additionally, the performance metrics consists of F1 score instead of ROC as the F1 score is harmonic mean of precision and recall, giving a balanced overview of the model's robustness. After training of *model_4*, the performance metrics implies that the model is under-fitting with relatively high loss values as the model is not able learn the underlying patterns due to the increased complexity of the problem.

Providing a model with more datasets to train, can improve the performance of the model. So, a new improved CNN model (*model_5*) will train on an extra dataset. However, the data in the extra dataset is unlabelled. Hence, to produce *model_5*, semi-supervised techniques is applied, where the model will go through the process of unfreezing and tuning of multiple layers over various iterations, monitoring the performance metrics. However, after few iterations of unfreezing layers, the computational of the model has increased rapidly, so that the performance metrics is suggesting that the model has not reached the desired goal (see figure 5).

Independent Ultimate Judgment

For task 1, model_3 is the best model to classify cell images into cancerous and non-cancerous classes because performance metrics evident a high-performance model. Although the loss values are not strictly meeting the goals, but the difference margin is small. Therefore, this CNN model is adequate to deploy in real-life with extra minor improvement techniques.

For task 2, model_5 is the best model to classify cells into cell types. However, this model requires a lot of enhancements as the performance metric implies a that the model is very substandard and should be deployed into the real-world.

A similar approach also is taken by researchers Korsuk Sirinukunwattana and Shan E Ahmed Raza, where they are also using a CNN model with SGD as an optimizer, ReLU activation and cross-entropy. Furthermore, they have increased the number dataset through data augmentation allowing the models to find more generalized patterns (Korsuk Sirinukunwattana, 2016).

However, they have slightly different architecture which only consists of two layers of convolution layers (Spatially Constrained Convolution Neural Network). Furthermore, they have implemented few regularization techniques such as dropout, etc. so that their mode does not over-fit. One major different approach used by other researchers is that they use more complex metric for performance evaluation (Korsuk Sirinukunwattana, 2016).

Jainam Doshi (s3825891)
Adhiraj Jain (s3821245)

Other investigations have used with multiple CNN architectures such as LeNet. AlexNet, ResNet using architectures RMSprop, SGD and SGD.

## Appendix

Figure 1: Performance metric of model 1

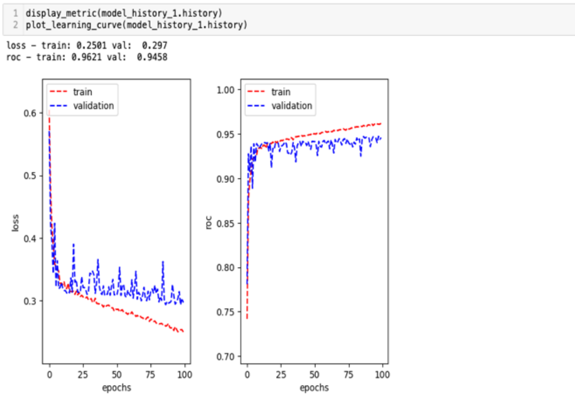



Figure 2: Performance metric of model 2

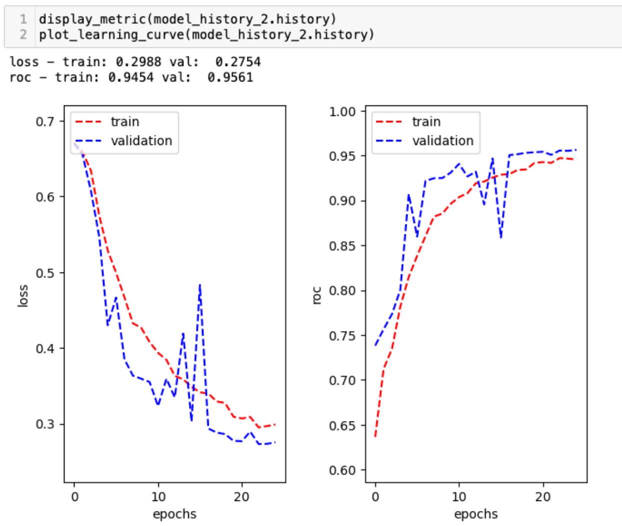



Figure 3: Performance metric of model 3

```
1 plot_learning_curve(model_3.history.history)
```

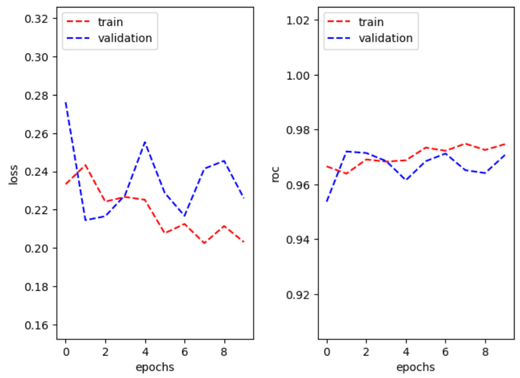
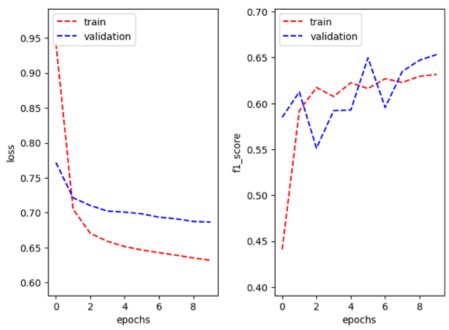



Figure 4: Performance metric of model_4

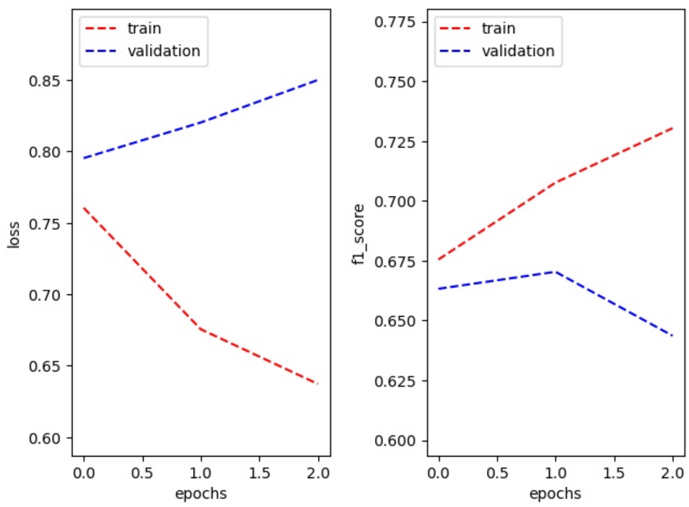
```
1 plot_learning_curve(model_4_history.history)
2 display_metric(model_4_history.history)
```




```
loss - train: 0.6319 val:  0.6866
f1_score - train: 0.6317 val:  0.6533
```

Figure 5: Performance metric model_5

```
1 plot_learning_curve(model_5_history.history)
2 display_metric(model_5_history.history)
```




```
loss - train: 0.6373 val:  0.8499
f1_score - train: 0.7304 val:  0.6436
```

## Bibliography

Ahmed, W. S. & Karim, A. a. A., 2020. The Impact of Filter Size and Number of Filters on Classification Accuracy in CNN. *2020 International Conference on Computer Science and Software Engineering (CSASE).*

AnalyticsVidhya, 2020. *Fundamentals of Deep Learning – Activation Functions and When to Use Them?.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/#:~:text=The%20ReLU%20function%20is%20another,neurons%20at%20the%20same%20time.
[Accessed 2023].

Bock, S., Goppold, J. & Weib, M., 2018. An improvement of the convergence proof of the ADAM-Optimizer. *CONFERENCE PAPER AT OTH CLUSTERKONFERENZ.*

Brownlee, J., 2019. *How Do Convolutional Layers Work in Deep Learning Neural Networks?.* [Online]
Available at: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/
[Accessed 2023].

Dinesh, 2019. *CNN vs MLP for Image Classification.* [Online]
Available at: https://medium.com/analytics-vidhya/cnn-convolutional-neural-network-8d0a292b4498
[Accessed 2023].

Korsuk Sirinukunwattana, . S. E. A. R., 2016. Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images. *IEEE TRANSACTIONS ON MEDICAL IMAGING.*

Li, Z. et al., 2022. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS,* 33(12), pp. 6999-7019.

Nellore, S. B., 2015. Various performance measures in Binary classification –An Overview of ROC study. *IJISET,* 2(9), p. 72.

Sanghvi, K., 2020. *Image Classification Techniques.* [Online]
Available at: https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac#id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6IjgyMjgzOGMxYzhiZjllZGNmMWY1MDUwNjYyZTU0YmNiMWFkYjViNWYiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJuYmYiOjE2OD
[Accessed 2023].

Sultan, H. H., Salem, N. M. & Al-Atabany, W., 2019. Multi-Classification of Brain Tumor Images Using Deep Neural Network. *IEEE,* Volume 7, pp. 69215-69225.
Vadapalli, P., 2022. *Types of Optimizers in Deep Learning Every AI Engineer Should Know.* [Online]
Available at: https://www.upgrad.com/blog/types-of-optimizers-in-deep-learning/#:~:text=An%20optimizer%20is%20an%20algorithm,appropriate%20weights%20for%20the%20model.
[Accessed 2023].
Yang, J. & Li, J., 2017. APPLICATION OF DEEP CONVOLUTION NEURAL NETWORK. *IEEE.*