Deep Learning

880663-M-6

Assignment

Using Deep Learning to Perform Multi-Class Classification on the Lung and Colon Cancer Histopathological Image Dataset (LC25000)

Report by:

Ivaylo Papazov (2051866)

March 2024

1. Problem Definition

The problem that this individual assignment will take a look at is of a classification nature. Using a perfectly balanced dataset of 25 000 instances of 5 different types of cancer images provided by Borkowski et al. (2019), the goal is to build a CNN (Convolutional Neural Network) and train it on the data using various hyperparameters, making sure that the accuracy is high while also avoiding overfitting. The CNN would later classify an image as one of the 5 classes.

2. Exploratory Data Analysis

Regarding exploratory data analysis, a number of steps were performed before the training of the first (Baseline) model.

Before explaining the steps that were taken to prepare the data for the analysis, it is also important to note that the original dataset has been augmented. Initially, the dataset contained only 750 images, which were later augmented using the Augmentor package, after which the dataset significantly increased to 25000 – the number that will be used in this assignment.

Firstly, all the images were resized from their original size (768 x 768) to a smaller one (120 x 120), for computational and memory constraint reasons. Furthermore, labels were assigned ('colon_aca', 'colon_c', 'lung_aca', 'lung_n', 'lung_scc') to the images according to the folder they belong to for easier further analysis and comprehension.

Secondly, the target variable – namely the labels, was first transformed into numerical format, after which they were one-hot encoded to continue the analysis.

Thirdly, several random images – 15 specifically, were visualized from the dataset, in order to get a better understanding of the inputs of the future models. The images were displayed along with their labels.

Moreover, the class distribution of the dataset was visualized using a bar chart. From it, we can conclude that the dataset is perfectly balanced – all five of the classes have 5000 images each, making accuracy a suitable metric for the upcoming analyses.

Finally, the dataset was divided using the train_test_split function from the scikit-learn Python package. Using it, the performed split was 60%/20%/20% for the training/validation/test sets, respectively. Stratification was performed on both splits to ensure that the distribution of the classes is kept the same as it is in the original dataset.

3. Results of the Baseline Model

After using the provided instructions (layers, activation function, etc.) for the assignment, the final model achieved a training accuracy of 0.9223 and a validation accuracy of 0.7556. Furthermore, the loss that the model outputted was 0.1933 and 0.6590 for the training and validation set, respectively. Additionally, the epoch that yielded the best accuracy was the 6th one, while the epoch that minimized the loss the most was also the 6th one. At first look these metrics seem relatively optimal.

However, after looking at the accuracies and losses plotted together (Figure 1), we see by the validation curves that the model is heavily overfitted. This is most likely due to the fact that no regularization techniques were used, as well as that no dropout rate was utilized.

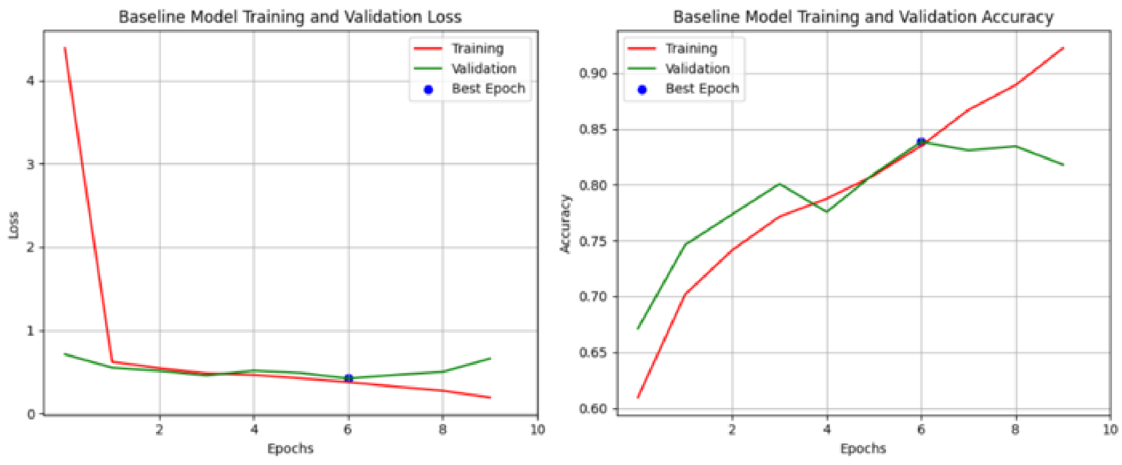
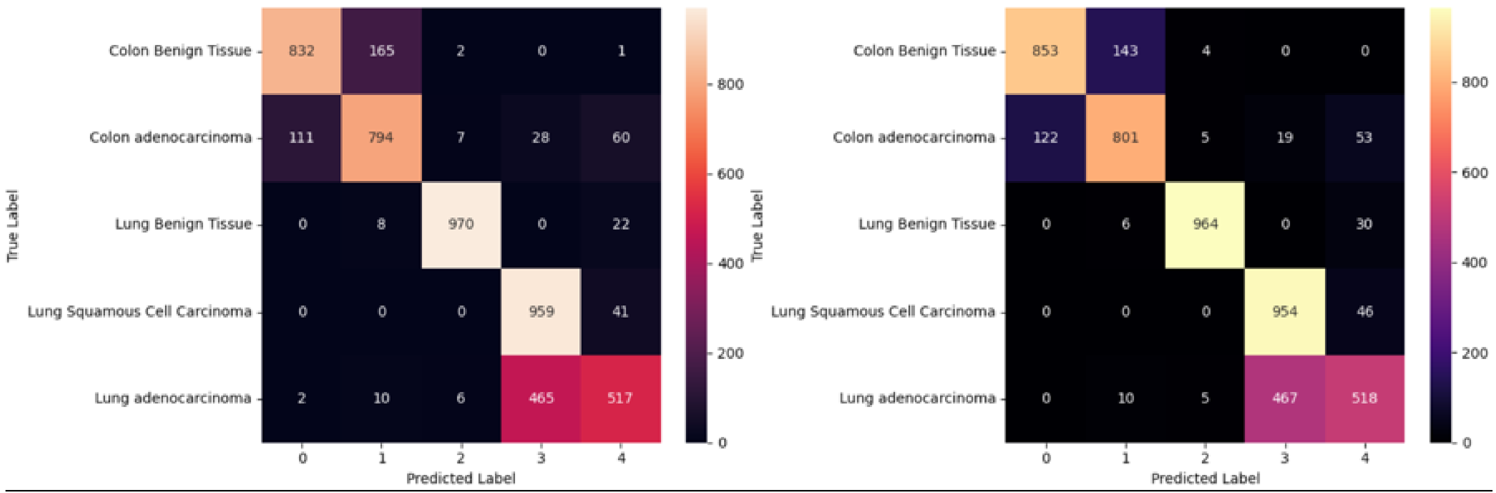


Figure 1 – Baseline Model Training and Validation Loss-Acc curves.



	0	1	2	3	4
Colon Benign Tissue	832	165	2	0	1
Colon adenocarcinoma	111	794	7	28	60
Lung Benign Tissue	0	8	970	0	22
Lung Squamous Cell Carcinoma	0	0	0	959	41
Lung adenocarcinoma	2	10	6	465	517

	0	1	2	3	4
Colon Benign Tissue	853	143	4	0	0
Colon adenocarcinoma	122	801	5	19	53
Lung Benign Tissue	0	6	964	0	30
Lung Squamous Cell Carcinoma	0	0	0	954	46
Lung adenocarcinoma	0	10	5	467	518

Figure 2 – Baseline Model Training and Validation Confusion Matrices.

Furthermore, after examining the confusion matrices of both the training and validation sets, we can also discover that most of the mistakes the model has done was when classifying the 5th label – “Lung adenocarcinoma”.

In order to fix those issues, an improved model, using fine-tuned hyperparameter, was introduced.

4. Improved (Fine-tuned) Model and Its Results

In order to build an improved version of the baseline model, the following fine-tuned model was based on the CNN created by Han et al. (2017), who also worked on a multi-classification problem related to cancer images. Furthermore, the CNN model built by them has been used in other highly cited papers as well (Nawaz et al., 2018).

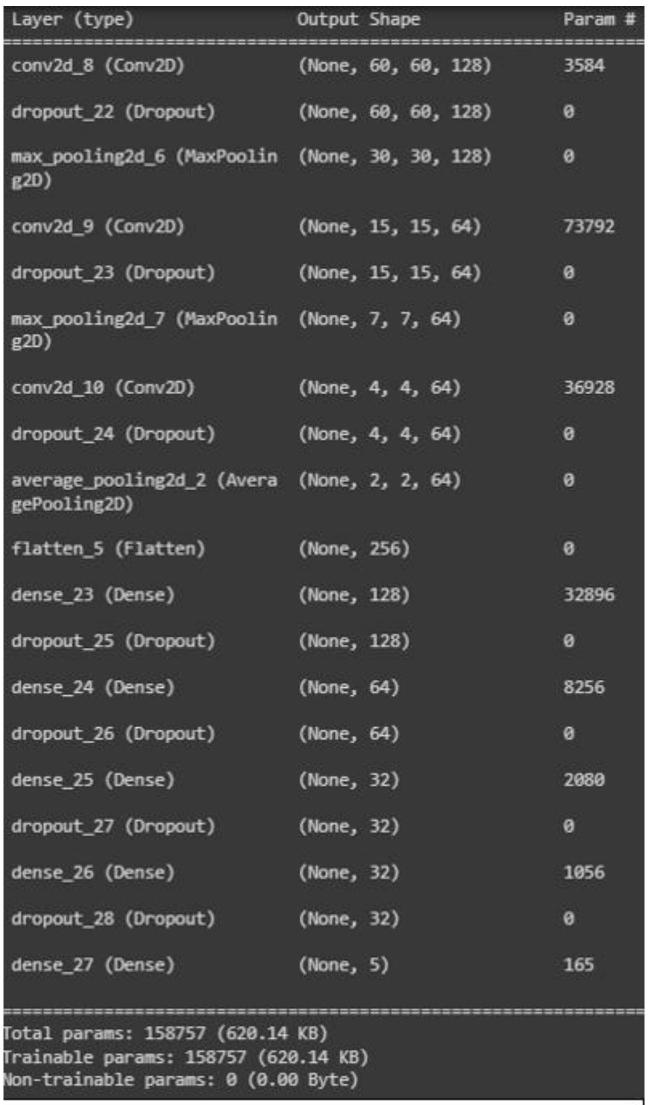
To begin, the first thing the improved model needed to do was increase the accuracy of the baseline model. This has been performed by additional layers – one convolutional, as well as two dense layers. The inputs of these layers have been inspired by the previously mentioned paper by Han et al. (2017).

Furthermore, when it comes to overfitting, two main steps have been taken to reduce it – an L2 regularization of 0.001 and a dropout rate of 0.2, respectively (Krizhevsky et al., 2017).

Moreover, as explained by the scientific papers (Han et al. (2017); Nawaz et al. (2018); Rawat et al. (2017), since there might be redundant information in the pictures, a stride of 2 has also been applied to all convolutional and dense layers.

Finally, an average pooling layer has been added before the fully connected layer (Han et al, 2017). The learning rate selected was 0.001.

Furthermore, other hyperparameter tunings were attempted, such as trying different regularizers (rmsprop, sgd), as well as introducing kernel initializers – both Gaussian and He. However, those parameters did not yield a better result in terms of accuracy and loss than the baseline.



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 60, 60, 128)       3584

dropout_22 (Dropout)         (None, 60, 60, 128)       0

max_pooling2d_6 (MaxPoolin   (None, 30, 30, 128)       0
g2D)

conv2d_9 (Conv2D)            (None, 15, 15, 64)        73792

dropout_23 (Dropout)         (None, 15, 15, 64)        0

max_pooling2d_7 (MaxPoolin   (None, 7, 7, 64)          0
g2D)

conv2d_10 (Conv2D)           (None, 4, 4, 64)          36928

dropout_24 (Dropout)         (None, 4, 4, 64)          0

average_pooling2d_2 (Avera   (None, 2, 2, 64)          0
gePooling2D)

flatten_5 (Flatten)          (None, 256)               0

dense_23 (Dense)             (None, 128)               32896

dropout_25 (Dropout)         (None, 128)               0

dense_24 (Dense)             (None, 64)                8256

dropout_26 (Dropout)         (None, 64)                0

dense_25 (Dense)             (None, 32)                2080

dropout_27 (Dropout)         (None, 32)                0

dense_26 (Dense)             (None, 32)                1056

dropout_28 (Dropout)         (None, 32)                0

dense_27 (Dense)             (None, 5)                 165

=================================================================
Total params: 158757 (620.14 KB)
Trainable params: 158757 (620.14 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 3: Optimized model summary.

The final result of the model outputted a training accuracy of 0.8987 and a validation accuracy of 0.9252. Moreover, the training loss was 0.3439 and the validation loss was 0.2678. To conclude the best epoch for both visualizations was 9.

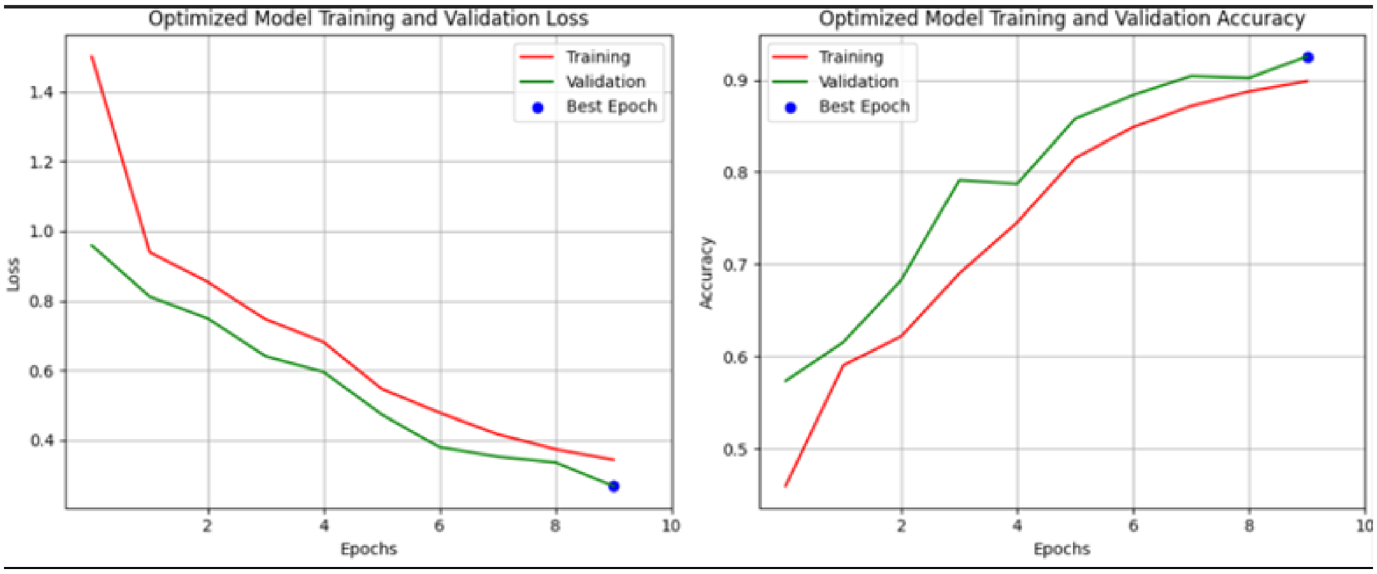


Figure 4: Optimized Model Training and Validation Acc-Loss curves.

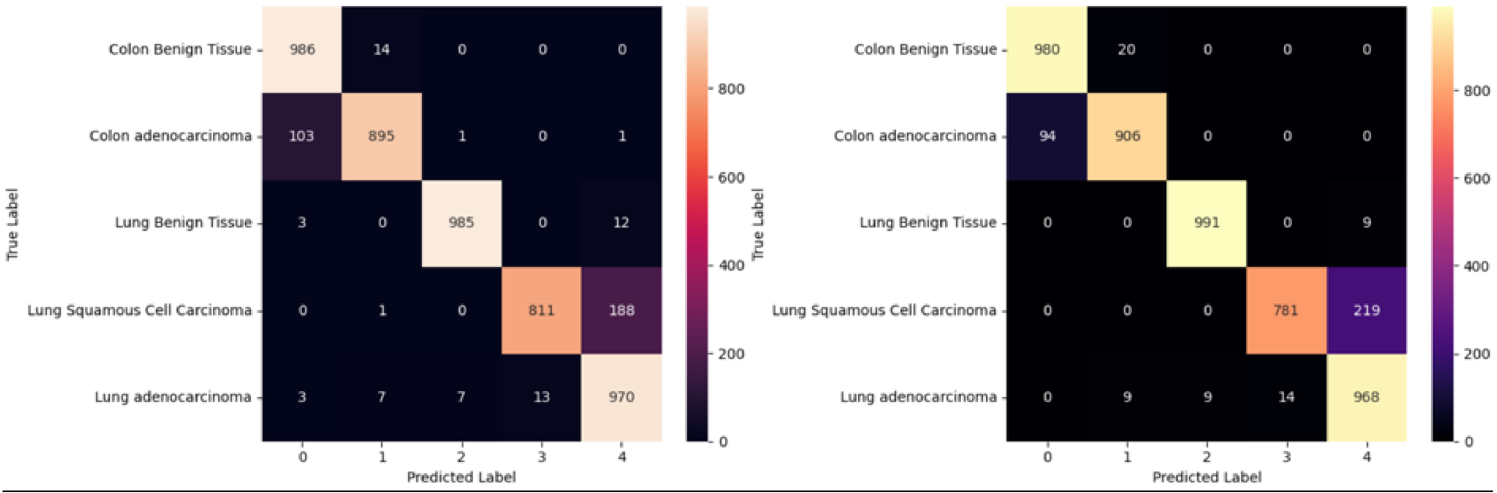
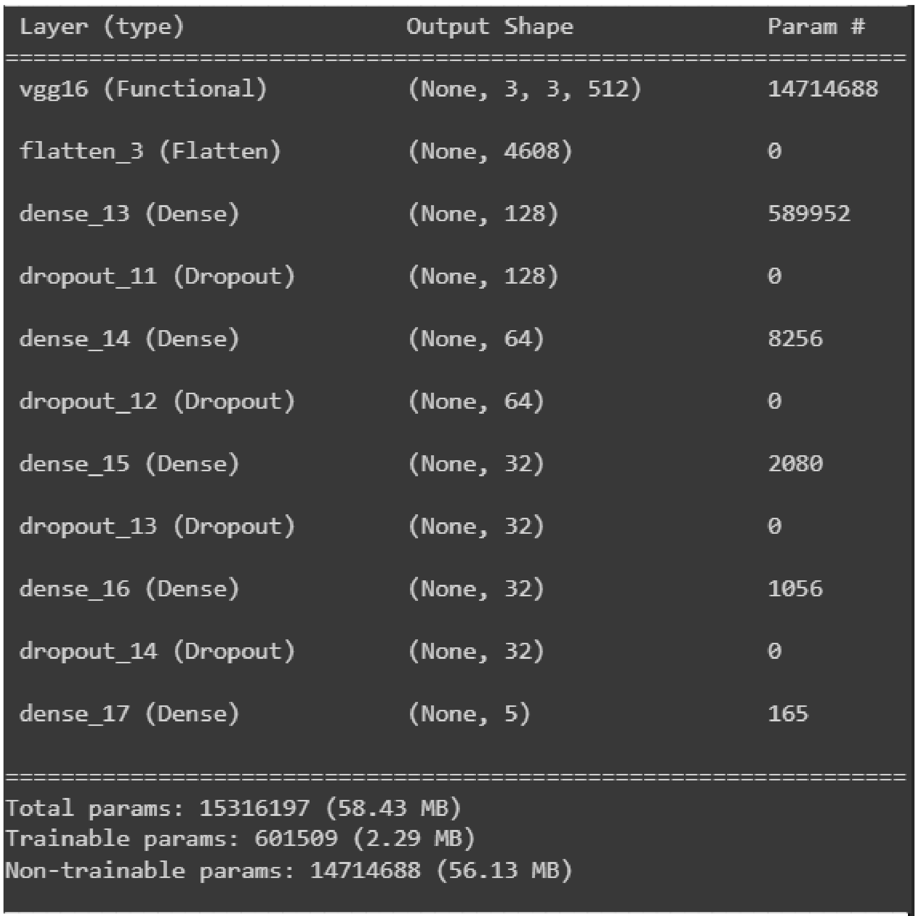


Figure 5 – Optimized Model Training and Validation Confusion Matrices.

5. Transfer Learning Model and Its Results



```
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 3, 3, 512)         14714688

 flatten_3 (Flatten)         (None, 4608)              0

 dense_13 (Dense)            (None, 128)               589952

 dropout_11 (Dropout)        (None, 128)               0

 dense_14 (Dense)            (None, 64)                8256

 dropout_12 (Dropout)        (None, 64)                0

 dense_15 (Dense)            (None, 32)                2080

 dropout_13 (Dropout)        (None, 32)                0

 dense_16 (Dense)            (None, 32)                1056

 dropout_14 (Dropout)        (None, 32)                0

 dense_17 (Dense)            (None, 5)                 165

=================================================================
Total params: 15316197 (58.43 MB)
Trainable params: 601509 (2.29 MB)
Non-trainable params: 14714688 (56.13 MB)
```

Figure 6: Summary of the Transfer Learning Model.

For the third and final model of this project, a Transfer Learning-based model will be utilized on the already optimized model described in section 4. A VGG-16 deep CNN will be implemented to perform this, as the paper by Tammina (2019) showed that adding this transfer learning method significantly betters the performance of the model.

In order for the transfer learning model to work, the model was first initiated without the top layer. The layers in this model were later frozen, up to the fully connected one, in order to keep the pre-trained weights intact, as well as prevent any overfitting.

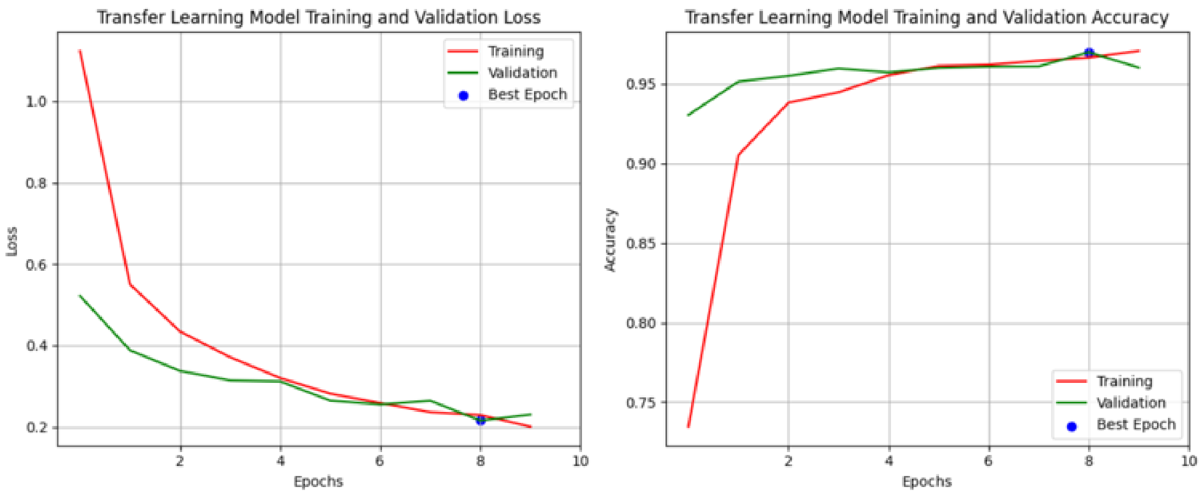


Figure 7: Transfer Learning Model Training and Validation Acc-Loss curves.

The final results of this model were a training accuracy and a validation accuracy of 0.9706 and 0.9602, respectively, as well as training and validation loss of 0.2007 and 0.2303, once again respectively. The best performing epoch of the model was the 8th one for both the training and validation data.

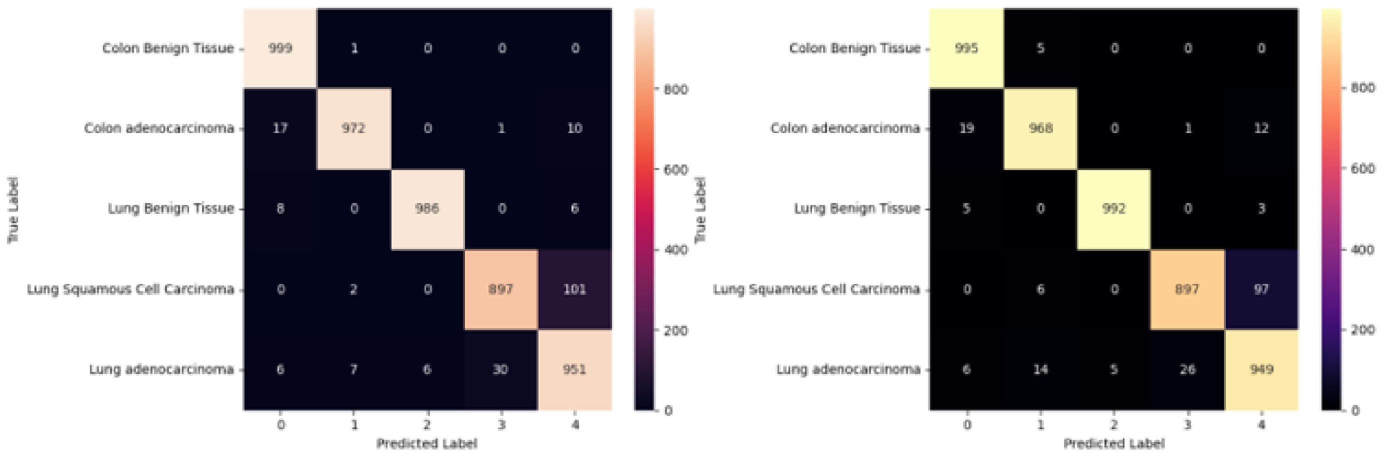


Figure 8: Transfer Learning Model Training and Validation Confusion Matrices.

6. Discussion

When it comes to the performance of the three models, it is safe to conclude that the Transfer Learning VGG-16 model described in section 5 yielded the best results.

The initial baseline model had a relatively high accuracy. However, from all the graphs presented in this paper, it can be safely assumed that it is largely due to overfitting. This can be seen from the big difference between the training and validation accuracy, as well as the curve from Figure 1. Furthermore, the model was deemed relatively unreliable, as during every run of the model, the evaluation metrics that were produced were significantly different and fluctuating.

Secondly, the optimized version of the baseline model, based around the CNN by Han et al. (2017) performed significantly better with the introduction of the L2 regularizer and dropout rate. The accuracy yielded was not much higher than the initial model, but from the graphs (both the Acc-Loss curve and the Confusion Matrices), it can be seen that the overfitting is much less of a problem in this instance. However, there is still room for improvement.

Finally, the third model – namely, the VGG-16 transfer learning one, had the best results so far. By freezing the initial layers, the model not only achieved an accuracy much higher than the previous two, but it also reduced the overfitting significantly as well.

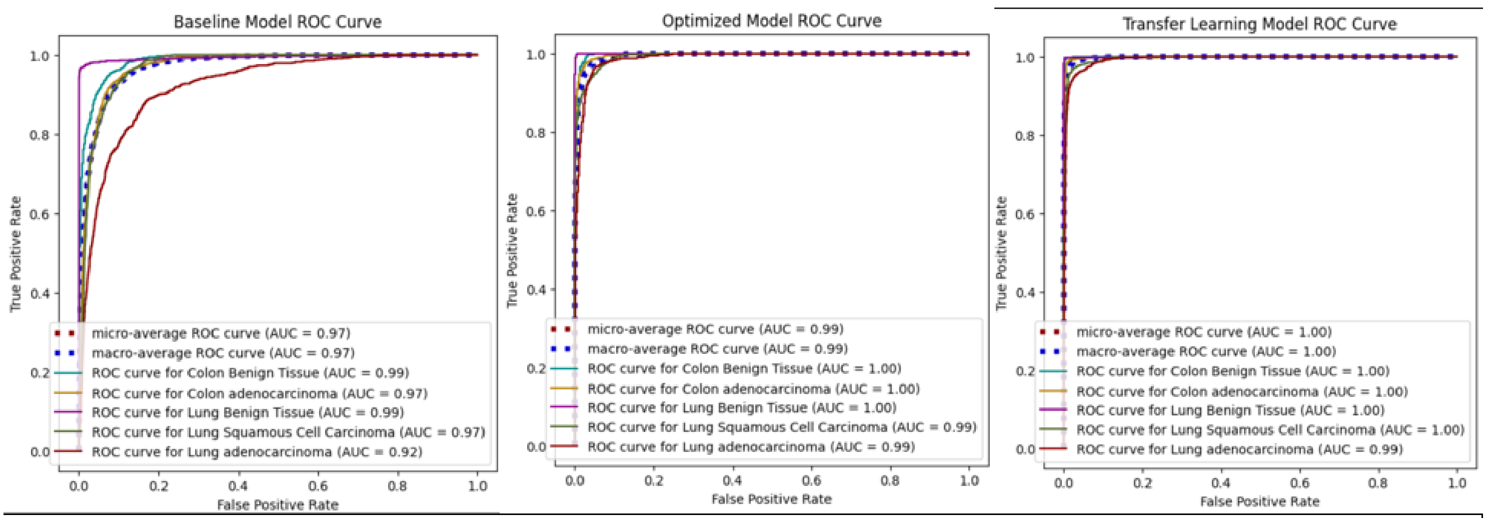


Figure 9: ROC Curve, along with AUC for each of the 5 classes, for all 3 models.

Furthermore, we can also examine the ROC Curves of all three models (Figure 9). Same as for the other evaluation metrics visualization, here it can also be concluded that the transfer learning model performed the best. Therefore, it is safe to wrap up and confirm that the last model should be the one to be used for this CNN multilabel classification problem.

7. References

Borkowski, A., Bui, M. M., Thomas, L. B., Wilson, C. P., Deland, L. A., & Mastorides, S. (2019). Lung and colon Cancer Histopathological Image Dataset (LC25000). *arXiv (Cornell University)*. https://arxiv.org/pdf/1912.12142.pdf

Han, Z., Wei, B., Zheng, Y., Yin, Y., Li, K., & Li, S. (2017). Breast Cancer Multi-classification from Histopathological Images with Structured Deep Learning Model. *Scientific Reports*, *7*(1). https://doi.org/10.1038/s41598-017-04075-z

Nawaz, M., Sewissy, A. A., & Soliman, T. H. A. (2018). Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network. *International Journal of Advanced Computer Science and Applications*, *9*(6). https://doi.org/10.14569/ijacsa.2018.090645

Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive review. *Neural Computation*, *29*(9), 2352–2449. https://doi.org/10.1162/neco_a_00990

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Tammina, S. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications*, *9*(10), p9420. https://doi.org/10.29322/ijsrp.9.10.2019.p9420