Lung and Colon Cancer Histopathological Image Classification Report

Section 1: Business Problem and Background

Problem Statement

The medical community is seeking an automated approach to accurately diagnose adenocarcinoma (ACA) and squamous cell carcinoma (SCC) in lung tissue and adenocarcinoma in colon tissue using histopathological images. An effective solution will be able to supplement the opinions of medical professionals and could also provide a cancer screening option for individuals/communities with limited access to healthcare services.

Background Information

Currently, the gold standard approach for diagnosing cancer, and identifying cancer/tumour type, is analysis of histopathological images by a pathologist. Effective automated classification of histopathological images could help to reduce pathologist workloads, provide a “second opinion” to supplement a pathologist’s conclusion, and serve as a cancer screening option for individuals/communities with limited access to healthcare services. Therefore, a model will be built in order to classify lung and colon cancer from histopathological slides with high accuracy. The model will be trained on histopathological images with five image classes: benign lung tissue, lung adenocarcinoma, lung squamous cell carcinoma, benign colon tissue, and colon adenocarcinoma. An effective model has the potential to reduce the workload on pathologists, supplement the opinions of pathologists, and provide cancer screening in situations where access to healthcare is limited. Therefore, benefits could be seen both within the medical community and on patient outcomes.

Success Criteria

A model will be developed that can classify lung and colon histopathological images into the following groups with a high degree of accuracy: benign lung tissue, lung adenocarcinoma, lung squamous cell carcinoma, benign colon tissue, and colon adenocarcinoma.

References:

- https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1685-x
- https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images

Section 2: Data Wrangling

Data Collection

Our model to classify cancer/tumour type using histopathology images was developed using an image dataset containing five classes of images: benign lung tissue, lung adenocarcinoma, lung squamous cell carcinoma, benign colon tissue, and colon adenocarcinoma.

The dataset, titled “Lung and Colon Cancer Histopathological Images” was obtained from Kaggle. It can be accessed at the following URL: https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images

Retrieval of the dataset from Kaggle was automated via code. An API key was generated in order to access the dataset. Shell commands were then used to download and unzip the dataset. Each class of images was stored in a separate directory within the following directory structure:

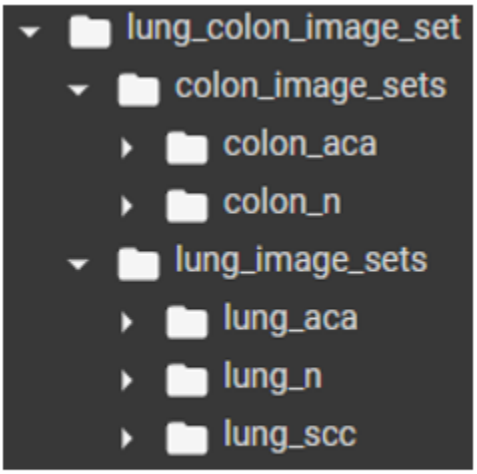

Figure 1: Directory structure for the raw dataset

Data Description

The Kaggle dataset contained 25000 images in total, with 5000 images belonging to each of the five image classes. All images were 768x768 pixels in size and were in RGB format (3 colour channels). The dataset was generated from 1250 original histopathological images (250 per image class). The 1250 images were augmented to 25000 using the Augmentor package.

References:

- https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images

Section 3: Exploratory Data Analysis (EDA)

Inspect the Images

An image from each class was inspected as a sanity check to ensure that there were no issues with the data.

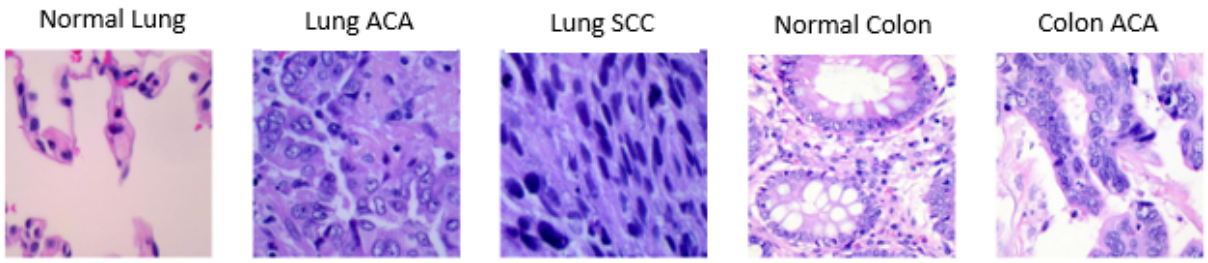

Figure 2: Sample image from each image class

Image Classes

The number of images per class and total number of image classes were examined to ensure the data was retrieved correctly. It was verified that there were 5000 images per class and a total of 5 image classes.

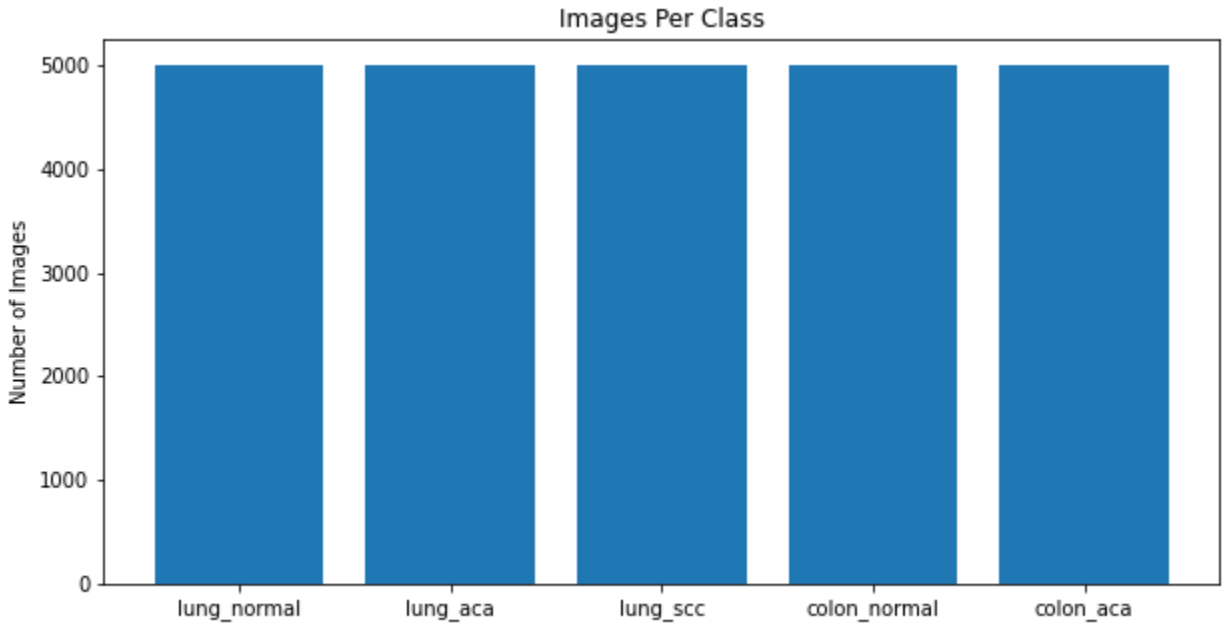

Figure 3: The dataset was perfectly balanced as it contained 5000 images per class.

Image Dimensions

The image dimensions were verified to be 768x768x3. This represented images that were 768x768 pixels in size with three colour channels (RGB format).

Principal Component Analysis: Inspect Potential Relationships Between Image Classes

120 images from each class were randomly selected (due to computational constraints). Principal Component Analysis (PCA) was then performed on the selected images. The first two principal components were selected to allow the data to be represented and visualized in two dimensions.

Principal Component 1 was plotted against Principal Component 2 for each selected image. The data points were coloured by image class in order to visualize potential relationships between the image classes. The PCA analysis found that among the five classes, the data was easily separated into the following three groups:

- Group 1: Benign Lung Tissue
- Group 2: Lung Cancer (Lung Adenocarcinoma and Lung Squamous Cell Carcinoma)
- Group 3: Colon Tissue (Benign Colon Tissue and Colon Adenocarcinoma)

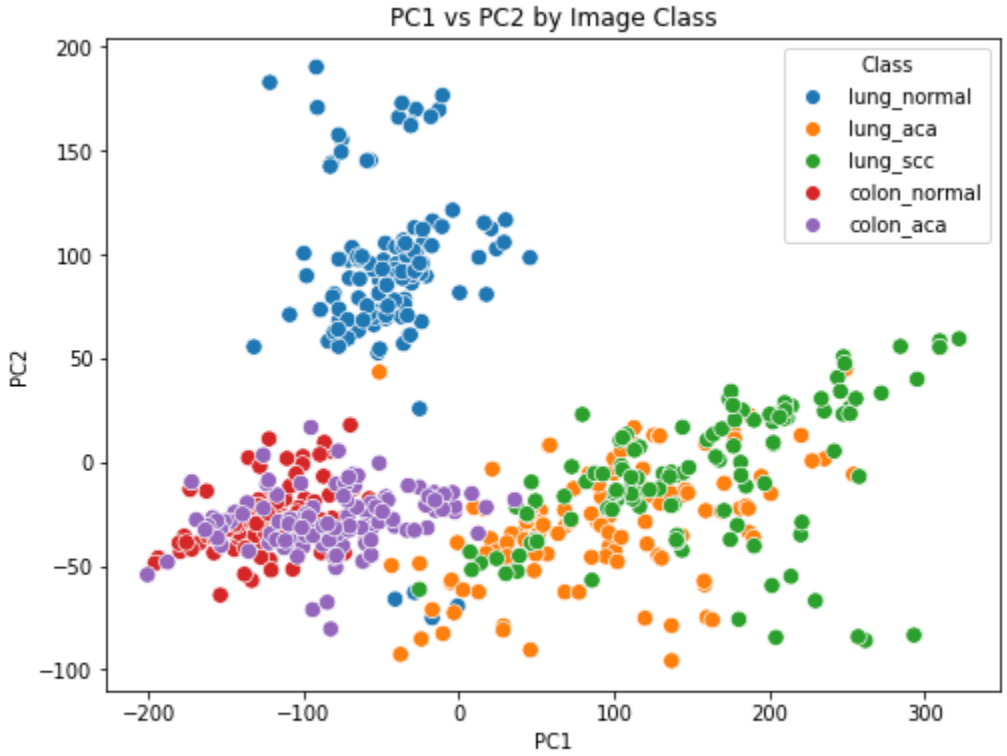

Figure 4: Two-dimensional representation of the relationships among the image classes, as identified by PCA analysis.

This analysis would suggest that models will have a high degree of success classifying benign lung tissue. It also suggests that models should be successful at identifying lung cancer, but may have some trouble differentiating between the two types of lung cancer. Finally, the analysis predicts that while models might be highly successful at identifying colon tissue they may have trouble differentiating between benign and cancerous tissue. If these relationships

hold true after evaluating models, then the colon tissue would pose the largest problem. This is because misclassifying cancerous colon tissue as benign tissue could result in a failure to detect colon cancer in patients. On the other hand, it would be less catastrophic to misclassify the lung cancer type, as long as lung cancer is detected. After lung cancer detection, it is likely that medical professionals would examine the patient and correctly identify the type of cancer before initiating a course of treatment.

It is important to keep in mind that the PCA analysis likely provides only a rough idea of important patterns in the data. Since the data was reduced to only two dimensions for visualization purposes much of the complexity in the data will have been lost. The deep learning models that will be trained will not be limited to a two dimensional representation of the data. Also, deep learning models are highly adept at identifying complex non-linear decision boundaries that may not be possible to visualize in two dimensions. Therefore, the problem areas identified by the PCA analysis may prove to be minimal, or non-existent, once models are trained and evaluated.

Section 4: Preprocessing

Train/Test Split

In order to train and evaluate models, the dataset was split in to train and test sets. An allocation of 90% training data and 10% test data was used. After generating the train/test split by randomly selecting images for each dataset, the list of images in each training/test folder was saved. This allowed the same train/test split to be reused across different notebooks. When fitting models, the training set was further subdivided into 90% training data and 10% validation data.

Directory Structure

For both the training and test sets, a directory was created for each class of image. This allowed TensorFlow to automatically generate the training, validation, and test image datasets from the directories. The datasets were represented as TensorFlow Datasets.

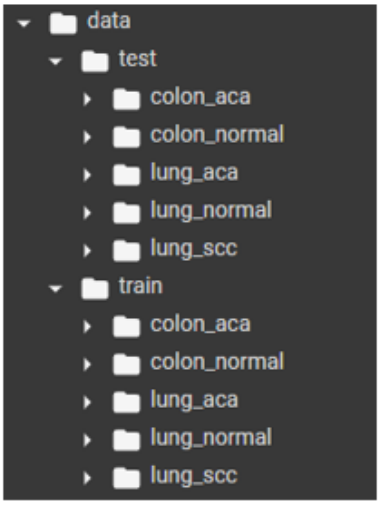

Figure 5: Directory structure for the train and test datasets

Batches

When assembling TensorFlow Datasets a batch size of 32 was used.

Normalization/Rescaling

The pixel values of all images were normalized to values between 0 and 1. This was expected to help reduce computation time when training the models.

Image Size Reduction

The image dimensions were reduced in order to reduce computation time. Images were rescaled from their original dimensions (768x768 pixels) to a size of 384x384 pixels.

Data Augmentation

Data augmentation can help make models more robust to variations in the data and can improve the ability of models to generalize to unseen data. Data augmentation was applied to the data prior to training two of the models: the "Data Augmentation Model" and the "Deep CNN Model with Data Augmentation". The following transformations were selected for augmentation, as they are among transformations that have previously been used successfully for classification of histopathology images of cancer tissue:

- Random Horizontal Flips
- Random Vertical Flips
- Random Clockwise Rotations
- Random Counterclockwise Rotation
- Random Zoom
- Random Contrast Change

References

- https://hal.archives-ouvertes.fr/hal-03321646/document
- https://arxiv.org/ftp/arxiv/papers/2112/2112.13553.pdf

Section 5: Modeling

Model Building and Training

All models were built using TensorFlow 2.0's Keras API. Each model was trained using Categorical Cross Entropy as the loss function and the Adam optimizer. Each model was set to train for a maximum of 100 epochs. However, early stopping with a patience of 5 was used to reduce overfitting. None of the models trained for the full 100 epochs. The training data was split into 90% training data and 10% validation data. The data was prepared using TensorFlow's

“image_dataset_from_directory” function. In addition to loss, validation accuracy was examined. 5 original models and 2 transfer learning models were trained.

5.1 Original (Non-Transfer Learning) Models

Model Architectures

Initially, a basic Convolutional Neural Network (CNN) was built as a baseline model. The model’s performance on training and validation data was analyzed. Modifications were then made to the baseline architecture in an effort to improve upon the baseline model. In total, five original models were trained. Each model’s architecture and performance on training and validation data is detailed below:

1. Baseline Model

The baseline model used a basic CNN architecture. The **inputs** to the model were images of shape **384x384x3**. The Baseline Model architecture consisted of the following layers:

1. **Rescaling Layer**: Divide pixel values by 255. Outputs 384x384x3 images with pixel values between 0 and 1.
2. **Convolutional Layer 1**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 32 feature maps with dimensions 382x382.
3. **Max Pooling Layer:** A max pooling layer with a 2x2 pooling size and 2x2 stride. Outputs 32 feature maps with dimensions 191x191.
4. **Convolutional Layer 2:** A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 64 feature maps with dimensions 189x189.
5. **Max Pooling Layer**: A max pooling layer with a 2x2 pooling size and 2x2 stride. Outputs 64 feature maps with dimensions 94x94.
6. **Convolutional Layer 3:** A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 64 feature maps with dimensions 92x92.
7. **Flattening Layer:** A flattening layer to convert the output of the convolutional layers into 1 dimension. Output contains 541,696 values.
8. **Dense Layer**: A dense layer with 64 nodes using ReLU activation. Outputs 64 values.
9. **Dense Layer**: A dense layer with K nodes (K=5 for this dataset) using Softmax activation. Each node corresponds to one of the image classes. Outputs K probabilities, one for each image class.

Baseline Model Training Analysis

The Baseline Model's best performance was 92.98% accuracy on the validation set. The model trained for 12 epochs meaning that the model performed best on epoch 7 (since early stopping with a patience of 5 was used). Given that the best model was trained after only 7 epochs, it is possible that the model was able to extract the underlying signal in the data very quickly. However, it may be worth using strategies to reduce the rate at which the model overfits

during training. This may provide more of an opportunity for useful features/patterns to be identified. Two strategies that may be useful are adding **dropout layers** to the Baseline Model and including **data augmentation** as a preprocessing step. These were the next two models that were tested.

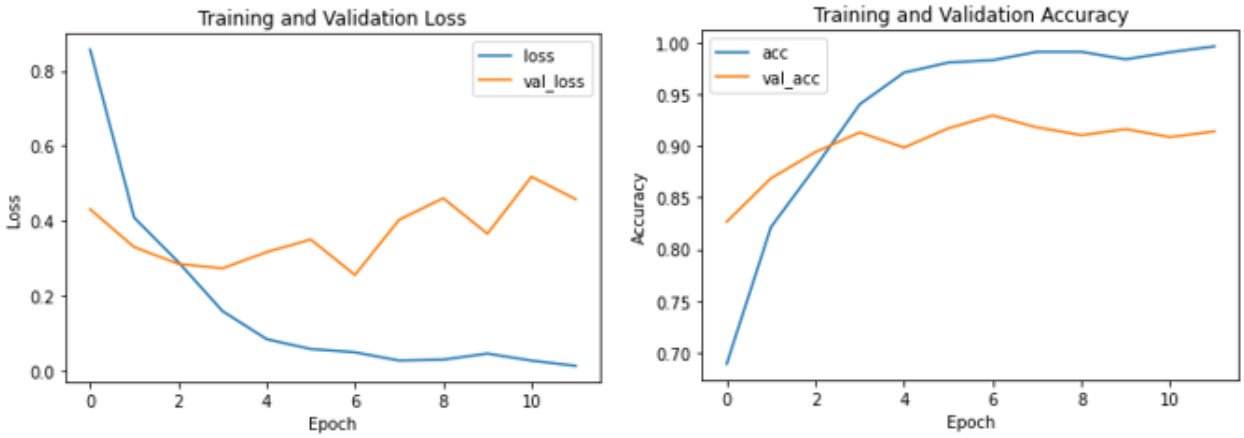

Figure 6: Baseline Model: training and validation loss (left), training and validation accuracy (right)

2. Dropout Model

The Dropout Model was built by adding dropout layers to the Baseline Model architecture. This was done in effort to reduce the rate at which the model overfits. The **inputs** to the model were images of shape **384x384x3**. The Dropout Model architecture consisted of the following layers:

1. **Rescaling Layer**: Divide pixel values by 255. Outputs 384x384x3 images with pixel values between 0 and 1.
2. **Convolutional Layer 1**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 32 feature maps with dimensions 382x382.
3. **Dropout Layer:** A dropout layer with a dropout rate of 0.2. Outputs 32 feature maps with dimensions 382x382.
4. **Max Pooling Layer:** A max pooling layer with a 2x2 pooling size and 2x2 stride. Outputs 32 feature maps with dimensions 191x191.
5. **Convolutional Layer 2:** A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 64 feature maps with dimensions 189x189.
6. **Dropout Layer:** A dropout layer with a dropout rate of 0.2. Outputs 64 feature maps with dimensions 189x189.
7. **Max Pooling Layer**: A max pooling layer with a 2x2 pooling size and 2x2 stride. Outputs 64 feature maps with dimensions 94x94.
8. **Convolutional Layer 3:** A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 64 feature maps with dimensions 92x92.

9. **Dropout Layer:** A dropout layer with a dropout rate of 0.2. Outputs 64 feature maps with dimensions 92x92.
10. **Flattening Layer:** A flattening layer to convert the output of the convolutional layers into 1 dimension. Output contains 541,696 values.
11. **Dense Layer**: A dense layer with 64 nodes using ReLU activation. Outputs 64 values.
12. **Dropout Layer:** A dropout layer with a dropout rate of 0.2. Outputs 64 values.
13. **Dense Layer**: A dense layer with K nodes (K=5 for this dataset) using Softmax activation. Each node corresponds to one of the image classes. Outputs K probabilities, one for each image class.

Dropout Model Training Analysis

The Dropout Model's best performance was 90.09% accuracy on the validation set. The model trained for 9 epochs meaning that the model performed best on epoch 4 (since early stopping with a patience of 5 was used). Dropout layers were used in an attempt to improve upon the Baseline Model, as the baseline model appeared to overfit fairly quickly during training. However, the Dropout Model actually appeared to overfit more quickly than the Baseline Model. Also, the best Dropout Model did not perform as well on the validation set compared to the best Baseline Model. It is possible that, in this case, the dropout layers actually impaired the model's ability to extract useful features from the data during training. However, it was also considered possible that the dropout model would perform better than the baseline when generalizing to unseen data. A possible area for future investigation is experimenting with different dropout architectures and/or different dropout rates.

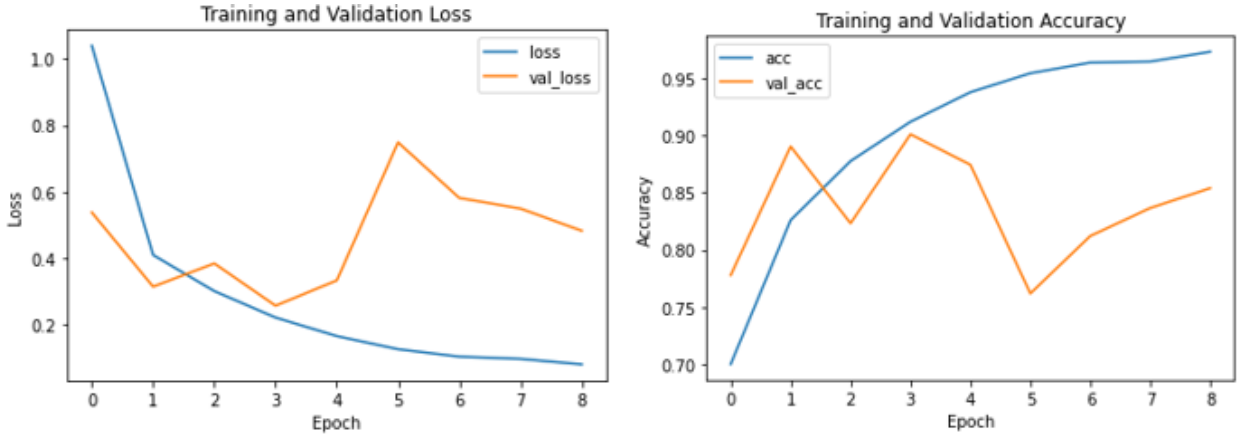

Figure 7: Dropout Model: training and validation loss (left), training and validation accuracy (right)

3. Model with Data Augmentation

Data augmentation was added as a preprocessing step. The model architecture was **identical to the Baseline Model** (see above). The following transformations were selected for

data augmentation because they have shown effectiveness when classifying histopathology images of cancer tissue:

- Random Horizontal and Vertical Flips
- Random Rotation
    - A factor of -0.5 to 0.5 was used for the random rotation. The factor represents the fraction of 2 Pi by which the image is rotated. Negative values represent clockwise rotation while positive values represent counter clockwise rotations.
- Random Zoom
    - A height factor of -0.3 to 0.3 was used for the random zoom. The height factor represents the percentage by which the image is zoomed. A positive value represents zooming out while a negative value represents zooming in. The original aspect ratio of the image was preserved.
- Random Contrast Change
    - A factor of 0.3 was used for the random contrast change. This means that a value between 0.7 (1 - contrast factor) and 1.3 (1 + contrast factor) will be randomly selected in order to modify the contrast. The formula to modify each pixel's value, x, is (x - mean) x factor + mean.

References

- https://hal.archives-ouvertes.fr/hal-03321646/document
- https://arxiv.org/ftp/arxiv/papers/2112/2112.13553.pdf
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomRotation
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomZoom
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomContrast

Model with Data Augmentation Training Analysis

The data augmentation model's best performance was 96.67% accuracy on the validation set. The model trained for 17 epochs meaning that the model performed best on epoch 12 (since early stopping with a patience of 5 was used). Data augmentation as a preprocessing step was used in an attempt to improve upon the Baseline Model. It appears that increasing the number of training examples, and the variability among the training examples, reduced the rate at which the model overfit during training. The Model with Data Augmentation showed improved performance on the validation set compared to the baseline model. This was not surprising because with more training data the model would be expected to have more of an opportunity to extract useful patterns/features. Based on the model's performance on the validation data, data augmentation appears to be a useful technique for improving model performance when working with this particular set of histopathological images.

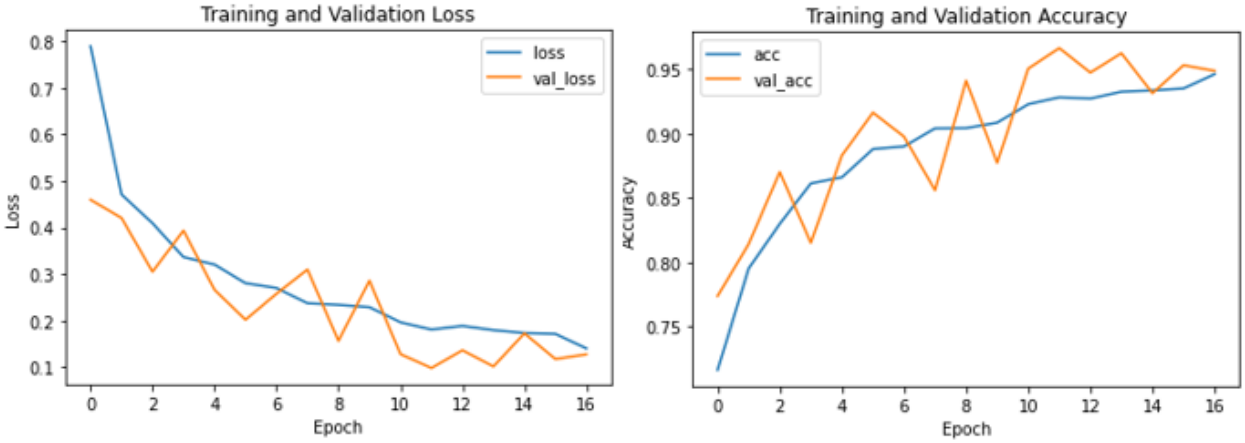

Figure 8: Model with Data Augmentation: training and validation loss (left), training and validation accuracy (right)

4. Deep Model

A deeper model (additional convolutional) layers was also tested. Additional convolutional layers can help a CNN extract more complex features from the training data. However, more depth can also result in a model that overfits quickly. The **inputs** to the model were images of shape **384x384x3**. The Deep Model architecture consisted of the following layers:

1. **Rescaling Layer**: Divide pixel values by 255. Outputs 384x384x3 images with pixel values between 0 and 1.
2. **Convolutional Layer 1**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 32 feature maps with dimensions 382x382.
3. **Convolutional Layer 2**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 32 feature maps with dimensions 380x380.
4. **Max Pooling Layer:** A max pooling layer with a 2x2 pooling size and 2x2 stride. Outputs 32 feature maps with dimensions 190x190.
5. **Convolutional Layer 3**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 64 feature maps with dimensions 188x188.
6. **Convolutional Layer 4**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 64 feature maps with dimensions 186x186.
7. **Max Pooling Layer:** A max pooling layer with a 2x2 pooling size and 2x2 stride. Outputs 64 feature maps with dimensions 93x93.
8. **Convolutional Layer 5**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 128 feature maps with dimensions 91x91.
9. **Convolutional Layer 6**: A 2-dimensional convolutional layer using a 3x3 kernel and ReLU activation. Outputs 128 feature maps with dimensions 89x89.

10. **Flattening Layer:** A flattening layer to convert the output of the convolutional layers into 1 dimension. Output contains 1,013,888 values.
11. **Dense Layer**: A dense layer with 128 nodes using ReLU activation. Outputs 128 values.
12. **Dense Layer**: A dense layer with K nodes (K=5 for this dataset) using Softmax activation. Each node corresponds to one of the image classes. Outputs K probabilities, one for each image class.

Deep Model Training Analysis

The Deep Model's best performance was 92.84% accuracy on the validation set. However, the model's best performance in terms of loss came on the second epoch where it achieved a validation accuracy of 90.53%. A deeper model was trained because in some cases adding more layers can help a model identify more complex features. However, more layers can also lead to models that overfit quickly. In this case, adding depth to the model did not appear to help. The best performing version of the model was trained after only 2 epochs. Additionally, the best Deep Model did not perform as well as the best Baseline Model on the validation data. However, it is possible that using data augmentation to provide the deeper model with more data could help reduce the rate of overfitting and provide the deeper model with more opportunity to extract useful features.

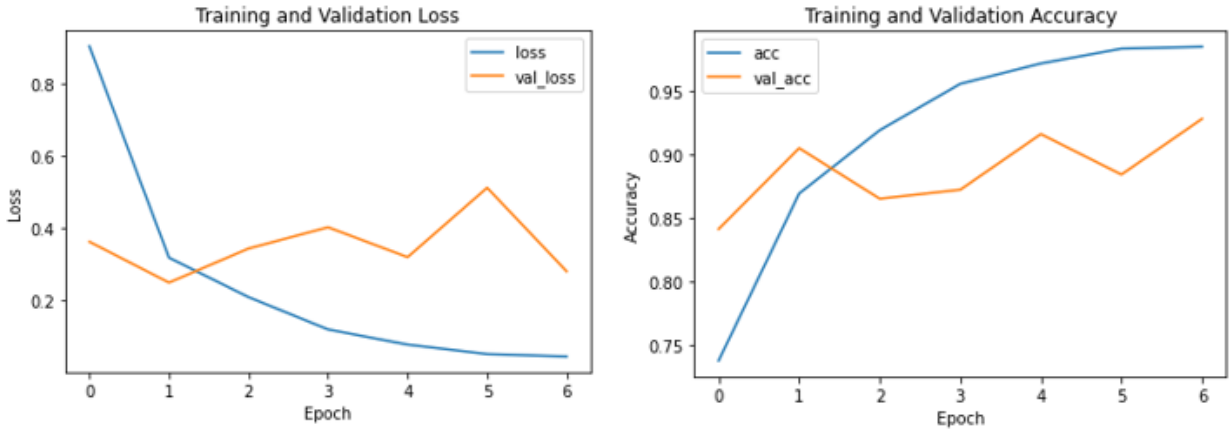

Figure 9: Deep Model: training and validation loss (left), training and validation accuracy (right)

5. Deep Model with Data Augmentation

The Deep Model was retrained with data augmentation as a preprocessing step. Although the Deep Model appeared to overfit very quickly, it was thought that the additional data provided by data augmentation may reduce the rate of overfitting and provide the Deep Model with more opportunity to extract useful features. This was important because, if the overfitting

issue was resolved, the Deep Model should be able to extract more complex, and potentially useful, features compared to the other original models. Additionally, data augmentation appeared to be effective at improving the baseline model. The Deep Model with Data Augmentation had architecture that was **identical to the Deep Model** (see above). The data augmentation transformations were identical to the transformations used when training the Model with Data Augmentation (see above).

Deep Model with Data Augmentation Training Analysis

The Deep Model with Data Augmentation had a best performance of 97.20% accuracy on the validation set. The model trained for 17 epochs meaning that the model performed best on epoch 12 (since early stopping with a patience of 5 was used). Originally, the Deep Model performed poorly on the validation set. However, using data augmentation as a preprocessing step appeared to be effective at improving the Baseline Model's performance. Therefore, it was thought that the Deep Model was also likely to benefit from more training data. The additional data appeared to lessen the rate of overfitting and drastically improve the Deep Model's ability to extract useful features. The deep CNN model with data augmentation was the best performing original model on the validation set. This is likely because, once the issue of overfitting was reduced, the Deep Model was able to extract more features, and more complex features, compared to the Baseline Model. The model's performance also appeared to be stable as it leveled off without much variation during the final training epochs.

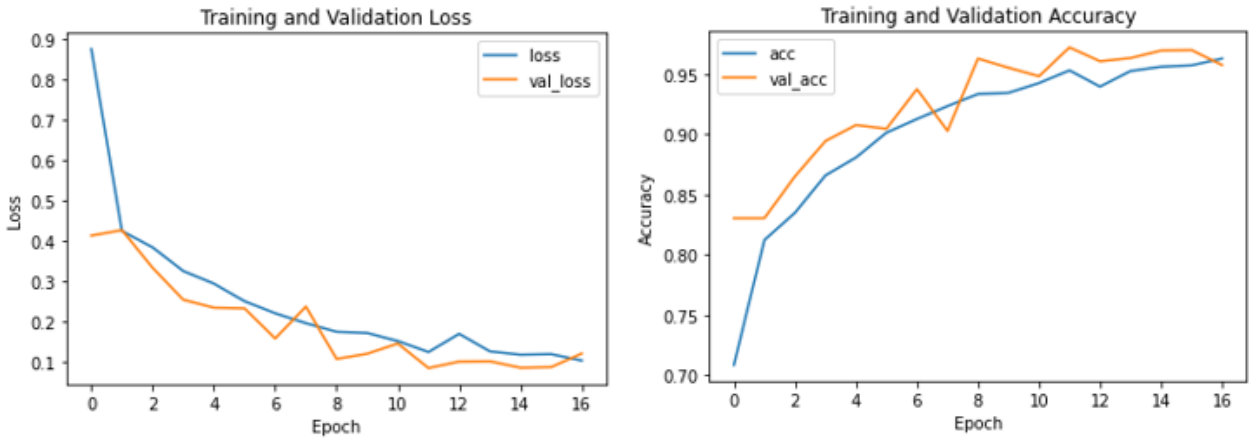

Figure 10: Deep Model with Data Augmentation: training and validation loss (left), training and validation accuracy (right)

5.2 Transfer Learning Models

Model Architectures

Two transfer learning models were trained: one model used the ResNet-50 architecture as a base model and the other used DenseNet-121 as a base model. Both architectures are accessible via TensorFlow 2.0's Keras Applications package. In both cases, weights trained on ImageNet were used for the base model. The ResNet-50 and DenseNet-121 architectures were selected because they had shown previous success when applied to classification of histopathological images. Additionally a custom head was applied to each base model so that we could make predictions on the "Lung and Colon Cancer Histopathological Images" dataset. For both models, the **custom head** consisted of the following layers:

1. **Flattening Layer:** A flattening layer to convert the output of the convolutional layers into 1 dimension.
    - For the **ResNet-50** model, the output contained 294912 values.
    - For the **DenseNet-121** model, the output contained 147456 values.
2. **Dense Layer**: A dense layer with 200 nodes using ReLU activation. Outputs 200 values.
3. **Dense Layer**: A dense layer with K nodes (K=5 for this dataset) using Softmax activation. Each node corresponds to one of the image classes. Outputs K probabilities, one for each image class.

ResNet-50

Adding layers to make CNNs deeper can help the network solve more complex problems because more layers allow more features, and more complex features, to be extracted. However, after a certain threshold additional layers actually tend to have a negative impact on model performance. This is often due to either the Vanishing Gradient or Exploding Gradient problem. Residual networks make it possible to avoid this problem and train deeper networks. ResNet-50 is a type of residual network that uses 50 layers. Residual networks allow deeper models to be trained effectively using residual blocks, which allow for "skip connections". Skip connections set up a "shortcut" for gradients to pass through and make it much easier for the layers in the model to learn identity functions. This allows outputs from earlier layers to be added to the outputs of stacked layers, ensuring that later layers perform at least as well as earlier layers. This allows deeper networks to be trained.

DenseNet-121

Like ResNet architectures, DenseNet architectures also help to overcome problems associated with increasing model depth (see above). DenseNet works by connecting every layer with every other layer. This helps to facilitate maximum information flow across the network. By maximizing connectivity, DenseNet architectures are able to have very narrow convolutional layers that contribute only a small number of feature maps to the model. The result is an avoidance of redundant feature maps and model layers that require fewer parameters than the

layers in a traditional CNN. Maximizing connectivity also means that each layer in a DenseNet model has access to the original input image.

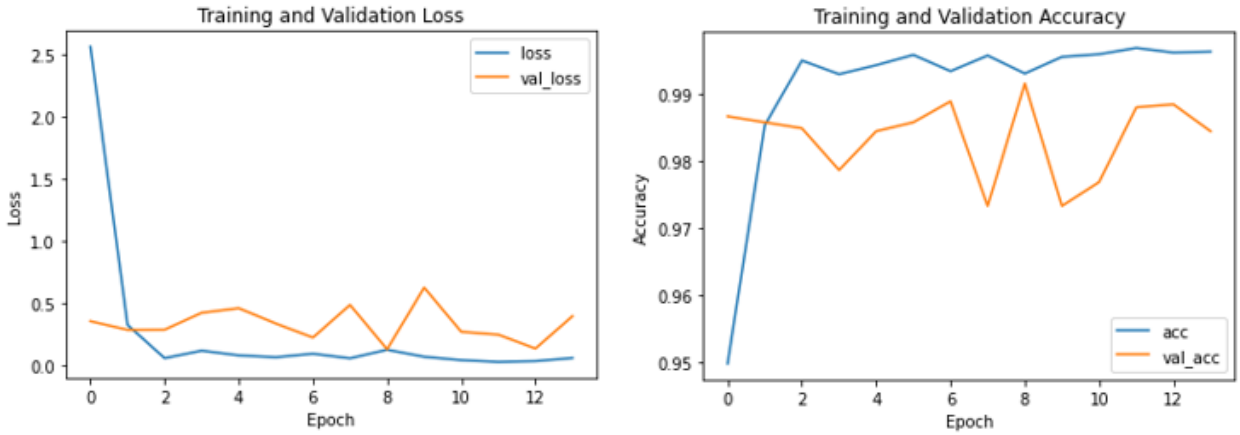

Figure 11: ResNet-50 Model: training and validation loss (left), training and validation accuracy (right)

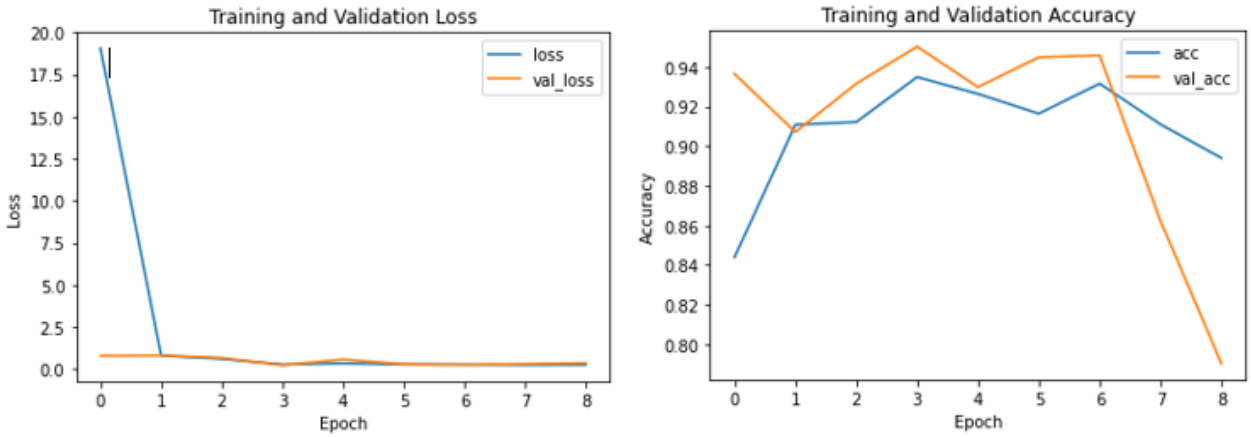

Figure 12: DenseNet-121 Model: training and validation loss (left), training and validation accuracy (right)

References:

- https://www.sciencedirect.com/science/article/pii/S0933365719307110
- https://arxiv.org/ftp/arxiv/papers/2112/2112.13553.pdf
- https://viso.ai/deep-learning/resnet-residual-neural-network/
- https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a

5.3 Model Evaluation

Model Evaluation Metric

Accuracy was selected as the performance metric to evaluate the models. Accuracy was an appropriate metric because the dataset was perfectly balanced (5000 images per image

class). The original model with the highest accuracy and the transfer learning model with the highest accuracy were selected for further investigation (this will be detailed below).

Model Accuracy Scores

After evaluating each model on the test dataset, the model accuracy scores (in descending order) were as follows:

	Model	Accuracy	Transfer Learning
5	ResNet50	0.9896	YES
4	Deep Augmented	0.9704	NO
2	Data Augmentation	0.9700	NO
0	Baseline	0.9200	NO
3	Deep	0.9072	NO
1	Dropout	0.9016	NO
6	DenseNet121	0.7776	YES

Figure 13: Model accuracy scores

The **Deep Model with Data Augmentation** (best original model) and the **ResNet50 Model** (best transfer learning model and best overall model) were analyzed further.

Deep Model with Data Augmentation: Further Analysis

1. Confusion Matrix

The Deep Augmented Model performed well overall on the test data. It did not misclassify any normal lung tissue. It was also 100% accurate when the predicted class was colon ACA. Interestingly, the model's strengths and weaknesses were those that were predicted by the PCA analysis during EDA. Specifically, the model's biggest problems were differentiating between the two types of lung cancer and, to a lesser degree, differentiating between normal and cancerous colon tissue. Also in agreement with the PCA analysis, the model was highly successful at classifying normal lung tissue. Overall, the biggest concern would be the cases where the model classifies colon ACA as normal colon tissue. This could lead to a failure to detect cancer in a patient. On the other hand, when the model predicts the wrong type of lung

cancer it is still detecting cancer. Once cancer is detected, the patient's case would likely be examined by medical experts who would be able to correctly classify the tumour type.

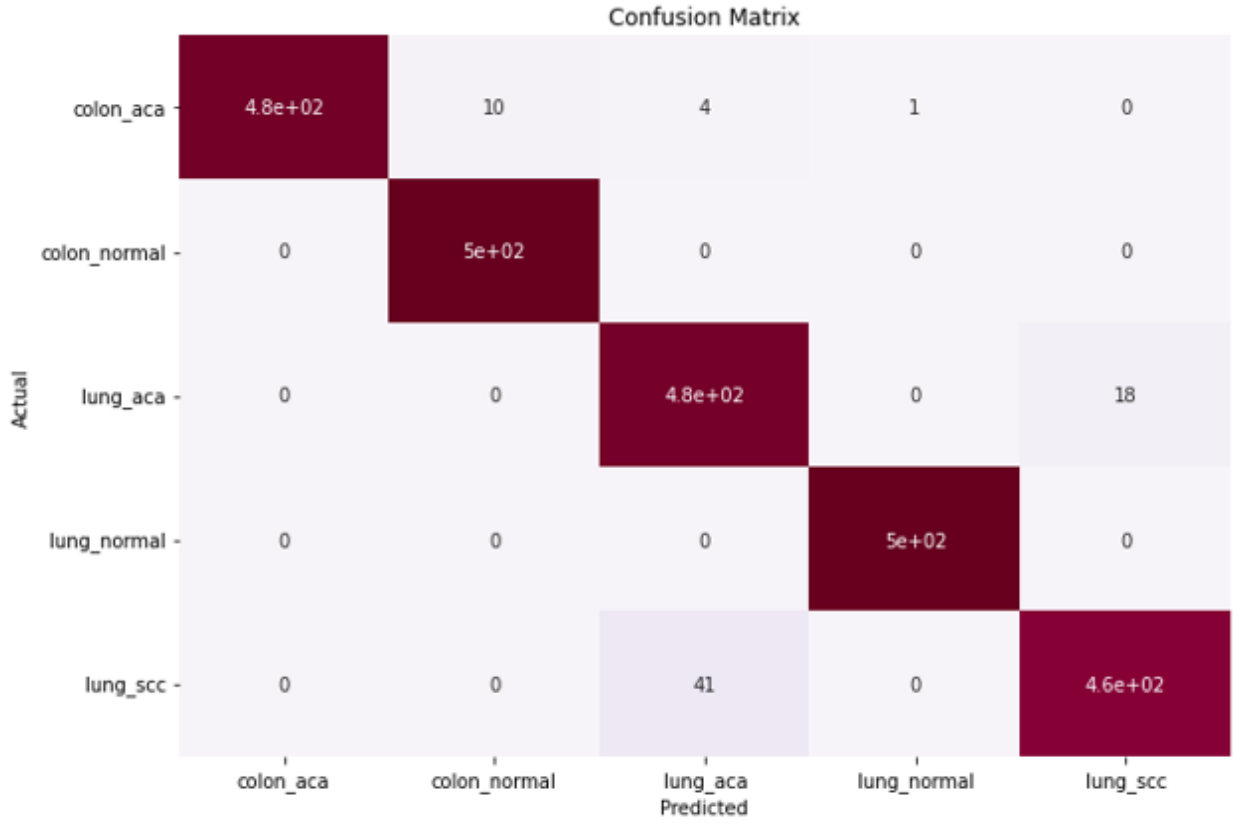

Figure 14: Confusion Matrix - Deep Model with Data Augmentation

2. Misclassified Images

Examples of misclassified images were examined for two of the model's main problem areas: colon ACA cases that were misclassified as normal colon tissue and lung SCC cases that were misclassified as lung ACA cases. It is difficult to comment on these images. However, a trained pathologist may be able to explain why the model struggles in these cases. There may be common features among the images that could explain the model's misclassification. It is also possible that these cases have very similar appearances to the predicted class and would be among the more challenging cases for a pathologist to classify.

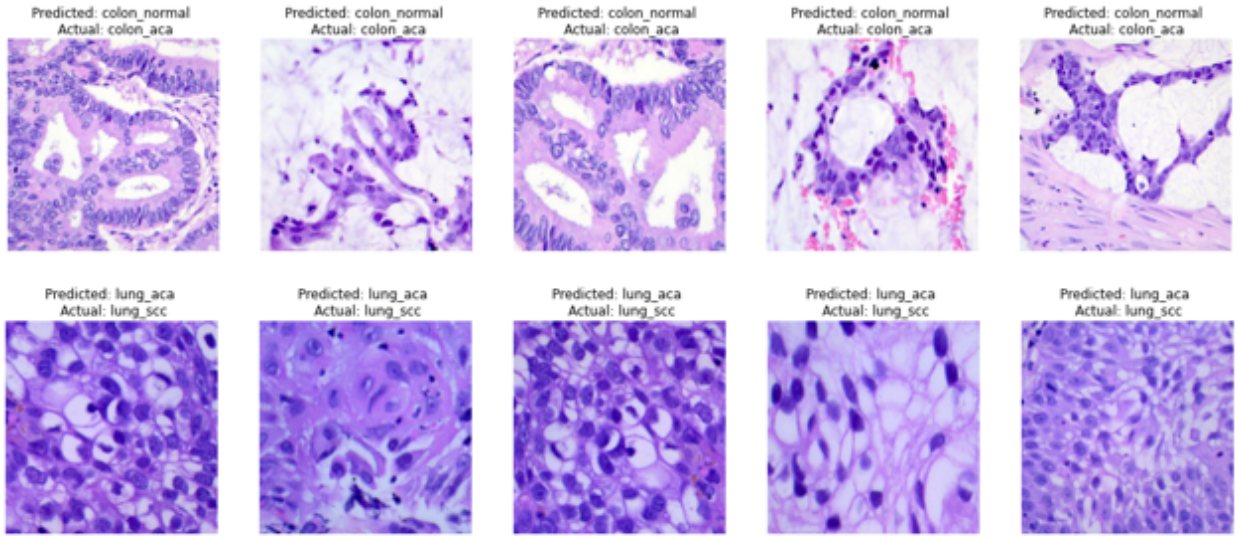

Figure 15: Examples of misclassified images: colon ACA misclassified as normal colon tissue (top row), lung ACA misclassified as lung SCC (bottom row)

3. Feature Maps

Feature maps for the first convolutional layer, last convolutional layer, and an intermediate convolutional layer were displayed for an example image from each image class. It is difficult to comment on the outputs. However, a trained pathologist may be able to provide some insights regarding the features that the model is extracting from the original images. That said, deep learning models tend to have low interpretability so it may not be possible to make sense of any of the extracted features. Below is an example of feature maps that were generated by the first convolutional layer of the model:

Figure 16: Sample feature maps generated by the first convolutional layer of the Deep Model with Data Augmentation

ResNet-50 Transfer Learning Model: Further Analysis

1. Confusion Matrix

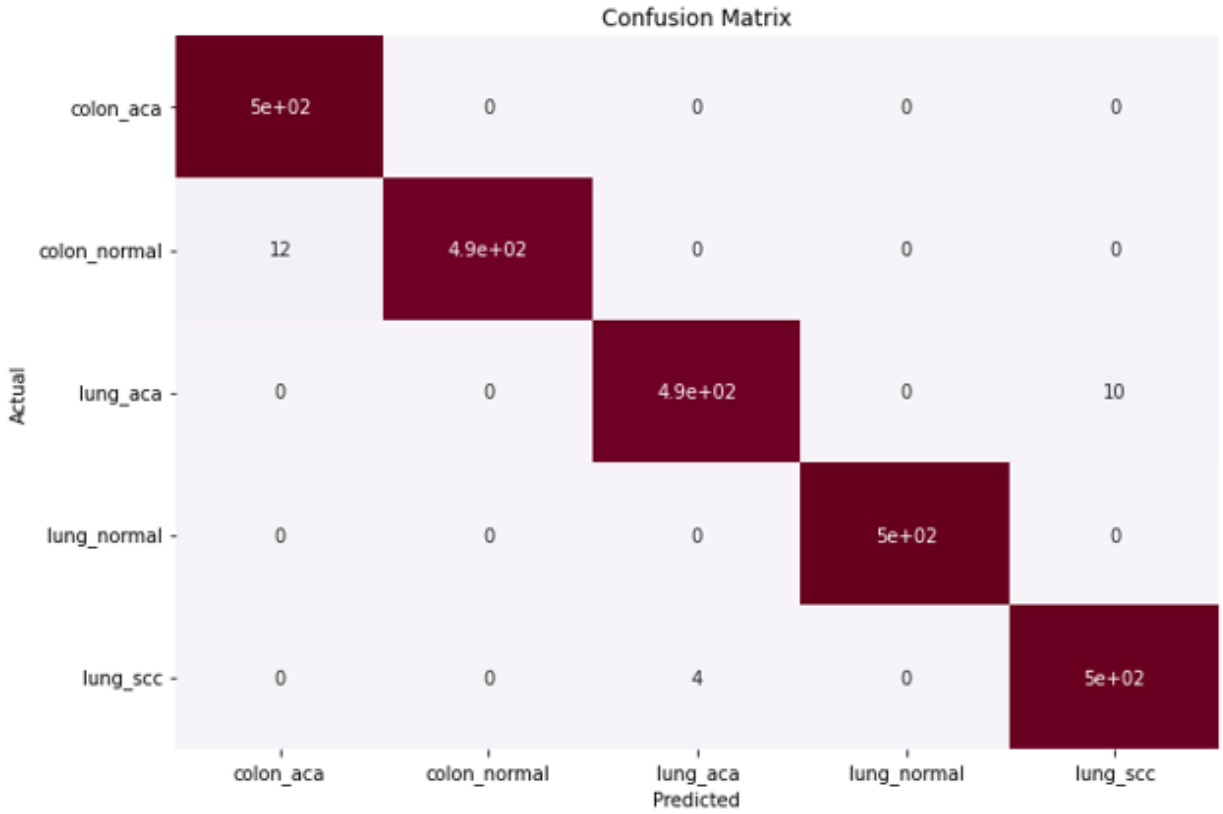

Figure 17: Confusion Matrix – ResNet-50 Model

The ResNet50 Model was the most accurate model when evaluated on the test set. It did not misclassify any normal lung or colon ACA tissue. It was also 100% accurate when the predicted class was normal colon or normal lung tissue. Interestingly, the model's strengths and weaknesses were those that were predicted by the PCA analysis during EDA. Specifically, the model's biggest problems were differentiating between the two types of lung cancer and differentiating between normal and cancerous colon tissue. Regardless, the model performed very well. Also in agreement with the PCA analysis, the model was highly successful at classifying normal lung tissue. Fortunately, this model's errors tended to be false positives, rather than false negatives, when predicting colon cancer. Therefore, it is less likely that colon cancer cases will go undetected compared to the Deep Model with Data Augmentation. There were also a few lung cancer cases where the model predicted the incorrect cancer type. However, since cancer was still predicted this would not lead to a patient's cancer going undetected.

2. Misclassified Images

Examples of misclassified images were examined for two of the model's main problem areas: normal colon tissue that was misclassified as colon ACA and lung ACA cases that were misclassified as lung SCC cases. It is difficult to comment on these images. However, a trained pathologist may be able to explain why the model struggles in these cases. There may be common features among the images that could explain the model's misclassification. It is also possible that these cases have very similar appearances to the predicted class and would be among the more challenging cases for a pathologist to classify.

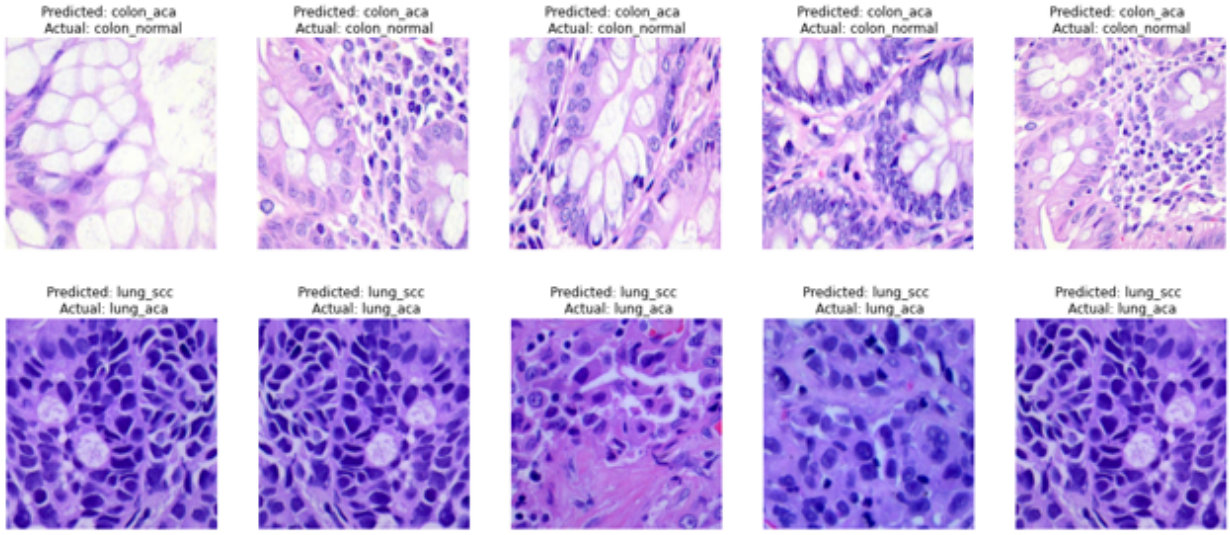

Figure 18: Confusion Matrix – ResNet-50 Model

3. Feature Maps

Feature maps for the first convolutional layer, last convolutional layer, and an intermediate convolutional layer were displayed for an example image from each image class. It is difficult to comment on the outputs. However, a trained pathologist may be able to provide some insights regarding the features that the model is extracting from the original images. That said, deep learning models tend to have low interpretability so it may not be possible to make sense of any of the extracted features. Below is an example of feature maps that were generated by the first convolutional layer of the model:

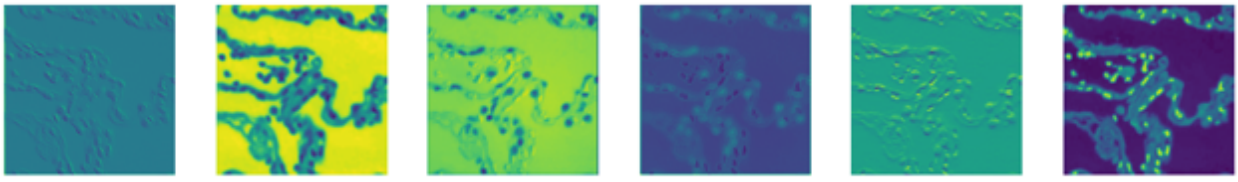

Figure 19: Sample feature maps generated by the first convolutional layer of the ResNet-50 Model

5.4 Final Model

The **ResNet-50 Model** was selected as the final model. The ResNet-50 model had the highest accuracy among all the models (**98.96% accuracy**). Further analysis revealed that the ResNet-50 model also minimized the number of missed cancer diagnoses on the test dataset. For a task such as cancer diagnosis, it is very important to minimize the number of false negatives predicted by the model.

Section 6: Conclusions

6.1 Effective Classification of Histopathological Images

When looking to classify histopathological images, it appears that it is possible to achieve a high degree of accuracy with CNNs. The ResNet-50 transfer learning model was able to achieve 98.96% accuracy. Even the Deep Model with Data Augmentation, a relatively simple model compared to the ResNet-50 model, was able to achieve an accuracy of 97.04%. Additionally, due to computational constraints when building and training the models, there is likely room for even further improvements (discussed below in section 6.2).

Dropout layers did not appear to be an effective technique for training a CNN on the "Lung and Colon Cancer Histopathological Images" dataset. However, it is possible that different dropout architectures would be more effective. Data augmentation was a highly effective technique and improved the performance of both the Baseline Model and the Deep Model. Adding depth (additional convolutional layers) was effective, but only when extra training data was generated via data augmentation.

Transfer learning models were expected to be the most accurate for classifying histopathological images. This was true in terms of the ResNet-50 model, which had the highest accuracy and was the selection for the final model. However, the DenseNet-121 performed surprisingly poorly.

6.2 Recommendations for Future Investigation

- There were limitations due to available computing power. If more computing power was available it would be possible to take the following steps:
    - Train more models, which would allow a wider variety of model architectures to be tested including deeper and wider models.
    - Train the models without scaling down the size of the images. This may allow the models to extract features more effectively resulting in improved model performance.

    - Perform hyperparameter tuning on parameters such as learning rate, CNN kernel size, number of nodes in CNN dense layers, dropout rate in the dropout layers, and optimizer selection.
        - The model with dropout layers performed particularly poorly. Testing different architectures with dropout layers or experimenting with different dropout rates may be beneficial.
- Although the dataset contained 25000 images, it contained 750 original images. The Augmentor package was used by the creators of the dataset to increase the size to 25000. Therefore, the model may be less successful when generalizing to images outside of the dataset. If at all possible, it would likely be beneficial to collect more raw data on which to train the models.
- Transfer learning models benefit from fine tuning in some cases. After the model is trained some, or all, of the non-trainable layers can be unfrozen and the model can be retrained. A very low learning rate should be used for fine tuning to avoid overfitting.
- Data augmentation as a preprocessing step was very effective at improving the performance of the Baseline Model and the Deep Model. Therefore, it may be worth training/testing the transfer learning models with data augmentation.
- It may be worth experimenting with training different models for each tissue type (i.e. separate models for lung tissue and colon tissue). The tissue type would typically be known ahead of time so feeding input data into the correct model should not pose a problem. This could be beneficial, as it might allow models to extract more specialized features. However, this approach would reduce the size of the training set for each model. Therefore, it could have the consequence of impairing the ability of some, or all, models to extract useful features from the image data.
- Collecting image data on more tissue and/or cancer types would allow a model to make predictions on a broader set of histopathological image classes. This would increase the utility of the model in a clinical setting. However, it would be important to ensure that the model's ability to make accurate predictions does not diminish as more classes are introduced. If this was to become a problem, it may be beneficial to experiment with different model architectures or acquire more training data if possible.
- Identify tumours within images (image segmentation) in addition to classification