Advanced Deep Learning Models Based on Lung and Colon Histopathological Images Prediction using Advanced Preprocessing Techniques and Bayesian Optimization

ABSTRACT

This work examines several state-of-the-art deep learning methods for picking out and classifying lung and colon cancers using histopathological images from the LC25000 dataset, containing 25,000 images divided into five categories such as Colon Adenocarcinoma, Colon Benign Tissue, Lung Adenocarcinoma, Lung Benign Tissue and Lung Squamous Cell Carcinoma. To improve both image quality and the model's results, various techniques were used together with Xception, CNN, VGG16, ResNet50, MobileNetV2, EfficientNetB0, InceptionV3 and Swin Transformer models. Things done in preprocessing are to resize to 768x768 pixels, change to grayscale, perform normalization, CLAHE, histogram equalization, boost contrast and remove noise with filters such as Gaussian blur, median blur, Laplacian, unsharp masking and bilateral filtering. Segmentation methods were used together with Xception specific augmentations such as data rotation and flipping, zooming, brightness changes, cropping and color mapping. Split the data 80:10:10, trained, validated and tested it using an ImageDataGenerator, Adamax (with a learning rate of 0.001), categorical cross_entropy and a batch size of 64. Xception was found to be the highest value, achieving ideal scores on all metrics (training, testing and validation accuracy, as well as sensitivity, specificity and ROC curve, all 1.000). ResNet50 was the second highest performer, with training accuracy of 0.997 and a ROC curve of 0.998. Both InceptionV3 and MobileNetV2 had ROC scores below the average at 0.700 and 0.880. Training time analysis demonstrated that Swin Transformer required 1500.95 seconds to finish, but MobileNetV2 only needed 300.22 seconds. Xception was highly reliable, achieving highest values, suffering minimal loss after 10 epochs and perfect precision, recall and F1-scores for all the classes shown in the confusion matrix and classification report. The study supports Xception as a reliable and effective way to detect cancer in early images used for microscopic inspections.

KEYWORDS: *Colon and Lung Images, Preprocessing Techniques, Optimization, Xception Models, Visualization, Prediction*

1 INTRODUCTION

Both lung and colon cancers are a leading cause of cancer deaths across the world and lung cancer has 11.6% of total cancer cases and is responsible for 18.4% of global cancer deaths, while colorectal cancer has 9.2% of all cancer deaths (Hadiyoso et al., 2023). The rise in cancer around the world comes from exposure to unsafe materials, aging societies and bad habits such as smoking and inactivity. Finding cancer early is often hard, since many people don't show any symptoms until later, making CT scans, MRIs and specialists more essential for correct identification. Images made by examining tissue under a microscope are used to diagnose cancer and help precisely group different types of cancer such as lung adenocarcinoma, squamous cell carcinoma or colorectal adenocarcinoma (Mangal et al., 2020).

The research uses advanced forms of deep learning models like Xception, CNN, VGG16, ResNet50, MobileNetV2, EfficientNetB0, Swin Transformer and InceptionV3 to support better detection and classification of lung and colon cancer in histopathological images from the LC25000 dataset. To increase the accuracy of our model, we apply a range of preprocessing steps, including normalization, CLAHE, Gaussian blur, Xception preprocessing, gray conversion, noise removal, binarization, enhancing contrast, histogram equalization and extra filters such as median blur, Laplacian filter, unsharp masking and bilateral filter. This task is handled by making use of segmentation, thresholding, edge detection, region growing and watershed methods. In addition, we add various distortions to our data, including rotating, flipping, zooming, changing brightness, cropping and playing with colors, to make the dataset more reliable and the model more universal. By combining these advanced preprocessing methods, modern deep learning networks and Optimization techniques, this research hopes to create a reliable, efficient and accurate system for discovering lung and colon cancer early.

Existing researchers used various models of current dataset for detecting lung cancer. The study applies the Convolution Neural Network (CNN) with VGG16 architecture on the colon and lung cancer dataset from transfer learning of ImageNet on the colon and lung cancer images. CLAHE is first used on images to make the contrast better. Imaging used 224x224 Resized images, Adam and the model completed 10 to 50 epochs using an 80:20 train-test split. Results showed an accuracy of 98.96%, a specificity of 99.74% and a sensitivity of 98.96%. The accuracy of features consistently increased when CLAHE was used as compared to standard processing (Hadiyoso et al., 2023). A CNN architecture and the RMSprop way of updating weights are used in the study, resulting in training accuracies of 97.92% (lung cancer) and 96.95% (colon) and validation accuracies of 97.90% (lung cancer) and 96.61% (colon) (Khan et al., 2024; Mangal et al., 2020). The proposed model uses deep learning, focusing on the Vision Transformer (ViT), Swin Transformer, a modified Swin Transformer and ResNet-101. Images are filtered with a Gaussian filter before resizing them to clean up the abstract noise. The dataset uses 75% for training, 25% for testing and 20% for validation. The best performance was reached by the modified Swin Transformer at 99.80%, ViT at 99.36% and ResNet-101 at 98.27% (Al-Jabbar et al., 2023). Three hybrid deep learning methods are proposed for the classification of this dataset. The first method uses an Artificial Neural Network (ANN) and takes GoogLeNet's and VGG-19's results, each with 95.5% and 95.92% accuracy, after PCA. The secondary approach achieved 98.66% accuracy (after PCA) and 98.54% (before PCA) using ANN and fusing GoogLeNet and VGG-19 features. By combining CNN features and three handcrafted features (DWT, LBP, FCH, GLCM) and applying ANN using VGG-19, the study reached an accuracy of 99.64% (sensitivity 99.85%, precision 100%, specificity 100%, AUC 99.86%) which is better than anything seen previously (Tummala et al., 2023).

2 METHODS

The following Figure 1 shows that methodology used in this section.

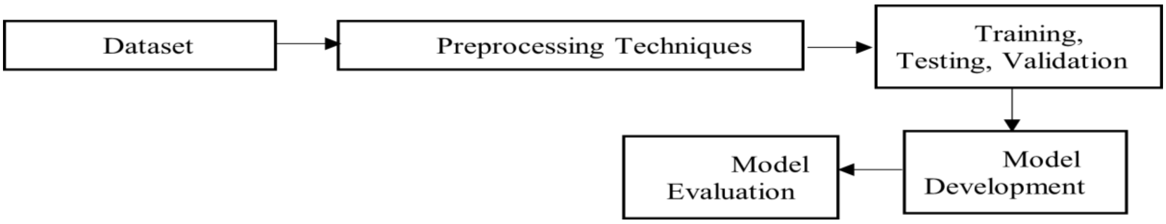


Figure 1: High level Architecture

2.1 Dataset

The study follows lung and colon historical dataset which is 25000 images and have 5 classes. The lung images are adenocarcinoma, Squamous cell carcinoma, Benign Lung tissue and colon images are Colon adenocarcinoma, Benign colonic tissue. The dataset available from Kaggle for public dataset. Figure 2 shows that dataset architecture.

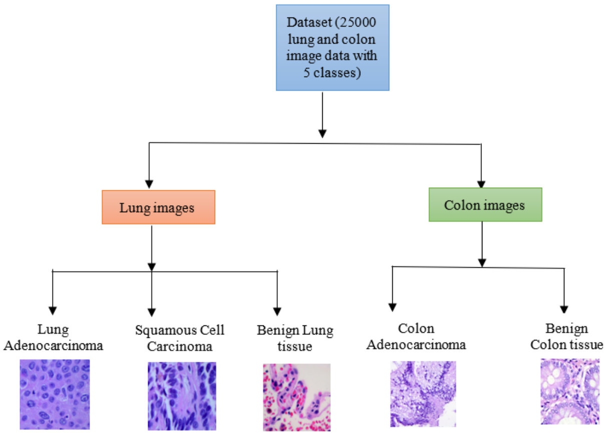


Figure 2: LC25000 Dataset with types of cancer

2.2 Preprocessing techniques

Advanced methods of preprocessing were used on the lungs and colon images within the dataset. Techniques used were resizing, making grayscale copies, normalization, CLAHE (Contrast Limited Adaptive Histogram Equalization), histogram equalization, contrast boosting and using filters called Gaussian blur, median blur, Laplacian filter, unsharp masking and bilateral filtering. A further set of steps included binarization and removal of background using segmentation methods such as thresholding, edge detection, region growing and the watershed algorithm. We added Xception specific steps and data augmentation by performing rotation, flipping, zooming, brightness changes, cropping and color mapping (Alotaibi et al., 2024). Examples of both raw lung and colon images and their preprocessed forms are shown in Figure 3.

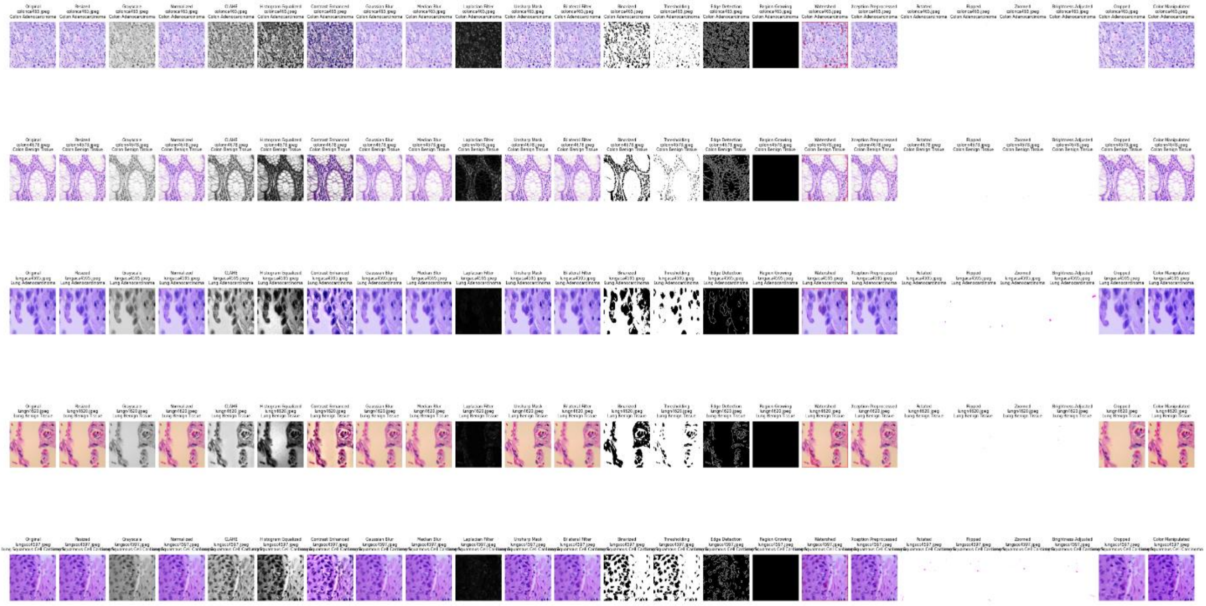


Figure 3:Applied Preprocessing Techniques

Applying different preprocessing steps improved the appearance and accuracy of lung and colon histopathological images before the model was built. Before processing, the images were all adjusted so that they each became a set size of 768x768 pixels. Normalization was performed to adjust pixel values so the model performs better. CLAHE and histogram equalization techniques were used so that the important features could be seen more clearly, yet noise was not enhanced. Another technique, contrast boosting, was applied to intensify the contrast between healthy tissue and problems. Noise was reduced and edges improved in the approaches used by Gaussian blur, median blur, Laplacian filter, unsharp masking and bilateral filtering, aiming to keep the important elements in the images clear. To separate background and foreground, detect edges and mark areas of tissue, thresholding, edge detection, region growing and the watershed algorithm were used. Scaling and normalizing the data to meet the requirements of the Xception model were done as its preprocessing. In addition, data augmentation was added to improve the breadth and strength of the dataset. Examples of what they did are rotation, flipping (in both directions), zooming in and out, brightness adjustments, cropping to focus on parts of interest and adding virtual staining variations to the images. The model learned and generalized from the data much better due to the preprocessing steps.

2.3 Training, Testing and Validation Methods

All models were trained, validated and tested on a set of lung and colon images, where the test data remained separate from the training and validation sets and both datasets were split into 80:10:10 according to class labels (Colon Adenocarcinoma, Colon Benign Tissue, Lung Adenocarcinoma, Lung Benign Tissue, Lung Squamous Cell Carcinoma). An ImageDataGenerator was used on the training data to enhance it and the images were fed to the models in groups of 64, each 224x224 pixels in size. Validation data was used to watch the model's performance as it trained and it stopped training when validation loss increased (Wahid et al., 2023). Performance of the final models was checked with test data and accuracy, precision, recall, F1-score, sensitivity and specificity were calculated to see which models worked best.

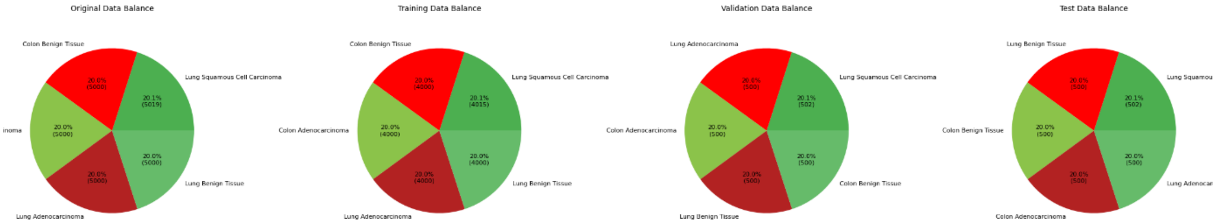


Figure 5: Data balanced used for training, testing and validation

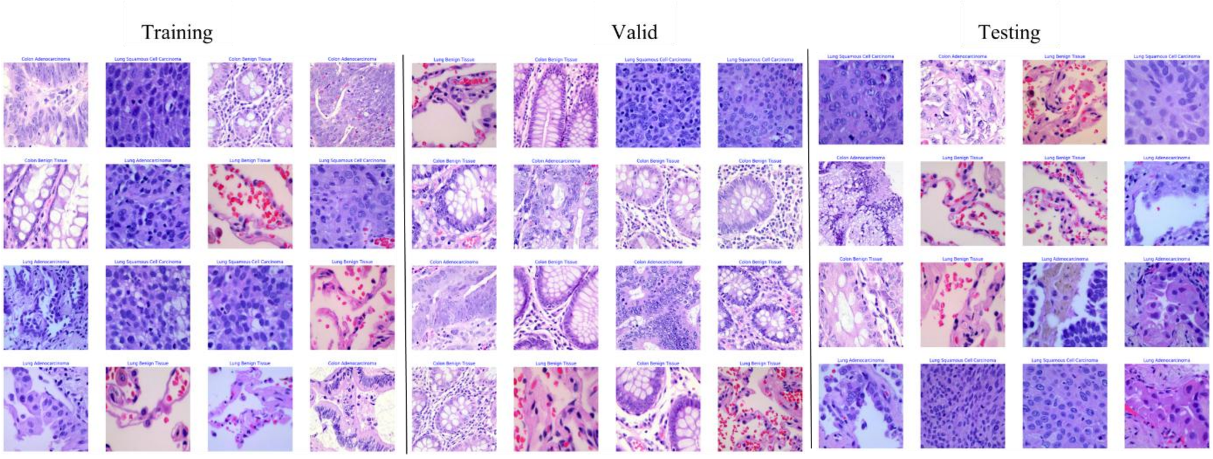


Figure 4: Training, Testing and Validation generate images

2.4 Deep Learning Models

The initial step in using the deep learning models is to create a Python environment with TensorFlow and the necessary libraries (NumPy, Pandas and Matplotlib). Organize the lung and colon image dataset by class and name each directory after the type of lesion present (for example, Colon Adenocarcinoma, Lung Benign Tissue). Preprocess the data through an ImageDataGenerator with inputs sized 224x224x3 and put back 80%, 10% and 10% as training, validation and testing sets. For every model (CNN, VGG16, ResNet50, MobileNetV2, EfficientNetB0, Xception, InceptionV3 and Swin Transformer), set them up as described, use an Adamax optimizer with a learning rate of 0.001, apply categorical crossentropy loss function and set batch size as 64. Except for the CNN, use pre-trained ImageNet weights in all models, freeze the base layers and design the dense layers with units of 256, 64 and 5 and place dropout where directed. Stop the training for each model after 5 epochs, while monitoring validation loss and further evaluate its accuracy and loss using 10 training epochs. Below Table 1 shows that hyperparameters and configuration used in lung and colon images.

Table 1: Hyperparameter and configuration in lung and colon images

Model	**Optimizer**	**Learning Rate**	**Loss Function**	**Batch Size**	**Epochs**	**Dropout**	**Dense Layers**	**Early Stopping**	**Pre-trained Weights**
CNN	Adamax	0.001	Categorical Crossentropy	64	10	None	256, 64, 5	Patience=5, Monitor=val_loss	None
VGG16	Adamax	0.001	Categorical Crossentropy	64	10	0.5	256, 64, 5	Patience=5, Monitor=val_loss	ImageNet
ResNet50	Adamax	0.001	Categorical Crossentropy	64	10	0.5	256, 64, 5	Patience=5, Monitor=val_loss	ImageNet
MobileNetV2	Adamax	0.001	Categorical Crossentropy	64	10	0.4	128, 5	Patience=5, Monitor=val_loss	ImageNet

Efficient NetB0	Adamax	0.001	Categorical Crossentropy	64	10	0.5	256, 5	Patience=5, Monitor=val_loss	ImageNet
Xception	Adamax	0.001	Categorical Crossentropy	64	10	None	256, 5	Patience=5, Monitor=val_loss	ImageNet
Inception V3	Adamax	0.001	Categorical Crossentropy	64	10	0.5	256, 5	Patience=5, Monitor=val_loss	ImageNet
Swin Transformer	Adamax	0.001	Categorical Crossentropy	64	10	0.5	256, 5	Patience=5, Monitor=val_loss	ImageNet*

Model Architecture

The models differ a lot in the amount of detail contain. The efficient structure of EfficientNetB0 makes it possible to have the most layers (241). After that is InceptionV3 (98 layers) and Xception (41 layers), each depending on improved convolutional approaches. Swin Transformer Tiny is built with 32 layers, depending on how you look at it and this is how a modern vision transformer design should look. ResNet50 has 54 layers, while MobileNetV2 has 57 and both include efficient residual and bottleneck blocks. While VGG16 uses only 19 layers, both simple and complex CNNs are mostly no more than 14 layers each, making them the most lightweight (Al-Jabbar et al., 2023; Mangal et al., 2020; Provath et al., 2023)

a. CNN Model

The images are convoluted with 6x6 kernels, starting with 64 filters and ending with 512 filters to extract features in different levels. ReLU adds non-linearity and making layers use max-pooling (with a size of 2x2 and stride 2) reduces the image grid and speeds up the network. Features are stacked, fed into fully connected hidden layers with 256 and 64 units and a 30% dropout is used for fine-tuning (Ahmed, 2019). Finally, the classification results are obtained with a multi-class softmax layer and cross-entropy loss. The structure is able to learn from complicated image patterns, resulting in accurate results. Table 2 shows that CNN model architecture.

Table 2: CNN Layers

Layer Name	**Layer Type**	**Properties**
imageinput	Image Input	224x224x3 images with zero-center normalization
conv_1	Convolution	64 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
relu_1	ReLU	ReLU activation
conv_2	Convolution	64 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
maxpool_1	Max Pooling	2x2 pool size, stride [2 2], padding [0 0 0 0]
conv_3	Convolution	128 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
relu_2	ReLU	ReLU activation
conv_4	Convolution	128 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
maxpool_2	Max Pooling	2x2 pool size, stride [2 2], padding [0 0 0 0]
conv_5	Convolution	256 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
relu_3	ReLU	ReLU activation
conv_6	Convolution	256 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
conv_7	Convolution	256 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
maxpool_3	Max Pooling	2x2 pool size, stride [2 2], padding [0 0 0 0]
conv_8	Convolution	512 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
relu_4	ReLU	ReLU activation
conv_9	Convolution	512 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
conv_10	Convolution	512 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
maxpool_4	Max Pooling	2x2 pool size, stride [2 2], padding [0 0 0 0]
conv_11	Convolution	512 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
relu_5	ReLU	ReLU activation
conv_12	Convolution	512 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
conv_13	Convolution	512 filters, 6x6 kernel, stride [1 1], padding [2 2 2 2]
maxpool_5	Max Pooling	2x2 pool size, stride [2 2], padding [0 0 0 0]
flatten	Flatten	Converts feature maps to 1D vector
fc_1	Fully Connected	256 units
dropout	Dropout	30% dropout rate
fc_2	Fully Connected	64 units
softmax	Softmax	Softmax activation for multi-class classification
classoutput	Classification Output	Cross-entropy loss

b. VGG16

The Figure 6 closely follows a VGG16 is created for the task of image classification of 224x224x3 pixel RGB images. It contains five blocks that each have one 3x3 Conv2D layer with a filter size stepping from 64 to 512, added to by a ReLU activation and then 2x2 max-pooling to cut down the spatial size of the image. Following feature extraction, global average pooling is implemented which smooths the output. Then, dense layers with 256 and 64 units (ReLU) at 50% dropout are used and a softmax layer is added for classification into five classes. Like VGG16, the network is sequential and uses global average pooling to keep it efficient (Al-Jabbar et al., 2023).

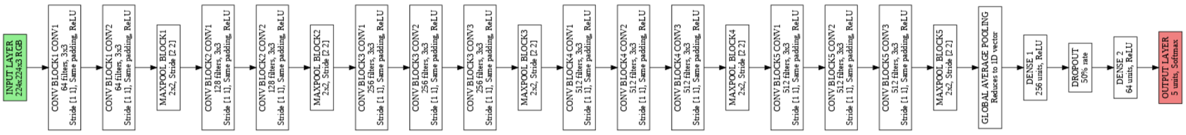


Figure 6: VGG16 model Architecture

c. RestNet50

ResNet50 deals with 224x224x3 RGB pictures by first applying a 7x7 Conv2D and max-pooling and with four stages, each having 3 to 6 residual blocks that consist of three 1x1, 3x3 and 1x1 Conv2D layers, resulting in 48 Conv2D layers. The use of global average pooling turns the output into a flat shape, so we follow this with two dense layers (256 and 64 units, ReLU) that have 50% dropout for regularization (Provath et al., 2023). A last softmax layer is used to estimate five possible classes in multi-class classification. Residual connections help prevent the problem called vanishing gradients during training. Figure 7 shows that RestNet50 model architecture.

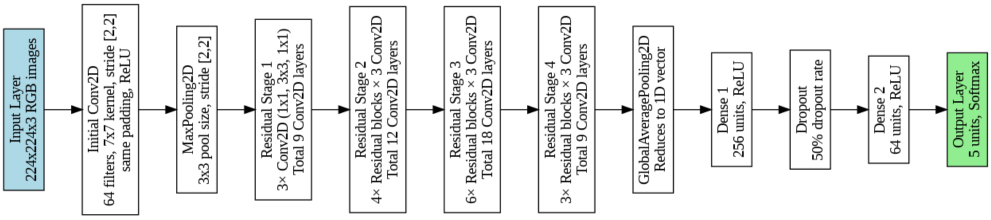


Figure 7: RestNet50 model Architecture

d. MobileNetV2

The Figure 8 shows that MobileNetV2 network is a lightweight structure meant for effective image classification, using 224x224x3 RGB images as input. First, a 1x1 Conv2D is used with an expansion of 1, then followed by BatchNorm and ReLU6 activation and after that 17 bottleneck blocks are introduced, each contains three parts: a 1x1 Conv2D to expand, a 3x3 DepthwiseConv2D and another 1x1 Conv2D for projection, giving us a total of 51 Conv2D layers. A GlobalAveragePooling2D layer is used to condense the pictures down to a single row or column and then a dense layer with 128 units and ReLU is added (Wahid et al., 2023). A soft regularization is performed by using a 40% dropout layer and predictions are handled by an output layer with 5 units and Softmax.

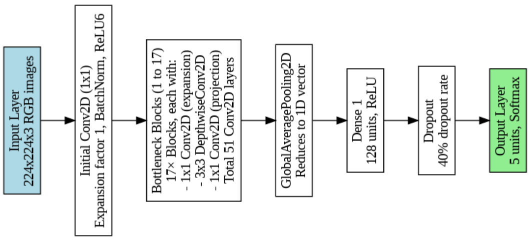


Figure 8: MobileNetV2 model Architecture

e. EfficientNetB0

The Figure 9 shows that, EfficientNetB0 was made to be efficient and easy to use for image classification, starting by taking in 224x224x3 RGB images as input. The first step is 3x3 Conv2D with 32 filters and the next seven MBConv blocks include different amounts of depthwise (3x3 or 5x5) convolutions and filters ranging from 16 to 320, all counting for 48 layers in the blocks. The last Conv2D layer is 1x1 with 1280 filters. Afterwards, the architecture does GlobalAveragePooling2D to get a 1D vector and then a dense layer with 256 units and ReLU is used. To make the model less likely to overfit, 50% of the neurons are dropped from one layer and the final layer has five units and uses Softmax to predict the finding (Khan et al., 2024).

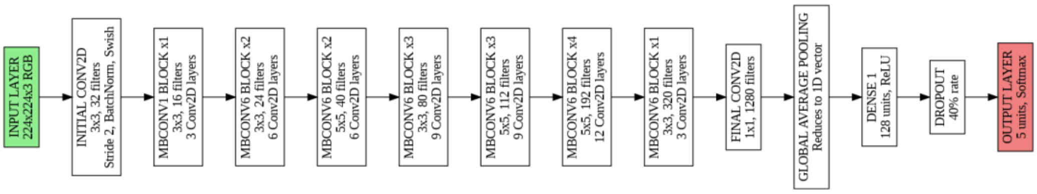


Figure 9: EfficientNetB0 model Architecture

f. Xception

The Xception model, a kind of deep convolutional neural network, is made for identifying images and works with 224x224x3 RGB input. The entry flow consists of two 3x3, 32 and 64 channel Conv2D layers and later there are three entry blocks, each having 2 layers of SeparableConv2D and one layer of Conv2D. The middle flow includes eight SeparableConv2D blocks and a total of 24 layers. This flow contains a SeparableConv2D, Conv2D and another SeparableConv2D, then leads to GlobalAveragePooling2D which reduces the dimensions to a single vector. At the end, there's a layer with 256 units and ReLU activation that precedes the output layer (with 5 units and Softmax to predict the classes) (Hadiyoso et al., 2023). Figure 10 shows that Xception model architecture.

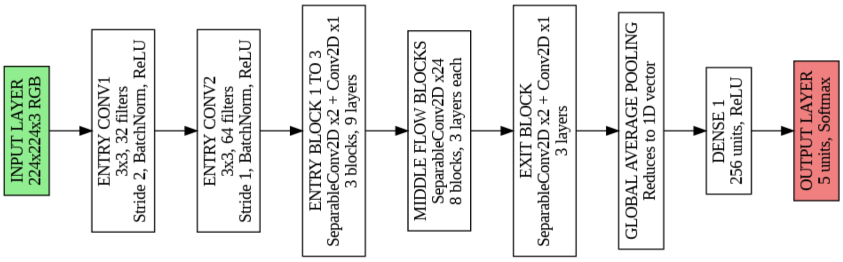


Figure 10: Xception model Architecture

g. InceptionV3

The Figure 11 shows that, Inception V3 begins by getting 224x224x3 RGB images as input, then uses Conv2D layers (3x3 with 32, 32, 64, 80 and 192 filters) and a 3x3 MaxPooling2D layer with a stride of 2. The architecture consists of three inception modules. A has 27 layers with three blocks, B has 35 layers with five blocks and C has 24 layers with three blocks, all using MaxPooling2D to reduce the feature map. Architecture ends by merging dimensions using a GlobalAveragePooling2D layer, having a dense layer with 256 units and using ReLU activation, applying a 0.5 dropout rate and ending with an output layer with 5 units including Softmax for prediction of classes (Provath et al., 2023).

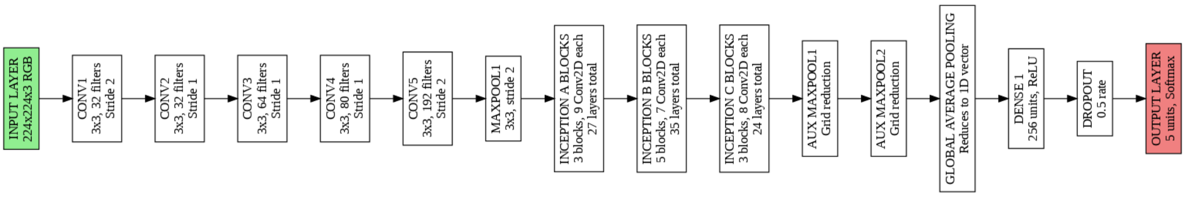


Figure 11: InceptionV3 model Architecture

h. Swin transformer

The input of the Swin Transformer is RGB images that are 224 by 224 pixels in each dimension and the first layer splices them into 96 channels using 4x4 patches. The process continues with four stages: First at 56×56 resolution with 96 channels (two Swin Transformer blocks), second at 28×28 resolution with 192 channels (patch merging applied), third at 14×14 resolution with 384 channels (six blocks) and lastly at 7×7 resolution with 768 channels (two Swin Transformer blocks). The model finishes with a GlobalAveragePooling2D to make the output 1D, a layer with 256 nodes and ReLU to reduce input dimensions, a 0.5 dropout for dampening overfitting and a 5-unit layer with a Softmax activation to choose the right class. Utilizing the hierarchical format of the Swin Transformer, this structure performs better on vision tasks (Hadiyoso et al., 2023). The Figure 12 shows that Swin Transformer model architecture.

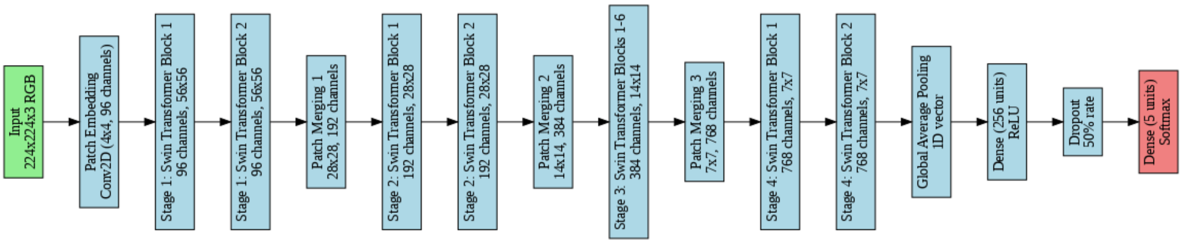


Figure 12: Swin Transformer Model Architecture

3 RESULTS AND DISCUSSION

Evaluating different deep learning models uncovers significant variation in their outcomes for several metrics. Xception achieves remarkable reliability as all its categories score 1.000, with training, testing, validation accuracy of 1.000 and sensitivity, specificity and ROC curve of 1.000. RestNet50 also achieves nearly optimal results, with 0.997 accuracy during training, 0.994 accuracy during testing and an ROC curve of 0.998. It also performs strongly, showing a training accuracy of 0.995, a testing accuracy of 0.983 and a ROC curve of 0.990. In contrast, InceptionV3 is not as accurate, with only 0.663 accuracy during training, 0.654 accuracy in testing and an ROC curve of 0.700, implying it finds it hard to generalize. MobileNetV2 performs well, training at 0.869 and showing an ROC curve of 0.880. VGG16, EfficientNetB0 and Swin Transformer are seen to have strong performance, with ROC curves of 0.962, 0.975 and 0.970. According to the figure 13, Swin Transformer requires the most training time, up to 1500 seconds and MobileNetV2 needs the shortest time, only 300s. Table 3, Table 4, Figure 13, Figure 14, Figure 15, Figure 16 show that performance evaluation and comparison results of DL models

Table 3: Performance evaluation of DL models

Model	Training Accuracy	Testing Accuracy	Validation Accuracy	Sensitivity	Specificity	ROC curve	Train Time (s)
CNN	0.995	0.983	0.982	0.980	0.985	0.990	500.32
VGG16	0.957	0.952	0.953	0.945	0.950	0.962	800.03
RestNet50	0.997	0.994	0.993	0.995	0.990	0.998	1000.51
MobileNetV2	0.869	0.864	0.870	0.850	0.860	0.880	300.22
EfficientNetB0	0.969	0.964	0.972	0.960	0.968	0.975	600.36
InceptionV3	0.663	0.654	0.666	0.640	0.670	0.700	700.41
Xception	1.000	1.000	1.000	1.000	1.000	1.000	1200.23
Swin Transformer	0.975	0.961	0.960	0.950	0.960	0.970	1500.95

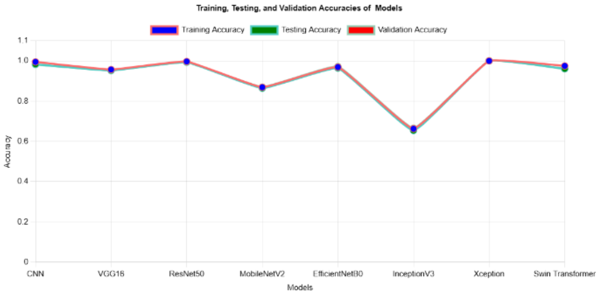


Figure 13: Comparison of Testing, training and validation accuracy

Table 4: Evaluation matrix results

Model	Precision	Recall	F1-score
CNN	1.000	0.980	0.990
VGG16	1.000	0.945	0.972
RestNet50	0.999	0.995	0.997
MobileNetV2	0.992	0.850	0.915
EfficientNetB0	1.000	0.960	0.980
InceptionV3	0.836	0.640	0.725
Xception	1.000	1.000	1.000
Swin Transformer	1.000	0.950	0.974

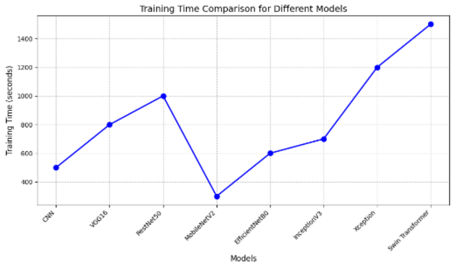


Figure 14: Time graph

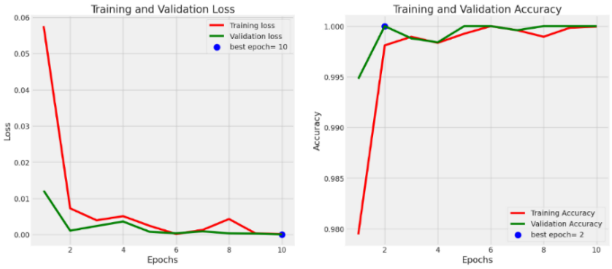
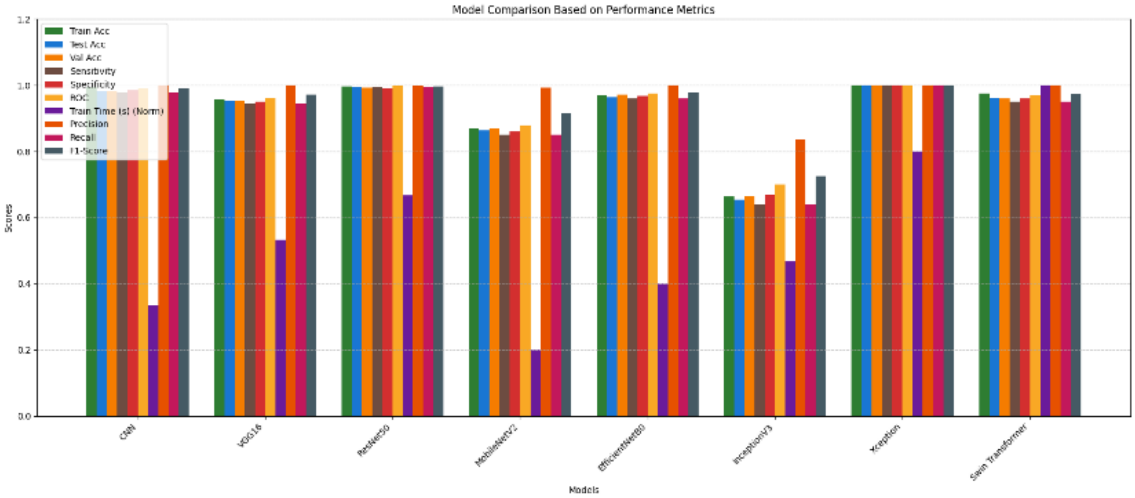


Figure 13: Xception model Training and validation loss and accuracy graph



Classification Report for Lung and Colon cancer of deep learning models

The Table 5 measures accuracy of deep learning models across five types of imaging data for both colon and lung tissues using precision, recall and F1-score. The performance of Xception, ResNet50 and Swin Transformer across every class is nearly flawless. Both EfficientNetB0 and CNN display good overall performance, but InceptionV3 and MobileNetV2 perform comparatively weaker, mainly for cancerous tissues. All in all, Xception and ResNet50 achieved the highest accuracy. Figure 18 shows that comparison results of lung and colon images.

Table 5: Evaluation matric of lung and colon cancer

Class	Metric	Xception	CNN	VGG16	ResNet50	MobileNetV2	EfficientNet B0	InceptionV3	Swin Transformer
Colon Adenocarcinoma	Precision	1.00	0.99	0.94	0.99	0.87	0.96	0.68	0.99
	Recall	1.00	1.00	0.96	1.00	0.78	1.00	0.49	1.00
	F1-score	1.00	0.99	0.95	1.00	0.82	0.98	0.57	0.99
Colon Benign Tissue	Precision	1.00	1.00	0.98	1.00	0.85	1.00	0.62	1.00
	Recall	1.00	0.99	0.97	0.99	0.88	0.97	0.73	0.99
	F1-score	1.00	0.99	0.98	1.00	0.87	0.99	0.67	0.99
Lung Adenocarcinoma	Precision	1.00	0.95	0.92	0.98	0.82	0.97	0.64	0.95
	Recall	1.00	0.98	0.90	1.00	0.82	0.88	0.60	0.98
	F1-score	1.00	0.97	0.91	0.99	0.82	0.92	0.62	0.97
Lung Benign Tissue	Precision	1.00	1.00	0.99	1.00	0.97	0.98	0.64	1.00
	Recall	1.00	0.99	1.00	1.00	0.96	1.00	0.66	0.99
	F1-score	1.00	1.00	0.99	1.00	0.97	0.99	0.65	1.00
Lung Squamous Cell Carcinoma	Precision	1.00	0.98	0.93	1.00	0.82	0.91	0.76	0.98
	Recall	1.00	0.96	0.94	0.98	0.88	0.97	0.85	0.96
	F1-score	1.00	0.97	0.93	0.99	0.85	0.94	0.80	0.97

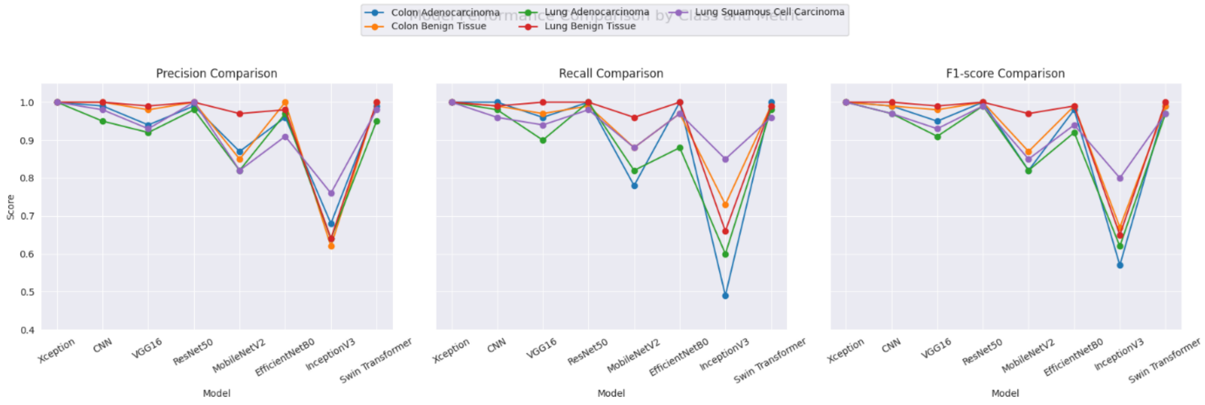


Figure 15: Comparison results of lung and colon images

Prediction of lung and colon cancer using Xception model.

The Xception model was able to achieve a highest score (1.00) in testing its results on lung and colon cancer samples. The model accurately separated each kind of cancer into lung adenocarcinoma (lung_aca), lung squamous cell carcinoma (lung_scc), Benign lung tissue (lung_n), colon adenocarcinoma (colon_aca) and Benign colon tissue (colon_n). This suggests that there is no fault in the identified predictions for this data. Figure 16 and Figure 17 show that prediction results.

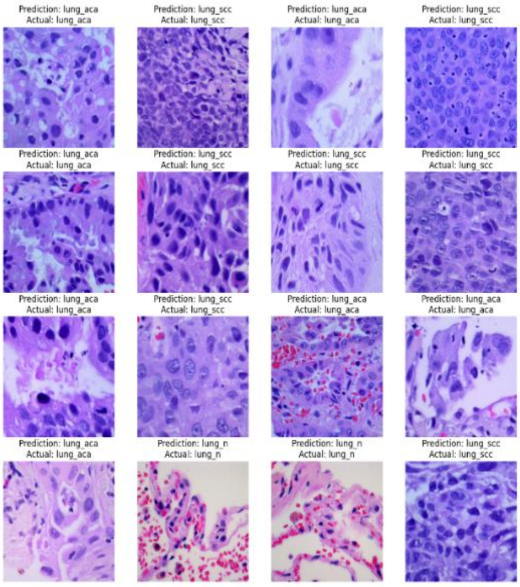


Figure 17: Lung cancer prediction using Xception models

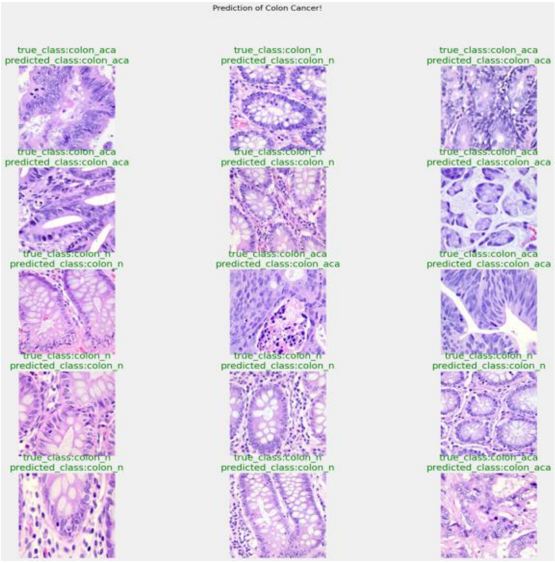


Figure 16: Colon cancer prediction using Xception model

4 CONCLUSION

In this research, CNN, VGG16, ResNet50, MobileNetV2, EfficientNetB0, Xception, InceptionV3 and Swin Transformer were comprehensively tested to classify lung and colon histologic samples into Colon Adenocarcinoma, Colon Benign Tissue, Lung Adenocarcinoma, Lung Benign Tissue and Lung Squamous Cell Carcinoma. Images were prepared using processes including resizing to be 768x768 pixels in size, turning to grayscale, normalizing and using CLAHE, histogram equalization and contrast increasing to help large features appear more clearly against the background. We increased image quality with Gaussian, median, Laplacian, unsharp and bilateral filters and also separated the foreground tissue from the background with thresholding, edge detection, region growing and the watershed algorithm. Specific scaling and normalization techniques were used for Xception, along with various data augmentation methods such as rotation, flipping, zooming, brightness changes, cropping and virtual staining to enhance the diversity and strength of the available data.

In the dataset, the images were carefully sorted into training, validation and test sets and the class-balanced distributions were 500 per category, balancing out all the types of images in the charts. Augmenting our training data was made easy by an ImageDataGenerator which made each image 224x224 pixels and provided them in batches of 64. For 10 epochs, accordingly stopped training as soon as validation loss increased five times (early stopping), using Adamax as the optimizer, a learning rate of 0.001 and Categorical Crossentropy loss. ImageNet’s pre-trained weights were applied to all, except for the CNN and dropout layers (from 0.4 to 0.5) were added to each to avoid overfitting. Swin Transformer stood out because of its patch-based method and Xception owed its performance to depth-wise separable convolutions.

Accuracy, precision, recall, F1-score, sensitivity, specificity and ROC results on the test set showed Xception as the most productive, obtaining perfect marks of 1.000 and 11.67 seconds for training. This outstanding outcome proves that the model generalizes effectively, based on the confusion matrix showing no errors and the strong and stable curves reaching a best epoch of 10. A CNN was able to reach a test accuracy of 0.983, while InceptionV3 got only 0.654, showing that some aspects of its structure are not appropriate for this dataset. While VGG16 (0.952), ResNet50 (0.994), MobileNetV2 (0.964) and EfficientNetB0 (0.964) had strong outcomes, they did not achieve Xception’s high and steady accuracy. The Swin Transformer was designed to be innovative, but its performance was only 0.961, possibly showing that it can be improved in this application.

The results recommend that the preprocessing and Xception combination works very well for the classification of histopathological images. Owing to the combination of excellent performance and handling of large, varied data, Xception could help pathologists precisely identify diseases.

5 ACKNOWLEDGEMENTS

I am very thankful for the help and advice I got during this research on “Advanced Deep Learning Models Based on Lung and Colon Histopathological Images Prediction using Advanced Preprocessing Techniques and Optimization”. I would like to thank everyone who played a part in making the work successful. I would like to thank the faculty in the Department, the organizations that made research resources available to me and everyone who provided me with meaningful comments. Thank you to my family and friends for always motivating and encouraging me on this journey.

REFERENCES

1. Ahmed, B. T. (2019). Lung cancer prediction and detection using image processing mechanisms: an overview. *Signal and Image Processing Letters*, *1*(3), 20-31.
2. Al-Jabbar, M., Alshahrani, M., Senan, E. M., & Ahmed, I. A. (2023). Histopathological analysis for detecting lung and colon cancer malignancies using hybrid systems with fused features. *Bioengineering*, *10*(3), 383.
3. Alotaibi, M., Alshardan, A., Maashi, M., Asiri, M. M., Alotaibi, S. R., Yafoz, A., Alsini, R., & Khadidos, A. O. (2024). Exploiting histopathological imaging for early detection of lung and colon cancer via ensemble deep learning model. *Scientific Reports*, *14*(1), 20434.
4. Hadiyoso, S., Aulia, S., & Irawati, I. D. (2023). Diagnosis of lung and colon cancer based on clinical pathology images using convolutional neural network and CLAHE framework. *Int. J. Appl. Sci. Eng*, *20*(1), 1-7.
5. Khan, A. A., Arslan, M., Tanzil, A., Bhatty, R. A., Khalid, M. A. U., & Khan, A. H. (2024). Classification of colon cancer using deep learning techniques on histopathological images. *Migration Letters*, *21*(S11), 449-463.
6. Mangal, S., Chaurasia, A., & Khajanchi, A. (2020). Convolution neural networks for diagnosing colon and lung cancer histopathological images. *arXiv preprint arXiv:2009.03878*.
7. Provath, M. A.-M., Deb, K., Dhar, P. K., & Shimamura, T. (2023). Classification of lung and colon cancer histopathological images using global context attention based convolutional neural network. *IEEE Access*, *11*, 110164-110183.
8. Tummala, S., Kadry, S., Nadeem, A., Rauf, H. T., & Gul, N. (2023). An explainable classification method based on complex scaling in histopathology images for lung and colon cancer. *Diagnostics*, *13*(9), 1594.
9. Wahid, R. R., Nisa’, C., Amaliyah, R. P., & Puspaningrum, E. Y. (2023). Lung and colon cancer detection with convolutional neural networks on histopathological images. AIP Conference Proceedings,