International Journal of Scientific Research and Management (IJSRM)
||Volume||09||Issue||11||Pages||EC-2021-621-634||2021||
Website: www.ijsrm.in ISSN (e): 2321-3418
DOI: 10.18535/ijsrm/v9i11.ec02

Application of Transfer Learning Technique for Detection and Classification of Lung Cancer using CT Images

[1]**Aashka Mohite**

[1] Department of Computer Engineering, Charotar University of Science and Technology, Anand 388421, India.

Abstract

Lung cancer is unquestionably a lung-influencing chronic condition that significantly hampers the respiratory system. It is the second most dangerous disease which causes increase in death rate. To resolve this issue, we had planned to create a very Convolutional Neural Network using Transfer learning to specifically classify the lung CT scans as normal, malignant, or benign in a subtle way. A dataset of 1100 lung CT scans is used for this purpose. For the most part, five Transfer Learning architectures are compared extensively in this classification such as MobileNet, VGG16, VGG19, DenseNet-201 and ResNet-101. Out of which, DenseNet-201 performed the best. The proposed strategy achieved a mean accuracy of 53 percent in the trials and 43% of mean F1-score, mean precision and mean recall.

Introduction

Lung Cancer is the second most normal malignancy in both males and females. If not identified at the early phase, it can cause passing. If the condition is diagnosed in time, the normal endurance rates for individuals with a cellular breakdown in the lungs ascend from 14 to 49 percent. As per WHO, 2.21million cases were recorded in 2020, though 1.80 million passed on because of lung cancer [1].

Lung cancer develops when cells in the lungs multiply uncontrolled way. Tumours grow because of this. These can impair a person's breathing and spread to other parts of the body. Smoking and alcohol use are by far the leading causes of lung cancer. Approximately 80% of lung cancer deaths are caused by smoking and alcohol consumption, and many others are caused by passive smoking Many people with lung cancer smoked in the past, but many others never smoked at all. Lung cancer in non-smokers can be caused by radon, second-hand smoke, environmental pollution, or other aspects [2]. Moreover, if cancer is present in another part of the body, then it may affect the lungs also.

As a result, early detection provides a reasonable chance of survival and can aid in the reduction of cancer malignancy. When compared to treatment to survive cancer therapy when diagnosed at an early stage, the chance of survival at the advanced stage is lower. As a result, the system is designed in such a way that it can distinguish between benign Tumours, which grow gradually and do not attack other body parts, malignant Tumours, which grow quickly and easily invade other body parts, and cancer-free patients.

Many imaging techniques, including computed tomography (CT), sputum cytology, chest X-ray, and magnetic resonance imaging (MRI), are used to detect lung Tumours early. Detection entails categorizing Tumours into two types: (i)non-cancerous (benign) Tumours and (ii)cancerous Tumours (malignant). In many cases, the prognosis of lung cancer is based on the knowledge of doctors, who may ignore some patients and cause problems. Furthermore, there is a reasonable chance of "false positives," which can have a negative impact on the patient's mental and financial well-being. As a result, by implementing image processing techniques, manual analysis can be improved. To achieve maximum efficiency and accuracy, Artificial Intelligence can be used.

This paper investigates the use of transfer learning in the task of detecting lung cancer from CT scans. Transfer learning is a strategy in which knowledge mined by a machine learning algorithm from one set of data is transferred to solve a different but related task involving new data, even if the volume of data available for the new task is limited. The knowledge gained by training a series of different convolutional neural networks (CNN) on a large scale, hierarchical dataset called ImageNet is applied to the task of detecting lung cancer from CT lung scans in this study.

Transfer learning is the process of using models that have been trained on one problem as a starting point for a related problem. It is adaptable, allowing pre-trained models to be used directly as feature extraction pre-processing, or integrated into entirely new models. In this experiment, a newly defined classifier that distinguishes between three categories is stacked on top of the pre-trained model. The study assesses and compares the results of five different transfer learning models, namely the MobileNet [3], VGG16 [4], VGG19 [5], DenseNet-201, and ResNet-101, which have been shown to be among the best and most consistent architectures in indicating the presence of cancer from lung scans in the literature.

The sections that follow present an analysis of recent scholarly work in the field of lung cancer detection. Following that, a description of the methodology used in the study's implementation is provided. Finally, an assessment of the results of the experiment is discussed.

Literature Review

W. Ausawalaithong, A. Thirach, S. Marukatat, and T. Wilaiprasitporn [6] employed deep learning in combination with a transfer learning approach to predict lung cancer from chest X-ray pictures gathered from several data sources. A fully connected layer was applied to a 224X224 image with a 121-layer Densely Connected Convolutional Network (DenseNet-121) and a single sigmoid node. For various image source datasets, the suggested model achieved 74.436.01 percent mean accuracy, 74.969.85 percent mean specificity, and 74.6815.33 percent mean sensitivity.

Matko Šarić, Mladen Russo, Maja Stella, Marjan Sikora[4] suggested CNN architectures based on VGG and ResNet for lung cancer diagnosis using whole-slide histopathology pictures, and the results were compared using the receiver operating characteristic (ROC) plot. Patch level accuracy for VGG16 and ResNet50 was 0.7541 and 0.7205, respectively, which is pretty low. The authors indicated that the low accuracy of the proposed models was owing to the considerable pattern variability between distinct slides.

S. Sasikala, M. Bharathi, and B. R. Sowmiya proposed that CNN be used on CT scan images to detect and categorize lung cancer [7]. They worked with MATLAB and had two steps in training: the first phase was to extract valuable volumetric features from the input data, and the second phase was classification. With 96 percent accuracy, their proposed technology could distinguish between cancerous and non-cancerous cells.

Raul Victor Medeiros da Nóbrega, Solon Alves Peixoto, Suane Pires P. da Silva and Pedro Pedrosa Rebouças Filho proposed a method to classify lung nodule malignancy using deep transfer learning [8]. VGG16, VGG19, MobileNet, Xception, InceptionV3, ResNet50, ResNet-ResNet-V2, DenseNet169, DenseNet201, NASNetMobile, and NASNetLarge were utilised as features extractors to process the Lung Image Database Consortium and Image Database Resource Initiativeon (LIDC/IDRI). Naive Bayes, MultiLayer Perceptron (MLP), Support Vector Machine (SVM), Near Neighbors (KNN), and Random Forest (RF) classifiers were used to classify the deep features returned. CNN-ResNet50 with SVM-RBF, which produced an ACC of 88.41% and an AUC of 93.19%, was the top deep extractor and classifier combination.

Methodology

As shown in Figure 1, the planned system follows a general flow image processing pipeline of data collection, preprocessing, segmentation, feature extraction, and classification.

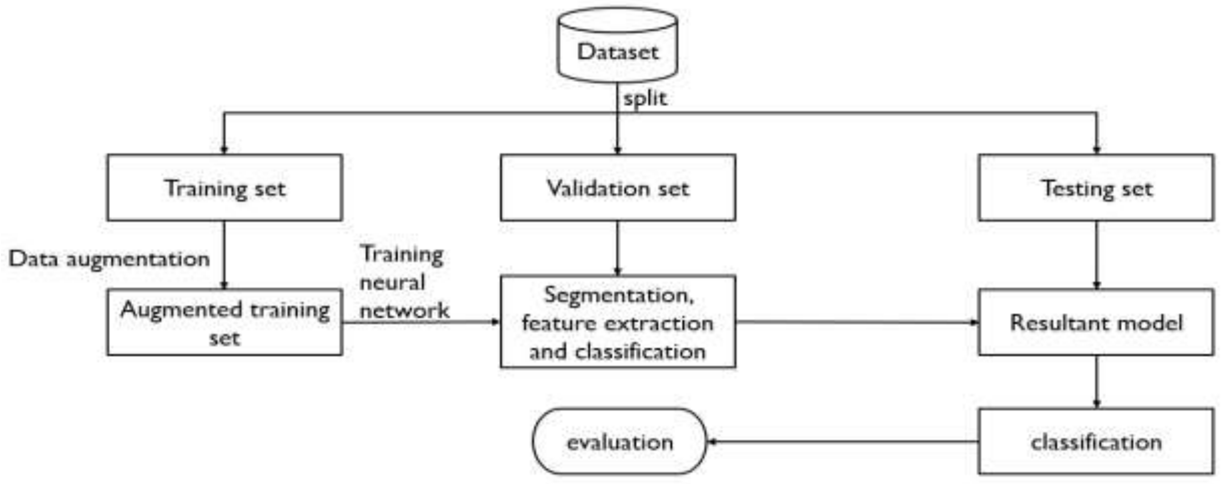


Figure 1: Flowchart for the Proposed System

Initially, the dataset was classified into three categories: benign, malignant, and cancer-free. Following that, the dataset is divided into three parts: training, validation, and testing. Data augmentation is done on the training set, which generates dummy data which makes the training of models less vulnerable to a phenomenon called overfitting. Following that, the transfer learning models' convolutional bases perform automatic segmentation and feature extraction, while the classifier is stacked on a fully connected artificial neural network with a dense layer and softmax as an activation function.

Dataset Acquisition

The proposed system makes use of the IQ-OTH/NCCD lung cancer dataset [9]. Over the course of three months, this dataset was compiled from Iraq-Oncology Teaching Hospital/National Centre for Cancer Diseases. The number of these slices varies between 80 and 200 slabs, each of which represents an image of the human chest with varying sides and angles. Gender, age, education, residential area, and living status are all different in the 110 cases. Some are representatives of Iraqi transport and oil services, while others are farmers and gainers.

It contains 1190 CT scan slices from 110 cases of lung cancer at various stages. There are three types of cancer cases: benign, malignant, and cancer-free. It contains 120 benign tumour CT scan slices, 561 malignant tumour CT scan slices, and 416 cancer-free CT scan slices (see Figure 2).

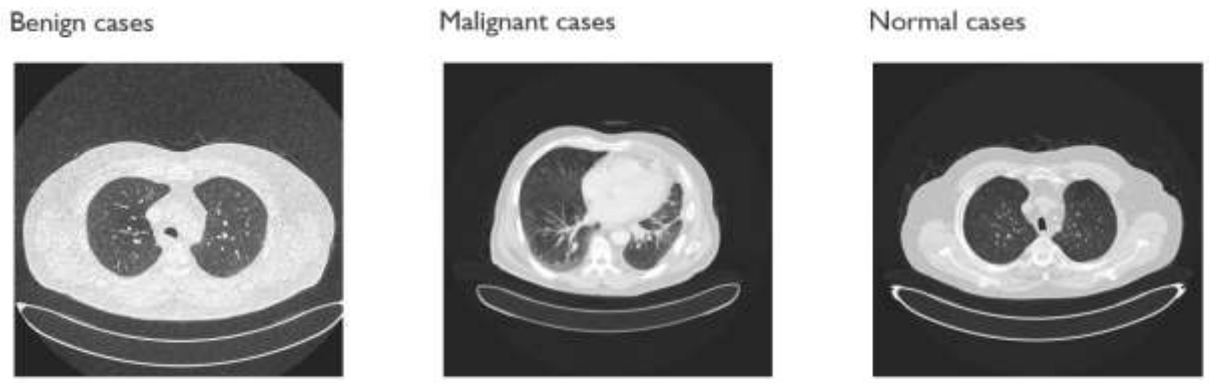


Figure 2: Different types of images in the dataset.

Splitting dataset

A 1,100-case imbalanced dataset was employed. Data was randomized and split into three sets for training, validation, and testing in a 70:15:15 ratio. A total of 70% of the data was used for training, 15% for validation, and 15% for testing

Data Augmentation

Deep learning models perform better when a large amount of data is used for training, and augmentation techniques can generate a variety of images, which can improve the efficiency of models in generalizing what they have learned to previously unseen images. Data augmentation is a method of creating new training data from an existing training set. It aids in expanding the size of a dataset without requiring the collection of new datasets. These images are treated as separate images by the network.

Data augmentation techniques include rotations, width shifts, height shifts, and zooming. The ImageDataGenerator package of Keras was used in this study to create images with rotations within a range of 40 to 40 degrees to the original images; width shift, height shift, shear, and zoom range within 20% of the original image; and horizontal flip.

Transfer Learning Models Architectures

Architectures such as MobileNet, VGG16, VGG19, DenseNet-201, and ResNet-101 are compared and studied in this proposed model. These architectures are stated to be among the best and most consistent architectures in the literatures and their inclusion in this experiment is due to their high performance in image processing.

MobileNet

To reduce model size and computation, MobileNet employs depthwise separable convolutions rather than standard convolutions. As a result, it can be used to construct lightweight deep neural networks for mobile and embedded vision applications.

Except for the first layer, mobilenet is built on depthwise separable convolutions. The first layer is a complete convolutional layer. After each layer, batch normalization and ReLU non-linearity are applied. The final layer, on the other hand, is a fully connected layer with no non-linearity and feeds to the softmax for classification. Depthwise convolution and the first fully convolutional layer in downsampling both use strided convolution. When depthwise and pointwise convolution are considered as separate layers, the total number of layers for mobilenet is 28 (see Figure 3)

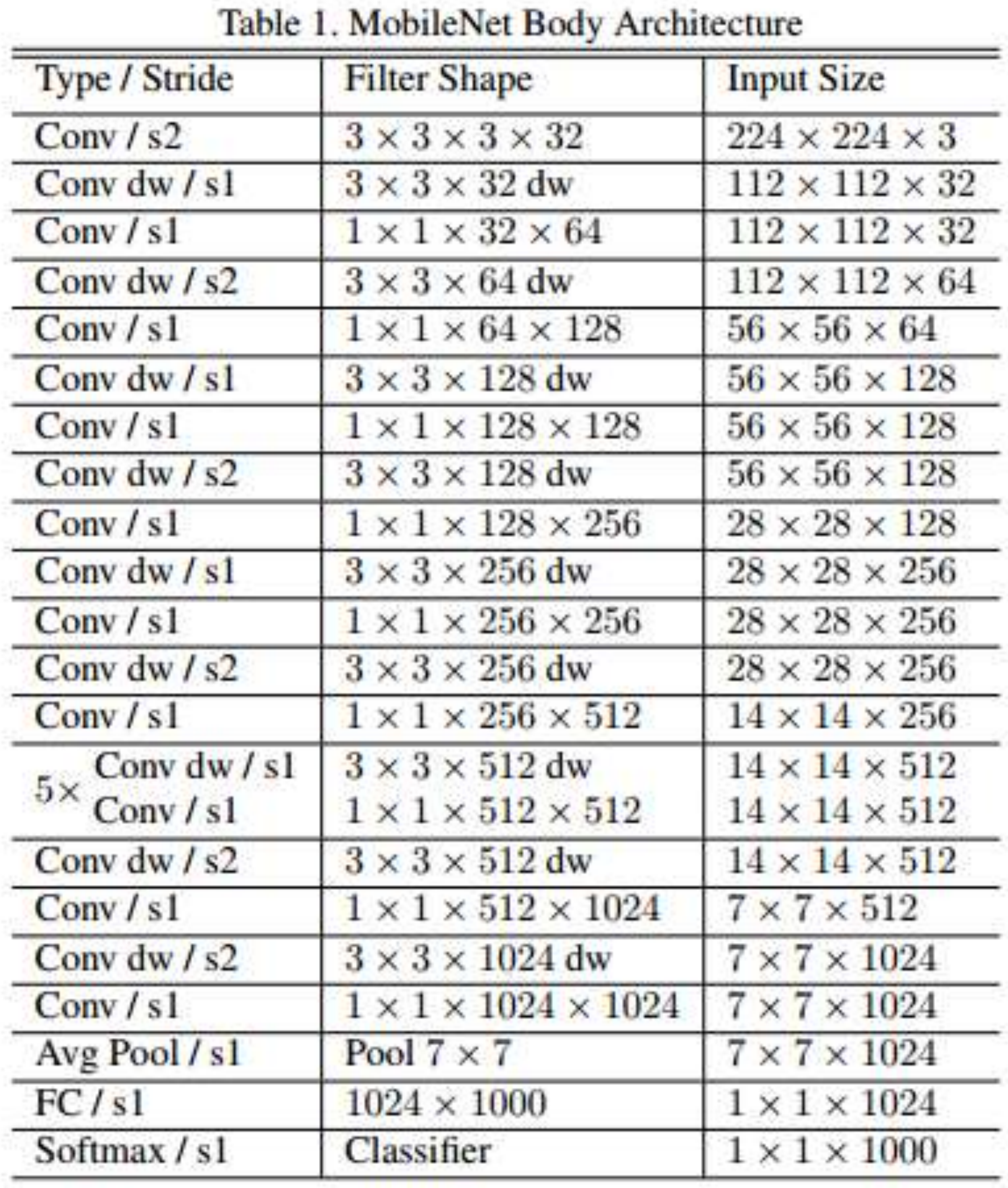


Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
5× Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 3: MobileNet architecture [10]

VGG16

VGG16 is a convolution neural net (CNN) architecture that won the 2014 ILSVR (ImageNet) competition. It is regarded as one of the best vision model architectures to date. The most distinguishing feature of VGG16 is that, rather than having a large number of hyper-parameters, they focused on having convolution layers of 3x3 filter with stride 1 and also use the identical padding and maxpool layer of 2x2 filter with stride 2. Throughout the architecture, this arrangement of convolution and max pool layers is consistent. Finally, it has two FC (fully connected layers) and a softmax for output. The 16 in VGG16 refers to the fact that it has 16 weighted layers.

The input of VGG is set to a 224x244 RGB image. The mean RGB value for all image data on the training set image is determined, and the image is then used as an input to the VGG convolution network. The convolution step is fixed and a 3x3 or 1x1 filter is used. There are three VGG fully connected layers, with the total number of convolutional layers + fully connected layers ranging from VGG11 to VGG19. VGG11

has a minimum of 8 convolutional layers and 3 fully connected layers. VGG19 has a maximum of 16 convolutional layers. +3 completely connected layers.
The overall structure consists of five sets of convolutional layers, followed by a MaxPool (See Figure 4).

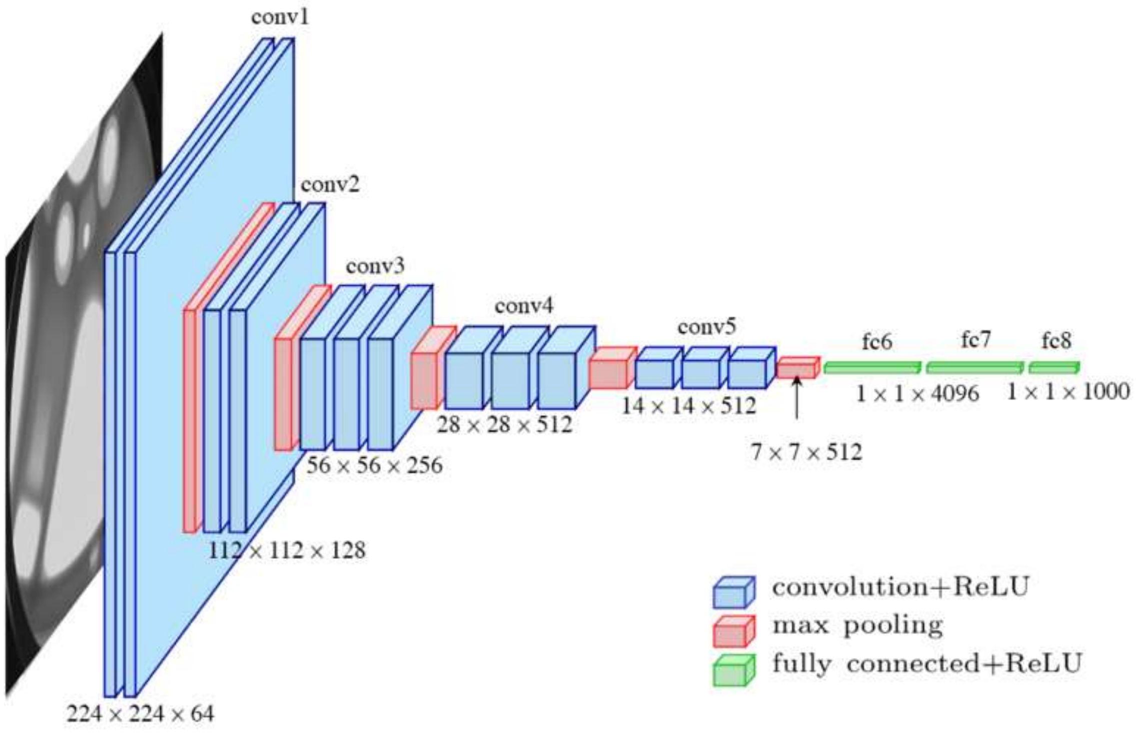


Figure 4: VGG16 architecture [11]

VGG19

VGG19 is a VGG model variant that consists of 19 layers, including 16 convolution layers, 3 fully connected layers, 5 MaxPool layers, and 1 SoftMax layer. Other VGG variants include VGG11, VGG16, and others. VGG19 has a total of 19.6 billion FLOPs.
This network was fed a fixed size (224 * 224) RGB image as input, implying that the matrix was of shape (224,224,3). The only pre-processing done was to subtract the mean RGB value from each pixel over the entire training set. They used kernels of (3 * 3) size with a stride size of 1 pixel to cover the entire image concept. To keep the image's spatial resolution, spatial padding was used. Stride 2 was used to perform max pooling over a 2 * 2 pixel window. This was followed by the Rectified linear unit (ReLu) to introduce non-linearity into the model to improve classification and computational time, as older versions used tanh or sigmoid functions, which managed to prove much preferable to any of those. Three fully connected layers were implemented, the first pair of which were 4096 in size, followed by a layer with 1000 channels for 1000-way ILSVRC classification, and the final layer is a softmax function. The architecture of VGG19 is as shown below in Figure 5.

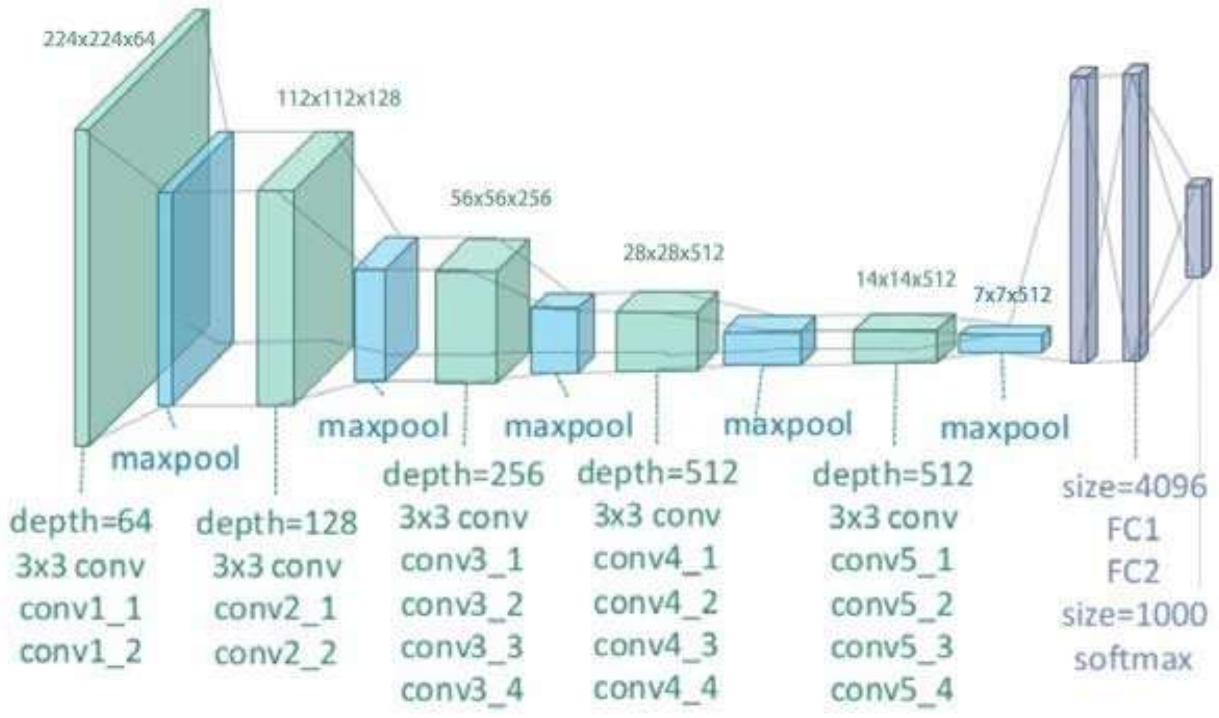
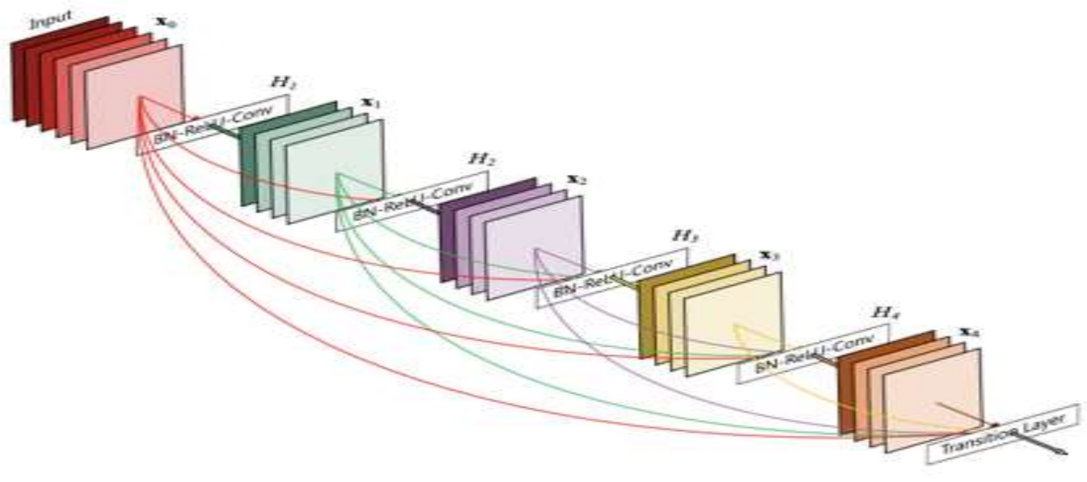


Figure 5: VGG19 architecture [12]

DenseNet-201

DenseNet-201 is a 201-layer deep convolutional neural network. A pre-trained model of a network trained on over a million of image data from the ImageNet database can be loaded. The pre-trained model can classify images into 1000 different object categories, including keyboards, mice, pencils, and a wide range of animals. As an outcome, the network has learned detailed feature representations for a diverse set of images. The network accepts images with a resolution of 224 by 224.

The illustration depicts a 5-layer dense block



DenseNet Structure

$$a^{[l]} = g([a^{[0]}, a^{[1]}, a^{[2]}, \dots \dots \dots \dots, a^{[l-1]}])$$

Figure 6: DenseNet 5-layer dense block [13]

Using the composite function operation, an output from the previous layer serves as an input to the second layer. The convolution layer, pooling layer, batch normalization, and non-linear activation layer are all part of this composite operation. The network has L(L+1)/2 direct connections as a result of these connections. The number of layers in the architecture is denoted by L. DenseNet comes in various versions, such as DenseNet-121, DenseNet-160, DenseNet-201, and so on. The numbers represent the number of neural network layers.

ResNet-101

ResNet, an abbreviation for Residual Network, is a type of neural network introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their paper "Deep Residual Learning for Image Recognition." ResNet-101 is a 101-layer deep convolutional neural network.

The ResNet network employs a VGG-19-inspired 34-layer basic network architecture, whereby the shortcut connection is added. As seen in the diagram below, these shortcut connections turn the design into the residual network (See Figure 7).

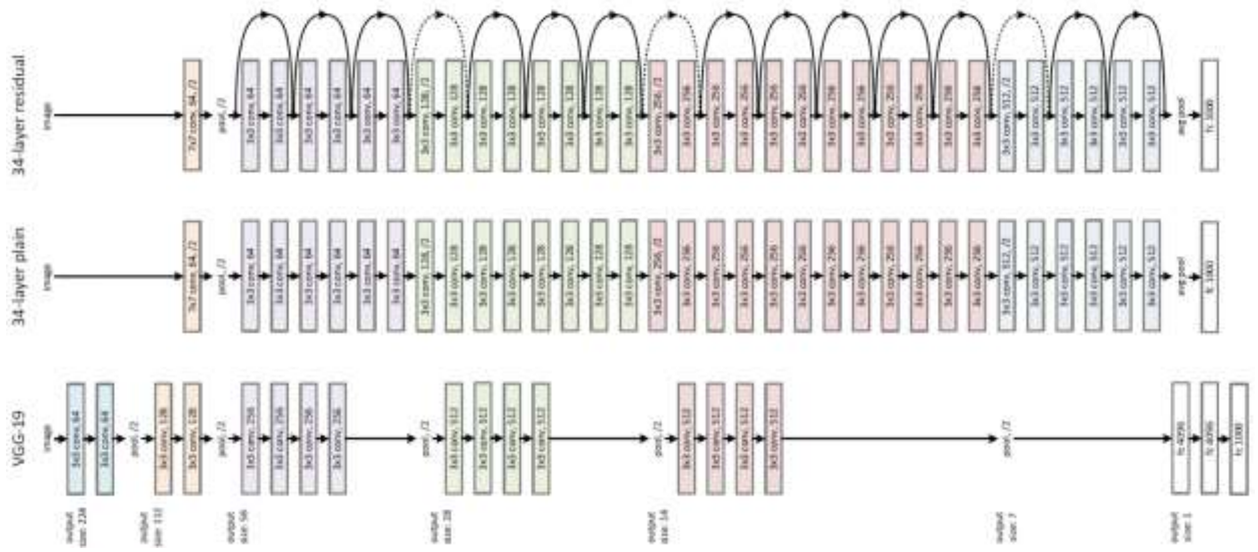


Figure 7: ResNet architecture [14]

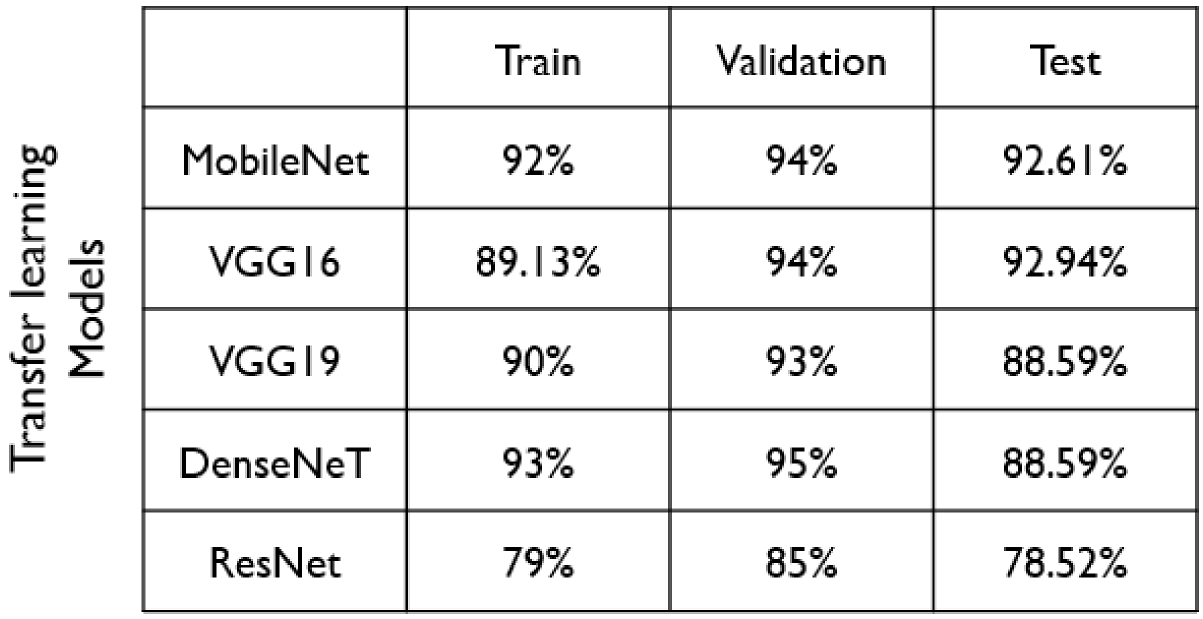
Training Process

The pre-trained models are employed as feature extraction models in this experiment, and the new classifier is combined with the pre-trained models. This newly designed classifier has a fully connected dense layer and an output layer that predicts three classes with softmax as the activation function.

After the model architecture is complete, it is constructed using categorical crossentropy as the loss function and RMSprop as the optimizer, with a learning rate of 1e-4. After that, the model is trained on the training set with a batch size of 10 and 200 epochs using fit_generator(). After a manual process of hyperparameter tuning, which began with 50 epochs until the model demonstrated maximum performance, the decision to use these hyperparameters was made. The model is then verified and tested using a batch size of 10 on the validation and test data, respectively.

The research is carried out in a Google Colab Notebook with runtime as GPU and using the Python programming language. The TensorFlow and sklearn libraries, which aid in model building, training, and evaluation, are used extensively throughout the code.

Figure 8 summarizes the accuracy of training, validation, and testing of all models.



Transfer learning Models

	Train	Validation	Test
MobileNet	92%	94%	92.61%
VGG16	89.13%	94%	92.94%
VGG19	90%	93%	88.59%
DenseNeT	93%	95%	88.59%
ResNet	79%	85%	78.52%

Figure 8: Accuracy of training, validation, and testing of all five models

The training and validation accuracy, as well as the training and validation loss, are determined for each epoch in the graph below. The graph depicts an increase in training accuracy over time, which is directly proportional to the decrease in loss over time (see Figures 9.1 - 9.5).

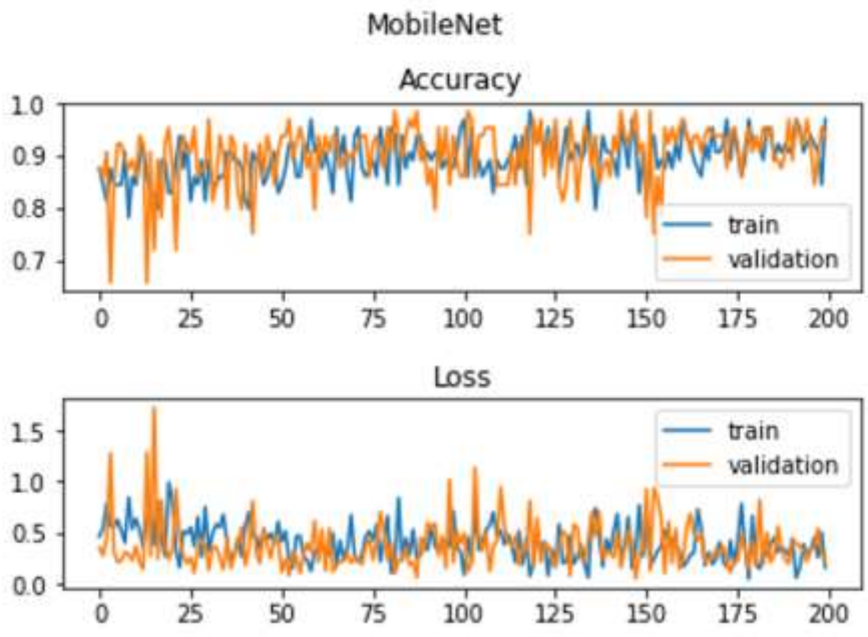


Figure 9.1: Accuracy and loss of training and validation set of MobileNet over 200 epochs

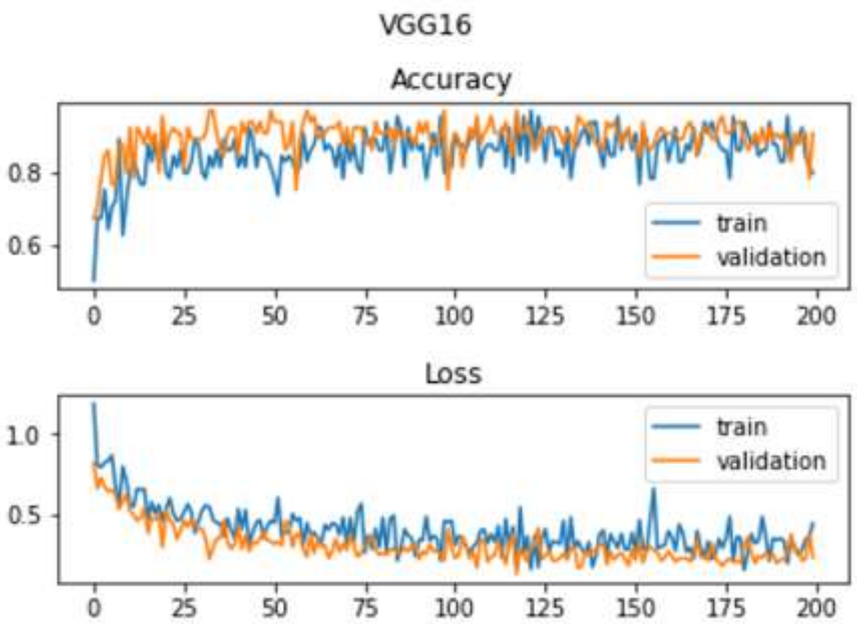


Figure 9.2: Accuracy and loss of training and validation set of VGG16 over 200 epochs

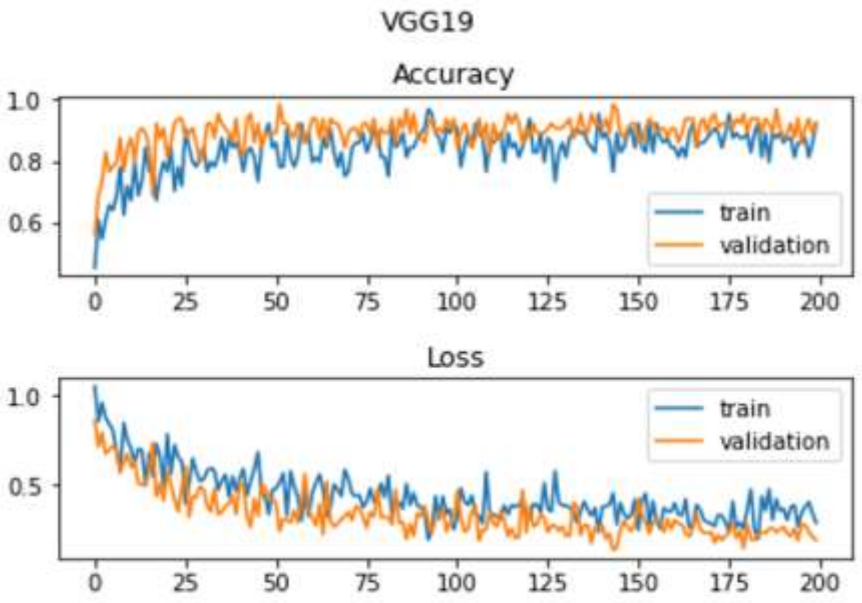


Figure 9.3: Accuracy and loss of training and validation set of VGG19 over 200 epochs

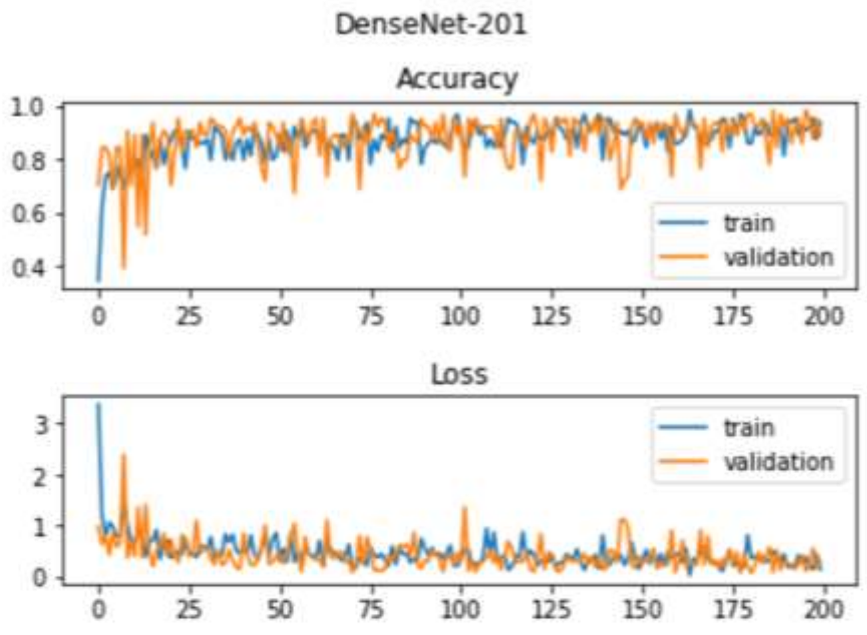


Figure 9.4: Accuracy and loss of training and validation set of DenseNet-201 over 200 epochs

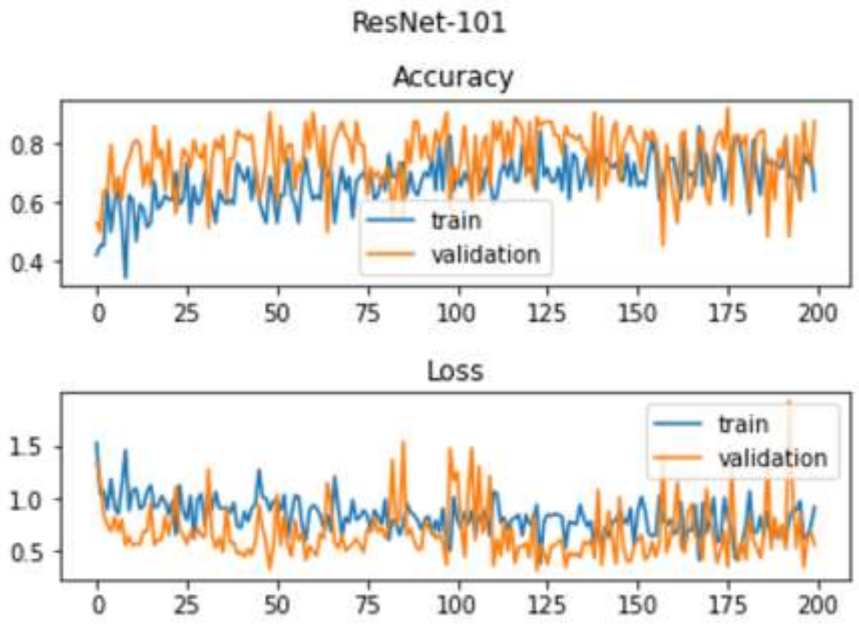


Figure 9.5: Accuracy and loss of training and validation set of ResNet-101 over 200 epochs

Results and Discussions

Confusion matrix-based performance metrics are used to evaluate the performance of all these models. These numbers correspond to the trained model's results on the testing set. Accuracy, precision, recall, and F1-score are the metrics evaluated here. Figures 10.1 – 10.5 show the confusion matrix used to evaluate the models, with True Positives (TP) indicating that the actual and predicted values are the same, and True Negatives (TP) indicating that the sum of values of all columns and rows except the values of the class for which we are calculating the values is the same, False-positives (FP) are the sum of the values in the corresponding column that are not True positive, and False-negatives (FN) are the sum of the values in the corresponding rows that are not True positive.

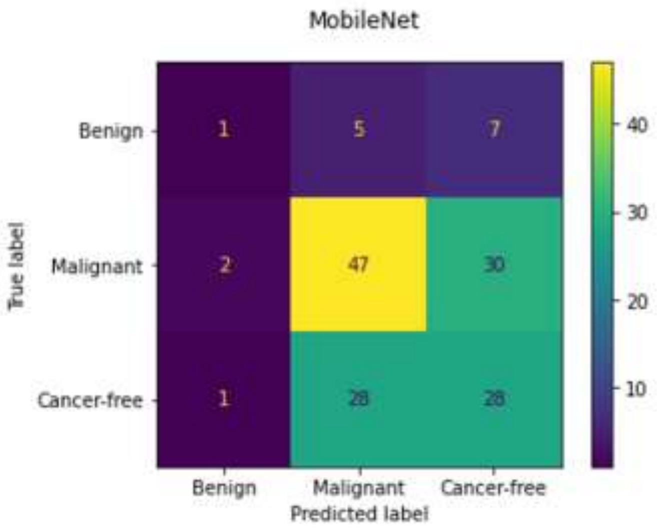


Figure 10.1: Confusion matrix of MobileNet

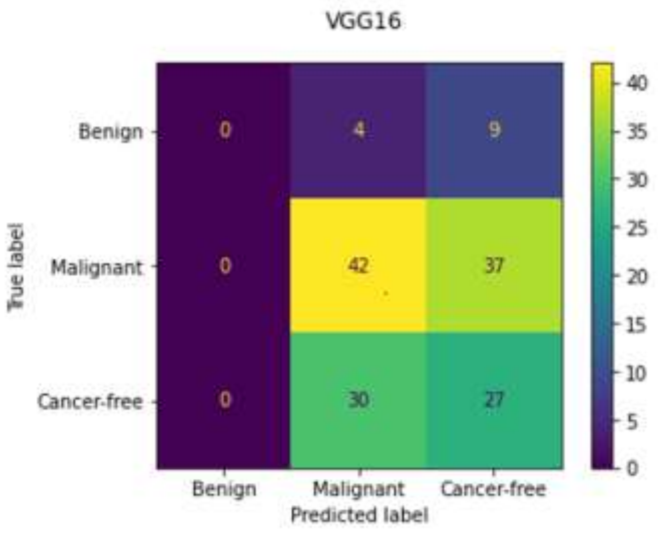


Figure 10.2: Confusion matrix of VGG16

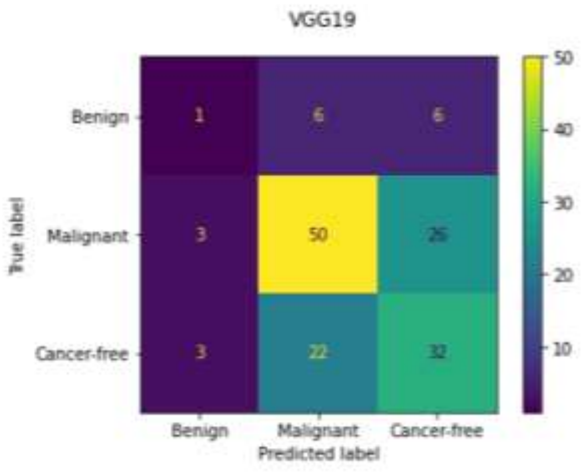


Figure 10.3: Confusion matrix of VGG19

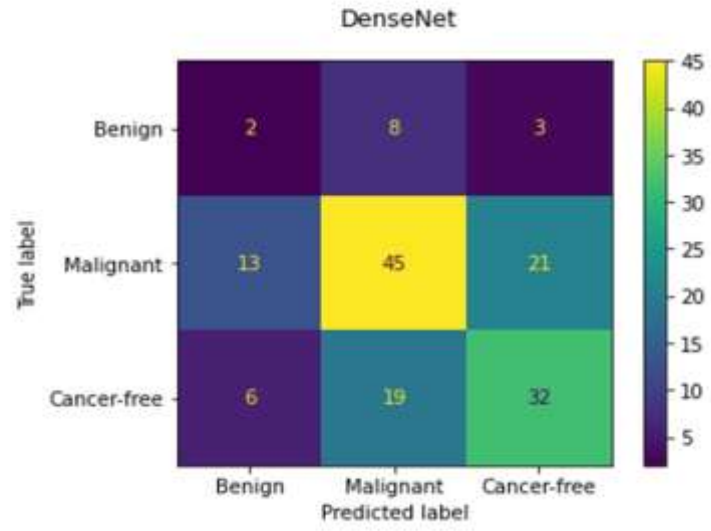


Figure 10.4: Confusion matrix of DenseNet-201

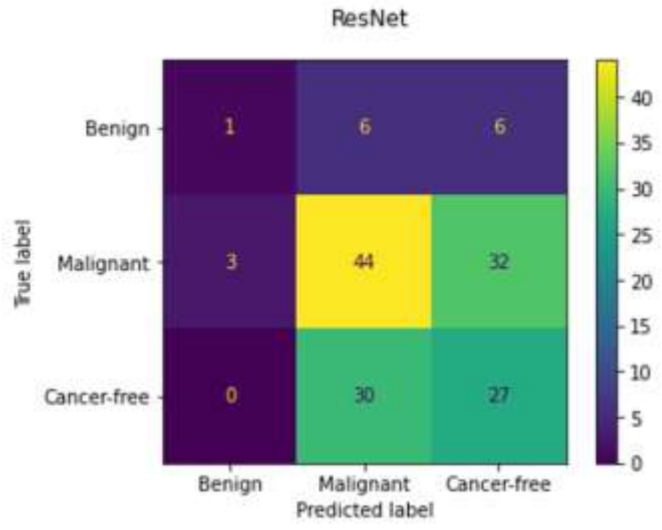
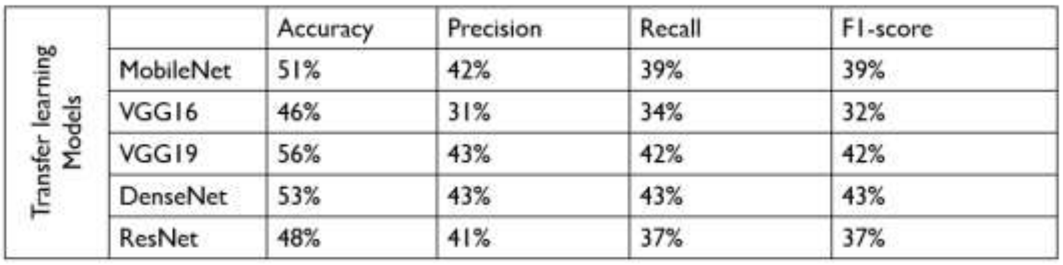


Figure 10.5: Confusion matrix of ResNet-101

The classification reports for each of the five architectures are shown in the figure 11.



		Accuracy	Precision	Recall	F1-score
Transfer learning Models	MobileNet	51%	42%	39%	39%
	VGG16	46%	31%	34%	32%
	VGG19	56%	43%	42%	42%
	DenseNet	53%	43%	43%	43%
	ResNet	48%	41%	37%	37%

Figure 11: Classification report of Five models and their metrics

F1-score is the measure that is evaluated for this proposed system based on the aforementioned reports because the system deals with uneven distribution of number of photos in dataset categories. As a result, DenseNet-201 has the greatest F1-score and hence can be chosen as the architecture for implementation of this system.

Conclusion and Future Work

A comparison of several transfer learning architectures for detecting the type of lung cancer existing in a person was conducted in this paper. DenseNet-201 was chosen as the architecture for the system's implementation with the accuracy of 53%, recall of 43%, precision of 43% and f1-score of 43%.
The created model only processes a single CT scan slide. The lungs, on the other hand, should be thoroughly checked in order to detect lung cancer more effectively. Future research on this subject should entail examining the lungs from all angles.
For the same aim, a 3D CNN can be employed.

References

1. https://www.who.int/news-room/fact-sheets/detail/cancer
2. https://www.cancer.org/cancer/lung-cancer/causes-risks-prevention/risk-factors.html
3. Abdelbaki Souid, Nizar Sakli, Hedi Sakli. 2021, 'Classification and Predictions of Lung Diseases from Chest X-rays Using MobileNet V2', MDPI
   View At: Publisher Site
4. Matko Šarić, Mladen Russo, Maja Stella, Marjan Sikora. 2019, 'CNN-based Method for Lung Cancer Detection in Whole Slide Histopathology Images', 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)
   View At: Google Scholar
5. Ruchita Tekade, K. Rajeswari. 2018, 'Lung Cancer Detection and Classification Using Deep Learning', 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)
   View At: Publisher Site
6. W. Ausawalaithong, A. Thirach, S. Marukatat, and T. Wilaiprasitpor. 2018, 'Automatic Lung Cancer Prediction from Chest X-ray Images Using the Deep Learning Approach,' 2018 11th Biomedical Engineering International Conference (BMEiCON), pp. 1-5
   View At: Publisher Site
7. S. Sasikala, M. Bharathi, B. R. Sowmiya. 2018, 'Lung Cancer Detection and Classification Using Deep CNN', International Journal of Innovative Technology and Exploring Engineering (IJITEE)
   View At: Publisher Site
8. Raul Victor Medeiros da Nóbrega, Solon Alves Peixoto, Suane Pires P. da Silva, Pedro Pedrosa Rebouças Filho. 2018, 'Lung Nodule Classification via Deep Transfer Learning in CT Lung Images', 2018 IEEE 31st International Symposium on Computer-Based Medical Systems
   View At: Google Scholar
9. https://www.kaggle.com/hamdallak/the-iqothnccd-lung-cancer-dataset
10. Ferhat Culfaz. 2018, 'Transfer Learning using Mobilenet and Keras', < https://towardsdatascience.com/transfer-learning-using-mobilenet-and-keras-c75daf7ff299>
11. Max Ferguson. 2017, 'The standard VGG-16 network architecture', <https://www.researchgate.net/figure/Fig-A1-The-standard-VGG-16-network-architecture-as-proposed-in-32-Note-that-only_fig3_322512435>
12. Clifford K. Yang. 2018, 'Illustration of the network architecture of VGG-19 model' https://www.researchgate.net/figure/llustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means_fig2_325137356
13. Gaurav Singhal. 2020, 'Introduction to DenseNet with TensorFlow', <https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow >
14. Sik-Ho Tsang. 2018, 'Review: ResNet — Winner of ILSVRC 2015 (Image Classification, Localization, Detection)', <https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8>