JOURNAL OF NORTHEASTERN UNIVERSITY

Volume 25 Issue 04, 2022 ISSN: 1005-3026 https://dbdxxb.cn/ Original Research Paper

ANALYSIS OF LUNG DISEASE CLASSIFICATION USING TRANSFER LEARNING APPROACH

Devavarapu Sreenivasarao
Research Scholor, Annamalai University and Assistant Professor, Department of CSE, Sreenidhi Institute of Science & Technology, Hyderabad, Email: sreenivasaraod@sreenidhi.edu.in

Dr. S. Anu H Nair
Assistant Professor, Department of CSE, Annamalai University, Chidambaram (Deputed to WPT Chennai). Email: anu_jul@yahoo.co.in

Dr. T Venkat Narayana Rao
Professor & HOD of CSE(IoT), Sreenidhi Institute of Science and Technology, Hyderabad, Email: venkatnarayanaraot@sreenidhi.edu.in

Dr. K P Sanal Kumar
Assistant Professor, Department of Computer Science, R.V Government Arts College, Chengalpattu, Email:sanalprabha@yahoo.co.in

Abstract
There is an increasing risk on health due to changing environment, climate and lifestyle. India tops the world in deaths due to lung diseases. They were the second highest cause of deaths in India after heart disease in 2017, killing nearly 1 million Indians that year. Early diagnosis and treatment of lung diseases is critical to prevent complications including death. Chest X-ray is currently the best available method for diagnosis, playing a crucial role in clinical care. Using Deep Learning to predict lung diseases from Chest X-rays can be a lifesaving factor for an individual suffering from the disease. This is possible as the results can be predicted with a high percentage of accuracy instantly. This paper presents an effective way for expert diagnosis of lung diseases using Deep Learning. It focuses on creating a system for assistance of Radiologists in detection of lung diseases. This will especially benefit rural areas where radiologists aren't easily available. Our system connects radiology labs with the radiologists who can diagnose faster and better with our model.

INTRODUCTION
Lung diseases are some of the most common cases found throughout the world. Some lung diseases can cause by smoking, infections, and genes which can be inherited by parents which cause most lung diseases. Lungs are the most complex system which expands and relaxes thousands of times each day to bring oxygen and send carbon dioxide out. The chest X-rays are said to be the most commonly accessible to diagnose and classify many lung diseases. These lung disease detection and classification needs to be done accurately because wrong

Submitted: 15/09/2022 **Accepted**: 17/11/2022

decision and treatment can cost the life of the patient. Integrating with advance technology with medical instructions can achieve improvement of the medical diagnosis and reduces the time in detection of disease. The main goal of this project is to implement deep-learning techniques to the chest x-ray images to analyze the performance of classification of type of lung disease and its efficiency.

The lung diseases can be miss-diagnosed with proper information and can be biased based on the prejudice done by the experience of the doctor. This prejudice might be beneficial for simple diseases but for complex cases it might mislead. The unbiased recognition with the help of technology is required in order to reduce difficulties for proper treatment for the patient.

To analyze and understand the chest x-ray images to classify by applying various classification techniques on the dataset.

Using CNN to train the images

LITERATURE SURVEY

Pneumonia are one of the most common infections that attack the human respiratory system. The goal of the literature review was to identify the most common factors and methodologies utilized in research about the virus's spread. People with chronic conditions including diabetes, hypertension, diabetes, stroke, heart failure, or kidney failure, as well as the elderly with weakened immune systems, are more susceptible to infection. Infection risk may be increased in enclosed spaces with poor ventilation and airflow. As with other respiratory viruses such as influenza and rhinoviruses, the virus is thought to spread through respiratory droplets from coughing and sneezing. Aerosol transmission can also occur when people are exposed to high aerosol concentrations for an extended period of time in enclosed areas.

The goal of this project is to train the dataset images with convolutional neural layers can help or enrich the ease of classifying the diseased images to its respective classes. In this project we have implemented deep learning models for training and followed by classification of those images. Recently detection of lung diseases like pneumonia was done by computer aided design (CAD) in order to classify using SVM classifier, K-Means algorithm, Convolutional Neural Networks. The problem with existing system is that the computational power is required more and the build model might not be able to predict properly. The techniques present inbuilt classification techniques based on Convolutional Neural Network were all perfectly tuned for coloured images where these techniques can be used for image classification and object detection but not on grey scale images like X-Ray’s where these images are prominent in medical field.

Related Work

Prediction and Classification of Lung Cancer Using Machine Learning Techniques (2020): Pragya Chaturvedi, Anuj Jhamb, Meet Vanani and Varsha Nemade, Algorithms used are feature extraction method like GLCM and used FCM for classification with 91.6% accuracy.

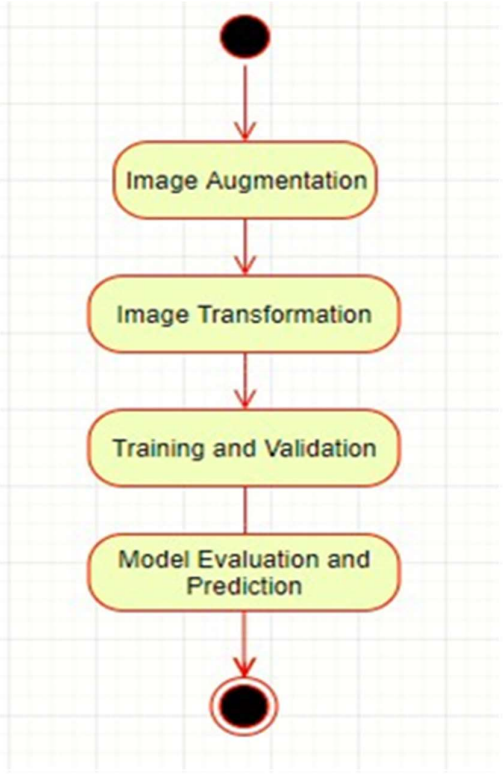
Classification of Lungs Diseases Using Machine Learning Technique (2021): Meet Diwan , Bhargav Patel, Jaykumar Shah. Algorithms used are Convolutional Neural Network. The accuracy of this model is 76%.

Detection and classification of lung diseases for pneumonia and Covid-19 using machine and deep learning techniques (2021): Shimpy Goyal and Rajiv Singh. Algorithms used are KNN (k-Nearest Neighbours) with accuracy of 86% in raw features and 89% in min-max normalization

Proposed System

The proposed system will use Convolution Neural Network that automatically learning decisive and discriminate features. But CNN requires large amount of data to train the network. Our dataset has not enough images to get higher accuracy. So we are implementing Transfer-Learning Approach which does not need a large amount of data to train the network. The benefits of transfer learning are decrease in time and computational cost of building a Deep Learning Model. Here we are going to determine the better deep learning model by using different deep learning models to classify lung disease.

Image Data augmentation techniques are used to increase the training dataset artificially by modifying the versions of images in the collected data. Here we took 700 images of real time field visited pepper plant leaves and then augmented them to 14,000 images. Using 14,000 images consisting of diseased and healthy plant leaves collected under controlled conditions.



Any image has a chance of having a noise. Even in the images that we have collected, there might be noise for some or the other reason. There are different types of noises like salt and pepper noise, gaussian noise, short tail noise, etc. It creates a disturbancefor the image and may

not get the accurate results. There are many image pre-processing techniques that help in removing the noise and give good performance by reducing noise. In total we have used 15 image pre-processing filters. Among those filters we select the best filter based on image quality metrics. These image processing filters help in restoring better features and reduce unwanted data/ noise in the image.

Convolutions of Image data is multiplying two pixel arrays, generally of different sizes but of the same dimensionality to produce a third array of the same dimensionality. Here the output pixels of image are simple linear combinations of certain input pixel value

The control flow of the system is illustrated by this activity diagram. First step is reading the images from the dataset. The dataset is the real time field visited lung x-rays and public dataset from Kaggle. Then applying various transformation techniques and computing various image quality metrics. After that applying model feature extraction and then applying various ML classification algorithms.

IMPLEMENTAION AND RESULTS

5.1 Language / Technology Used

Python language is used to write the code. Python provides a wide variety of libraries for scientific and computational usage. Libraries such as opencv, os, numpy, skimage, math, scipy, pandas etc are used for image processing techniques such as filtering and segmentation. The main functionalities in the project are

- Data Augmentation: increase the dataset artificially by modifying the orientation (angle, degree, rotate etc) of images in the collected data.
- Image Filtering: Applying various filtering techniques on the data
- Image Convolutions: Applying convolutions layers to the image dataset.
- Optimizers: Applying optimizers on the convolutions to reduce loss and improve accuracy
- Classification: Applying DL models on images to classify them.
- The libraries or packages used for the above functionalities are:

Data Augmentation

- from keras preprocessing.image we import Image Data Generator
- opencv library is used to capture the images
- os libraries are used to read the labels, rename the images and store them
- PIL (python image library) is used to manipulate the images

Image Filtering

- opencv library is used for image processing
- os library are used to retrieve images and perform operations on them
- matplotlib and seaborn are data visualization libraries.
- sklearn is a library which is used for machine learning and statistical modelling of data

Image Convolutions

- numpy is a library used for numerical calculations for image pixel data manipulation

- keras.layers is used for applying convolutions on images.

Optimizers

- keras includes different types of optimizers for the models.
- It has functions compile() and fit() for model to train dataset.

Classification

- keras.applications contain all deep learning models.
- Each model has its own input image size.

5.2 Methods / Algorithms used

The language used in this project is python and various filtering and segmentation techniques.

Optimizer Techniques

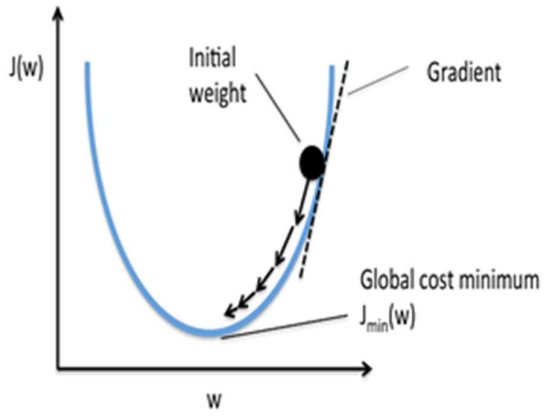
SGD

Stochastic gradient descent (often abbreviated SGD) is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or sub differentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the computational burden, achieving faster iterations in trade for a lower convergence rate.

- for one or more epochs:
  - for each weight j
    - $w_j := w + \Delta w_j$, where: $\Delta w_j = \eta \sum_i (\text{target}^{(i)} - \text{output}^{(i)}) x_j^{(i)}$

Instead, we update the weights after each training sample:

- for one or more epochs, or until approx. cost minimum is reached:
  - for training sample i:
    - for each weight j
      - $w_j := w + \Delta w_j$, where: $\Delta w_j = \eta (\text{target}^{(i)} - \text{output}^{(i)}) x_j^{(i)}$



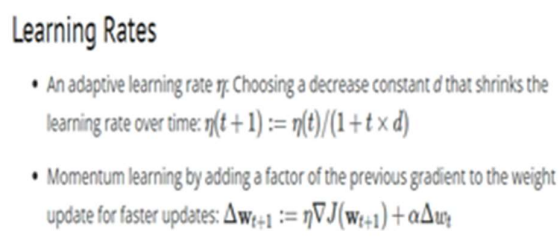


Fig 5.2.1 Stochastic Gradient Descent Graph

5.2.1.2 RMSprop

Root Mean Squared Propagation, or RMSProp, is an extension of gradient descent and the AdaGrad version of gradient descent that uses a decaying average of partial gradients in the adaptation of the step size for each parameter. The use of a decaying moving average allowsthe algorithm to forget early gradients and focus on the most recently observed partial gradients seen during the progress of the search, overcoming the limitation of AdaGrad.

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)\left(\frac{\delta C}{\delta w}\right)^2$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \frac{\delta C}{\delta w}$$

5.2.1.3 Adam

It is a replacement algorithm for training deep learning models which is in very optimised manner. To handle sparse gradients on noisy problems we will implement both AdaGrad and RMSProp to optimise the algorithm. Configures the default configuration parameters well on most problems.

5.2.1.4 Adadelta

Adadelta is an extension of Adagrad and instead of accumulating all past gradients., it can inhabits learning rates based on gradient updates, So that Adagrad has very popular even there are many updates.

5.2.1.5 Adagrad

Adagrad is an optimizer with parameter-specific learning rates, which are inhabits to how frequently a parameter gets updated during training. If there are smaller updates then there will be more updates to be received by a parameter.

$$\text{SGD} \Rightarrow w_t = w_{t-1} - \eta \frac{\partial L}{\partial w_{t-1}}$$

$$\text{Adagrad} \Rightarrow w_t = w_{t-1} - \eta_t^{'} \frac{\partial L}{\partial w_{t-1}}$$

$$\text{where } \eta_t^{'} = \frac{\eta}{\sqrt{\alpha_t + \varepsilon}}$$

ε is a small +ve number to avoid divisibilty by 0

$$\alpha_t = \sum_{i=1}^{t} (\frac{\partial L}{\partial w_{t-1}})^2 \quad \textit{summation of gradient square}$$

5.2.1.6 Adamax

Based on the infinity norm, a variant and an optimiser called Adamax implement the Adamax algorithm. Default parameters follow those provided in the paper. In models with embeddings adam is superior to Adamax.

5.2.1.7 Nadam

Nadam Optimizer is an extension of Adaptive Movement Estimation (Adam) to add NAG which is an improve type of momentum

5.2.2 Deep Learning Models

5.2.2.1 Convolutional Neural Networks (CNN)

To extract higher representations for the image content, we will use a neural network model called CNN. It is contrast to the classical image recognition In CNN, it takes the image's raw pixel data and the train the model. Finally it extracts the characteristics to classify.

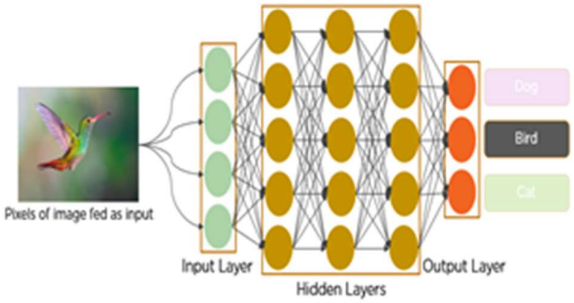


Fig 5.2.2 Convolutions Neural Network

5.2.2.2 Transfer learning Approach

Transfer Learning is a machine learning method Initially a new task can be pretrained by a model which is already used . By implementing transfer learning to a new task, performance should be higher by training only a small amount of data. Inception, Mobile Net, Dense Net are typical examples models of Transfer Learning.

In python, Keras Applications are Transfer learning models with pre-trained weights. Prediction, feature extraction and fine-tuning can be used from these models. We are going to use MobileNet, Inception-ResNet V2, DenseNet, EfficientNet-B0 and comparing among them.

Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3
DenseNet201	80	77.3%	93.6%	20.2M	402	127.2	6.7
NASNetMobile	23	74.4%	91.9%	5.3M	389	27.0	6.7
NASNetLarge	343	82.5%	96.0%	88.9M	533	344.5	20.0
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9
EfficientNetB1	31	79.1%	94.4%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.1%	94.9%	9.2M	186	80.8	6.5
EfficientNetB3	48	81.6%	95.7%	12.3M	210	140.0	8.8
EfficientNetB4	75	82.9%	96.4%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.6%	96.7%	30.6M	312	579.2	25.3
EfficientNetB6	166	84.0%	96.8%	43.3M	360	958.1	40.4
EfficientNetB7	256	84.3%	97.0%	66.7M	438	1578.9	61.6

Table 5.2.1 Keras Models

5.2.2.3 MobileNet

MobileNet are the popular architectures used for image classification, face detection, segmentation and many more. They are known for their latency on mobile and embedded devices. You may infer this from the name "MobileNet". They have far less number of trainable parameters as they use Separable Convolutions.

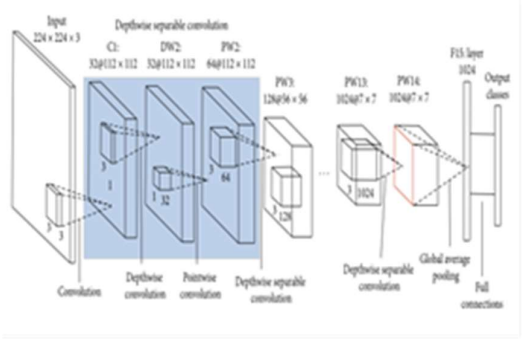


Fig 5.2.3 MobileNet

5.2.2.4 InceptionResNetV2

From the ImageNet database we will train various images based on one of the convolutional neural network called Inception-ResNet-v2 which is 164 layers deep and the images can be categorised in various categories, such as the keyboard, mouse, pencil, and many things. Finally the network has enriched with various feature representations for no. of images. The size of an input can be 299-by-299, for a network and a list can be obtained as an output containing estimated class probabilities.

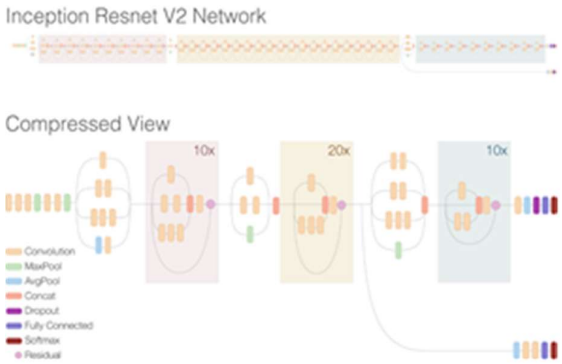


Fig 5.2.4 InceptionResNetV2

5.2.2.5 Inception v3

One of the image recognition model Inception v3 which has attained greater than 78.1% accuracy on the ImageNet dataset. The model is developed by grouping so many ideas of various researchers from past years. "Rethinking the Inception Architecture for Computer Vision" by Szegedy, et. al.has been taken as the base paper for this.The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using Softmax.

5.2.2.6 DenseNet

A DenseNet is a type of convolutional neural network that utilizes dense connections between layers, through Dense Blocks, where we connect all layers (with matching feature-map sizes) directly with each other. To preserve the feed-forward nature, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers.

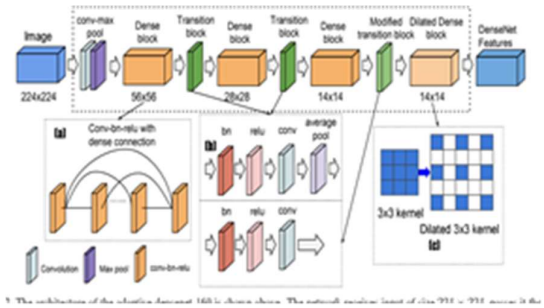


Fig 5.2.5 DenseNet

5.2.2.7.Xception

Xception is one of the convolutional neural network having 71 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database we will load a version which is pretrained into the network and it can be implemented on various images. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich

feature representations for a wide range of images. The network has an image input size of 299-by-299.

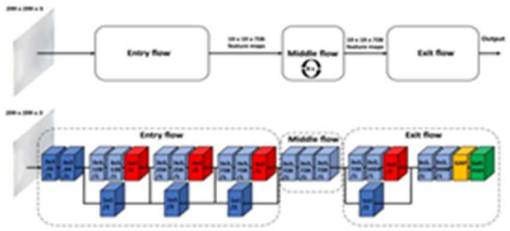


Fig 5.2.6 Xception

5.2.2.8.VGG-19

VGG-19 is one of the convolutional neural network which is having 16 layers deep. You can load a We can easily load a version which is used to train the network by implementing various images from the ImageNet database [1]. The pretrained network can classify images into various object categories, such as keyboard, mouse, pencil, and many other. Finally, the network has enriched with various feature representations for any image. The size of an input for an image for a network.

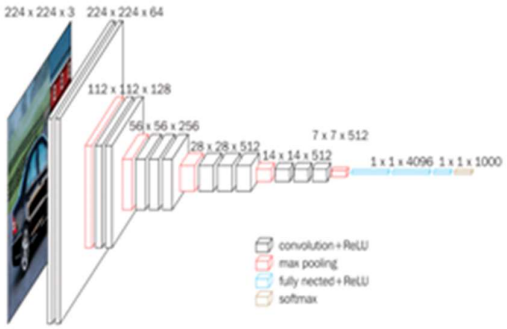


Fig 5.2.7 VGG-16

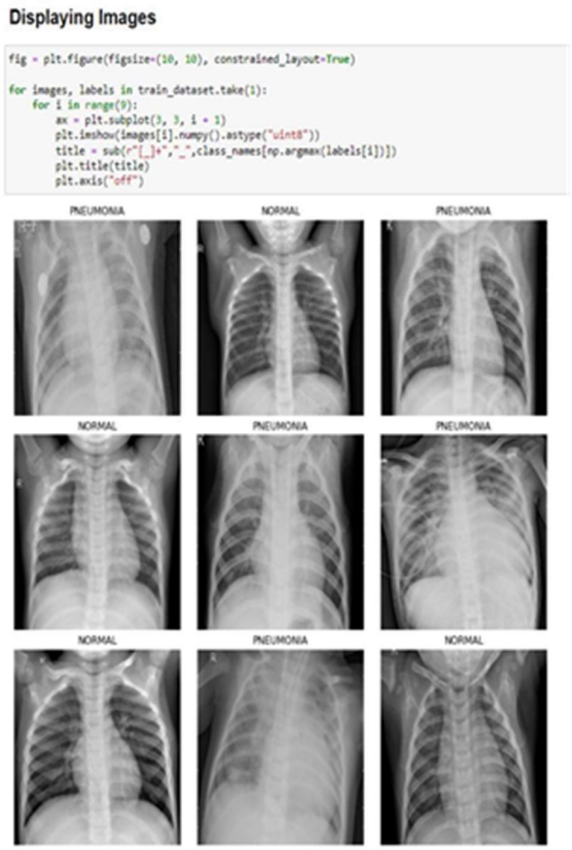
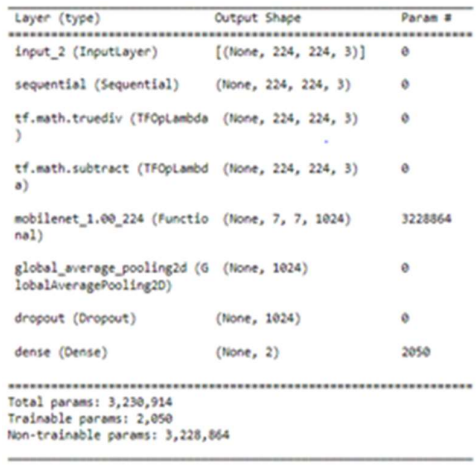


Fig 5.3.4 Images of dataset

5.4 Convolutions and Parameters of each Model

Basic vanilla CNN where every convolutions and parameters are included, In transfer learning here the methods are already trained by imagenet weights so the parameters and convolutions are far less in compated to CNN

5.4.1 MobileNet



Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
sequential (Sequential)	(None, 224, 224, 3)	0
tf.math.truediv (TFOpLambda)	(None, 224, 224, 3)	0
tf.math.subtract (TFOpLambda)	(None, 224, 224, 3)	0
mobilenet_1.00_224 (Functional)	(None, 7, 7, 1024)	3228864
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0
dropout (Dropout)	(None, 1024)	0
dense (Dense)	(None, 2)	2050

Total params: 3,230,914
Trainable params: 2,050
Non-trainable params: 3,228,864

Fig 5.4.1.1 mobilenet summary

5.4.2 Inception-ResNet V2

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
sequential (Sequential)	(None, 224, 224, 3)	0
tf.math.truediv (TFOpLambda)	(None, 224, 224, 3)	0
tf.math.subtract (TFOpLambda)	(None, 224, 224, 3)	0
inception_resnet_v2 (Functional)	(None, 5, 5, 1536)	54336736
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1536)	0
dropout (Dropout)	(None, 1536)	0
dense (Dense)	(None, 2)	3074

Total params: 54,339,810
Trainable params: 3,074
Non-trainable params: 54,336,736

Fi5.4.2.1 Inception_resnet_v2 summary

5.4.3 Inception V3:

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
sequential (Sequential)	(None, 224, 224, 3)	0
tf.math.truediv (TFOpLambda)	(None, 224, 224, 3)	0
tf.math.subtract (TFOpLambda)	(None, 224, 224, 3)	0
inception_v3 (Functional)	(None, 5, 5, 2048)	21802784
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dropout (Dropout)	(None, 2048)	0
dense (Dense)	(None, 2)	4098

Total params: 21,806,882
Trainable params: 4,098
Non-trainable params: 21,802,784

Fig 5.4.3.1 Inception_v3

5.4.4 Densenet:

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
sequential (Sequential)	(None, 224, 224, 3)	0
tf.math.truediv (TFOpLambda)	(None, 224, 224, 3)	0
tf.nn.bias_add (TFOpLambda)	(None, 224, 224, 3)	0
tf.math.truediv_1 (TFOpLambda)	(None, 224, 224, 3)	0
densenet121 (Functional)	(None, 7, 7, 1024)	7037504
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0
dropout (Dropout)	(None, 1024)	0
dense (Dense)	(None, 2)	2050

Total params: 7,039,554
Trainable params: 2,050
Non-trainable params: 7,037,504

Fig 5.4.4.1 Densenet

5.4.5 VGG19:

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
sequential (Sequential)	(None, 224, 224, 3)	0
tf.__operators__.getitem (SlicingOpLambda)	(None, 224, 224, 3)	0
tf.nn.bias_add (TFOpLambda)	(None, 224, 224, 3)	0
vgg19 (Functional)	(None, 7, 7, 512)	20024384
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense (Dense)	(None, 2)	1026

Total params: 20,025,410
Trainable params: 1,026
Non-trainable params: 20,024,384

Fig 5.4.5.1 vgg19

5.4.6 Xception

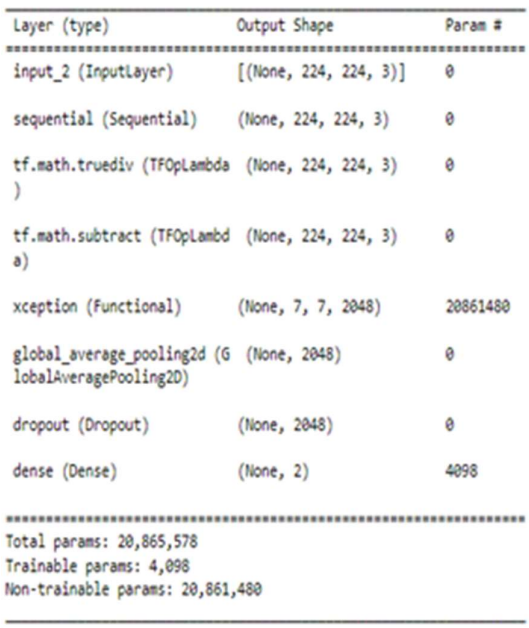


Fig 5.4.6.1 Xception

5.5 Results

We will compare the models as per the accuracy and the loss based on the testing. The performance metrics are considered as below

- Performance accuracy can be defined as the total number of correctly classified images to the total number of images
- Loss function can be defined as how well the data can be developed by the model
- Precision can be defined as the ratio of the number of correctly predicted observations to the total number of positive predictions
- Recall can be defined as the ratio of correctly predicted observations to all observations in that class
- F1 score can be defined as the harmonic mean between precision and recall
- The time required per epoch for training a particular DL model.

Table 5.5.1 Models

Model	Epoch	Time	Accuracy	Loss	Precision	Recall
MobileNet	**50**	**15m 1sec**	**0.962**	**0.104**	**0.962**	**0.962**
InceptionV2	**50**	**50m 52sec**	**0.9653**	**0.09653**	**0.9653**	**0.9653**
InceptionV3	**50**	**21m 26sec**	**0.9149**	**0.2.23**	**0.9149**	**0.9149**
DenseNet	**50**	**38m 2sec**	**0.95**	**0.14**	**0.94**	**0.94**
Xception	**50**	**37m 23sec**	**0.951**	**0.145**	**0.951**	**0.951**
Vgg	**50**	**59m26 sec**	**0.944**	**0.122**	**0.944**	**0.944**

Determining the time taken for each models

The time taken by each model depends the algorithm and convolutional layers of that model. More the convolutional layers more will be computing cost and more time will be taken.

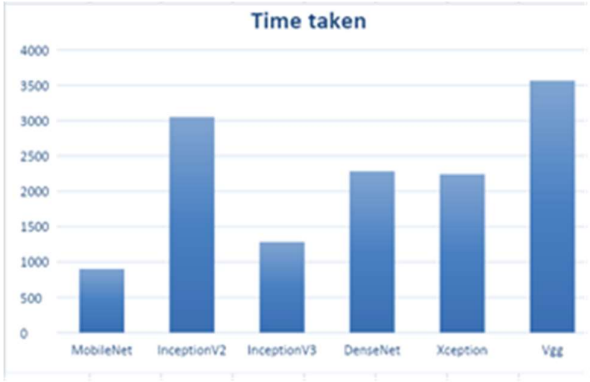


Fig 5.5.1 Time Performance

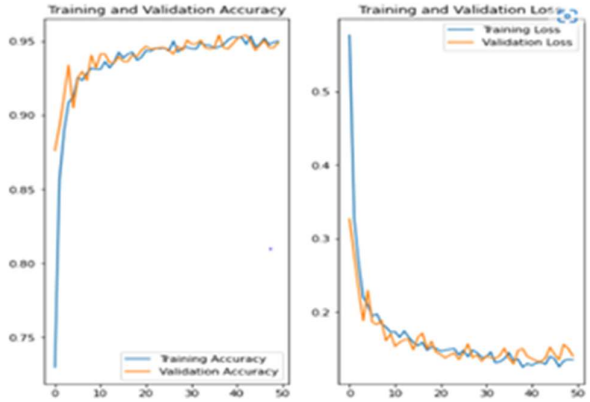


Fig 5.6.1 Mobile Net

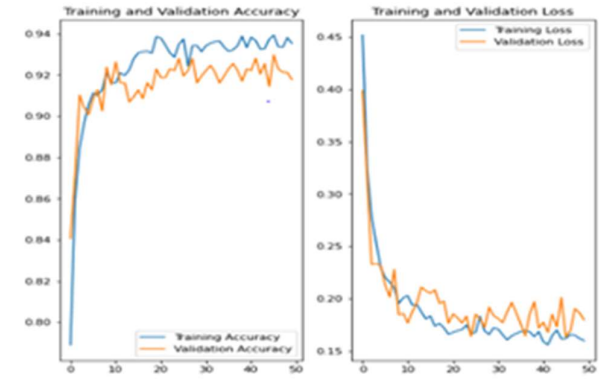


Fig 5.6.2 DenseNet

Fig 5.6.3 Inception V2

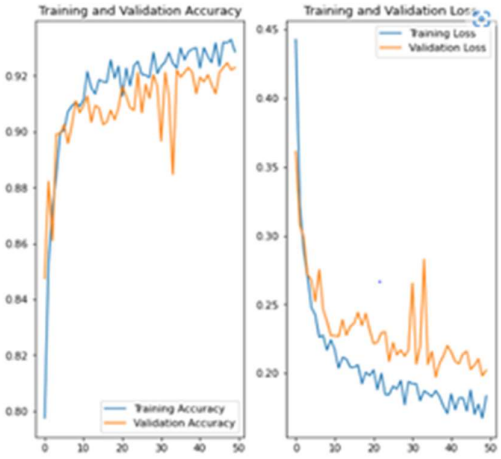


Fig 5.6.4 Inception V3

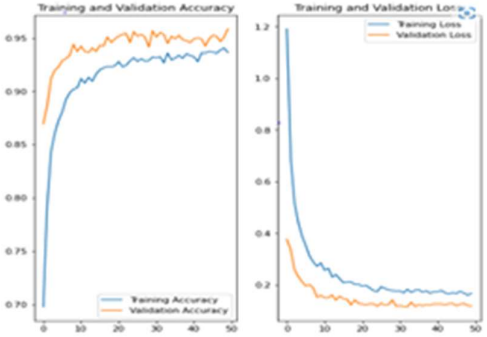


Fig 5.6.5 VGG

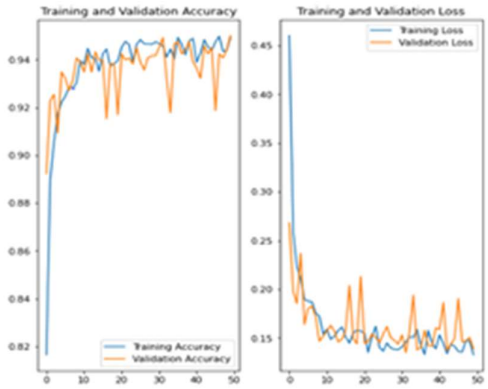


Fig 5.6.6 Xception

CONCLUSION

We will conclude that we will get an overview that the various techniques can be used in detecting lung disease (x-rays) based on deep learning. We have used data argumentation and data transformation techniques and analyzed them. Later we implemented algorithms based on deep learning to extract the features and classification can be done based on the algorithm used. We were able to select the best algorithm and apply model on them. On Classification, we got absolute and best accuracy for MobileNet .On comparing with the existing models by the frameworks , we conclude that by transfer learning approach and simple classifiers such as shallow neural networks we will pretrain the models which can successfully compete with the complex systems. The convolutional neural network might work for CT-Scan and X-Ray images but there is a problem in proper training considering these deep-leering techniques are used for coloured images. The present both vanilla CNN and transfer learning models are not perfectly fit for greyscale images. This can be improved in later models with better Computer Aided Graphics. Feature Extraction methods can be used to extend or to add additional features to the x-rays that can bring more information that can be dependable for CNN models. The present models can be improved by training with more refined images which have proper dimensions.

References

1. Anuj Rohilla RH, Mittal A. TB detection in chest radiograph using deep learning architecture. Int. J. Adv. Res. Sci. Eng. 2017;6:1073–1084. [Google Scholar]

2. Candemir S, et al. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. IEEE Trans. Med. Imaging. 2014;33:577–590. doi: 10.1109/TMI.2013.2290491. [PubMed] [CrossRef] [Google Scholar]

3. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 248–255 (2009)

4. Gordienko, Y., et al.: Deep learning with lung segmentation and bone shadow exclusion techniques for chest X-ray analysis of lung cancer. Computing Research Repository, vol. abs/1712.07632 (2017)

5. Islam, M.T., Aowal, M.A., Minhaz, A.T., Ashraf, K.: Abnormality detection and localization in chest X-rays using deep convolutional neural networks. ArXiv, vol. abs/1705.09850 (2017)
6. Jaeger S, Candemir S, Antani S, Wang Y-X, Lu P-X, Thoma G. Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. Quant. Imaging Med. Surg. 2014;4:475–477. [PMC free article] [PubMed] [Google Scholar]
7. Jaeger S, Karargyris A, Candemir S, Folio L, Siegelman J, Callaghan F, Xue Z, Palaniappan K, Singh RK, Antani S, Thoma G, Wang Y, Lu P, McDonald CJ. Automatic tuberculosis screening using chest radiographs. IEEE Trans. Med. Imaging. 2014;33:233–245. doi: 10.1109/TMI.2013.2284099. [PubMed] [CrossRef] [Google Scholar]
8. Kang Q, Lao Q, Fevens T, et al. Nuclei segmentation in histopathological images using two-stage learning. In: Shen D, et al., editors. Medical Image Computing and Computer Assisted Intervention – MICCAI 2019; Cham: Springer; 2019. pp. 703–711. [Google Scholar]
9. Kermany KZD, Goldbaum M. Large dataset of labeled optical coherence tomography (OCT) and chest X-ray images. Cell. 2018;172:1122–1131. doi: 10.1016/j.cell.2018.02.010. [PubMed] [CrossRef] [Google Scholar]
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS 2012), vol. 25, pp. 1097–1105 (2012)
11. Li F, et al. Computer-aided detection of peripheral lung cancers missed at CT: ROC analyses without and with localization. Radiology. 2005;237(2):684–90. doi: 10.1148/radiol.2372041555. [PubMed] [CrossRef] [Google Scholar]
12. Lo S-CB, Li H, Wang YJ, Kinnard L, Freedman MT. A multiple circular path convolution neural network system for detection of mammographic masses. IEEE Trans. Med. Imaging. 2002;21:150–158. doi: 10.1109/42.993133. [PubMed] [CrossRef] [Google Scholar]
13. Lo S-CB, Chan H-P, Lin J-S, Li H, Freedman MT, Mun SK. Artificial convolution neural network for medical image pattern recognition. Neural Netw. 1995;8:1201–1214. doi: 10.1016/0893-6080(95)00061-5. [CrossRef] [Google Scholar]
14. Pasa F, Golkov V, Pfeiffer F, Cremers D, Pfeiffer D. Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization. Sci. Rep. 2019;9:1–9. doi: 10.1038/s41598-019-42557-4. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
15. Hwang, S., Kim, H.-E.: A novel approach for tuberculosis screening based on deep convolutional neural networks. In: Medical Imaging 2016: Computer-Aided Diagnosis, vol. 9785, pp. 1–23 (2016)
16. Suzuki K. Pixel-based machine learning in medical imaging. Int. J. Biomed. Imaging. 2012;2012:1. [PMC free article] [PubMed] [Google Scholar]
17. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 2818–2826, June 2016
18. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Medical Image

Computing and Computer-Assisted Intervention – MICCAI 2015; Cham: Springer; 2015. pp. 234–241. [Google Scholar]