INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD
SUMMER PROJECT REPORT

Lung Inflammation Detection

Under the supervision of:
Dr. Pavan Chakraborty

Under the guidance of:
Ms. Snigdha Sen

Team: **Arya Krishnan, Ashutosh Kaushik, Priyal Gupta**

CERTIFICATE FROM SUPERVISOR

I do hereby recommend that the summer project report of project under my supervision Lung Inflammation Detection be accepted as complete fulfillment of the requirements needed as per the curriculum for the completion of summer project of seventh semester of Bachelor Of Technology in Information Technology.

Date : 19-09-2020 Dr. Pavan Chakraborty

Place : IIIT Allahabad

Contents

Abstract

The aim of this study is to automate the process of Lung Inflammation detection. We will explore the concept of transfer learning in the medical field as well as several CNN features and optimisation techniques. We will fine-tune different classifiers pretrained on ImageNet and draw up a comparision of their performances.

1 Introduction

Pneumonia is a life threatening lung inflammatory disease caused by viral or bacterial infection of the lungs. Early diagnosis is vital for efficient treatment. When detecting Lung Inflammation, CT scans are analyzed. The CT screening of millions of people puts a huge burden on radiologists and manual detection may be faulty. Hence recent efforts are to come up with an algorithm to automatically detect risks with professional accuracy.

Deep Learning indicates high diagnostic performance in early diagnosis of pneumonia from normal radiographs, also indicating a high accuracy in distinguishing between viral and bacterial pneumonia [4].

AI has the potential to review an immense amount of images and diagnose cases difficult for human experts. A convolutional network holds the ability to learn an abstract representation of the images within each layer. This is learnt by processing the images in the form of pixel values and classifying it as output [5]. This allows a simple end-to-end image-to-classification approach.

Since deep learning models are always data hungry, one way is to use the technique of transfer learning, especially when faced with a lack of labeled data in the given domain. Instead of training from scratch, the model can use the findings from a similar domain as initialisation for further fine-tuning.

2 Problem Definition

In this study we will try to get a CNN to achieve the same level of performance on a small dataset that it would on a large dataset. We will employ the method of transfer learning using pretrained architectures on ImageNet along with also trying different optimizers, initialization techniques, and addition of dropout and batch normalization layers to achieve optimal learning. We will use five different models and analyze their performances.

3 Motivation

Pneumonia is a serious disease which if not treated on time could lead to adverse conditions. It is basically lung inflammation disease caused by viral or bacterial infection of the lungs. At an early stage the disease can be treated. Detection of this disease by analyzing CT-Scan images by radiologists is very time taking and is less efficient. Hence researchers came up with the solution to perform detection using machine learning.
Here we make an effort on analyzing the performance of different models which could ease the task of radiologists. Also there are different previous research and models published for similar problems but there is none for the comparison and analysis of different approaches, which is a huge motivation for us to work on this problem and come up with the best results.
Deep Learning is however limited by the amount of data available in the domain. Training models requiring more than a million images on a few thousand can cause overfitting.

4 Analysis of previous research

4.1 Pneumonia Detection Using CNN based Feature Extraction

To automate something has always been a very exciting task. The authors in this paper [6] tried to automate the process of pneumonia detection. They tried to detect the presence of pneumonia in individuals using chest X-Ray images. They used CNN for feature extraction and combined it with other classifiers such as SVM to detect whether the person is infected or not. They divided the architecture of the model in three different stages. The first one was the pre-processing stage. In this stage they tried to reduce the computational complexity of their model by reducing the size of the images. The second stage was the feature extraction stage. Here they used different variants of the pre-trained CNN models and chose the optimal one. The third stage was the classification stage where they used different classifiers (for example SVM) to perform classification of images. They achieved the accuracy as 80.02%, which was highest among all the combinations of convolutional neural network architectures for feature extraction and classifiers.

4.2 A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images

In the paper [7], the authors have tried to come up with a novel deep learning model. They propose a majority voting ensemble of five different classifiers pretrained on ImageNet. They then proceed to compare and contrast their performances and show that their proposed ensemble outperforms all individual classifiers reaching an accuracy of 96.4%. They also attribute the success of deep learning to its ability to gain insights from raw data fed directly through a common learning process [8].
The authors mention that they intend to get the same level of performance from CNNs on a small dataset as a large one. For this purpose they propose using pretraining. They've included an AlexNet, InceptionV3, GoogLeNet, ResNet18 and a DenseNet121 all of which they mention are large models and could overfit easily. To avoid this they perform data augmentation and also add noise to the images.
The performance measure used is the cross-entropy loss function alongwith the Adam optimizer. They find that ResNet18 performed the best second to their proposed model at 94.23% and InceptionV3 performed the worst at 92.01%.

4.3 Pneumonia Identification in Chest X-Ray Images Using EMD

In this paper, [9] the authors tried to use a different algorithm for determining the presence of pneumonia. They calculated EMD(Earth's Mover Distance) to correctly identify the images of infected lungs. They first pre-processed the image and performed rotation, scaling and normalization of intensity. After that they obtained images of uniform shape and size. After that they calculated EMD and compared the results. The maximum accuracy they could achieve using this approach was about 83.33%.

4.4 Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray

In the paper [10] by Rahman et. al., four pre-trained CNNs namely AlexBet, ResNet18, DenseNet201 and SqueezeNet were used for transfer learning and performance analysis. The authors have followed three schemes of classification: normal vs Pneumonia, bacterial vs viral Pneumonia and normal, bacterial and viral Pneumonia.
The authors also note that a CNN typically performs well on a larger dataset than a smaller one justifying their preference to use transfer learning. They also make use of data augmentation such as rotation, scaling and translation to increase the size of their dataset. Their comparative analysis showed that DenseNet201 outperformed all the other models in all three classification schemes. They conclude by suggesting using a larger database and working on an ensemble as future work to increase accuracy further.

4.5 Inflammatory mechanisms in the lung

The authors of [11] have discussed different reasons and causes of inflammation. First they discussed immunity and how it's related to inflammation. Immunity is of two types, innate and acquired, innate is the one which is with us since birth and which also changes through evolution from generation to generation, whereas the other one is what we acquire while facing different antigens and then our immune system tends to learn the type of attacks. This way our immunity works. Now inflammation also falls into major two categories, that is acute and chronic. Acute is the case of pneumonia and chronic occurs in the case of asthma and other major diseases. Actually, inflammation in the lungs is the response of our immune system to the different pathogens and antigens entered in our respiratory system and tend to be harmful for the function and gas exchanges. During the inflammation various cells of this type become active which helps in reducing the irritation caused by these polluting particles. Authors have given a full understanding and details about lung inflammation and what's the importance of it.

5 Methodology

5.1 Preliminaries

5.1.1 Convolution Neural Network

Convolutional Neural Networks are a type of feedforward neural network. The network consists of two parts. The first part of the network consists of convolutional and max-pooling layers which act as feature extractor. The second part consists of the fully connected layer which performs non-linear transformations of the extracted features and acts as the classifier. In CNN network learns through filters and a bit of preprocessing over the image. That way it's a bit different from other image classification algorithms. We have used different architectures and would try to compare their performances and get the best out of them.

5.1.2 Lung Inflammation

Inflammation is the response of the immune system against some foreign element or something that irritates the particular part of the body. In lung Inflammation, air bags (alveoli) are swelled up, makes the breathing difficult and also the transfer of oxygen to the blood gets hindered. This might also occur in the case of lung cancer, pneumonia or some other lung disease. We are trying to automate the process of it's detection, which directly leads to the early detection of these fatal disease.

5.2 Batch Normalization, Dropout and He Initialization

Batch Normalization :- It is used to increase the stability of a neural network. It normalises the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. Batch Normalisation, basically allows each layer of a network to learn by itself a little bit more independently of other layers.

Dropout :- It is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase. It's computationally not too heavy and effective, nodes are dropped layer wise. It also helps in thinning and reducing the capacity of the network and ultimately producing better results.

He initialization :- This is an improved version of Random initialization (which was to assign weights randomly) where we initialize layer wise, the range of value of the initialization depends on the size of the previous layer directly. It has better results as compared to zero or random initialization.

5.3 Resnet50 [1]

This is a CNN architecture, 50 stands for the number of layers in the architecture, then ResNet breaks down into two, "res" and "net" which is res-idual neural net-work.

The most important feature of this architecture is the skip connection, first introduced by resnet. Deep neural networks have a vanishing gradient problem while backpropagating error from deeper in the network, the gradient saturates at a very small value from repeated multiplication with small weights.

This is resolved in ResNet50 by jumping over (or skipping) some layers whenever required. In a skip connection other than stacking convolutional layers we also add the original input to the output of the convolutional block. A residual block is when the activation is fed directly to a deeper layer skipping over the block.

5.4 InceptionV3 [2]

The Inception Network uses Inception layers which parallely run different sizes of filters on the image and lets each layer pick which filter size learns best. The network has 42 layers constituting of Inception modules. Every module consists of different filters in parallel each of which contribute to the output of the module.

Inception V3(version 3) is improved from v2, choosing the best kernel size is way more difficult for varying pictures, and an earlier network was made deep to get the best CNN which was computation heavy. Inception v1 is a new idea in which instead of using a single kernel size we use different (which are more suitable) sizes which make the network wider instead of making it deep and ultimately helping us in improving the overall performance.

5.5 VGG16 [3]

Visual geometry group 16 (VGG 16) is a convolution neural network architecture, 16 stands for the number of layers having weights. It is a highly dense network with around 140,000,000 parameters. It does not work on a large number of hyper-parameters, instead it contains the same convolution 3x3 layer with the stride length 1, and the exact same other parameters such as max-pool layer and padding. Finally we have 2 fully connected layers having the output of soft-max function. This VGG was proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper [3].

5.6 Dataset

We will be using the Kaggle dataset Chest X-Ray Images (Pneumonia). The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (Pneumonia/Normal). There are over 5,000 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal).
The training set: Images we're going to train the neural network on.
The validation set: Images we're going to use to check if the model is underfitting or overfitting.
The test set: These are images we're going to use to check how good our neural network is with data it has not seen before.

- 5216 images belonging to 2 classes in the training set.
- 16 images belonging to 2 classes in the validation set.
- 624 images belonging to 2 classes in the test set.

Preprocessing and data augmentation have been performed. For our train, validation and test sets, we will zoom the image randomly by a factor of 0.2, Rescale pixel values to 0 to 1, randomly flip half the images horizontally and vertically, and apply shear based transformations randomly.

We will run most models for 326 steps per epoch (number of batches per epoch) with a batch size of 16.

5.7 Evaluation Metrics

The networks are evaluated on their Accuracy measure and Binary Cross Entropy loss.

6 Experiments and Results

6.1 Experiment 1: Simple CNN

In this experiment, we train a Simple CNN with Default initialization, No Padding, No Dropout and No Batch Normalization.

Here is a summary of the Simple CNN model we trained and tested on Keras using the Adam Optimizer for 10 epochs.

Layer (type)	Output Shape	Param #
conv2d_3 (**Conv2D**)	(None, 62, 62, 32)	896
max_pooling2d_3 (**MaxPooling2**)	(None, 31, 31, 32)	0
conv2d_4 (**Conv2D**)	(None, 29, 29, 32)	9248
max_pooling2d_4 (**MaxPooling2**)	(None, 14, 14, 32)	0
flatten_2 (**Flatten**)	(None, 6272)	0
dense_3 (**Dense**)	(None, 128)	802944
dense_4 (**Dense**)	(None, 1)	129

Total params: 813,217
Trainable params: 813,217
Non-trainable params: 0

Figure 1: **Experiment 1**: Model architecture

- Training loss: 0.1356, Training accuracy: 94.86%
- Validation loss: 0.5137, Validation accuracy: 87.50%
- Test Loss: 0.49, Test Accuracy: 83.2%

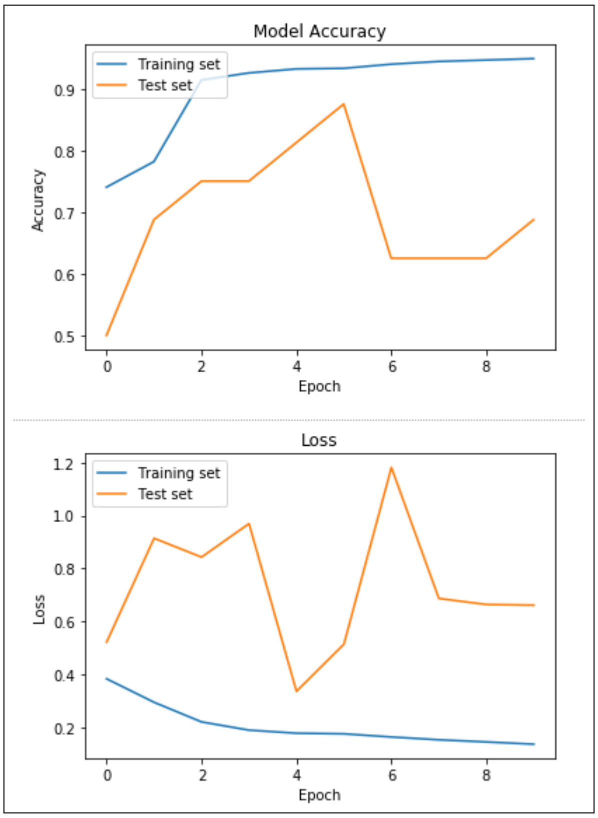

Figure 2: **Experiment 1**: Loss and Accuracy graphs

Note: the test set refers to the validation set in the graphs.

6.2 Experiment 2: Modified CNN

In this experiment, we train the simple CNN modified with He initialization, Padding added of size 1 on each side, Adam optimizer, Dropout and Batch Normalization.
The batch size used here is 32 and the training is done 652 steps per epoch. We run the training for 10 epochs.

Layer (type)	Output Shape	Param #
zero_padding2d_1 (ZeroPadding)	(None, 66, 66, 3)	0
conv2d_1 (**Conv2D**)	(None, 64, 64, 64)	1792
max_pooling2d_1 (**MaxPooling2**)	(None, 32, 32, 64)	0
dropout_1 (Dropout)	(None, 32, 32, 64)	0
zero_padding2d_2 (ZeroPadding)	(None, 34, 34, 64)	0
conv2d_2 (**Conv2D**)	(None, 32, 32, 64)	36928
dropout_2 (Dropout)	(None, 32, 32, 64)	0
flatten_1 (**Flatten**)	(None, 65536)	0
dense_1 (**Dense**)	(None, 256)	16777472
batch_normalization_1 (Batch	(None, 256)	1024
dropout_3 (Dropout)	(None, 256)	0
dense_2 (**Dense**)	(None, 128)	32896
batch_normalization_2 (Batch	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
dense_3 (**Dense**)	(None, 1)	129

Total params: 16,850,753
Trainable params: 16,849,985
Non-trainable params: 768

Figure 3: **Experiment 2**: Model architecture

- Training loss: 0.1290, Training accuracy: 95.25%
- Validation loss: 0.6409, Validation accuracy: 75.00%
- Test Loss: 0.28, Test Accuracy: 88.94%

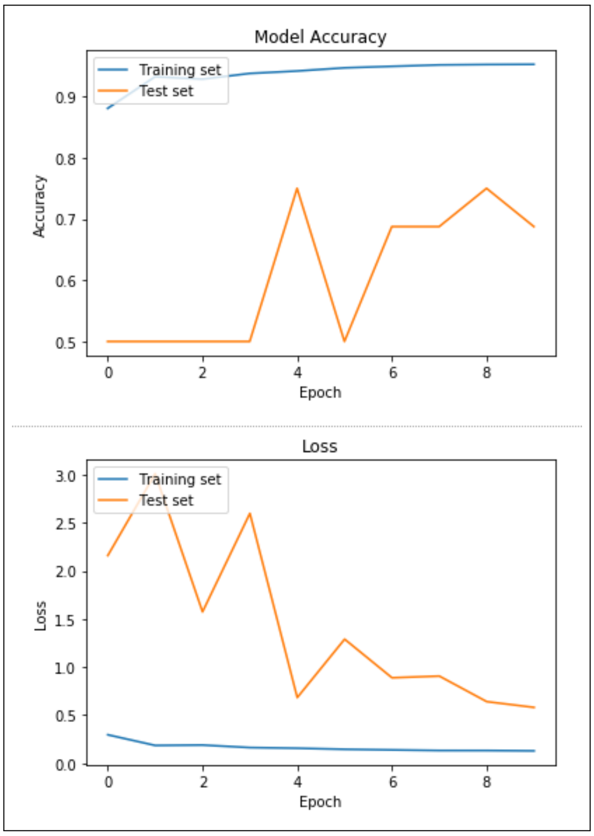

Figure 4: **Experiment 2**: Loss and Accuracy graphs

Note: the test set refers to the validation set in the graphs.

6.3 Experiment 3: ResNet50

In this experiment, we fine-tune the ResNet50 architecture.
We will run it for 16 epochs.
The optimizer used is the Stochastic Gradient Descent optimiser with a learning rate of 1e-4.
Resnet50 is often used in Transfer Learning in the medical field, pretrained on ImageNet as a feature extractor.
It is then fine-tuned using the SGD optimizer to fit the given data distribution.
We have also added a Global Pooling, Dense and sigmoid activation layer for the purpose of binary classification.

Layer (type)	Output Shape	Param #
resnet50 (Model)	(None, None, None, 2048)	23587712
global_average_pooling2d_1	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
dense_2 (Dense)	(None, 2)	1026

Total params: 24,637,826
Trainable params: 24,584,706
Non-trainable params: 53,120

Figure 5: **Experiment 3**: Model architecture

- Training loss: 0.0429, Training accuracy: 98.50%
- Validation loss: 0.1161, Validation accuracy: 93.75%
- Test Loss: 0.277, Test Accuracy: 89.74%

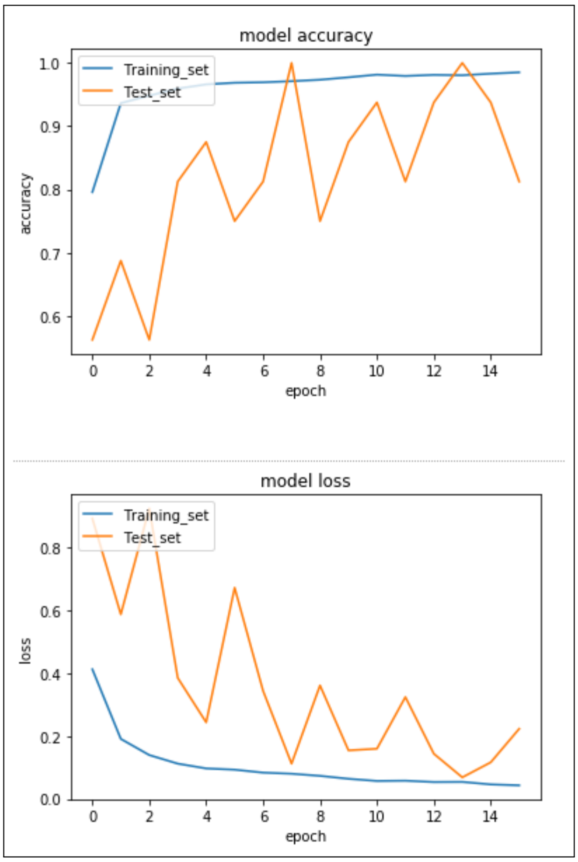

Figure 6: **Experiment 3**: Loss and Accuracy graphs

Note: the test set refers to the validation set in the graphs.

6.4 Experiment 4: InceptionV3

In this experiment, we fine-tune the InceptionV3 architecture.
We will run it for 16 epochs.
The optimizer used is the Stochastic Gradient Descent optimiser with a learning rate of 1e-4.
Inception V3 is often used in Transfer Learning in the medical field, pretrained on ImageNet as a feature extractor.
It is then fine tuned using the SGD optimizer to fit the given data distribution.
We have also added a Global Pooling, Dense and sigmoid activation layer for the purpose of binary classification.

Layer (type)	Output Shape	Param #
inception_v3 (Model)	(None, 4, 4, 2048)	21802784
global_average_pooling2d_1	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
dense_2 (Dense)	(None, 2)	1026

Total params: 22,852,898
Trainable params: 22,818,466
Non-trainable params: 34,432

Figure 7: **Experiment 4**: Model architecture

- Training loss: 0.0619, Training accuracy: 97.99%
- Validation loss: 0.2208, Validation accuracy: 93.75%
- Test Loss: 0.2204, Test Accuracy: 91.66%

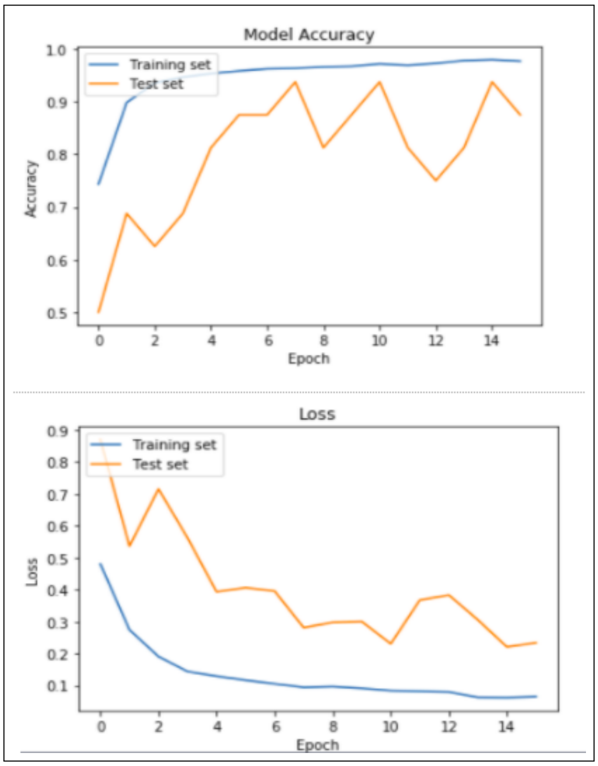

Figure 8: **Experiment 4**: Loss and Accuracy graphs

Note: the test set refers to the validation set in the graphs.

6.5 Experiment 5: VGG16

In this experiment, we fine-tune the VGG16 architecture.
We will run it for 16 epochs.
The optimizer used is the Stochastic Gradient Descent optimiser with a learning rate of 1e-4.
VGG16 is often used in Transfer Learning in the medical field, pretrained on ImageNet as a feature extractor.
It is then fine tuned using SGD optimizer to fit the given data distribution.
We have also added a Global Pooling, Dense and sigmoid activation layer for the purpose of binary classification.

Layer (type)	Output Shape	Param #
vgg16 (Model)	(None, None, None, 512)	14714688
global_average_pooling2d_3	(None, 512)	0
dense_5 (Dense)	(None, 512)	262656
dense_6 (Dense)	(None, 2)	1026

Total params: 14,978,370
Trainable params: 14,978,370
Non-trainable params: 0

Figure 9: **Experiment 5**: Model architecture

- Training loss: 0.0677, Training accuracy: 97.39%
- Validation loss: 0.1278, Validation accuracy: 93.75%
- Test Loss: 0.167, Test Accuracy: 93.75%

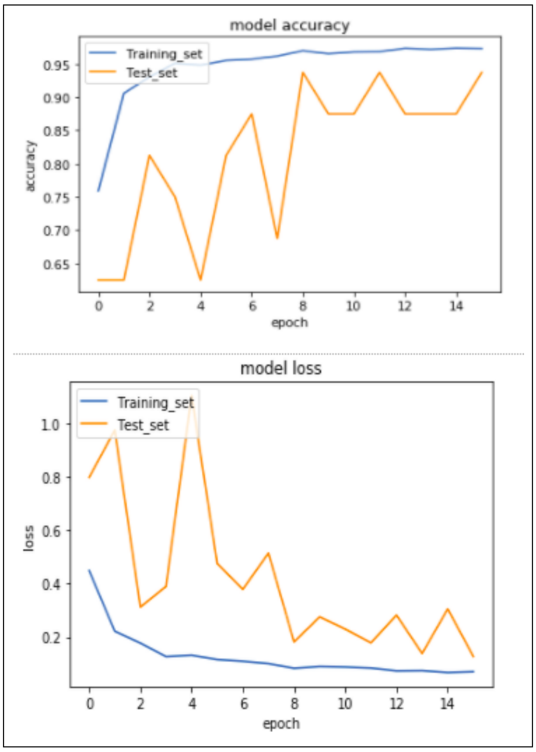

Figure 10: **Experiment 5**: Loss and Accuracy graphs

Note: the test set refers to the validation set in the graphs.

7 Discussion

The goal of this work is to apply transfer learning and compare and contrast the performances of multiple models common to the medical imaging domain to try and diagnose Pneumonia accurately.

Given below are the accuracies obtained by the above models by splitting the dataset into 3 parts, one for training, second for validation and the third one for testing.

Model	Training Accuracy	Validation Accuracy	Test Accuracy
1. Simple CNN	94.86	87.50	83.20
2. Modified CNN	95.25	75.00	88.94
3. ResNet50	98.50	93.75	89.74
4. InceptionV3	97.99	93.75	91.66
5. VGG16	97.39	93.75	93.75

Figure 11: Tabulation of the results

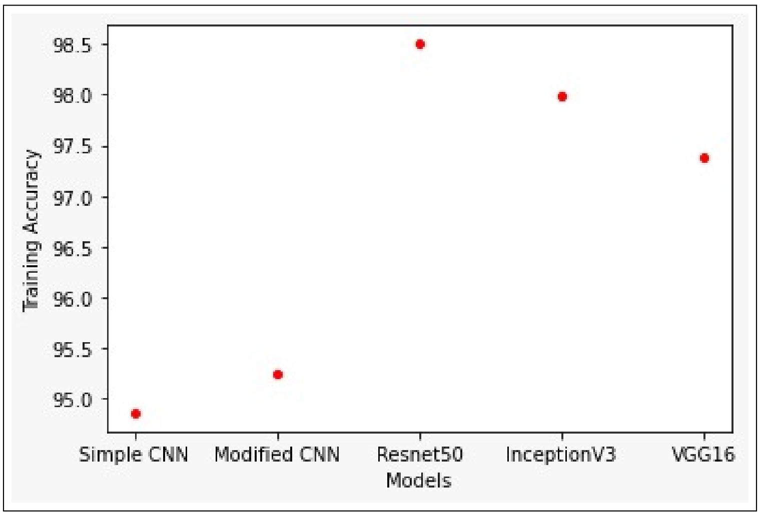

Figure 12: Comparison of train accuracies of all models

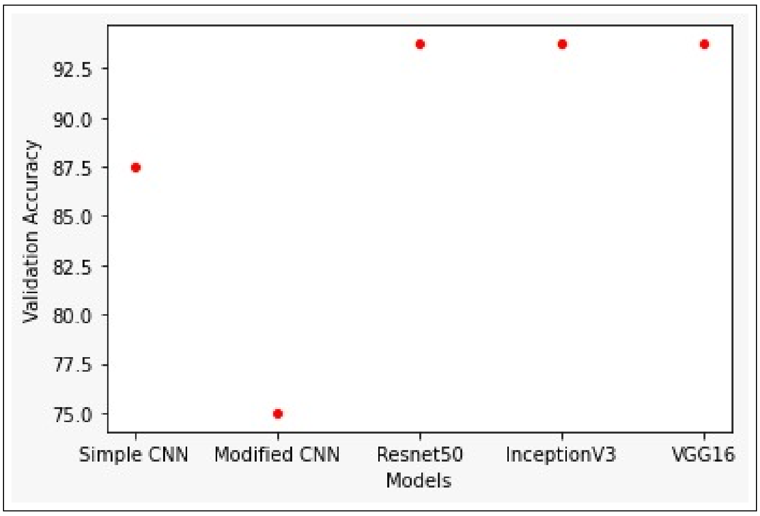

Figure 13: Comparison of validation accuracies of all models

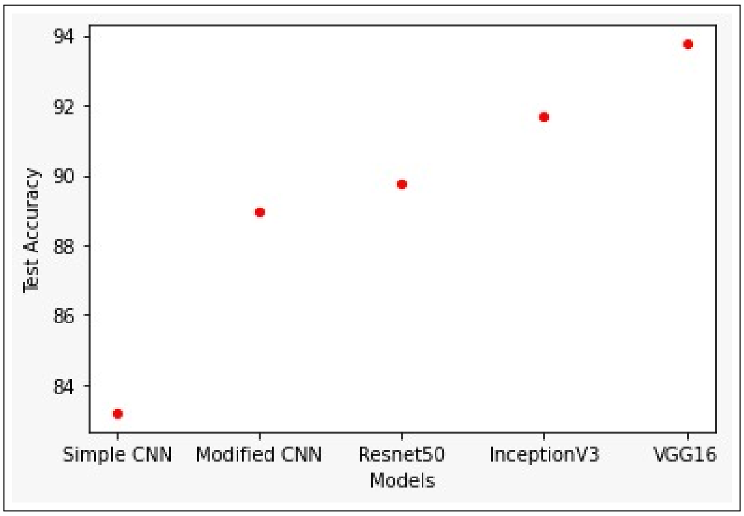

Figure 14: Comparison of test accuracies of all models

All the pre-trained models along with the modified CNN model are quite large for this dataset and hence overfit easily. These models can require more than a million images if trained from scratch, training on a few thousand can cause overfitting. To help our models generalize better we've used image augmentation and transfer learning with the pre-trained weights of ImageNet.

- We first started with training a simple CNN on our dataset, which resulted in a decent accuracy but couldn't fully capture the complexity of the data.
- Then we applied a few modifications on our first version, we used "he initialization" and dropout which resulted in an improvement. Dropout helps prevent overfitting and using "He" is a better initialization than random and zero, since it depends on the size of the previous layer. We also did batch normalization in which we divide it into batches and normalize the input to the current batch, ultimately making different batches a bit more self dependent.
- Since a ResNet can afford to be much deeper than the other models due to its residual blocks avoiding the Vanishing Gradient problem, it is capable of learning a new abstraction of the input in every layer.
- A ResNet can find a simpler mapping if it exists since it can even represent an identity mapping.
- Deciding the right depth of the model for learning a specific distribution is difficult, ResNet with its identity mappings gives some flexibility allowing us to add layers without any ramifications.
- ResNet is computationally more efficient than VGG given it is 8 times deeper but has fewer parameters than VGG.
- Similarly InceptionV3 provides flexibility in deciding the right convolutional filter size (instead of having to use fixed filter sizes in every layer) using Inception blocks that parallely apply kernels of varying sizes.

- Since different filter sizes capture different information, Inception can extract and concatenate different perspectives. The network then decides during training, which filter's weights it wants to reinforce.
- VGG is best when there is a need to reduce the number of parameters to be trained.
- This is done by using multiple fixed size kernels which are mostly of size 3x3. For example, a 5x5 kernel size could easily be replaced by two 3x3 size kernels, which could reduce the parameters from 25(5x5) to 18(3x3x2).
- This reduction of parameters in the convolutional layers results in faster learning.

Comparison				
Network	**Year**	**Salient Features**	**Top accuracy**	**Parameters**
VGGNet	2014	Fixed-size kernels	92.30%	138M
ResNet50	2015	Shortcut Connections	95.51%	23M
InceptionV3	2014	Wider-Parallel filters	93.30%	6.4M

Figure 15

Resnet was an improvement to VGG and is expected to perform better in our case, still results are otherwise. We believe this is due to the small fixed kernel size since the features being detected in the X-ray are small and distributed. VGG has a small fixed kernel size of 3x3, this fact may have helped it outperform the others and finally VGG did better as compared to Resnet. So based on our experimental work and result we can say that VGG is a better choice in our case for inflammation detection.

8 Conclusion and Future Work

In this work, we studied the different classifiers currently popular in the medical field along with modifying and pretraining them. We have shown an analysis and comparison of the classifiers on augmented and preprocessed data. We have implemented transfer learning using the ImageNet dataset and the SGD optimizer for finetuning to overcome overfitting on a very limited dataset. Although deep learning cannot at its current level outperform a medical diagnosis by doctors, it can, however, provide support in performing tedious tasks such as examining radiographs. It can provide a cheap and accurate diagnosis in countries with few medical resources and experts since the majority of the cases are found in underdeveloped countries. The accuracy achieved in this paper could be improved further perhaps by modifying the models, using other models or by introducing a larger dataset. For future research we will try different ensemble techniques to try and combine the most unique and effective characteristics of individual classifiers.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[2] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[4] Y. Li, Z. Zhang, C. Dai, Q. Dong, and S. Badrigilan, "Accuracy of deep learning for automated detection of pneumonia using chest x-ray images: A systematic review and meta-analysis," *Computers in Biology and Medicine*, vol. 123, p. 103898, 2020.

[5] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122 – 1131.e9, 2018.

[6] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, and A. Mittal, "Pneumonia detection using cnn based feature extraction," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–7, 2019.

[7] V. Chouhan, S. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damasevicius, and V. Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest x-ray images," *Applied Sciences*, vol. 10, p. 559, 01 2020.

[8] M. Bakator and D. Radosav, "Deep learning and medical diagnosis: A review of literature," *Multimodal Technologies and Interaction*, vol. 2, p. 47, 08 2018.

[9] R. Jain, H. Vashista, N. Mittal, P. Ranjan, and R. Janardhanan, "Pneumonia identification in chest x-ray images using emd," *Springer*, 01 2020.

[10] T. Rahman, D. M. Chowdhury, A. Khandakar, K. Islam, K. Islam, Z. Mahbub, M. Kadir, and S. Kashem, "Transfer learning with deep convolutional neural network (cnn) for pneumonia detection using chest x-ray," *Applied Sciences*, vol. 10, p. 3233, 05 2020.

[11] B. Moldoveanu, P. Otmishi, P. Jani, J. Walker, X. Sarmiento, J. Guardiola, M. Saad, and J. Yu, "Inflammatory mechanisms in the lung," *Journal of inflammation research*, vol. 2, pp. 1–11, 01 2009.

Comments and Suggestions