Improving Disease Prediction System Using Chest X-Ray Scans

Submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

Roll No	Names of Students
20154144	Akshay Sharma
20154061	Dipunj Gupta
20154080	Manmeet Singh
20154031	Gunjan Singh
20154001	Arjit Naresh Gupta

to the

Computer Science and Engineering Department
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY ALLAHABAD
Allahabad, UP, India – 211004

March 2018

Undertaking

I declare that the work presented in this report titled **Improving Disease Prediction System Using Chest X-Ray Scans** , submitted to the Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology, Allahabad, for the award of the Bachelor of Technology degree in Computer Science & Engineering, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

April 2018
Allahabad

Akshay Sharma

Dipunj Gupta

Manmeet Singh

Gunjan Singh

Arjit Naresh Gupta

Certificate

Certified that the work contained in the report titled **Improving Disease Prediction System Using Chest X-Ray Scans**, by
Group: CS02
Akshay Sharma
Dipunj Gupta
Manmeet Singh
Gunjan Singh
Arjit Naresh Gupta
goes here, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Ranvijay
Computer Science & Engineering Department
M.N.N.I.T. Allahabad

April 2018

Preface

In this report we are implementing machine learning models to improve accuracy in disease prediction using chest X-ray scans.
For the project, we wanted to do something that would use cutting-edge technology in the domain of our study as well as be something that could benefit the common people if implemented as a product. We thus got the idea of using machine learning techniques to automate the detection of diseases from X-ray scans.
Report gives attention to feature selection techniques, data augmentation, data preprocessing and previous research work done on the same.
The blend of learning and knowledge acquired during project work and its implementation is presented in this report. Besides, this report also presents the import results and conclusions inferred from the experiments carried out.

Acknowledgement

It is a great pleasure to thank the giants on whose shoulders we stand. We wish to express our sincere gratitude to Dr. Ranvijay for providing us an opportunity to work on this project and for his guidance and encouragement at every step of the project.
His advice, encouragement and critique are sources of innovative ideas, inspiration and causes behind this report. He exposed us to the intricacies of relevant topics through paper counselling and discussions and always showed great interest in providing timely support and suitable suggestions.
We would also like to thank other groups which were willing to share information about various features as viewed in practical life. We would like to express our gratitude for our colleagues and friends for their constant support and encouragement.

Akshay Sharma
Dipunj Gupta
Manmeet Singh
Gunjan Singh
Arjit Naresh Gupta

Contents

List of Figures

Chapter 1

Introduction

Lung cancer is the most common cancer worldwide and is estimated to be the leading cause of cancer death and is typically at an advanced stage at diagnosis with very low survival rate due to delay in diagnosis. The chest radiograph is the most commonly performed radiological investigation because it is quick, inexpensive, and associated with low radiation doses. Detecting pulmonary lesions on plain film is challenging because, despite high spatial resolution, there is limited contrast resolution. The planar nature means that the heart, diaphragm, and mediastinum obscure a large part of lungs, and patients often have several co-existing pathologies visible on each chest radiograph. Furthermore, many benign findings can mimic a nodule, e.g. opacity due to composite shadowing or skin lesions. Studies have shown that in new lung cancer diagnoses where the lesion can be seen on plain film, with hindsight the lesion was appreciable on previous studies but had been missed in 19 to 40% of the cases.[12]
Computer-aided detection (CAD) systems using machine learning techniques have facilitated automated detection of lung nodules and provided a cost-effective double-reporting mechanism.
This report presents the underlying details of the mini project done in attempts to find new methods to improve accuracy of detecting a particular disease from X-ray scans provided using various image recognition models. The dataset has 14 classes of disease definitions and an additional no disease class.

1.1 Motivation

- **Improve Accuracy**- There has been significant research in the field of disease prediction using X-ray scans but the accuracy achieved has not been so appealing. This is one of the facts that encouraged us to carry our project work in this field.
- **Portable Devices**- Besides, we also intend to prepare a model that is compatible with portable devices. The model would be lightweight to work alongside minimum computational availabilty.
- **Improve Runtime**- We also intend to improve the runtime of the model for providing quicker results.
- **Multilabel Classification**- One of the most encouraging points was to create a multilabel classification model which can diagnose the presence of mutilple diseases from a single scan.

1.2 Some Wonderful Minds

- Andrew Ng & his group tried to develop an algorithm that can detect pneumonia from chest X-rays at a level exceeding practicing radiologists[13].
- The next section will provide further implementation details and related work about the dataset and gains in accuracy level by various researchers.

Chapter 2

Related Work

2.1 ChestX-ray

The chest X-ray is one of the most commonly accessible radiological examinations for screening and diagnosis of many lung diseases. A tremendous number of X-ray imaging studies accompanied by radiological reports are accumulated and stored in many modern hospitals Picture Archiving and Communication Systems (PACS). A chest X-ray database, which comprises 112,120 frontalview X-ray images of more than 30,000 unique patients with the textmined 14 disease image labels (where each image can have multi-labels), from the associated radiological reports using natural language processing has been created. In the dataset, X-ray images are directly extracted from the DICOM file and resized as 1024*1024 bitmap images without significantly losing the detail contents
Deep learning yields has given a rise in performance in the medical image analysis domain for object (often human anatomical or pathological structures in radiology imaging) detection and segmentation tasks. The given dataset was analysed by training with multilabel DCNN classification models. Fig. 2.1 illustrates the DCNN architecture adapted earlier, with similarity to several previous weakly-supervised object localization methods. As shown in Fig. 2.1, the network surgery on the pre-trained models (using ImageNet), e.g., AlexNet, GoogLeNet, VGGNet-16 and ResNet-50, by leaving out the fully-connected layers and the final classification layers has been done. Instead a transition layer, a global pooling layer, a prediction layer and a loss layer was inserted in the end (after the last convolutional layer)[1].

AUCs of ROC curves for multi-label classification in different DCNN model settings have been shown on table[1].

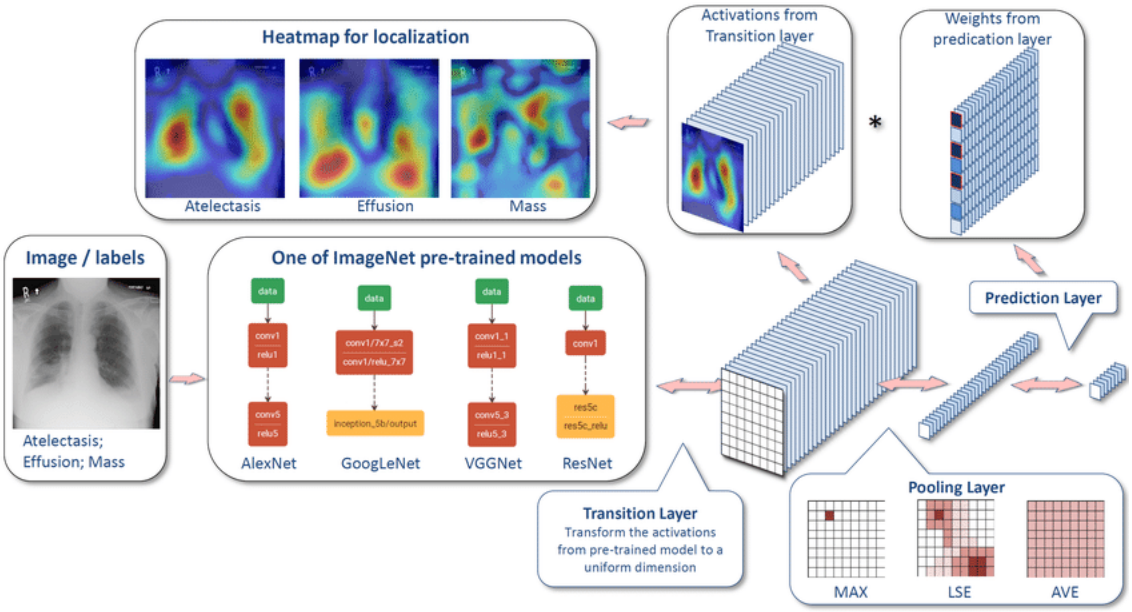


Figure 2.1: The overall flow-chart of unified DCNN framework and disease localization process.

Setting	Atelectasis	Cardiomegaly	Effusion	Infiltration	Mass	Nodule	Pneumonia	Pnumothorax
AlexNet	0.6458	0.6925	0.6642	0.6041	0.5644	0.6487	0.5493	0.7425
GoogLeNet	0.6307	0.7056	0.6876	0.6088	0.5363	0.5579	0.5990	0.7824
VGGNet-16	0.6281	0.7084	0.6502	0.5896	0.5103	0.6556	0.5100	0.7516
ResNet-50	0.7069	0.8141	0.7362	0.6128	0.5609	0.7164	0.6333	0.7891

Chapter 3

Methodolgy Used

The original dataset consisted of over 110,000 images and training on that did not seem computationally feasible. We trained on a smaller dataset of 5606 images with all the classes.
After the preprocessing of the images, following models were being trained and tested upon the dataset.

3.1 Machine Learning Approach

During the initial phase of our project, we tried to achieve our goal of improving the accuracy with the help of traditional machine learning algorithms like:

3.1.1 Decision Tree Classifier

This algorithm gave a very little accuracy on the dataset and hence we moved towards trying more complex algorithms.

3.1.2 Random Forest Classifier

Random Forest Classifier was being trained on the model with proper hyper-parameter tuning for improving the accuracy.

3.1.3 K-Nearest Neighbor Classifer

This algorithm gave almost the same accuracy as Random Forest Classifier.

3.2 Deep Learning Approach

We used different Convolutional Neural Network models alongwith data augmentation techniques for achieving accuracy:

3.2.1 Convolutional Neural Networks in Brief

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing.[1] They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

3.2.2 Design

A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers[6].

- **Convolution Layer**- The convolution layer emulates the response of an individual neuron to visual stimuli.
- **Pooling Layer**- Convolutional networks may include local or global pooling layers[clarification needed], which combine the outputs of neuron clusters at one layer into a single neuron in the next layer.
- **Fully Connected Layer**-Fully connected layers connect every neuron in one layer to every neuron in another layer.

3.2.3 Four-layered CNN

We used a ReLU based CNN involving 4 layers with filters of 4x4 and 2x2 dimensions. This was the beginning of our quest to solve this problem and we moved on to other models to increase accuracy.

3.2.4 AlexNet

We used Rectified linear unit activation function which sped up learning upto 6 times with same accuracy. A total of 8 layers were implemented, the first 5 were convolutional layers followed by 3 fully connected ones.

3.2.5 VGGNet

A typical VGGNet consists of 11-19 layers. We implemented a 16 layer model and found a slight increase in accuracy from AlexNet. Using lesser layers, we reached a trade-off where our accuracy wasn't affected as much and the training could be done on the dataset given our computation constraints.

3.2.6 DenseNet

The fact that CNN's can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output improved accuracy on X-ray scans using DenseNet.

3.2.7 MobileNet

Our last (and the best) results came from the implementation of MobileNet designed to perform well even under strict constraints on memory and computational budget.

The next section will provide further implementation details about the dataset and gains in accuracy on a numerical level by these models.

subfig

Chapter 4

Proposed Work

4.1 Problem Description

In the modern era, Computer-aided detection (CAD) systems using machine learning techniques have facilitated the automated detection of diseases, identifiable from the Chest X-ray scans. But, the achieved results are not that convincing and moreover, they involve large computation requirements which make them unsuitable for usage in hand-held device applications. Besides, multilabel classification of the diseases from a single scan also needs to be improved. The main aim of this project is to improve the accuracy of the disease prediction system using X-ray scans in an optimized and efficient way.

4.2 Solution Framework

4.2.1 Dataset

Due to the limited availability of resources for computation, we are using only 5% of the dataset available, prepared after uniform assortment of images from the complete dataset. This dataset contains 5606 images from the 15 classes (14 diseases, and one for N̈o findings)̈. Images can be classified as "No findings" or one or more disease classes:

- Atelectasis
- Consolidation
- Infiltration
- Pneumothorax

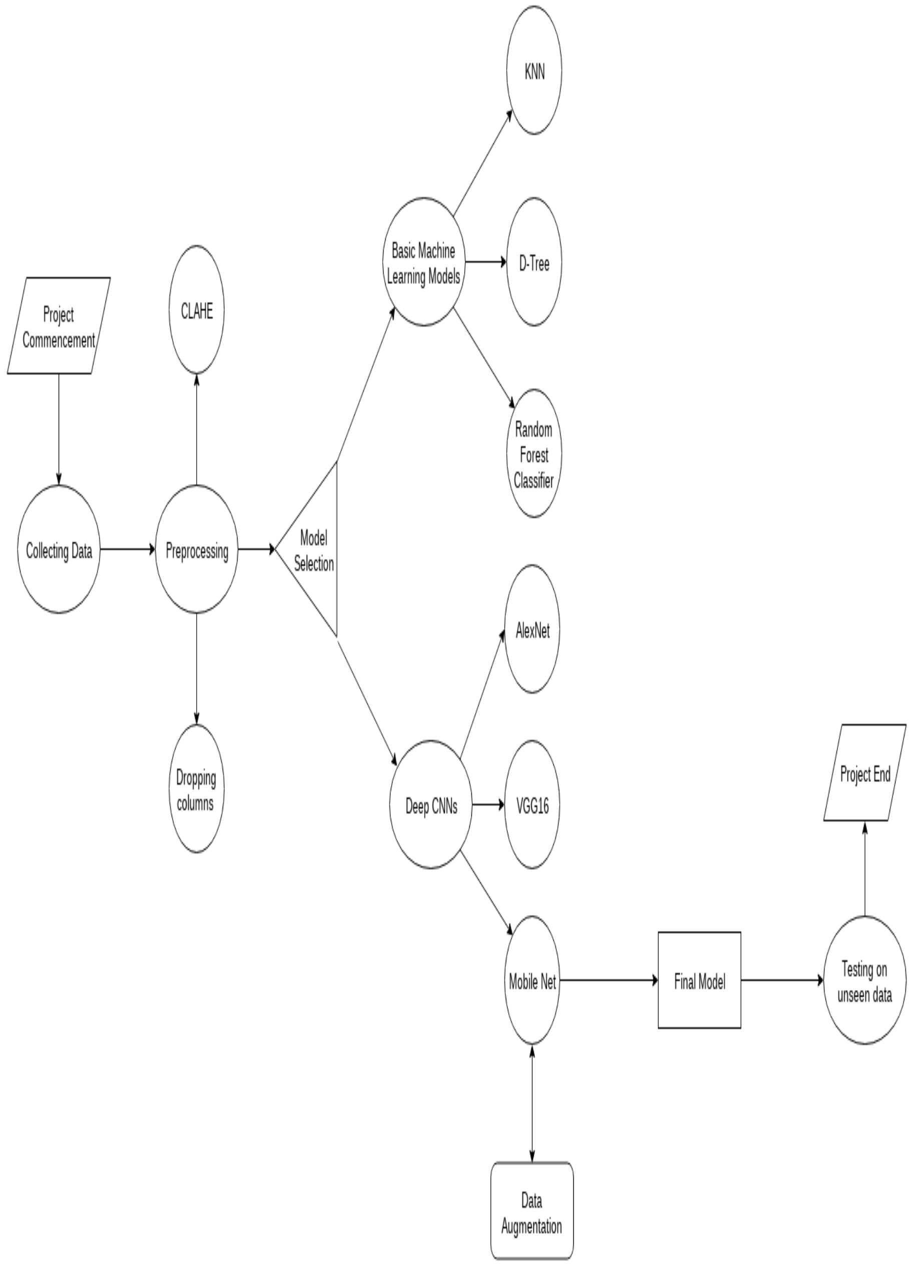


Figure 4.1: Solution Framework & Workflow

- Edema
- Emphysema
- Fibrosis
- Effusion
- Pneumonia
- Pleural Thickening
- Cardiomegaly
- Nodule Mass
- Hernia[3]

4.2.2 Preprocessing Of Dataset

Primary Processing Of Images

The dataset consists of RGB images of dimensions 1024*1024 which were resized for fitting on various Machine Learning & Deep Learning models/algorithms. The images were resized and also various coloring standards were being tested. Following dimensions of images were being tested upon in combination with RGB and Grayscale coloring schemes:

Dimensions	Coloring Scheme
128*128	Grayscale
227*227	Grayscale
227*227	RGB
224*224	Grayscale
224*224	RGB

Moreover, we have used INTER_CUBIC interpolation for getting sharp images at a faster transformation speed.[2]

Image-Disease Mapping & One Hot Encoding

According to the given dataset, we are given a CSV file containing the information related to the X-Ray scan images specifying the diagnosed diseases, patient information, etc. So, for applying various algorithms the data was structured and the information related to the images was mapped against them. The diseases were numbered and the target variable(predicted disease) was one-hot encoded. Moreover, the images were converted to numpy arrays for processing.

Contrast Limited Adaptive Histogram Equalisation(CLAHE)

In the previous research, images have been preprocessed with histogram equalization. The histogram equalization considers the global contrast of the image. In many cases, it is not a good idea. In certain cases, we lose most of the information due to over-brightness. It is because the histogram is not confined to a particular region. So to solve this problem, adaptive histogram equalization is used. In this, image is divided into small blocks called tiles (tile size is 8x8 by default in OpenCV). Then each of these blocks are histogram equalized as usual. So in a small area, histogram would confine to a small region (unless there is noise). If noise is there, it will be amplified. To avoid this, contrast limiting is applied. If any histogram bin is above the specified contrast limit (by default 40 in OpenCV), those pixels are clipped and distributed uniformly to other bins before applying histogram equalization. After equalization, to remove artifacts in tile borders, bilinear interpolation is applied[4].

4.2.3 Data Augmentation

In order to build a powerful image classifier using very little training data, image augmentation is usually required to boost the performance of deep networks. Image augmentation artificially creates training images through different ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips, etc. We have applied data augmentation techniques like:

- Samplewise data normalization
- Horizontal Flipping
- Vertical Shifting
- Horizontal Shifting
- Zooming
- Rotation, etc.

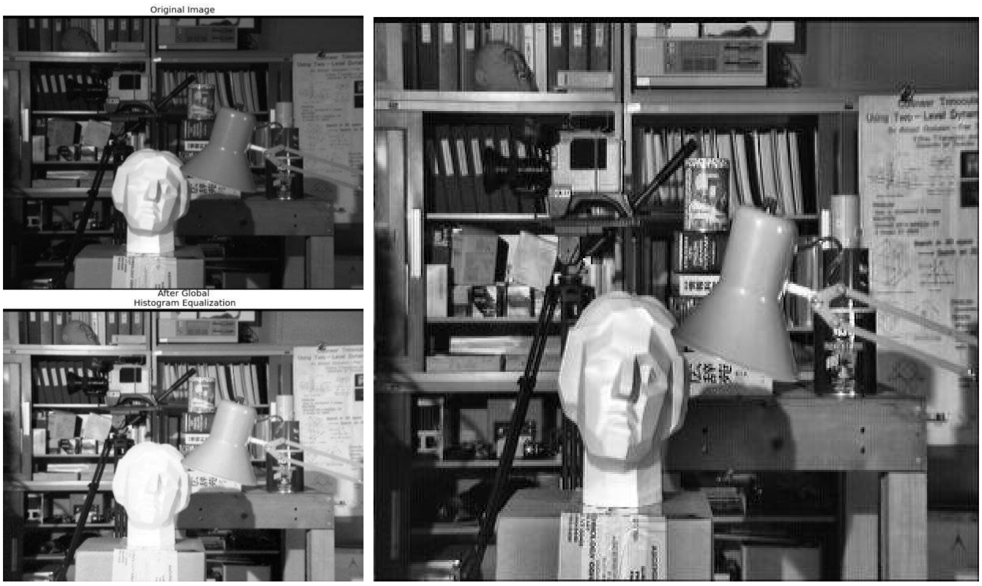


Figure 4.2: Contrast Limited Adaptive Histogram Equalization

4.2.4 Training On Dataset

The dataset was divided into training set and testing set having a split of 20% that is training set having 4204 images and testing set having 1402 images.

4.2.5 Traditional Machine Learning Approach

1. **Decision Tree Classifier**- Decision Tree Classifier, repetitively divides the working area(plot) into sub areas/classes by identifying various features. It repetitively divides the tree into branches until purity condition is reached or information gain becomes 0. We analysed the accuracy of Decision Tree Classifier on our data by varying the minimum number of samples required to split an internal node(min_samples_split). Maximum accuracy was found at min_samples_split=40 of 47%.

2. **Random Forest Classifier**- A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement. Initially, the model was trained on the training set and then

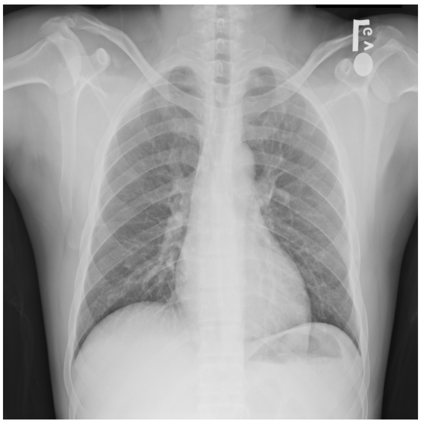


Figure 4.3: Original Image

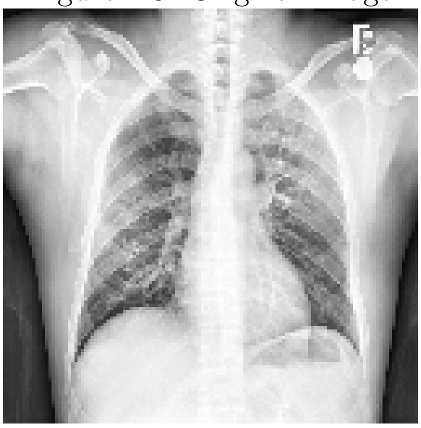


Figure 4.4: Preprocessed Image

Figure 4.5: Difference between Original & Preprocessed Images

a graph was plotted by tuning the parameters. The major changes in the accuracy were observed on tuning the maximum depth of the decision tree classifiers. The maximum accuracy obtained using this model is 65% at maximum depth=2 and number of decision tree classifiers(n_estimators)=10. A graph was also constructed by varying the number of decision tree classifiers.

3. **K-Nearest Neighbors(KNN)**- The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. We tried to achieve accuracy using KNN on our dataset by tuning the values of K(n_neighbors) and giving more weightage to the closer points(weight). The maximum accuracy of 65% was achieved at n_estimators=50 & weight='distance'[5].

4.2.6 Deep Learning Approach involving Convolutional Neural Network

1. **Initial Model (4-Layer CNN)**- The layer structure of our initial Convolutional Neural Network is as follows:
   CONV2D ->RELU ->MAXPOOL ->CONV2D ->RELU ->MAXPOOL ->FLATTEN ->FULLYCONNECTED
   After training the model on the dataset, we found a maximum accuracy of 67% at learning rate of 0.001 after running 10-15 epochs.

2. **AlexNet(Modified)**- AlexNet contains 5 convolutional layers and 3 fully connected layers[7]. Relu is applied after very convolutional and fully connected layer. Dropout is applied before the first and the second fully connected year. The image size in the following architecutre chart should be 227 * 227. In our model, we have avoided the dropout to speed up the training process. After training the model on the dataset, we found a maximum accuracy of 68.4% at learning rate of 0.009 after running 10-15 epochs.

3. **VGG16 Net**- VGGNet consists of 16 convolutional layers and is very appealing because of its very uniform architecture. It only performs 3*3 convolutions and 2*2 pooling all the way through[8]. It is currently the most preferred choice in the community for extracting features from images. The weight configuration of the VGGNet is publicly available and has been used in many other applications and challenges as a baseline feature extractor. After training the model on the dataset,

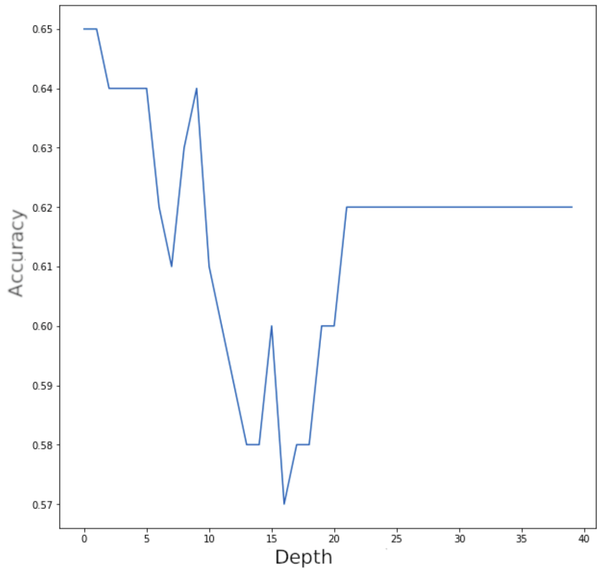
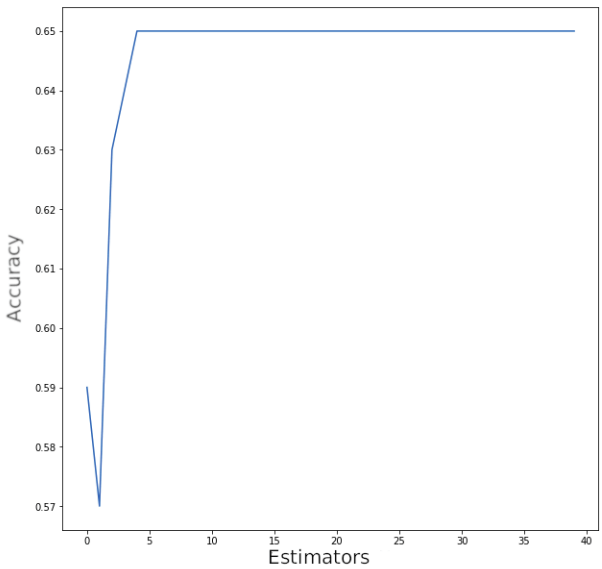


Figure 4.6: Random Forest Classifier: Accuracy variation with maximum depth of decision tree classifier & estimators



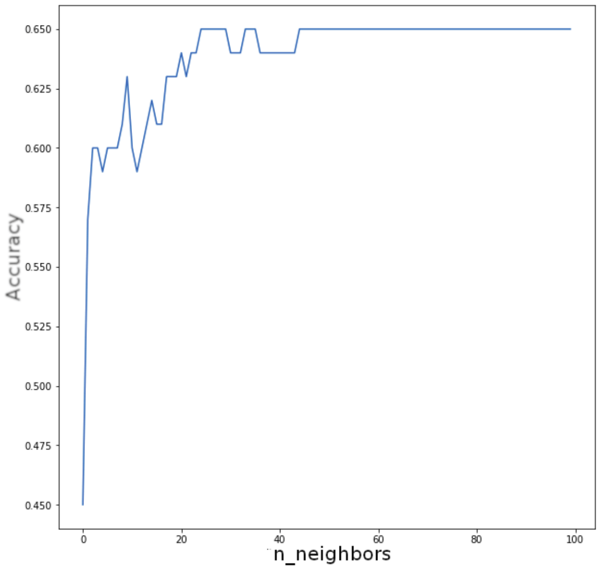


Figure 4.7: KNN: Accuracy variation with number of neighbors

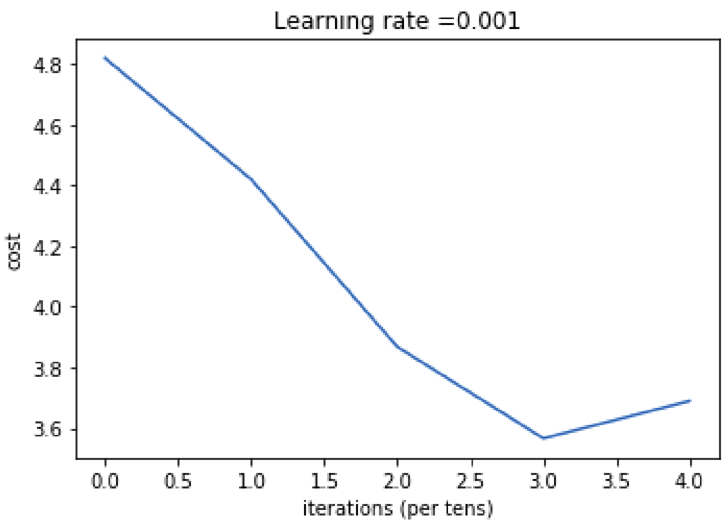


Figure 4.8: 4-layer CNN graph plotted for first 4 epochs

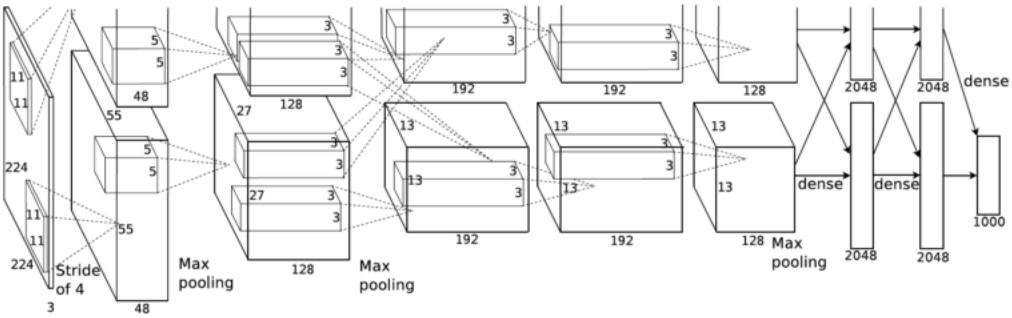


Figure 4.9: Architecture Of AlexNet

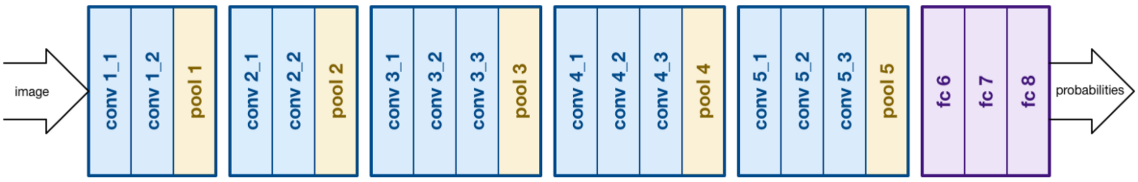


Figure 4.10: Architecture Of VGG16

we found an accuracy of 66-68% at learning rate of 0.009 after running 20-30 epochs.

4. **DenseNet121**- In the research so far, DenseNet has shown the most impressive results when trained on the dataset. Densenet contains a feature layer (convolutional layer) capturing low-level features from images, serveral dense blocks, and transition layers between adjacent dense blocks. Densenet121 has four dense blocks, which have 6, 12, 24, 16 dense layers respectively. DenseNet121 connects each layer to every other layer in a feed-forward fashion[9]. Whereas traditional convolutional networks with L layers have L connections - one between each layer and its subsequent layer - our network has L(L+1)/2 direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters[10]. After training the model on the dataset, the accu-

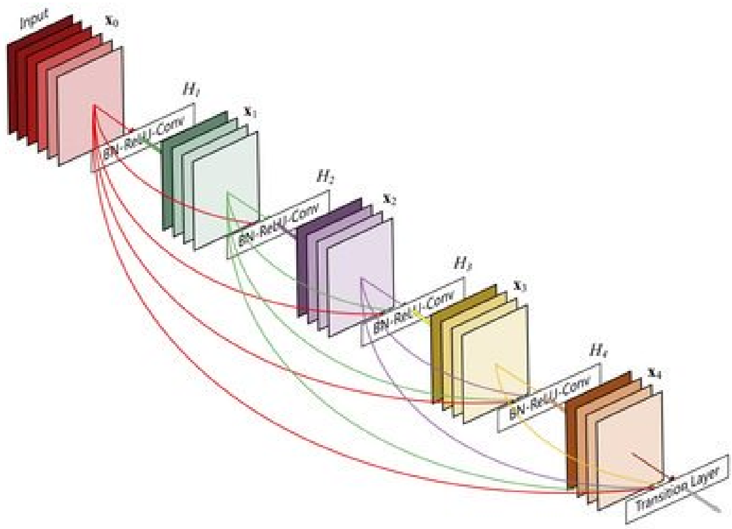
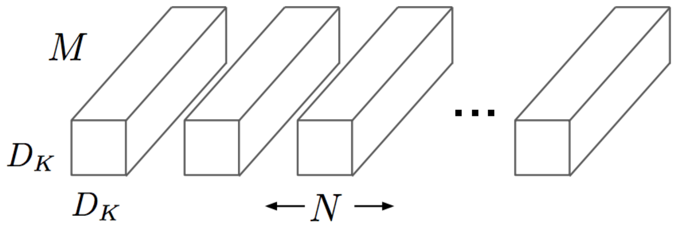


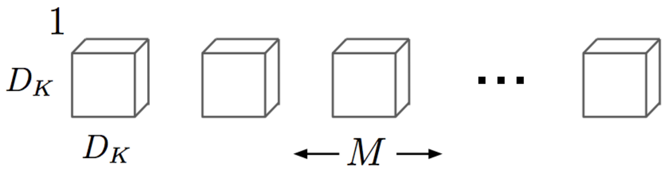
Figure 4.11: Architecture Of DenseNet

racy still remained approximately 66-68% at learning rate of 0.009 after running 20-30 epochs.

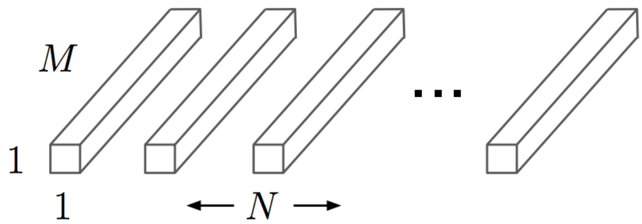
5. **MobileNet with Data Augmentation**-MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. MobileNets primarily focus on optimizing for latency but also yield small networks.
   The MobileNet model is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1*1 convolution called a pointwise convolution. We use depthwise convolutions to apply a single filter per each input channel (input depth). Pointwise convolution, a simple 1*1 convolution, is then used to create a linear combination of the output of the depthwise layer. MobileNets use both batchnorm and ReLU nonlinearities for both layers. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size. MobileNet has 28 layers. MobileNet uses 3*3 depthwise separable convolutions which uses between 8 to 9 times less computation than standard convolutions at only a small reduction in accuracy[11].



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 4.12: Depthwise Separable Convolution & Standard Convolution

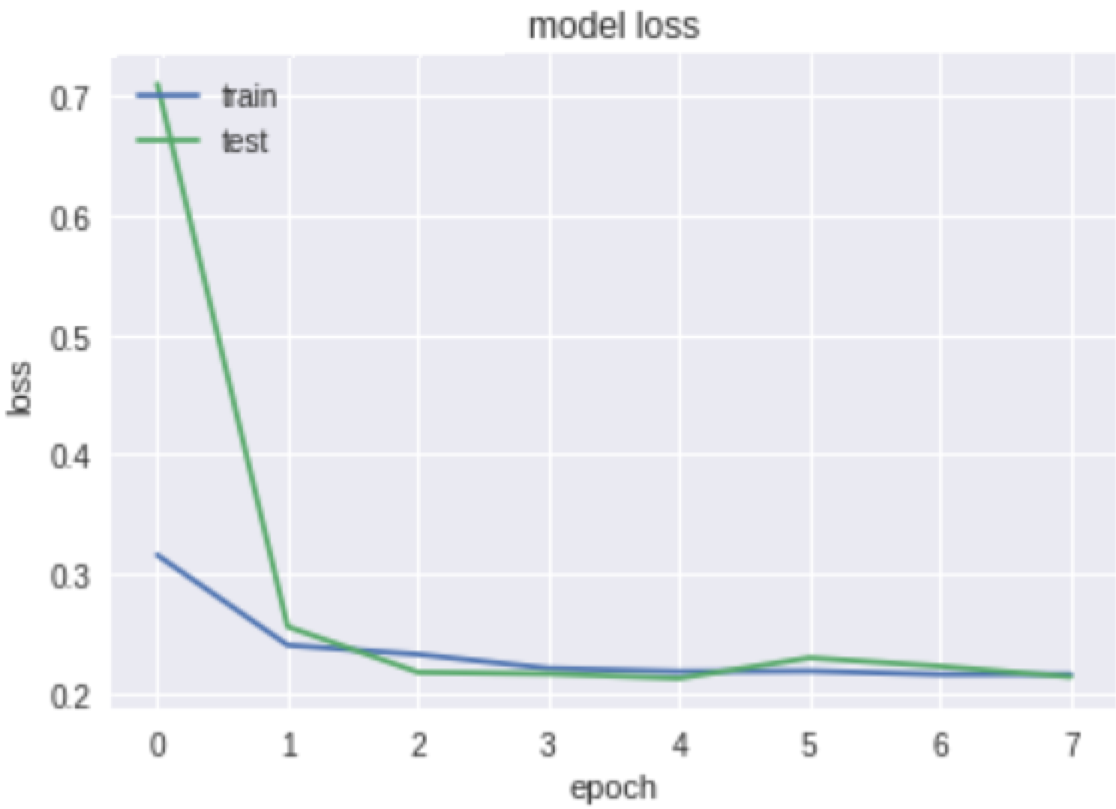


Figure 4.13: MobileNet: Loss versus Epoch Graph

Before training the model on the dataset, we implemented data augmentation techniques to improve the accuracy of the model. Besides, Grayscale images of dimensions 128*128 were used. Moreover, we added 2 fully connected layers alongwith dropout to perform regularization for avoiding the overfitting of the model. Optimization was done using Adam optimizer and loss calculated was binary_crossentropy loss due to multi-label classification. The accuracy(binary) achieved on validation set after training improved from 68% to about 91-93% after just 7 epochs.

Chapter 5

Experimental Setup and Results Analysis

5.1 Experimental Setup

5.1.1 Google Colab

For such a huge image dataset we required powerful GPUs to train the complex model that we would be using during the course of the project. But, due to unavailability of local GPUs we decided to opt for a cloud based service that offers GPU environment. Google recently released its Colab platform which allows FREE GPU compute time on an Nvidia K80 GPU. It is built on top of Jupyter Notebook. But because of the experimental nature of the service, we only get a GPU for 12 hours at once. We dealt with this non-persistent nature of Google Colab by writing a Jupyter Notebook commands as a script that when executed would quickly import the data into the environment with minimum hassle.

5.1.2 Importing Data On Google Colab

To import the data, all we needed was the authenticated download URL from the kaggle's website. To generate the said URL, login into kaggle and press download on the dataset. We then used wget as a built-in magic command to fetch the dataset from kaggle directly to Colab notebook's server. This saved us lot of bandwidth and thanks to the fast network connectivity at Colab Servers, and the download was blazing fast. Then we extracted the zip file into a single DATA folder which had an organized structure, so that whenever we needed to establish the environment, we could be sure that it had a predefined consistent structure.

Moreover, to counter the connectivity issues we also uploaded the dataset on Google Drive which can be used directly.

5.1.3 Kaggle

In addition to the experimental setup on Google Colab, we also tried to make the best use of the GPU provided by kaggle.com. Kaggle provides limited access to Nvidia Tesla K80 GPU. Since, the dataset was available on the website itself, so we didn't need to import it explicitly for carrying out our preprocessing, training and validation work. Kaggle also provides Jupyter Notebook environment for carrying out the experimental work. But, the access is limited to only one notebook for a period of 12 hours.

5.2 Results Analysis

After carrying out various Machine Learning & Deep Learning Algorithms, we strived for improving the accuracy of our Model on the given dataset and here are the results for the same:

5.2.1 Basic Models

1. Decision Tree Classifier- 47%
2. Random Forest Classifier- 65%
3. K-Nearest Neighbours- 65%

5.2.2 Neural Network Architectures

1. 4-Layer CNN- 66-67%
2. Alexnet- 67-68%
3. Vgg-16- 66-68%
4. DenseNet- 66-68.4%
5. MobileNet- 91-93%

5.2.3 Outcomes

From the research work carried out during the project duration, we are able to achieve the following:

1. We are able to improve the AUC-ROC accuracy metric of the following diseases:
    - Effusion
    - Infiltration
    - Mass

    See figure 5.1. We have achieved these results by training on just 5% of the total dataset that is, 5606 images. This accuracy will probably increase with increase in the size of dataset for training.

2. Our focus was on the multilabel classification of the diseases against the multiclass classification and have succeeded in achieving the accuracy of 92-93%(approx.) for the same.

3. We are able to achieve these results by using a deep learning approach(CNN), that is, MobileNet, which offers much effective and efficient results with minimum computational requirements.

4. We are working on a dataset having 15 classes(14 diseases & 1 no finding) in the target variable. In the previous research, this number has been limited to 8. We have managed to achieve significant results on the remaining classes as well.

The Accuracy improved by almost 30% by moving from the basic 4-layer CNN to MobileNet. The accuracy was approximately 92.6% at last in mobileNet.

Setting	Atelectasis	Cardiomegaly	Effusion	Infiltration	Mass	Nodule	Pneumonia	Pnumothorax
AlexNet	0.6458	0.6925	0.6642	0.6041	0.5644	0.6487	0.5493	0.7425
GoogLeNet	0.6307	0.7056	0.6876	0.6088	0.5363	0.5579	0.5990	0.7824
VGGNet-16	0.6281	0.7084	0.6502	0.5896	0.5103	0.6556	0.5100	0.7516
ResNet-50	0.7069	0.8141	0.7362	0.6128	0.5609	0.7164	0.6333	0.7891
MobileNet(Project Result)*	0.67	0.73	**0.75**	**0.63**	**0.58**	0.48	0.56	0.65

MobileNet(Project Result)$\hat{*}$- The results have been obtained on 5% of the total dataset.

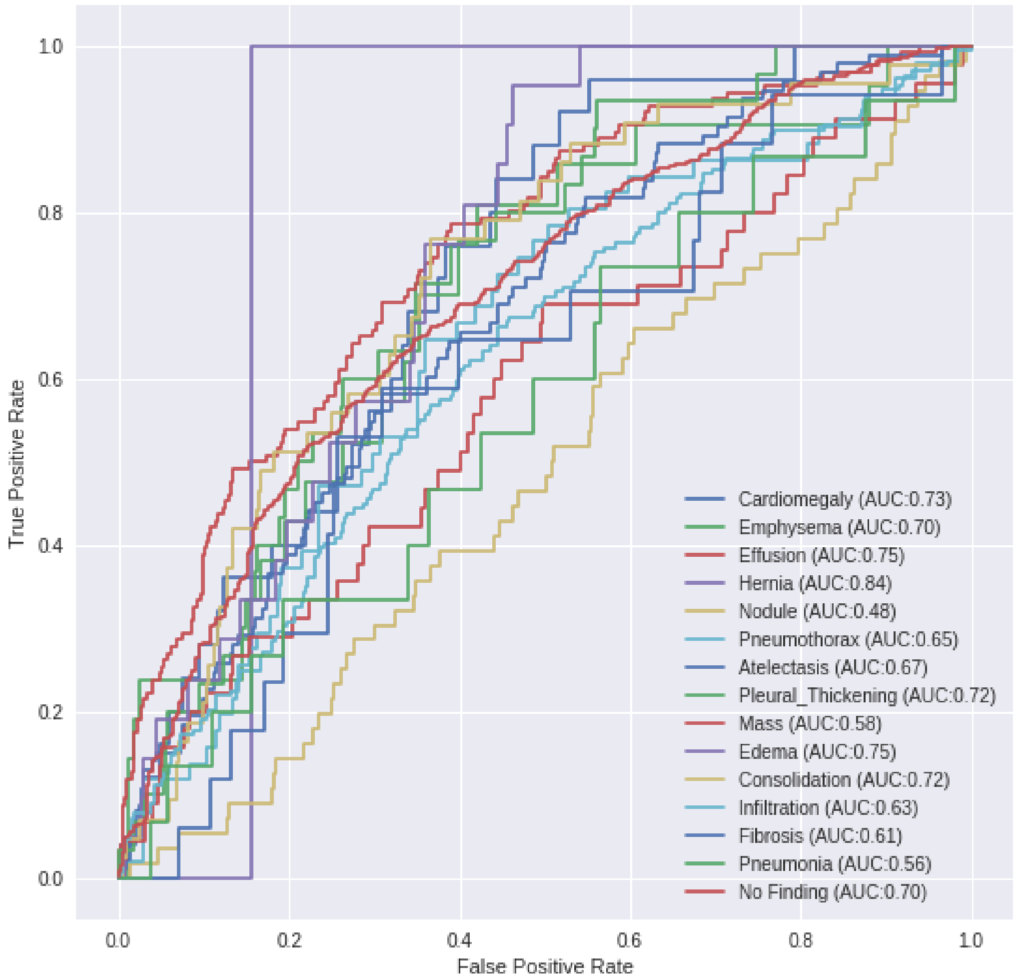


Figure 5.1: ROC curves for different diseases & area under them

Setting	Consolidation	Edema	Emphysema	Fibrosis	Pleural Thickening	Hernia	No Finding
AlexNet	NA	NA	NA	NA	NA	NA	NA
GoogLeNet	NA	NA	NA	NA	NA	NA	NA
VGGNet-16	NA	NA	NA	NA	NA	NA	NA
ResNet-50	NA	NA	NA	NA	NA	NA	NA
MobileNet(Project Result)*	**0.72**	**0.75**	**0.70**	**0.61**	**0.72**	**0.84**	**0.70**

NA- Not Available/Not Tested

Chapter 6

Conclusion & Future Work

During the research period of our project, we studied the prevailing research and development on diagnosis of Chest X-rays. From the research carried out during the 6th semester of our undergraduate course, we learnt a lot about Neural Networks and specially about convolution neural networks. We undertook courses outside our curriculum to gain more information about the convolution networks, like the deeplearning.ai course on Andrew Ng on coursera, Introduction to deep learning on udacity.com, Deep Learning by Santdex, etc. We started this project with the motive of improving accuracy and runtime performance of disease detection in Chest X-Rays. The method we propose is to use MobileNet with Data Augmentation so that it can learn more effectively at train time and run efficiently at run time. The model is lightweight so that it can be installed on handheld devices which can be used in remote areas which have minimum resources. Our secondary motive was to dilute the mixed opinions of Radiologists in very rare and hard to diagnose cases, as our model can be used to provide an assistance to them or the patient himself. With the availability of more computation resources we would like to train our model on the entire dataset. Moreover we can ensure that the model stays accurate with the advent of new diseases or change in the pattern of existing diseases by collecting labelled data from the front-end of the implementation of our model via reputed and trustworthy sources like experienced radiologists or a group of radiologitsts.

References

[1] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R. M. (2017). ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.369,

[2] Tanbakuchi, A. (2016, February 21). Comparison of OpenCV Interpolation Algorithms. , http://tanbakuchi.com/posts/comparison-of-openv-interpolation-algorithms/

[3] National Institutes of Health Chest. (2018, February 21), https://www.kaggle.com/nih-chest-xrays/data

[4] OpenCV: Histograms - 2: Histogram Equalization. (n.d.)., https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html

[5] Scikit-learn. (n.d.)., http://scikit-learn.org/stable/index.html

[6] Convolutional neural network., https://en.wikipedia.org/wiki/Convolutional_neural_network

[7] Gao, H. (2017, August 7). A Walk-through of AlexNet - Hao Gao - Medium., https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637

[8] Das, S. (2017, November 16). CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more., https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

[9] Gao, H. (2017, August 15). The Efficiency of Densenet - Hao Gao - Medium., https://medium.com/@smallfishbigsea/densenet-2b0889854a92

[10] [1608.06993] Densely Connected Convolutional Networks. (n.d.)., https://arxiv.org/abs/1608.06993

[11] MobileNets: efficient convolutional neural networks for mobile vision applications. (n.d.)., https://arxiv.org/pdf/1704.04861.pdf

[12] Emanuele Pesce, Petros-Pavlos YpsilantisLearning to detect chest radiographs containing lung nodules using visual attention networks https://arxiv.org/pdf/1712.00996.pdf

[13] Stanford Machine Learning, https://stanfordmlgroup.github.io/projects/chexnet/