



A Deep learning approach was used to automate the diagnosis of diseases from a chest X-ray

Arjun Choudhary¹, Dr. Kalpna Sharma², Dr. Prakash Choudhary³

¹Research Scholar, Department of Computer Science, Bhagwant University, Ajmer, Rajasthan-305001, India.

²HOD & Assistant Professor, Department of Computer Science and Engineering, Bhagwant University, Ajmer, Rajasthan-305001, India.

³HOD & Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology Hamirpur, Himachal Pradesh, Hamirpur, HP-177005, India.

Abstract: Pathology The diagnosis of pathology in a chest X-ray is generally complicated, even for experienced practitioners. A system that can automatically diagnose the findings in images obtained through X-rays of the chest can be useful in the medical examination of the patient as there is a shortage of experienced doctors. Classifying the chest X-ray is a multi-label classification task as a patient may have multiple diseases. In this research, we aim to develop an algorithm using deep learning techniques to identify the condition in the chest X-ray with high accuracy. In this research, we fine-tuned a pre-trained CNN architecture named DenseNet-121 to extract the features from the chest X-ray and to classify the extracted features into the pathology. The weights of the model are initially set with the weights of a model that is trained on ImageNet. Then the model is trained on a sample of the "ChestXray14" dataset.

Keywords: Chest X-ray; CNN; CAD; DenseNet.

1. INTRODUCTION

Chest X-ray is the most common and currently the best available method for medical imaging, which is an increased demand for diagnosis and treatment. For detecting lung disease chest X-rays plays a big role. There are various types of chest diseases like asthma, Chronic Obstructive Pulmonary Disease (COPD), Chronic Bronchitis, Pneumonia, tuberculosis and many other diseases that can be examined with the help of chest X-rays by experienced radiologists. The Chest X-rays can detect cancer, infection in a lung, which causes lungs to collapse. They shows chronic lung conditions, such as emphysema as well as complications related to these conditions.

Sheffield et al. [1]. Proposed that there are more than 65 million of peoples in the world which suffers from Chronic Obstructive Pulmonary Disease. The most common deadly disease is tuberculosis or TB and more than 10 million of peoples suffer from TB, and 1.6 million people die every year. In the world, 1.6 million people die from lung cancer each year, which is the most common neoplasm. Most children are affected by pneumonia and asthma. It kills millions of people yearly.

All over the world, nearly 4 million deaths are caused by chronic respiratory diseases. Exposure to toxic smoke is also dangerous and has a hazardous effect on human health. About 2 billion people come into contact with indoor toxic smoke and around a billion with tobacco smoke. In addition to this, inhalation of outdoor pollutants has a harmful effect on the respiratory system of the human body. Because of industrialization, more people are exposed to polluted air, as a result of which more people are likely to develop respiratory diseases.

CAD (Computer-Aided Design) can help in diagnosing patients quickly. In recent years, computer vision has come a long way because of the enhancements in deep learning algorithms and improvements in computational power. CNN has come up as one of the best ways to deal with the image classification problems. CNN has proved good at learning local and global structures in an image. In study, we will be using CNN to examine the chest X-ray. Task of diagnosing the disease in a chest X-ray can be divided into three parts: preprocessing, feature extraction and classification.



The basic diagram is shown below in figure No. 1.

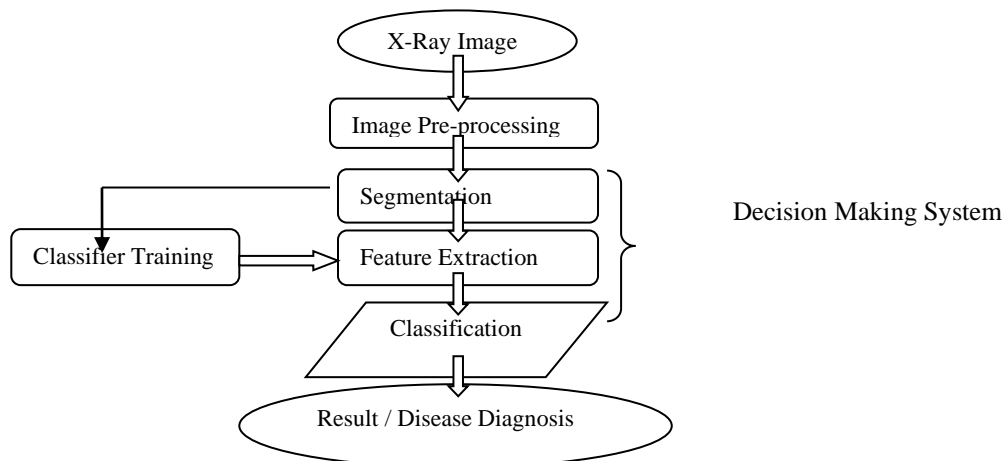


FIG.1. THE BASIC BLOCK DIAGRAM OF CAD

The preprocessing phase includes resizing of the image, improving contrast, removing noise, flipping of the images, and some other techniques.

With feature extraction, we will fine tune a pre-trained CNN architecture. A CNN uses convolution layers, max pooling layers, activation functions, and dropout for extracting the features from an image.

For classification of the extracted features, we will use an artificial neural network. ANN takes the flattened output of the CNN as an input and passes it through several hidden layers and into the output layer.

The number of healthy people is much higher than the number of diseased people, so the number of samples of healthy X-rays is higher than the number of diseased X-rays. Furthermore, there is a difference in the number of patients within the diseases, as some conditions are rare compared to others. The development of the system for CAD of chest X-rays poses the problem of an imbalanced dataset. Training a model with imbalanced data has many disadvantages. Suppose a model learns from a dataset that has 95% of the samples of class 1 and 5% of class 2. The model will overfit on class 2, and the model will always predict class 2 as an output no matter what input is given. To address this issue, we have studied various ways of handling the class imbalance and have chosen cost-sensitive learning for this research. The basic concept of cost-sensitive learning is to penalize CNN for wrongly classifying the samples of the minority class.

2. RELATED WORK OR SURVEY

Many types of research have been performed around medical imaging and CAD in recent years. Advancement in deep learning algorithms and architectures has accomplished great feats in the tasks related to computer vision. In particular, the CNNs have outperformed all of the other techniques in computer vision and enabled computers to process, evaluate images and extract details similar to the human mind. Research in the field of CAD has been used by CNN to their advantage and achieved good results. The existing work on automatic disease detection in chest X-rays has also shown that transfer learning is a very handy technique while training a model on a small dataset.

Stefan Jaegar et al. [2]. proposed a method for automatically diagnosing tuberculosis (commonly known as TB) in chest radiographs. In this method, they first identified the lungs using the cut segmentation method, in which they first used a lung model representing the shape of a normal lung, and after that, the graph approach was applied, in which the lung area has to satisfy certain conditions to be considered as a lung region. Then, from the identified region, features (shape and texture) are extracted. Two types of feature sets, which are CBIR-based image features and object detection-inspired features, were obtained separately. With the use of extracted features, the chest X-ray is classified as positive or negative for TB. The researchers used SVM for the classification of the features extracted from the images.

Abhishek Hazra et al. [3]. experimented with designing a method to automatically diagnose the disease in a chest X-ray using CNN. In this study, the researchers compared the performance of a pre-trained on ImageNet and, after that, a fine-tuned on the chest X-ray dataset. The custom model proposed by the researcher was trained on the dataset of 5606 images and achieved an accuracy of 83.67%. The VGG16 model achieved better accuracy than the custom model on the sample of the chest X-ray 14 dataset. A Softmax function was used at the end of the fully connected layer, which converts the logits into the probability distribution among the classes. In the trained model, if an image is fed as an



input, the model shows the top 3 diseases by probability. The researcher used Grad CAM for the visualisation and analysis of the prediction results.

Yaniv Bar et al. [4] proposed an automated method for diagnosing a chest X-ray for diseases. They used a dataset of 443 images, which had three types of disease classes. In the proposed method, they used a unique technique of pre-training by fusing the features of a decaf CNN and GIST features. They proposed that the combination of the features results in a much better classification as the combination of the features would capture the hidden features. The results were improved a little more by the late fusion of the Decaf5, Decaf6 and GIST descriptions. The method achieved a mean AUC of 0.87 to 0.94 for different disease classes.

Gao Huang et al. [5] performed a detailed study on densely connected convolutional neural networks and proposed a new type of neural network architecture named DenseNet. The paper proposes that the CNN can be made much deeper if the connections between the layers close to the input and the layers close to the output are made shorter. DenseNet exploits this fact by connecting each layer to every other layer in a feed-forward manner. It proposes to concatenate the outputs of the previous layers. DenseNets have many plus points over the existing CNN architectures: mitigation of the vanishing gradient problem, improved flow of information, and a reduction in the number of parameters. The performance of the network was evaluated using the image recognition tasks using SVHN, CIFAR-10, CIFAR-100, and ImageNet datasets. The deeper neural networks have a problem with vanishing gradients as due to a large number of layers, the gradientishes out by the time it reaches the end or the beginning. The dense CNNs solve this problem. " The layers in DenseNet are narrow as they add very small numbers of features and do not change the existing feature maps. A classifier is used to classify the feature maps into a class. A densely connected CNN is divided into dense blocks and has a transition layer to connect the dense blocks. The DenseNets outperformed the ResNets in all of the experiments, despite having a lower number of parameters. The DenseNets are also less prone to the problem of over-fitting.

Ankita Shelke, et al. [6]. It is proposed that chest X-ray images be obtained after obtaining from the X-ray screening machine be passed through a VGG-16 model. The results are bracketed as normal, pneumonia, or tuberculosis. The Pneumonia images are then classified using a DenseNet-161 model as either normal Pneumonia or COVID-19. Then, the COVID-19 X-Ray images are passed through a ResNet-18 model and categorized as severe, medium, or mild.

Subrato Bharati, Hossain Mondal, et al. [7]. The basic CNN has poor performance for rotated, tilted, or other abnormal image orientations. Therefore, we propose a new hybrid deep learning framework by combining VGG, data augmentation, and spatial transformer network (STN) with CNN. This new hybrid method is termed here as VGG Data STN with CNN (VDSNet).

3. PROBLEM STATEMENT AND PROPOSED METHOD

In this work, various methods were used, which are explained below in each sub-section. Deep visual recognition CNN architectures based on supervised learning are made up of multiple convolutional stages stacked on top of each other to learn hierarchical visual features [8, as shown in Figure 1].Regularization approaches such as stochastic pooling, dropout, and data augmentation have been utilised to enhance the recognition accuracy. Recently, the faster convergence of these architectures has been attributed to the inclusion of Rectified Linear Units [ReLU] nonlinearity into each of the layers with weights.

3.1 Platform and framework for disease diagnosis simulation on Chest X-ray: Scikit-learn, keras, numpy, pandas, and matplotlib are all used by the Anaconda 3 IDE platform tool.

3.2 Data Availability: The data used in this study is the third-party ChestX-Ray 14 dataset and is publicly hosted by the National Institutes of Health Clinical [9].

3.3 Google Colab: Google provides a free cloudservice Colaboratory also known as GoogleColab that is based on Jupyter notebook environment. It also provides hardware accelerators like GPU and TPU for faster execution. This enables developers and researchers to develop deep learning projects which require high computational power. Colab supports python2 and python3 programming language. We use Python3 to program the proposed algorithm. Google drive can also be mounted on colab to provide the permanent storage as the sessions in colab resets every 12 hours, whenever the session resets all the local variables and the local data is lost so we have to save it in google drive. We save our dataset in google drive and whenever we want to access the dataset we need to mount the google drive. We can also save the notebooks in drive. In this research we use google colab for training, validating and testing the model.

3.4 Python libraries used in simulation: The major libraries needed in the simulation of this research work are explained in this section.

Keras: It is useful for working with machine learning and implementing neural network models. It is an open-source library written in Python. It is designed to be easy to use, extensible and modular. It is able to run on top of various



machine learning libraries such as Tensor Flow, R, Microsoft Cognitive Toolkit, PlaidML, and Theano. Keras provides high-level abstractions, which makes it much easier to use and much easier to understand. For example, To add a convolutional layer to our network, we just need to use the function convolution 2D and pass the parameters such as filter size, number of filters, stride, etc. But we don't need to programme the entire functionality of the convolutional operation. This is what makes Keras so popular. Keras also provides popular CNN architectures such as VGG16, ResNet, DenseNet, and many more with their pre-trained weights on "ImageNet". In this research, we use Keras to build, compile, train, and test our CNN.

Scikit-image: It is an open-source library built for image processing. We have used this library mainly for preprocessing the images. The main reason for using this library is to apply CLAHE to the chest X-ray images.

This library provides support for large and multidimensional arrays and also provides a variety of complex mathematical operations to perform on these arrays. And as we know that CNN includes a lot of multidimensional arrays, NumPy becomes an irreplaceable library.

Pandas: This library is built for the manipulation of data and analysing data. In this research, the dataset used has a csvfile which contains the details about the dataset, such as image index and the pathology of the images. Pandas play a crucial role in the preprocessing of the dataset; it is helpful for reading the images from their locations.

3.5 Simulation Specifications

In this section, we will discuss the parameters of the different techniques that are used in this research. This section also elaborates on the implementation details of the research.

3.6 Chest X-ray Image Dataset: The "ChestXray14" dataset was released by the NIH. A sample of this dataset was used in this research. The sample dataset consists of 5,606 labelled chest X-ray images. A chest X-ray can reveal multiple diseases. Figure 3 shows the number of samples in each class. The dataset consists of chest X-rays belonging to 14 diseases. The names of the diseases are: Pneumonia, Fibrosis, Edema, Cardiomegaly, Atelectasis, Emphysema, Pneumothorax, Infiltration, Consolidation, Effusion, Mass, Hernia, Nodule, and Pleural Thickening. In figure 4.4, it can be easily seen that the number of samples belonging to each class varies by a large margin. This kind of dataset is called a dataset with imbalanced class distribution.

No. of images vs. Disease

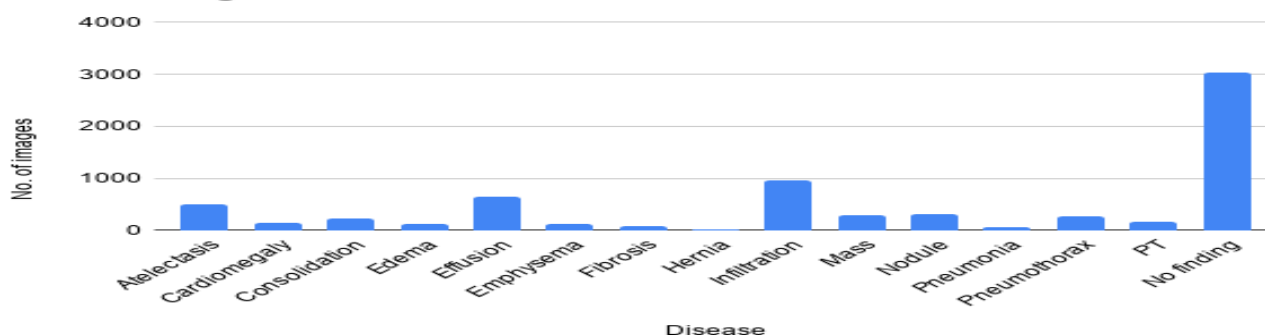


FIG. 2. DISTRIBUTION OF THE SAMPLES AMONG DIFFERENT DISEASES.

Pre-processing of the Dataset

Preprocessing of the data plays an integral part in the performance of a trained model. In this phase, the data gets transformed and encoded into a format that the computer can process. In this phase, we have covered the operations performed on the dataset before feeding it into our model.

Pranav Rajpurkar et al. [10] developed and validated a deep learning algorithm that classified clinically important abnormalities in chest radiographs at a performance level comparable to that of practising radiologists. Once tested prospectively in clinical settings, the algorithm could have the potential to expand patient access to chest radiograph diagnostics.

Yu Xing Tang, You Bao Tang et al. [11] demonstrate the effectiveness of deep convolutional neural networks in classifying normal and abnormal chest radiographs. A single best convolutional neural network trained on a moderate-sized (approximately 8500 radiographs) dataset with moderate image resolution (256 256 pixels) achieves a high AUC of 0.98, with a sensitivity of 96.50% and specificity of 92.86% (i.e., ResNet18).

Ebrahim Mohammed Senan et al. [12]. want to help doctors and radiologists detect patients with COVID-19 using deep-learning techniques. Two pretrained deep-learning modeling algorithms, namely, ResNet-50 and AlexNet, were used to extract the most important distinguishing features from X-ray images. In training discriminative DL models, data augmentation has been shown to be effective. Examples of data augmentation techniques are flipping, rotating,

colour jittering, random cropping, elastic distortions, and generative adversarial network (GAN) based synthetic data generation.

T Rahman, MEH Chowdhury, A Khandakar, KR Islam et al. [14]. The comparative performance of training and testing accuracy for different CNNs for three classification schemes shows that DenseNet201 is producing the highest accuracy for both training and testing. For normal and pneumonia classification, the test accuracy was 98%, while for normal, bacterial, and viral pneumonia classification, it was 93.3%, and for bacterial and viral pneumonia classification, it was found to be 95%.

The motivation behind this work was to help doctors and radiologists detect patients with COVID-19 using deep-learning techniques. Two pretrained deep learning algorithms, ResNet-50 and AlexNet, were used to extract the most important distinguishing features from X-ray images.

(i) Greyscale to RGB conversion

We have used transfer learning in this research, and the model is pre-trained on the ImageNet. The ImageNet contains RGB images. The images in the "ChestXray14" dataset are in grayscale format, i.e., they have a single channel, but our model is designed to classify pictures with three channels, so we need to convert the grayscale image into an RGB image. To do so, we have used a simple method in which we have just duplicated the single-channel values into three channels to convert images into RGB format.

(ii) Image data enhancement

Image data augmentation applies various transformations to images, such as flipping, shifting, zooming, etc. Image data augmentation aims to feed different variations of the training set.

(iii) Dividing the dataset into sections

The dataset that we have used in this research has 5,606 labelled images of the chest X-rays. The dataset is split into three parts: training data (70%), validation data (validation data) (10%), and test data (20%). Almost 4 thousand images are used for training the model. The validation data is used for selecting the most appropriate model. The test set is used to evaluate the performance of the trained model.

4. CONVOLUTIONAL NEURAL NETWORK (CNN)

CNNs are very effective at utilising spatial and temporal dependencies, which makes CNNs optimal for images. Because in pictures, the position of the pixels is equally important as their values. Primary CNNs are built using convolutional layers, pooling layers, activation functions, and fully connected layers. A sequence of the above layers can be used to create a CNN architecture. The convolutional layer can be said to be the core of a CNN. The convolutional layer has learnable filters (kernels), which are also known as the parameters of the convolutional layer. A filter has three dimensions (height, width, and depth). At the time of forward pass, the filter is kept at a pixel and the dot product of the pixel values and filter is taken, and the result of element wise dot product is replaced at the centre pixel. This process is done at all the pixels, which gives a 2D feature map as an output. Padding is often used to keep the dimensions unchanged. Fig. 3.

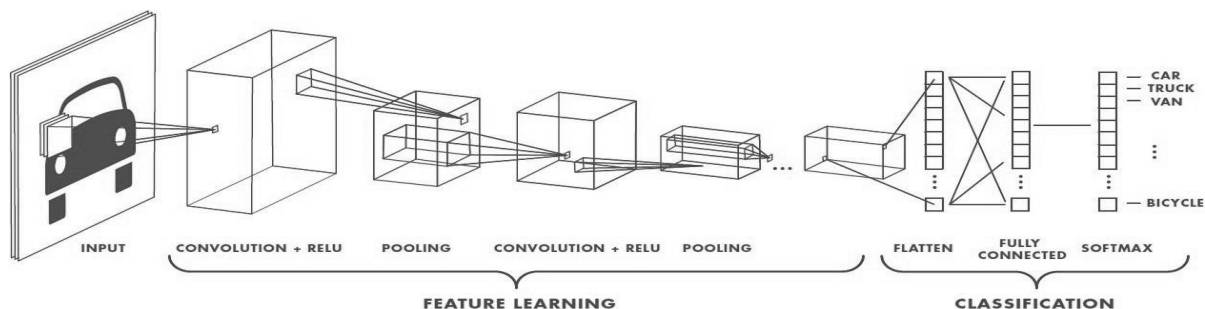


FIG. 3. CONVOLUTION OPERATION WITH RELU ACTIVATION.

G Prasanth, R Sivarama krishnan, and L Rodney [12]. Proposed that data augmentation has been proven to be effective in training discriminative DL models. Examples of data augmentation techniques are flipping, rotating, colour jittering, random cropping, elastic distortions, and generative adversarial network (GAN) based synthetic data generation.



Convolution Layer: Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs, such as an image matrix and a filter or kernel. Then the convolution of the 5 x 5 image matrix is multiplied with a 3 x 3 filter matrix, which is called the "Feature Map" as an output. Convolution of an image with different filters can perform operations such as edge detection, blurring, and sharpening by applying filters. The various convolution images after applying different types of filters (kernels)

A pooling layer is used in a convolutional neural network to cut down the size of the output of the convolutional layer. This is used after convolutional layers, which causes a reduction in the number of parameters, and this decrease leads to a rise in the computation efficiency. The pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling, also called subsampling or downsampling, reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pool. The sum of all the elements in the feature map is called "sum pooling."

Strides: Stride is the number of pixels shifted over the input matrix. When the stride is 1, then we move the filters to 1 pixel at a time. When the stride is 2, we move the filters to 2 pixels at a time and so on.

Sometimes the filter does not perfectly fit the input image. We have two options:

- Pad the picture with zeros (zero-padding) so that it fits.
- Drop the part of the image where the filter did not fit. This is called "valid padding," which keeps only the valid parts of the image.

The Fully Connected Layer: The layer we call the FC layer, we flattened our matrix into a vector and fed it into a fully connected layer like a neural network.

In the above diagram, the feature map matrix will be converted into a vector (x1, x2, x3,...) With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cancerous, non-cancerous, diseased, normal, etc.

The convolutional layer has four parameters:

- (i) Number of filters: This is determined by the size of the required output.
- (ii) Step: It defines how many steps a filter moves after each dot product.
- (iii) Zero padding amount: Determines whether the size will be reduced or kept the same.
- (iv) Filter sizes

Non-Linearity (ReLU): The ReLU function keeps the positive values as is and changes the negative values in the feature map to 0. ReLU is computationally efficient and also makes the CNN converge faster. Batch normalisation is also used in CNN to normalise the values. Normalization is done according to the statistics of the mini batch. It has certain advantages: it decreases the chance of getting large values; the chance of overfitting is reduced; the learning rate can be increased, making the model learn faster. Activation functions are used in CNN after the convolutional layer. They are used to bring the values in the feature map between ranges. There are a wide variety of activation functions available, such as sigmoid, tanh, gaussian, rectified linear unit (ReLU), etc. Among these activation functions, the most commonly used activation function after the convolutional layers is ReLU. Rectified Linear Unit is used in non-linear operations. The output is $(x) = \max(0, x)$.

ReLU's purpose is to introduce non-linearity into our ConvNet. Because the real-world data we want our ConvNet to learn is non-negative linear values. There are other non-linear functions such as tanh or sigmoid that can also be used instead of ReLU. Most data scientists use ReLU since, performance-wise, ReLU is better than the other two.

5. DENSENET

The DenseNet is considered a logical extension of the famous architecture named ResNet. It uses the concept of concatenation of the outputs of the previous layers. DenseNet is gaining popularity because of its improved flow of information (and gradients), reuse of feature maps, and a lower number of parameters. This makes the training computationally efficient, and it is also less prone to overfitting.

In this research, we have used the DenseNet-121 architecture as the feature extraction unit. We have finetuned DenseNet-121, which was initially trained on ImageNet. In this section, we will discuss the features of DenseNet in detail and explain the architecture of DenseNet. DenseNet introduces "Dense Connections," which improve the flow of

gradients and information within the same network. In the dense connection, every layer is directly connected to all of the following layers. Figs. 4 and 5 show how dense connections work.

The dense connections model requires fewer layers, as there is no need to learn redundant feature maps, allowing the collective knowledge to be reused. The proposed architecture has narrow layers, which provide state of the art results for as low as 12 channel feature maps. Fewer and narrower layers mean that the model has fewer parameters to learn, making it easier to train. The input layers are concatenated as a result of concatenated feature maps, which prevents the model from over-fitting the training data. I have explained the concepts with their standard network (DenseNet-121).

(i) Dense layer

Each dense layer consists of two convolutional operations:

- **1 X 1 CONV** (conventional convolutional operation for extracting features)
- **3 X 3 CONV** (bringing down the feature depth/channel count)

The DenseNet-121 comprises six such dense layers in a dense block. The depth of the output of each dense layer is equal to the growth rate of the dense block.

(ii) Transition layer

At the end of each dense block, the number of feature-maps accumulates to a value equal to the input features + (number of dense layers x growth rate). So, for 64 channel features entering a dense block of 6 dense-layers of growth rate 32, the number of channels accumulated at the end of the block will be $64 + (6 \times 32) = 256$. To reduce this channel count, a **transition layer** (or block) is added between two dense blocks. The transition layer consists of:

- **1 x 1 CONV** operation.
- **AVERAGE POOL OPERATION 2 X 2**

The 1 x 1 CONV operation reduces the channel count by half.

The 2 x 2 AVG POOL layer is responsible for downsampling the features in terms of their width and height.

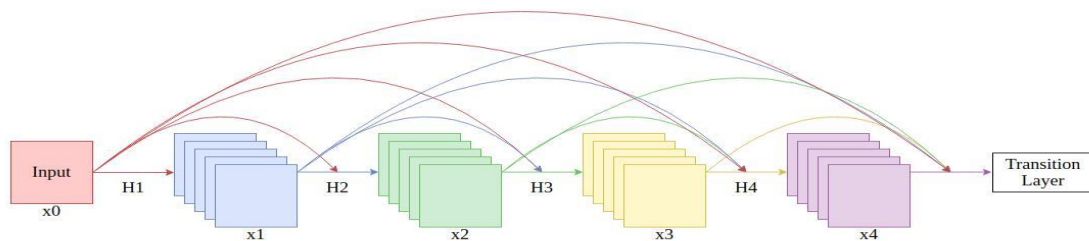


FIG. 4. CONNECTIONS BETWEEN DENSELY CONNECTED LAYERS.

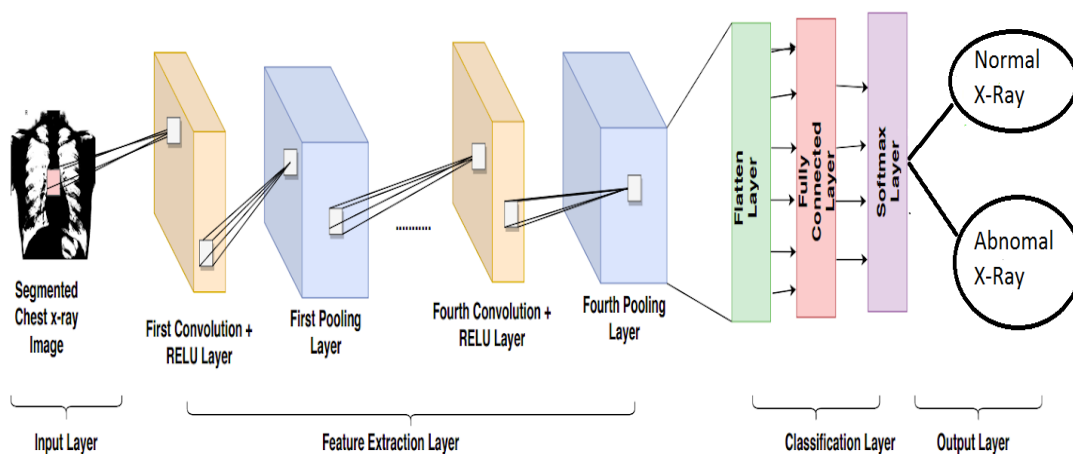


FIG. 5. CNN ARCHITECTURE OF DEEP LEARNING



6. METHODS AND MATERIALS:

Supervised training, testing and validation is used.

6.1 Training the model: Architecture of the model and details of CNN used is given in the table below.

Table 1. Details of CNN used in this research.

Layers	Output size	Layer Details	Trainable
Convolution Layer	112*112	7*7 conv, stride 2	False
Pooling Layer	56*56	3*3 max pool, stride 2	False
Dense Block (1)	56*56	[1*1 conv + 3*3 conv]*6	False
Transition Layer (1)	56*56	1*1 conv	False
	28*28	2*2 average pool, stride 2	False
Dense Block (2)	28*28	[1*1 conv + 3*3 conv]*12	True
Transition Layer (2)	28*28	1*1 conv	True
	14*14	2*2 average pool, stride 2	True
Dense Block (3)	14*14	[1*1 conv + 3*3 conv]*24	True
Transition Layer (3)	14*14	1*1 conv	True
	7*7	2*2 average pool, stride 2	True
Dense Block (4)	7*7	[1*1 conv + 3*3 conv]*16	True
Classification Layer	1*1	7*7 global average pool	True
		Fully connected layer Sigmoid	True

The above table shows the details of the layers of the CNN architecture that we have used in this research. The details of the dense block and transition block are explained in the previous chapter.

7. RESULTS AND DISCUSSION

In this section, we will discuss the parameters of the different techniques that are used in this research. This section also elaborates on the implementation details of the research. In his section, we will discuss the parameters of the different techniques that are used in this research. This section also elaborates on the implementation details of the research.

The "ChestXray14" dataset was released by the NIH. A sample of this dataset was used in this research. The sample dataset consists of 5,606 labelled chest X-ray images. A chest X-ray can reveal multiple diseases. The dataset consists of chest X-rays belonging to 14 diseases. The names of the diseases are: Pneumonia, Fibrosis, Edema, Cardiomegaly, Atelectasis, Emphysema, Pneumothorax, Infiltration, Consolidation, Effusion, Mass, Hernia, Nodule, and Pleural Thickening.

Preprocessing: In preprocessing, we have first resized the images to 224*224 resolution. After resizing the images, we have applied image data augmentation, in which the images are flipped horizontally and zoomed. For image augmentation, we have used the Image Data Generator class of the Keras library with parameters set to "horizontalflip=True", "zoom range=0.2". The preprocessing phase also includes applying CLAHE (for local contrast enhancement) to the X-ray images. Fig. 6. shows the result of applying CLAHE to a chest X-ray image. For applying CLAHE, we have used the equalise adaphthist class of the scikit-image library with the parameters:

(1) kernel size = 28*28. The kernel size represents the size of the region of the image which comes under one histogram.

(2) clip limit = 0.05. The clip limit controls the amount of contrast and higher values mean more contrast.

(3) nbins = 256. nbins shows the number of bins in a histogram among which the equalisation occurs.

In Fig. 3, we can see that there is an enhancement in the visual quality of the chest X-ray image.

Preparing the dataset for CNN: We have split the dataset randomly into three parts: training data, validation data, and testing data.

Training data is 70%, validation data is 10%, and test data is 20% of the total im-ages. For splitting the dataset, we have used the train test split method of the SKELARN library. After splitting the dataset, we have created three separate data generators (for training, validation, and testing) using the data from the csv file which contains the mapping from the image indices to the disease labels.

Training: For fine tuning the pre-trained model, we feed the training data generator of pre-processed images into the model, monitor the performance of the model on the validation data generator based on certain metrics, and choose the

best model according to the values of the metrics. Finally, we check the performance of the selected model on the test data generator and get the results.

We use class weights, the range of the loss changes. The optimizer like SGD (stochastic gradient descent), whose step size relies on the gradient, may tend to fail. So, for this classification task, we used the Adam optimizer.

The metrics on which we evaluate our model are:

$$(i) \quad \text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

(ii) **Loss:** It tells the difference between the predicted value by the model and the original value of the sample. If it is 0 then the model is perfect higher values mean more deviation from the actual labels.

(iii) **AUC:** It tells us about the area enclosed under the ROC (receiver operating characteristic) curve. ROC curve is a graphical plot between true positive rate (TPR) and false positive rate (FPR).

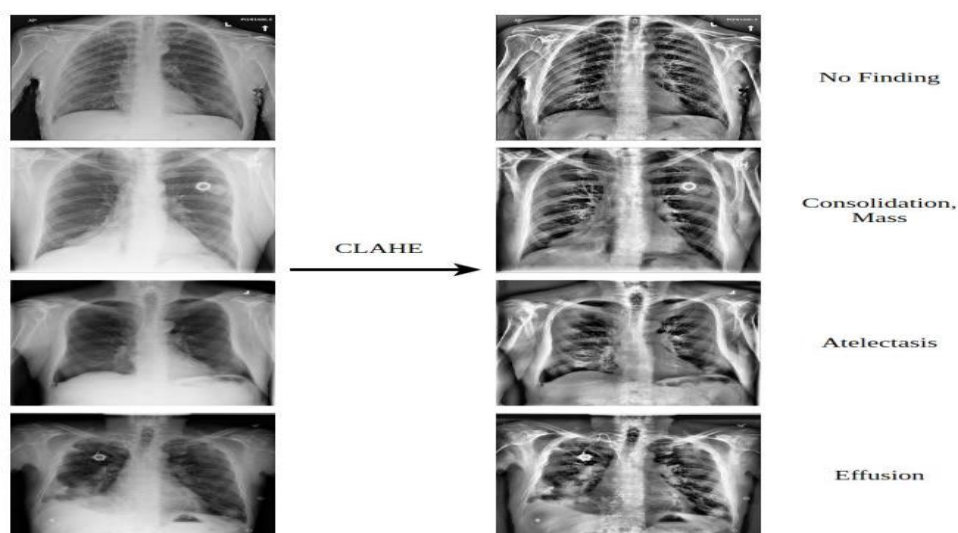


FIG. 6. EFFECTS OF APPLYING CACHE IN CHEST X-RAY.

The values of AUC vary from 0 to 1, higher the value of AUC means a better model. We have trained all the 6 models on the sample dataset of ChestX-ray14 for 29 epochs with the default learning rate of Adam optimizer and tracked the progress of the model over 29 epochs on the basis of accuracy, loss and auc. The following table shows the values of the metrics of the models after training each of them for 29 epochs.

Table.2 shows the various metrics over training and validation. The accuracy and auc on the test data shown in Table.3.

TABLE. 2. THE VALUES OF VARIOUS METRICS OVER TRAINING AND VALIDATION.

Metric / Model	1	2	3	4(a)	4(b)	4(c)
Training accuracy	92.57	88.83	90.37	90.45	90.14	91.26
Training loss	0.1768	1.6984	0.2554	0.4050	0.4099	0.5343
Training auc	0.8202	0.6337	0.6283	0.7976	0.7993	0.7923
Validation accuracy	92.60	88.24	89.99	90.46	90.16	91.22
Validation loss	0.2223	1.7335	0.3138	0.2619	0.3007	0.4052
Validation auc	0.8211	0.6301	0.6298	0.7983	0.7964	0.7923

TABLE. 3. ACCURACY AND AUC ON THE TEST DATA.

Metric / Model	1	2	3	4(a)	4(b)	4(c)
Testing accuracy	93.39	90.57	91.13	90.55	88.61	74.96
Testing AUC	0.8269	0.6535	0.6526	0.8121	0.7540	0.6983



8. CONCLUSION

This research shows how the deeper CNNs help with better feature extraction from the chest X-rays. CLACHE is used for the local histogram of the X-ray images. CLACHE makes images clearer, and it would be very helpful for the doctors. CAD has made tremendous progress, but the scope of CAD in chest X-rays remains limited because of the scarcity of large and well-labeled X-ray datasets. Another big problem is that the performance of CAD is limited as compared to practitioners. The CAD systems do not know the medical history of the patient, whereas the practitioners get help from the patient's old records. DenseNet is a network that portrays the importance of having dense connections in a network using dense blocks. This helps with feature-reuse, better gradient flow, reduced parameter count, and better transmission of features across the network. Such an implementation can help in training deeper networks using fewer computational resources and with better results.

ACKNOWLEDGMENTS

The authors would like to thank the Editor-in-Chief and anonymous referees for their suggestions and helpful comments that have improved the papers quality and clarity.

REFERENCES

- [1] Sheffield, European Respiratory Society, (2017): The Global Impact of Respiratory Disease – Second Edition Forum of International Respiratory Societies Print ISBN:9781849840873; e-ISBN:9781849840880.
- [2] Jaegar, S, et al. (2013): Automatic screening for tuberculosis in chest radiographs: a survey. *Quantitative imaging in medicine and surgery*, 3(2):89, 2013.
- [3] Choudhary, P; Hazra, A, (2019): Chest disease radiography in twofold: using convolutional neural networks and transfer learning. *Evolving Systems*, pages 2097-2106, 2017.
- [4] Bar, Y, et al. (2015): Chest pathology detection using deep learning with non-medical training. In 2015 IEEE 12th international symposium on biomedical imaging (ISBI), pages 294–297, IEEE.
- [5] Huang, G, et al. (2017): Abnormality detection and localization in chest x-rays using deep convolutional neural networks. arXiv preprint arXiv:1705.09850.
- [6] Ankita Shelke, et al. (2021): Chest X-ray Classification Using Deep Learning for Automated COVID-19 Screening. DOI: 10.1007/s42979-021-00695-5.
- [7] Subrato Bharati and Hossain Mondal et al. (2020): Hybrid deep learning for detecting lung diseases from X-ray images. <https://doi.org/10.1016/j.imu.2020.100391>.
- [8] George E. Dahl, Dong Yu, Li Deng, Alex Acero, (2012): Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition in *IEEE Transactions on Audio, Speech & Language Processing* 20(1): 30-42.
- [9] <https://nihcc.app.box.com/v/ChestXray-NIHCC>.
- [10] Pranav Rajpurkar et.al. (2018): Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. <https://doi.org/10.1371/journal.pmed.1002686>.
- [11] Yu Xing Tang, You Bao Tang et.al. (2020): Automated abnormality classification of chest radiographs using deep convolutional neural networks. <https://doi.org/10.1038/s41746-020-0273-z>.
- [12] Ebrahim Mohammed Senan et. Al. (2021). Automated Diagnosis of Chest X-Ray for Early Detection of COVID-19 Disease. <https://doi.org/10.1155/2021/6919483>.
- [13] G. Prasanth, R. Sivaramakrishnan, and L. Rodney, (2019): Assessment of data augmentation strategies toward performance improvement of abnormality classification in chest radiographs, 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), p. 4, Berlin, Germany, 2019.
- [14] T Rahman, MEH Chowdhury, A Khandakar, KR Islam, (2020): Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray. <https://doi.org/10.3390/app10093233>.
- [15] Mooney P, Chest x-Ray Images (Pneumonia), 2018.
- [16] Simonyan K, Zisserman A, (2014): Very deep convolutional networks for large-scale image recognition, arxiv preprint arxiv 1409.1556.
- [17] Ioffe S, Szegedy C, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arxiv preprint arxiv 1502.03167 (2015).