IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 August 2024, accepted 2 September 2024, date of publication 4 September 2024,
date of current version 13 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3454537

RESEARCH ARTICLE

Multi-Label Classification of Lung Diseases Using Deep Learning

MUHAMMAD IRTAZA[1], ARSHAD ALI[1], MARYAM GULZAR[2], AND AAMIR WALI[1]
[1]FAST School of Computing, National University of Computer and Emerging Sciences, Lahore 54770, Pakistan
[2]SE Department, LUT University, 53850 Lappeenranta, Finland

Corresponding authors: Maryam Gulzar (Maryam.Gulzar@lut.fi) and Arshad Ali (arshad.ali1@nu.edu.pk)

ABSTRACT Assistance for doctors in disease detection can be very useful in environments with scarce resources and personnel. Historically, many patients could have been cured with early detection of the disease. The application of deep learning techniques in the fields of medical imaging, on large datasets, has allowed computer algorithms to produce as effective results as medical professionals. To assist doctors, it is essential to have a versatile system that can timely detect multiple diseases in the lungs with high accuracy. Over time, although many classifiers and algorithms have been implemented, however, deep learning models (i.e., CNN, Deep-CNN, and R-CNN) are known to offer better results. After a thorough literature review of the state-of-the-art techniques, this work applies various models such as MobileNet, DenseNet, VGG-16, EfficientNet, Xception, and InceptionV3 to the selected large dataset. The goal is to enhance the accuracy of these algorithms by experimenting with parameter optimizations. We observe that MobileNet produces better results as compared to other models. We implement a deep convolutional GAN to produce synthetic X-ray images containing various pathologies already included in the chosen imbalanced dataset namely NIH Chest X-ray containing 14 classes. The synthetic dataset contains 1193 samples belonging to five classes. We test the suggested model using evaluation measures like recall, precision, and F1-score, along with binary accuracy. The suggested deep learning model produces recall as high as 57%, binary accuracy as 93.4%, F1-Score as 0.553, and AUC as 81. After the inclusion of generated synthetic samples, the value of the F1-score becomes 0.582 resulting in a 5% increase. Though, Generative Adversarial Network (GAN) shows lower performance, however, we encourage further research and experiments to find the versatility of GANs in the field of medical imaging.

INDEX TERMS Lung disease, mobile-net, image-augmentation, GAN, class imbalance, multi-class classification.

I. INTRODUCTION

Health disorders and conditions affecting the lungs are referred to as lung diseases. Diseases such as pneumonia, asthma, tuberculosis, emphysema, malignancies in the lungs, and a few more make the lungs lose their versatility and hence decrease the overall volume of air. Around 2 million chest radiography images are used by doctors around the globe to examine various types of lung diseases. Diseases targeting the human chest, one major cause leading to death, essentially require this technology for diagnosis and treatment. Computer systems can be used for the interpretation of radiographs, as effectively as the actual radiologists, and for clinical support in health programs as well as chest disease diagnosis. This can prove to be an effective tool for countless clinical environments, where, for instance, sufficient medical staff is not available.

Lung diseases like pneumonia, emphysema, asthma, COVID-19, and others are very contagious with quite a high death rate. Pulmonary fibrosis is a long-lasting infection that is quite hard to diagnose because symptoms may appear at either the initial stage or after many years. It causes difficulty in breathing and as of now, it does not have any cure. Early diagnosis is a very crucial and challenging task because some diseases show only ignorable symptoms like common flu, cough, and fever. Therefore, lung disease diagnosis at early stages cannot be achieved via the symptoms only.

The associate editor coordinating the review of this manuscript and approving it for publication was Vishal Srivastava.

At the initial stages, Chest X-rays have proved to be very helpful in highlighting the infection status as they can show the early symptoms of such lung infections that can lead the human body to respiratory failure. CT scans are helpful for surgery and can also be beneficial for lung cancer detection or heart failure. However, X-rays are much more popular due to their cost-effectiveness and simplicity in capturing.

The advancement in the emerging deep learning techniques and models has proven quite beneficial in the diagnoses and detection of numerous lethal diseases more efficiently. Application of state-of-the-art deep learning techniques, on large datasets, has allowed computer algorithms to produce as effective results as medical professionals, in the fields of medical imaging tasks involving skin cancer classification [1], lymph node metastases [2], and diabetic retinopathy detection [3]. A keen interest of researchers, in the advanced methods for automatic detection of chest imaging [4], [5], has led to the development of such algorithms that can detect pulmonary nodules [6]. Nevertheless, it is suggested that there must be a system that can classify multiple pathologies, including pneumonia and pneumothorax.

A versatile and accurate system is required to classify multiple pathologies, including pneumonia and pneumothorax. The research is still in its early stages to find the models that provide efficient and accurate classification of lung diseases by examining various pathologies. This research work conducts an experimental analysis for the assessment and evaluation of deep learning-based state-of-the-art models on large datasets with multiple lung diseases, for classification. The deep learning models have been planned to train on a large chest x-ray-based image dataset having various classes for the early detection of fatal lung diseases. For the assessment purpose, evaluation measures named Binary Accuracy, Recall, Precision, F1 Score, and AUC were calculated to compare the results of recent research models. Furthermore, synthetic X-ray images, generated with GANs, to tackle the scarcity of medical X-rays are also tested. Our study introduces a groundbreaking novel approach to lung disease diagnosis through multi-label classification, enabling the simultaneous identification of multiple lung diseases within a single chest X-ray. This contrasts with traditional methods, which are limited to single disease detection. By focusing on enhancing the recall metric, crucial in the medical field, our method significantly improves diagnostic accuracy. Additionally, we leverage a DCGAN to address class imbalance by generating synthetic images for various disease classes, further enhancing the robustness and effectiveness of our diagnostic system. Although the GAN results were not as promising as expected, our research establishes a foundation for future exploration of generative AI in lung disease classification.

A. RESEARCH GOALS

- Implementation of data processing techniques such as normalization, and image preprocessing.
- Tackling the problem of class imbalance with the help of Generative Adversarial Networks (GANs).
- Implementation of Image Augmentation techniques like geometric transformation, colour transformation, and image enhancement to examine the effect of overfitting.
- Implementation of MobileNet, EfficientNet, InceptionV3, Xception, ResNet, and DenseNet.
- Examination of training time of Deep transfer learning-based CNN models to find an efficient model.
- Comparing the classification results of synthetic data, produced with GANs against the original data combined with geometric image augmentation.

B. RESEARCH QUESTIONS

- Can we improve the classification results, along with the efficiency of the algorithm for large datasets by using the latest deep transfer learning models?
- Can data generated with Generative Adversarial Networks produce better results than conventional geometric image augmentation techniques?
- What effects can image preprocessing techniques and hyper-parameters including model, batch size, input image resolution, and loss function, have on the classification fitness functions?

C. RESEARCH CONTRIBUTIONS

- A transfer-learning model, combining MobileNetV1 and a three-layered deep neural network classifier incorporating Geometric Image Augmentation, for multi-label classification of lung diseases. This model aims to achieve superior performance compared to existing approaches in terms of both accuracy and efficiency.
- A comprehensive analysis of data preprocessing techniques (e.g., normalization, threshold segmentation) and hyperparameter tuning for X-ray classification, with a focus on optimizing classification performance.
- Generation and evaluation of a synthetic X-ray images dataset belonging to five classes using deep convolutional Generative Adversarial Networks (DCGANs).

The remainder of the paper is structured as follows: In Section II, we describe the existing literature from different perspectives considered in this study. Section III presents methodology of the work. In section IV, we provide experimental settings and experimental results. Results are discussed in Section V. In Section VI, we conclude this article.

II. LITERATURE REVIEW

Since the widespread of contagious diseases such as COVID-19, the researchers have been keenly working to develop reliable and efficient methods, for disease detection and classification, to bring down the exposure of human resources to such outbreaks. hese papers involved different datasets named NIH ChestX-ray-14 [1], [7] [8], [9] [10], [11], ChestX-ray2017 [7], PLCO [8], ICBHI 2017 [12], Subregion demarked parenchymal-lung disease of ILD Dataset [13], Chexpert [14], COVIDx Dataset [15] JSRT dataset [16],

Montgomery dataset [16], Shenzhen Hospital data [17], and various deep CNN-based techniques like MobileNetV2 [1], [7] DenseNet-121 [8], ResNet-38 [10],ResNet-50 [10], [14],ResNet-101 [10], CNN [13], [14], [18], [19], VGG [12], [14], [17], SVM Classifier [12], GAN [15], [16] and InceptionV3 [14].

A. CNN

Convolutional neural networks (CNN) are probably the most popular deep learning model used for both segmentation [20] and classification of all kinds of data: image [21], text [21] and speech. Within lung disease classification, the use of CNN is also very common. The research work of [9] is solely based on the diagnosis of tuberculosis & lung cancer disease, which comes under thoracic diseases. This task is quite time-consuming and labour-intensive, leading to diagnostic errors. Usually, expert radiologists are required to analyze the images. Recent deep learning-based approaches are powered by huge network architectures and have been proven quite fruitful in medical imaging interpretation tasks. However, to obtain expert-level performance there is a need for a large amount of image-based labeled datasets, which is a bottleneck.

In 2018, Rajpurkar et al. [9] evaluated the deep learning-based models and investigated the problem of pathologies detection. For this problem, the pathologies in the chest radiographs were compared with the practising radiologists. A CNN-based model named CheXNeXt was developed to detect the 14 different types of pathologies simultaneously. The ChestX-ray14 dataset was used to train the deep learning-based CNN algorithm. The dataset consists of a total of 112,120 chest radiographs based on frontal-view labeled images of 30,805 different patients. The automatic extraction approaches were used to obtain the labels of the images. The dataset was divided into three partitions named training set (images = 98637, patients = 28744), validation set (images = 420, patients = 389), and tuning set (images = 6351, patients = 1672). The parameters of the model were optimized using the training set, the validation set was used to validate the CheXNeXt model, and the tuning set was used to evaluate and compare the networks. The AUC was used as the evaluation metric to evaluate the performance of the CheXNeXt model.

The experimental results showed that the newly proposed model gave better results in the case of atelectasis detection giving an AUC of 86.2% as compared to the radiologist's AUC, which was 80.8%. The key limitations of the research were that the CheXNeXt and radiologists were not allowed to take any benefit by looking at the patient history and the whole dataset was collected from a single institute.

B. MobileNet

Souid et al. [1] proposed a classification solution for lung pathologies using state-of-the-art deep learning and transfer learning techniques. A modified version of MobileNet V2 was used on the NIH Chest-Xray-14 dataset along with some metadata provided with the dataset i.e. age, gender, etc. Due to the significant class imbalance in this dataset, the researchers employed resampling techniques to address this issue.

One of the major goal of [1] was to develop a solution for IoT systems, requiring the system to be trained and performed on devices with low computing power. The authors specify that the deep learning model used in this work was designed particularly to develop low latency and small applications in the field of computer vision and IoT. After data augmentation, the authors divided the samples into training, validation, and testing subsets. The training subset included 38,819 images, and the validation, as well as testing subsets, included 12,940 image samples. It can be noted that the entire dataset was not used in this study, as the total samples in the dataset are 112,120. The training was done with 10 epochs and a batch size of 32.

The proposed solution was evaluated on the accuracy, sensitivity, area_under_the_curve (AUC), specificity, and time consumption, as fitness measures. The solution achieved 90% accuracy, 0.45 sensitivity, 0.97 specificities, and 0.55 f1-score. Since details, such as tissue structure and texture, are of utmost importance in medical image classification, researchers have derived a module to capture multi-scale input features in convolutional neural networks. Abnormalities in the tissues can be of variable sizes; hence capturing the spatial information is very important.

Hu et al. [7], proposed a solution, MD-Conv, which has multiple depth-wise convolution filters with variable kernel sizes in a single depth-wise convolution layer. This paper compares the classification results i.e. accuracy, AUC, and floating point operations per second (FLOPs), using this technique, on two datasets to other state-of-the-art methods. The proposed solution achieved AUC of 0.78 and the FLOPs were much lower than ResNet50 and DenseNet121. However, accuracy was lower than other techniques i.e. DenseNet+LSTM and DenseNet121.

C. DenseNet

The computational cost of models can increase exponentially when the input contains images. To handle this problem, the images are resized to a lower resolution which leads to losing details. Siemens et al. [8], proposed a solution based on the location-aware technique of Dense Networks, known as DNetLoc to incorporate high-resolution images for maximum possible detail. The authors combined two datasets i.e. Chest X-ray 14 and PLCO to have cumulative samples equal to 297,541 images, although the PLCO dataset has 12 classes along with class imbalance. This research makes use of pre-trained models on ImageNet. For the first dataset, cross_entropy was used as a loss function along with some additional weights, as there is a class imbalance. The batch size used was 128 samples per iteration. According to the authors, the reason for having a large batch size was to make the probability of samples belonging to all

classes higher. Adaptive learning rate along with Adam as the model optimizer was used. Furthermore, the input image size was the same as the original size i.e. 1024 * 1024. Histogram normalization was also applied to the PLCO dataset to tune the contrast and brightness of the images. Further normalization techniques involved mean as well as standard deviation normalization. The results with the Chest X-ray dataset were 0.84 AUC and for PLCO the AUC was 0.87.

D. VGG

Lung Disease is a very commonly found disease around the world. The impact of this illness on health is escalating quickly due to environmental changes, climate change, adjustments in lifestyle, and other reasons. Since these fatal diseases are spreading swiftly, so their timely diagnosis is quite essential. To resolve the classification problem in this field a lot of research has been done in the field of image processing, deep learning, and machine learning.

In [22] Subrato Bharati et al presented a novel idea of a deep learning-based hybrid model named VDSNet. The model was a combination of CNN-based model VGG and an image augmentation technique named Spatial_Transformer_Network (STN). The newly proposed model was trained on a well-known and publicly available lung disease dataset (NIH Chest X-ray).

The experiments were done on two versions of the dataset (Full and Sample), at both versions, VDSNet outperformed the existing models. For the evaluation purpose, validation accuracy, precision, F0.5 score, and recall were considered as the evaluation metrics. With the fuller version of the NIH Chest x-ray dataset, their proposed model VDSNet gave the validation accuracy of 73% and outperformed the existing models named hybrid CNN (validation accuracy = 69.5%), vanilla gray (validation accuracy = 67.8%), VGG (validation accuracy = 63.8%), vanilla RGB (validation accuracy = 69%), and modified capsule network.

E. ResNet-50

In another study, Baltruschat et al. [10], explored the domain of transfer learning and weight initializations on the model ResNet-50, on the NIH Chest X-Ray 14 dataset. They experimented with the network architecture and the input size passed to the model. They also incorporated non-image features from the dataset such as age, gender, etc. Their methodology section states that they introduced class balancing and positive/negative balancing in the loss function, but there was no significant difference in the results. So, they used class-averaged binary_cross_entropy as their objective function.

They also investigated the effect of two distinct strategies of network initialization for ResNet50. In the first technique, they used weights trained on the dataset of ImageNet, whereas, in the second technique, they randomly assigned weights to the network. The researchers concluded that making use of non-image features proves to help increase the classification results of some of the pathologies. They achieved an average AUC of 0.80 for all 15 classes. This research suggests that fine-tuning the models can enhance results. In this study, the average AUC improved from 0.73 to 0.80.

F. MULTI-LABEL CLASSIFICATION

In [14] Aravind Sasidharan Pillai et al introduced a new deep learning-based approach for multi-label classification of chest X-ray images, which can simultaneously detect multiple pathological conditions. The authors trained & evaluated the technique on a publicly available dataset named Chexpert of chest X-ray images, which included 14 different pathological labels such as pneumonia, emphysema, and lung mass. They down-sampled the dataset to 11 GB which was originally of the size 439 GB and it was divided into two sets train set 80% & validation set 20%.

The proposed approach consists of CNN (Custom Net, DenseNet121, ResNet-50, Inception_V3, Vgg16) architecture with residual connections and attention mechanisms for feature extraction and classification. The residual connections help to alleviate the vanishing gradient problem in deep networks, while the attention mechanism allows the network to focus on specific regions of the image that are most relevant for each label. The proposed approach has the potential to be a valuable tool for automatic multi-label classification of chest X-ray images, which can aid in the diagnosis and management of various lung diseases. The authors suggest that their approach can be further improved by incorporating additional clinical information or by using more sophisticated deep-learning architectures.

The authors conducted extensive experiments to evaluate the performance of their approach and compared it to several other state-of-the-art methods for multi-label chest X-ray classification. The results showed that their approach outperformed all other methods in terms of both accuracy and area_under_the_receiver_operating_characteristic curve (DenseNet training AUROC 78 & training accuracy 87%). The authors also conducted a detailed analysis of the performance of their approach on different subsets of the dataset, including cases with multiple co-occurring pathologies. The results showed that their approach was able to accurately detect specific pathologies, even in the presence of other pathologies.

In [23], Shamrat et al. through customized MobileNetV2 from chest X-ray images, presented the MobileLungNetV2 model for classifying 14 lung diseases using chest X-rays. The model is based on MobileNetV2 but fine-tuned to improve accuracy. Pre-processing steps like contrast enhancement and noise reduction were applied to the ChestX-ray14 dataset. This study was about a single label classification.

In [19], K. V. Priya et al proposed a federated learning approach for detecting chest diseases in chest X-ray images

using DenseNet for multi-label classification. The authors note that chest diseases are a major cause of death worldwide, and early detection is critical for effective treatment. However, due to the large number of images and the need for specialized expertise, manual interpretation of chest X-ray images can be time-consuming and error-prone. Therefore, automated classification of chest X-ray images can help to improve diagnosis and reduce the burden on radiologists. The proposed federated learning approach involves training a deep neural network on local datasets at multiple hospitals or clinics, with each dataset containing images from patients at that location. The model is trained using the federated averaging algorithm, which aggregates the gradients of the local models to update a global model while maintaining the privacy of the local datasets. The authors use DenseNet, a popular deep-learning architecture, for the multi-label classification of chest diseases.

The authors evaluate the performance of the federated approach on a publicly available dataset of chest X-ray images with 14 different chest diseases. They compare the performance of their approach with a centralized approach and a baseline approach using a single local dataset. The results show that the federated approach achieves higher accuracy and F1-score compared to the baseline and centralized approaches, indicating that the federated approach can improve the performance of chest disease classification while maintaining data privacy. The authors conclude that federated learning can be a useful approach for large-scale multi-institutional studies in medical imaging.

In another study, Al-Sheikh et al. [24] introduced an automated system for multi-lung disease detection using X-rays and CT scans. It employed a customized convolutional neural network (CNN) and two pre-trained deep learning models (AlexNet and VGG16Net) with a new image enhancement model based on k-symbol Lerch transcendent functions. The system involved two main steps: pre-processing with image enhancement and classification using the CNN models. This study did not address the issue of multi-label classification.

G. GANs (GENERATIVE ADVERSARIAL NETWORKS)

Data augmentation techniques and generative adversarial networks (GANs) have been used extensively for synthetically generating all kinds of data such as images [25], [26], speech [27] and videos [28]. Within images, GANs have also been used to generate medical images [29]. Similarly, in [15] Saman Motamed et al. used the GAN for the data augmentation of chest X-ray images to detect Pneumonia and COVID-19 diseases. A GAN was trained to generate accurate X-ray pictures to be combined with initial training data. The authors demonstrated that using GAN-generated pictures for data augmentation could improve the efficiency of detection models by evaluating their method on two publicly accessible datasets.

The first utilized chest X-ray dataset named ChestXRay2017 consists of two categories named Normal (having 1575 images) and Pneumonia (having 4265 images) in JPEG format. The images were resized to 128 $\times$ 128 and they chose Pneumonia (the larger class) as the training set and 500 images were randomly selected per class to test the performance of the model. The rest of the images were used for augmentation and to train the different models. Wang et al. created the second publicly available dataset named COVIDx by mixing the Covid-chest X-ray dataset with the four other datasets. Covid-chest X-ray was developed by Cohen et al which includes chest X-rays having radiological readings for COVID-19. The COVIDx dataset has 3 classes named normal (having 8066 images), pneumonia (having 5559 images), and COVID-19 (having 589 images). All images were converted in grayscale and resized to the size of 128 by 128. The test set consists of a total of 589 images randomly taken from each of the 3 classes.

The results showed that IAGAN model achieved the highest results for dataset-I (Sensitivity/Recall = 0.82, Specificity = 0.84 & Accuracy = 0.80) and dataset-II (Sensitivity/Recall = 0.69, Specificity = 0.69 & Accuracy = 0.69). They conclude that using GAN-based data augmentation is an excellent method for enhancing the performance of deep learning models in applications involving medical imaging.

Malygina et al. [11] investigated how GANs can be utilized to improve the performance of deep learning-based models to detect the pathologies in the chest X-rays in the case of an imbalanced dataset. The authors used a publicly available dataset named ChestXray14 which consists of a total of 112,120 x-ray images taken from 30,805 patients having 14 labels. The total dataset is divided into three categories of data for the pathology classification & localization task: train set (70%), validation set (10%), & test set (20%).

To reduce the computational power to detect pneumonia the images of the dataset were resized to 256 $\times$ 256 from the 1024 $\times$ 1024 to train both the GAN & the image classifier. They created synthetic pictures using a GAN that was trained on the normal and pneumonia categories, adding them to the training dataset to create more evenly distributed classes. After that, the authors analyzed the performance of several deep learning models that had been trained on both the initial and supplemented datasets. Model without augmentation showed the AUC (Pneumonia = 0.9745, Pleural-Thickening = 0.9792, & Fibrosis = 0.9745), and PR AUC (Pneumonia = 0.9580, Pleural-Thickening = 0.9637, & Fibrosis = 0.9446) while the model with the augmentation showed the AUC (Pneumonia = 0.9929, Pleural-Thickening = 0.9822, & Fibrosis = 0.9697), and PR AUC (Pneumonia = 0.9865, Pleural-Thickening = 0.9680, & Fibrosis = 0.9294). The augmented dataset by GAN increased the model's accuracy in the case of pneumonia & COVID-19 recognition.

Munawar et al. [16] proposed a two-stage approach, in which firstly, a GAN is trained to create synthetic lung images, and then these images are used to train a segmentation network. The authors used a publicly available dataset

of chest X-ray images named the JSRT dataset, Montgomery dataset & Shenzhen dataset, which was preprocessed and augmented to create a large training set. The datasets were divided into three categories: the JSRT dataset (train set: 200, validation set: 20, test set: 20), the Montgomery dataset (train set: 110, validation set: 10, test set: 18), and Shenzhen dataset (train set: 200, validation set: 40, test set: 40). The GAN was then trained on this dataset to generate realistic lung images. The segmentation network was then trained on the synthetic lung images and evaluated on a separate test set.

The results showed that the proposed method outperformed a few other state-of-the-art methods for lung segmentation on the same dataset. The authors also conducted a detailed analysis of the performance of the method and showed that it was able to segment lungs accurately in a variety of challenging scenarios, including images with significant pathology and those with low image quality. The authors conclude that the proposed method has the potential to be a valuable tool for automatic lung segmentation in chest X-ray images and can aid in the diagnosis and monitoring of lung diseases. Table 1 summarizes the existing works.

H. RESEARCH GAPS

Thoroughly examining the literature revealed a few research problems and gaps such as class imbalance problems, generalization problems, and the challenge of computational complexity. Improvement in these areas, which can enhance our capabilities in the field of medical imaging.

1) REMOVING CLASS IMBALANCE

NIH Chest X-ray 14 dataset has 14 disease classes with 60000 samples in total and 1 non-disease class with 60000 samples. There are about 10,000 images of just the Infiltration class. Class of Hernia has the least amount of samples i.e. around 0.3% of the total samples. This indicates a huge class imbalance which leads to non-uniform results for each class. So there is a need to reduce this imbalance with augmentation by using Generative Adversarial Networks (GANs), or by merging some other datasets of the same type with it. Finally, we can evaluate the results by applying various deep learning or transfer learning models.

2) BETTER GENERALIZATION OF MODEL

Another gap that our team came up with is lower values of accuracy or F1 score or area_under_the_curve (AUC). For example, the results from research work [1] show that the same model that provides good results in some classes produces quite low results for others. One of the tasks in the future is to find such a technique that is generalized enough to produce high results for all diseases. One of the classes that have lower classification results is the Infiltration class. Although this class has a large number of samples, the results do not reflect that. One reason for suboptimal results might be the quality of the samples, which can be examined in future research.

3) IMPROVING COMPUTATIONAL COST

The NIH Chest X-ray 14 dataset comprises 112,120 samples and a size of 45 GB. The image size is 1024 by 1024 obtained from x-ray tests of around 30,000 patients. Many dense layered deep learning models struggle in the training phase Since this dataset has a large size. The computational time is so high that a single training execution of around 20 epochs costs around 5 hours. These rounded-off results were obtained from our implementation of the EfficientNet B4 model on this dataset. Such long computation times can hinder in-depth experimental research. Hence, one of the future research can be decreasing the execution complexity of models or applying some preprocessing techniques that keep the results good as well as decrease the execution time. One such way of decreasing the execution time is to use such models that have a low no of layers such as MobileNet, GoogleNet, etc.

III. RESEARCH METHOD

This section provides insights towards our methodology for the classification of lung diseases. Figure 1 provides a flowchart of the methodology followed in this research.

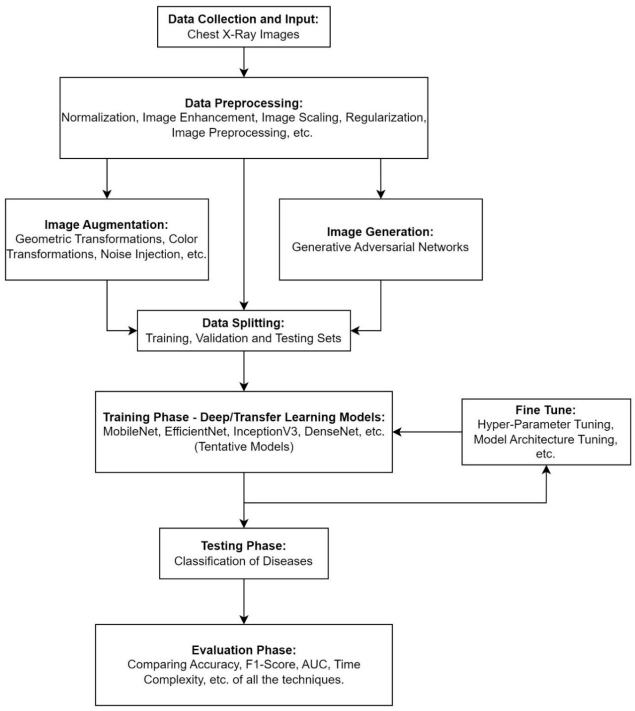


FIGURE 1. Flow chart of research methodology.

A. DATA COLLECTION AND INPUT

Human chest exams with X-rays are one of the most common and cost-effective processes for examination used by doctors. However, since X-rays have low quality as compared to other sophisticated methods such as CT scans, it is a difficult task to perform disease classification using X-ray images.

TABLE 1. Summarized review of existing works.

Ref#	Year	Technique/Models	Dataset	Train & Test Set	Results
[1]	2021	MobileNet V2 (with some additional hidden layers)	NIH ChestX-ray (ChestX-ray14) 14 class dataset and used only 64,699 samples after data augmentation	Training: 38819 (60%), Validation: 12940 (20%), Testing: 12940 (20%)	AUC: 0.81, Accuracy: 90%, Sensitivity: 45%, Specificity: 97%, F1 Score: 55%
[7]	2020	MobileNet V2 (with Multi-Kernel Depthwise Convolution (MD-Conv))	ChestX-ray14 (112,120 samples) ChestX-ray2017 (5856 samples)	-	AUC: 0.78
[8]	2018	DNet: DenseNet-121 & DNet-Loc: DenseNet-121 Location Aware	Combined 2 datasets (297,541 samples), ChestX-Ray14 (112,120), PLCO (185,421 samples), PLCO is 12 Class dataset with imbalance	-	ChestX-Ray14 – AUC: 0.84, PLCO – AUC: 0.87
[9]	2018	CheXNeXt	ChestX-ray14	-	Average AUC: 0.84
[10]	2019	ResNet-38, ResNet-50, ResNet-101, with Multi-label loss function	ChestX-ray14	Training: 70%, Validation: 10%, Testing: 20%	ResNet-38 – AUC: 0.80
[13]	2018	CNN	Subregion_demarked parenchymal-lung disease of ILD	Training set = 960 samples, Test set = 240 samples	Accuracy at CNN classifier = 95.12%
[22]	2020	Hybrid Model (VGG + spatial transformer network)	NIH chest X-ray image dataset	-	Val_acc at VDSNet = 73%, vanilla gray = 67.8%, vanilla RGB = 69%, hybrid CNN = 69.5% and VGG = 63.8%
[12]	2020	Pre-trained VGG16 + SVM Classifier & VGG16(TL) + Softmax Classifier	ICBHI 2017 Database	-	Accuracies Method 1 = 7.62%, Method 2 = 5.18%
[17]	2020	VGG16, ResNet-50, InceptionV3	Dataset from Shenzhen Hospital	Training set 80%, Validation set 10% & Test set 10%	At Segmented CXRs: VGG16 has 95%, ResNet-50 has 78%, InceptionV3 has 75% At Non-Segmented CXRs: VGG16 has 77%, ResNet-50 has 68%, InceptionV3 has 77%
[18]	2019	2D CNN as Classifier	Dataset consisting of 6 categories	Training set = 70% and test set = 30%	Accuracy = 97%
[14]	2022	Custom Net (CNN), DenseNet121, ResNet-50, Inception_V3, Vgg16	Chexpert (downsampled version-11 GB) original size 439 GB	train & validation split 80:20	DenseNet training AUROC 78 & training accuracy 87
[19]	2020	DenseNet121 visualization technique: gradCAMS	Chest X-ray 14, Total pics 112,120 from 30,805 people, Labels = 14 diseases NIH chest X-ray (due to class imbalance issue)	-	-
[15]	2021	CNN & GAN (IAGAN, DCGAN)	ChestXRay2017 & COVIDx dataset	Dataset I (Train,Test) Normal: (-,500), Pneumonia: (33885,500), COVID_19: -, Dataset II: (67293,589 (42300,589), (0,589)	IAGAN: (Datasets I, II) Sensitivity: (0.82,0.69), Specificity: (0.84,0.69), Accuracy: (0.80,0.69)
[11]	2019	GAN (CycleGAN), DenseNet for classification	ChestXray14	Total = 112120, Labels = 14 diseases, Train Set = 70%, Validation Set = 10%, & Test Set = 20%	AUC: (Pneumonia 0.9929, Pleural-Thickening 0.9822, Fibrosis 0.9697), PR ACU (Pneumonia 0.9865, Pleural-Thickening 0.9680, Fibrosis 0.9294)
[16]	2020	GAN & U-net	JSRT dataset, Montgomery dataset & Shenzhen dataset	JSRT (train: 200, valid: 20, test: 20), Montgomery (train: 110, valid: 10, test: 18), Shenzhen (train: 200, valid: 40, test: 40)	dice-score = 0.9740, and IOU score of 0.943

Arranging an adequate number of samples for deep learning models is a difficult, if not impossible, task to do. due to patient privacy and confidentiality concerns, finding publicly available datasets is challenging. However, the National Institute of Health has made available, publicly, a dataset that is convenient to download and work with. This dataset is called NIH Chest X-ray 14 dataset, containing 112,120 samples of multiclass images along with annotated labels up to an accuracy of 90%. The labels were generated by the authors using natural language techniques. This data belongs to 30,805 patients.

The NIH CXR 14 dataset is available in the form of.png images with extension, over the internet. The size of the dataset is huge i.e., 45 GBs. Nevertheless, since it can be downloaded to the local machine and is already available on Kaggle, it makes the experiments convenient. For this research work, we used image resolution between 224 * 224 and 500 * 500, although the original is 1024 * 1024.

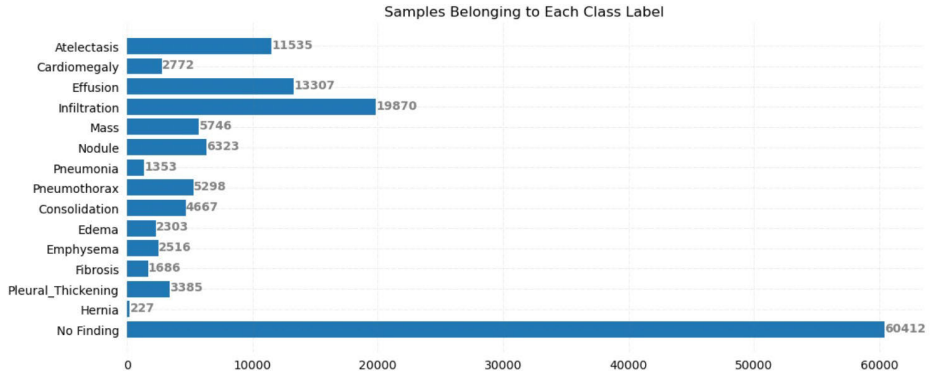


FIGURE 2. Visual class imbalance of NIH Chest Xray 14 dataset [1].

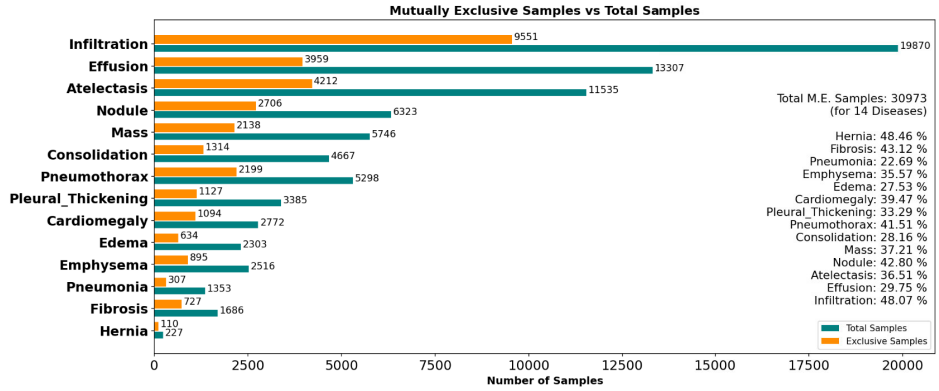


FIGURE 3. Mutually exclusive samples vs total samples.

The labels, corresponding to each image, are available in a comma-separated-value (CSV) file in the form of strings. We first converted these labels to a one-hot-encoded format and then divided the data into 3 subsets for training, validation and testing phases using Python's library named Sklearn. We used the train_test_split function with random_state equal to 0. The ratio for these divisions was 7:1:2 for 3 subsets. 80726 images were used in training, 8970 for validation and 22424 for testing the model's performance.

Figure 2 indicates the number of samples belonging to each class. It is evident how imbalanced the dataset is.

Figure 3 shows the number of samples that belong to a single class and the total samples belonging to each class. Each class has less than 50% mutually exclusive images, with the lowest as 22% for Pneumonia.

Figure 4 presents a few sample images of the dataset. All images are labelled with a disease(s) from the 14 lung pathologies or no-finding that represents the no-disease class.

B. DATA PREPROCESSING AND DATA AUGMENTATION

In data processing, data fed as input is scaled/normalized which helps the model in learning the objective function quickly and effectively. Data augmentation can be applied to a variety of data types, including images, audio, text, and time series. Some common data augmentation techniques for image data include:

- Flipping or rotating the image horizontally or vertically
- Cropping or resizing the image
- Adding noise or distortions to the image
- Changing the brightness, contrast, or color of the image
- Applying geometric transformations such as scaling, shearing, or perspective warping

1) IMAGE PREPROCESSING

It refers to the set of operations or techniques applied to raw images to prepare them for further analysis or processing. The goal of image preprocessing is to enhance or extract meaningful information from the image while reducing noise, artifacts, or other unwanted components that may interfere with subsequent analysis. The preprocessing step typically includes a series of operations such as image cropping, resizing, color correction, filtering, noise reduction, image enhancement, segmentation, and feature extraction. These operations are chosen based on the characteristics of the images and the specific application requirements.

- In this research work, the image input size was kept to either 224 * 224 or 300 * 300 which caused blurriness.
- To improve the quality, image sharpening was applied to all the dataset subsets.
- Since images are read in RGB format, we applied various rescaling techniques including multiplying each

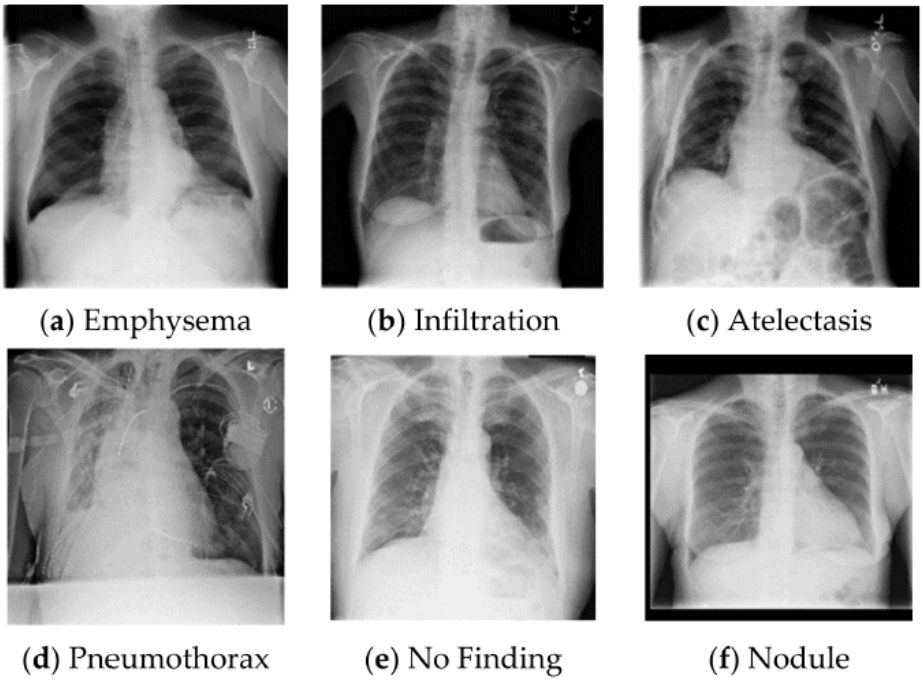


FIGURE 4. Sample images of NIH Chest X-ray 14 dataset [1].

pixel with 1/255 or converting each pixel value between −1 and 1. This not only generalizes the input but also saves memory and decreases the computational cost of operations applied. Furthermore, it also smoothens the learning of the objective function.
- The filter used for sharpening was: [0 −1 0] [−1 5 −1] [0 −1 0]

2) IMAGE AUGMENTATION

This technique is used to produce data samples from the existing samples. It can be useful when the dataset has a low number of examples or if there is a class imbalance in the dataset. We can apply various geometric methods to produce augmented data. This not only aids in learning a class imbalanced dataset but also generalizes the model's learning, hence decreasing overfitting. The model learns to handle new variants of training examples. We used the ImageDataGenerator library of Keras, to apply image augmentation.

Following are the methods that were implemented for this purpose:

- Random Rotation: It randomly rotates the input image according to the value provided. In this work, the value was set to 20 degrees.
- Sheering: It shifts the viewing angle of the image.
- Random Zoom: It randomly zooms into the input image and the value was set to 0.1 percent in this work.
- Width Shift: It shifts the image's width horizontally and its value was set to 0.1 percent of the total width of the image.
- Height Shift: It shifts the image's height vertically and its value was set to 0.1 percentage to the fraction of the image's total height.
- Horizontal Flip: This option randomly flips the image horizontally to change the sides of the image.

3) IMAGES BEFORE AND AFTER APPLYING VARIOUS PREPROCESSING AND AUGMENTATION TECHNIQUES

Figure 5 and Figure 6 present the original and processed images, respectively, from the dataset. We can see the images in smaller resolution (Figure 6) are a little blurry.

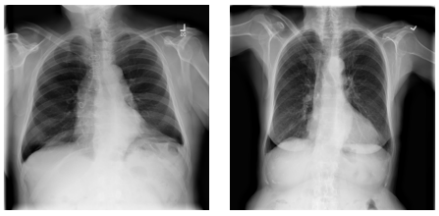


FIGURE 5. Original Images with a resolution of 1024 * 1024.

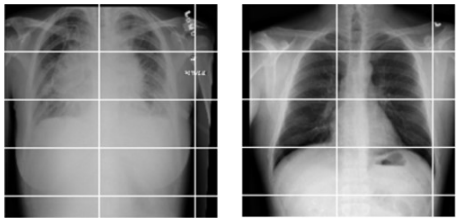


FIGURE 6. Augmented images with a resolution of 224 * 224.

Figure 7 demonstrates augmented images after image sharpening with a resolution of 224 * 224. (The left image has height shifted whereas the right image has width shifted).

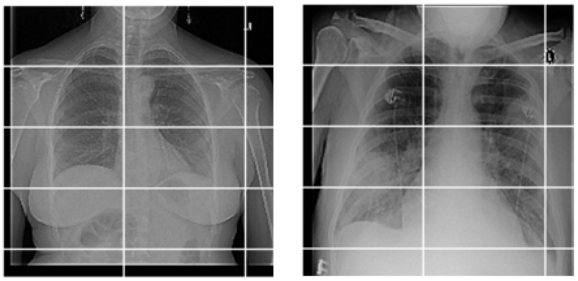


FIGURE 7. Augmented images after image sharpening with resolution of 224 * 224.

Figure 9 presents X-ray images after applying image augmentation and sharpening in RGB format; these images are fed to the network.

4) DATA GENERATION WITH GENERATIVE ADVERSARIAL NETWORK (GAN)

GAN's [30] can be used to produce data from already available samples. It can be vital for scenarios such as medical image classification wherein arranging enough data to train a large deep learning model, can be tedious. By using current available data, we can generate new samples with GAN and these synthetic samples do not belong to real patients in any direct sense. This technique has accelerated the use of deep learning in the medical field; however, it still needs a lot of improvement in terms of capturing delicate tissues from the images. In this work, we have implemented deep convolutional GAN (DCGAN) [31] to produce synthetic X-ray images containing various pathologies that are already included in the used dataset. A deep generative model with around 5 layers was implemented against a rather shallow discriminator network. The deep architecture helps in capturing minor tissues from the X-ray images. DCGAN was chosen for this study due to its balance of simplicity and effectiveness in generating high-quality images. Unlike more complex GAN variants such as WGAN, CGAN, and BEGAN, DCGAN provides a straightforward architecture that is easier to implement and optimize, making it suitable for our application in medical image generation. Additionally, DCGAN has been widely validated in various image synthesis tasks, demonstrating its robustness and efficiency in generating realistic images, which aligns well with our goal of augmenting medical datasets.

The DCGAN generates synthetic X-ray images by training a generator and a discriminator in an adversarial manner. The generator creates images from random noise, while the discriminator evaluates whether the images are real or synthetic. During training, the generator tries to produce images that are increasingly realistic, thereby 'fooling' the discriminator. This adversarial process iteratively improves the quality of the generated images. Specifically, the generator learns to map random noise vectors to image spaces that resemble the distribution of the real X-ray images, whereas the discriminator learns to distinguish between real and synthetic images. Over time, this results in the generator producing high-quality synthetic X-ray images that closely resemble real ones.

We used 3 layers in this network with 512, 256 and 128 neurons in each layer respectively. We used LeakyReLu with alpha equal to 0.2, as the activation function in all the layers except the output layer, where we used sigmoid due to binary classification. As for the generator network, it consisted of 6 layers. The number of neurons in each layer was 128, 256, 512, 512, 1024, and 1024, respectively. All the layers used LeakyReLU except the output layer, which had tanh activation function. Figure 8 shows the structure of GAN used.

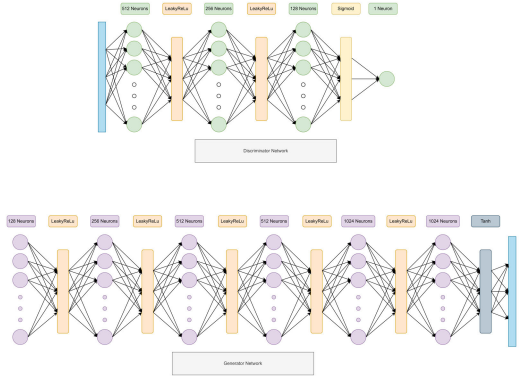


FIGURE 8. GAN architecture.

The resolution of the original images fed to the GAN was 300 by 300 with a batch size of 32. The training was done for 300k epochs for each class. It took 30 hours for each class on our local PC with a 1050Ti GPU and i7-8750H CPU. Due to time and hardware constraints, it was a challenge to train the GAN for further epochs. In Figure 10, we can see some of the samples generated for the Cardiomegaly class after 300k epochs.

Figure 11 shows individual images belonging to the class of Hernia. The images are a little pixelated, which is due to the low resolution of the latent vector and input size of the original images.

Our GAN implementation aimed to generate synthetic X-rays that closely resemble real images within the training data. This approach doesn't introduce any external data and solely augments the existing dataset to train our model.

C. TRAINING PHASE

This section highlights some of the major details involved in the training phase such as types of models, hyper-parameters, and architectures of classifiers.

1) DATASET SPLIT

The dataset split is 70% for Training, 10% for Validation and 20% for Testing.

- Training Images: 80726
- Validation Images: 8970
- Testing Images: 22424

Table 2 presents the distribution of samples belonging to each class in the training subset of the dataset.

TABLE 2. Training set sample distribution.

Class Name	Training Samples	Total Samples
Atelectasis	8276	11535
Cardiomegaly	2015	2772
Consolidation	3324	4667
Edma	1656	2303
Effusion	9717	13307
Emphysema	1758	2516
Fibrosis	1213	1686
Hernia	152	227
Infiltration	14190	19870
Mass	4105	5746
No Finding	43543	60412
Nodule	4517	6323
Pleural_Thickening	2478	3385
Pneumonia	987	1353
Pneumothorax	3784	5298

2) MODEL ARCHITECTURES

A few deep learning models are applied to the dataset to examine the performance based on different architectures. Ten models were implemented with Python's library named Keras. In terms of model initialization, we used transfer learning for the convolution part of the models and random initialization for the fully connected network. Off-the-shelf weights that were trained on the ImageNet dataset were used for quick learning of convolution network. Deep as well as shallow models were used to test which convolutional networks produce better results in feature extraction. As for the classifier network, a custom architecture was implemented. This custom architecture included 2 hidden layers and an output layer comprising 15 neuron units, i.e. a dedicated neuron for each class. 256 and 50 neurons were assigned to the first and second hidden layers, respectively. Fewer layers and neurons were used to avoid overfitting.

3) HYPER-PARAMETERS

The hyper-parameters used in the implementation were kept the same for all the models, to have a proper comparison of performance.

- Batch Size: The memory constraint of RAM and VRAM of GPU didn't allow many options to be tried in terms of batch size. Hence, the value of 32 was used for all the models trained.
- Optimizer: At first two options were tried i.e., Stochastic Gradient Descent (SGC) and Adaptive Momentum (Adam), but after some executions no significant difference was found. Therefore, we settled for the Adam optimizer for the training of models, as it is one of the advanced methods of learning the weights of the model.
- Learning Rate: Since we used already trained weights (i.e., transfer learning) for convolutional networks in our models, the learning rate was kept low. Its value was set at 0.0001.
- Activation Functions: For its simple computation and effectiveness, ReLU was applied in all the layers, except the output layer, for activation of neurons and to add non-linearity into our models. As for the output layer, sigmoid was used, since our problem is a multi-label classification problem, and each neuron has to output an independent value.
- Loss Function: Since the dataset has non-mutually exclusive labels, we need such a loss function that can consider each class independently. For this purpose, Binary_Cross_Entropy (BCS) is used to measure the loss of all the classes.

D. TESTING AND EVALUATION PHASE

In this section, we discuss the metrics that are used for the evaluation of our trained models.

1) BINARY ACCURACY

In general, accuracy states a percentage of correctly classified samples divided by all the classification results, including the correct as well as the misclassified results. Since the dataset, we worked on belongs to a multi-label classification problem, we cannot just consider the one-hot encoded labels as a single label. For example, if there are three classes in total and the sample belongs to classes 1 and 2 (label: 1 1 0). However, our model prediction states that the sample belongs to classes 1 and 3 (label: 1 0 1), so using regular accuracy would be considered incorrect. Whereas, binary accuracy calculates the percentage of labels that matched in the entire one-hot encoded label. Therefore, the binary accuracy of the above example would be 2 divided by 3, i.e. 66% accuracy.

2) RECALL

When a dataset has a class imbalance, evaluating the model based on simple accuracy is not enough. Recall is a measure of how many actual positive examples in the dataset, were classified as truly positive by the model. The recall is an important measure in the medical field since we aim to classify all the actual diseased samples as truly diseased, by the model. It is defined in [1], as follows:

$$Recall(orSensitivity) = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{1}$$

3) PRECISION

Another metric that can evaluate the performance of the trained model on the imbalanced dataset is Precision. It is a measure of how many samples were actually positive from all the samples that the model predicted as positive. It is defined

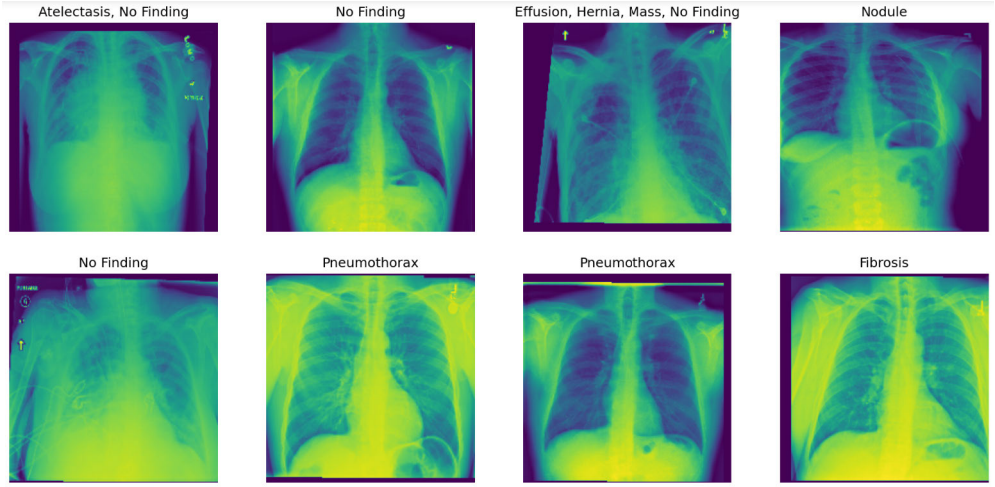


FIGURE 9. Sharpening of X-Ray images in RGB format.

FIGURE 10. Images generated after 300k epochs belonging to Cardiomegaly class.

in [1], as follows:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (2)$$

4) SPECIFICITY

This measure can be considered as a recall but for negative samples, i.e. it gives a measure of how many actual negative examples were classified as negative by the model. It is defined in [1], as follows:

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives} \quad (3)$$

5) F1 SCORE

This measure considers both recall and precision of the model and indicates more of a balanced performance value. It is a harmonic mean of precision & recall. The F1 Score of a

FIGURE 11. Images generated by GAN belonging to Hernia class.

model will be high if more samples predicted as positive by the model were actually positive and more samples predicted as negative by the model were actually negative. It is defined in [1] with the following formula:

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

As the F1 Score gives equal importance to both precision & recall, it might sometimes fail to indicate the performance of the model where one is more important than the other. For example, in the medical field, we need a higher recall rate of models and can give up on the importance of precision. Therefore, F-Beta Score can be used where we assign a weight that defines the importance of each measure. A higher Beta value i.e. more than 1, gives more weightage to recall, and a lower beta value i.e. lower than 1 gives more weightage to precision.

6) AUC (ROC)

The Receiver Operating Characteristic (ROC) curve is used to examine the capability, of a model, of distinguishing between classes and is plotted using measures named True Positive Rate (TPR) and False Positive Rate (FPR). Both of these measures are directly proportional i.e. if one increases the other increases as well. The ROC curve is plotted against a number of thresholds and indicates an area_under_the_curve (AUC). The higher the area, the better the model's performance is.

TABLE 3. MobileNetV1 results without geometric image augmentation with adam optimizer.

Epochs	5	10	15
Binary Accuracy	93.2	92	92
Recall	55.2	47.7	50
Precision	52.8	42.5	46.5
F1 Score	54	45	48.2
Specificity	98.2	97	96.4
Average AUC	78.6	77.1	76

IV. EXPERIMENTAL RESULTS

In this section, the results of our experimental research and the programming environments, that were used to obtain, are documented in separate sections.

A. EXPERIMENTAL SETUP

The size of the dataset played a significant role in deciding the programming environments to conduct experiments. Deep learning models were implemented and trained on two platforms. The first one was a free GPU-based platform called ''Kaggle''. Its free version provides 13 Giga Bytes of RAM, a 2-core CPU, and a P100 GPU. The second platform was of our own i.e. an ASUS ROG laptop with 16 Giga Bytes of RAM, an 8th generation 6-core Intel CPU i.e. i7-8750H, and NVIDIA's GTX 1050Ti GPU. Although we had multiple options, these were not enough to explore the multi-dimensional space of experimental analysis thoroughly. Nevertheless, we obtained decent results with different models.

B. EXPERIMENTAL RESULTS (WITHOUT GEOMETRIC IMAGE AUGMENTATION)

This section presents the results of deep learning models without any geometric augmentation techniques. The original images from the dataset were passed as it is to the deep learning model. MobileNetV1 was used for this experiment. Table 3 indicates the model starts to overfit as the number of epochs is increased. The batch size was set to 16 and the input resolution of the images was 300 * 300. We scaled the pixels' values between 0 and 1 and to optimize the model we used binary cross entropy loss. To tackle this overfitting problem we use augmentation for the rest of the experiments in the research.

1) VGG16

In medical CXR data, spatial information can be very vital in the classification of diseases. Therefore, it is common to use deep learning-based networks to extract the features having important information from the images. VGG16 is one such model that has a high no of layers in its convolution network. We implemented this model with already learned weights for its feature extractor and a custom classifier with 3 layers. There were 4 executions with 5, 10, 15, and 30 epochs, having batch size 32. The model gave an above-average performance, with recall as high as 54.5 and a maximum AUC hitting mark of 67.

2) INCEPTION-V3

This model focuses on a rather optimized approach for producing good results without a huge network architecture. It has less number of learnable parameters as compared to VGG16 but produces quite good classification results. In this work, we trained it on 5, 10, and 15 epochs with a batch size of 32. It gave recall as high as 55 and AUC as high as 69, which are higher than VGG16 with a lesser number of epochs, training time, and parameters. Although the graphs and results indicate that, the model started to overfit as the epochs were increased.

3) Xception

Similar to Inception, this model also follows the idea of a shallow-depth network with optimized performance. We ran this model on 5, 10, and 15 epochs with 32 batch sizes and Adam optimizer for weight learning. It gave recall as high as

56.3 and an AUC of 67.7. The results were quite constant for all 3 executions.

4) DenseNet-121

It has many number of layers, although the number of learnable parameters is not high. It uses transition layers to pass the information from one layer to the layers ahead by skipping some of them. We implemented this model with a custom classifier of 3 layers and trained it on 5, 10, and 13 epochs. The results show that the model starts to overfit as the epochs are increased. Hence, the best results that this model produced were on 5 epochs. The recall was 55 and AUC was 68.4.

5) MobileNet

A lightweight model, as the name suggests, with only 4.3 million learnable parameters (as compared to VGG16's 138 million). This is the best model in terms of results so far in this research work. It gave recall as high as 57 and an AUC of 70 with 10 epochs.

6) MobileNet-V2

This is a newer version of the previous model and has a slightly less number of learnable parameters. The model was trained on 5, 10, and 15 epochs, and it produced slightly lower results than its predecessor did. The model has an AUC of 69.1 and a maximum recall of 55.

7) EfficientNet-B2

The efficientNet model introduces an idea of systematic increment of the model's width, depth, and input resolution. This model was trained on multiple epochs but it did not produce good enough results. With recall only as high as 48 and AUC just hitting 57, it was not as promising as other models.

8) EfficientNet-B4

It is another version of the EfficientNet model, with a slightly larger network architecture. It does not necessarily have a huge number of parameters to learn, nonetheless, it takes quite a while to train. Just like the previous model, it did not produce good results with a maximum AUC of 52 and a recall of 43. Due to high training hours, we were not able to train it on a higher number of epochs.

9) EfficientNet-B6

Similar to EfficientNet B4, this model does not produce good enough results to carry on with further experimentation. It was trained on only 5 and 10 epochs due to its high training hours. It produced 56.2 AUC and 46.8 recall. It did give slightly better results on higher epochs, which highlights the question of whether this model produces good results with fine-tuning.

10) ResNet50-V2

This model has a very large no of layers and uses residual blocks to pass information to the layers ahead. We trained it on 5, 10, and 15 epochs. It gave almost similar results as MobileNet V2, although it takes more time to train. It gave a recall of 55 and an AUC of 67.4 on 10 epochs.

C. SUMMARY OF EXPERIMENTAL RESULTS (MODEL EXPERIMENTATION)

Table 4 provides the best results for each of the implemented models regardless of the number of epochs.

Observation 1: MobileNet (version 1 and 2) models produce slightly better results than other CNN-based models i.e., VGG-16, DenseNet, InceptionNet, etc.

D. INITIAL COMPARISON RESULTS

This section demonstrates how our preliminary results turned out to be in comparison to the existing work [reference to base paper].

1) CLASS-WISE AUC

Table 5 shows a comparison between the existing models [base paper] and some of the well-performing suggested models. It was observed that MobileNet performs better mostly in terms of accuracy for 4 classes.

2) AUC RESULTS COMPARISON

The suggested MobileNet produces better AUC for 9 diseases when compared with the MobileNetV2's class-wise AUC as shown in Table 5. However, the average AUC does not improve due to MobileNet's lower AUC for diseases such as Emphysema, Cardiomegaly, Hernia, etc. We concluded that this is because of the lower number of mutually exclusive samples for these classes.

3) EXPERIMENTAL RESULTS (AFTER HYPER-PARAMETER TUNING, IMAGE PROCESSING, AND FINE TUNING)

We experimented with various aspects of deep learning including image-preprocessing techniques, hyper-parameter tuning, fine-tuning the models, and generating synthetic samples with Generative Adversarial Networks (GANs). We documented the obtained results after applying each mentioned technique.

4) RESULTS AFTER HYPER-PARAMETER TUNING

The results of various hyper-parameter tuning options are demonstrated with the MobileNet model. From an infinite number of factors, that can affect a model's performance, we tested a small number of options from the hyper-parameter list.

5) LOSS FUNCTION

Two types of loss functions i.e., conventional Binary Cross Entropy (BCE) and Multi-Label Binary Cross Entropy (MLBCE) were tested. In MLBCE, the loss of a specific

TABLE 4. Summary of best results (model experimentation).

Model Name	Epochs	Binary Accuracy	Recall	Precision	F1-Score	Specificity	AUC
VGG16	30	93.3	54.5	55.5	55	98.2	80
Inception-V3	10	93	55	54.5	55	98.5	79
Xception	10	93.3	56.3	54.2	55.2	98	80
MobileNet	10	**93.4**	**57**	54	**55.3**	98.4	**81**
MobileNet-V2	15	93	55	53.3	54	**98.6**	78
DenseNet121	5	93.3	55	**56**	55.2	98.3	80
EfficientNetB2	10	93	48	54	53	97	67
EfficientNetB4	10	92.2	43	54	48	97	68
EfficientNetB6	10	92.6	46.8	53	50	97.4	66.2
ResNet50-V2	10	**93.4**	55	54	54.4	98.5	67.4

TABLE 5. Preliminary results: class wise AUC.

	MobileNetV2	VGG16	Xception	DenseNet121	ResNet50V2	MobileNetV1	MobileNetV2
Epochs	10	30	10	5	10	10	10
Atelectasis	0.79	0.71	0.7	0.75	0.78	**0.88**	0.66
Cardiomegaly	0.88	0.77	0.86	0.73	0.75	0.73	**0.89**
Consolidation	0.79	0.81	0.87	0.85	0.77	**0.89**	0.74
Edema	**0.88**	0.74	0.82	0.76	0.87	0.82	0.84
Effusion	0.87	**0.89**	0.79	0.87	**0.89**	0.88	0.7
Emphysema	**0.89**	0.78	0.88	0.78	0.86	0.69	0.79
Fibroses	0.76	0.83	0.74	0.82	0.79	0.8	**0.87**
Hernia	0.81	**0.89**	0.8	0.84	0.75	0.72	0.74
Infiltration	0.71	0.8	**0.87**	0.81	0.81	**0.87**	0.75
Mass	0.82	**0.89**	0.75	**0.89**	**0.89**	0.78	0.85
No Finding	-	0.77	0.79	0.79	**0.88**	0.81	0.71
Nodule	0.74	0.86	0.78	0.72	0.73	0.75	0.77
Pleural Thickening	0.76	0.88	0.86	0.77	0.79	0.9	**0.87**
Pneumonia	0.73	0.7	0.72	0.69	0.77	**0.8**	0.76
Pneumothorax	**0.88**	0.75	0.76	**0.88**	0.69	0.79	0.75
Average	**0.81**	0.8	0.8	0.8	**0.8**	**0.81**	0.78

TABLE 6. Results with MLBCE loss with 224 * 224 image resolution.

Epochs	5	5	5	5	5	10	10	10	10	10
Batch Size	4	8	16	32	64	4	8	16	32	64
Binary Accuracy	72.8	74.5	67.4	63.7	73	64.4	71	69.8	70	68.7
Recall	64.7	54.7	44.4	49.4	55	67.3	66.8	61.6	57.3	68
Precision	28.8	35	47.5	41.6	38.2	33.8	35.1	34.3	35.8	28.1
F1 Score	40	42.6	46	45.2	45.1	45	46	44.1	44.1	39.8
Specificity	72	73.5	65.6	62.2	71.7	62	69.5	68	68.4	66.8
Average AUC	77.3	80.3	80.5	80.8	80	78.6	81.1	80.7	80.5	81.2

class was calculated by multiplying the ratio of positive and negative samples of the class, in the simple BCE formula. This loss function pushed the Recall higher but the Precision and F1-Score remained very low at a smaller classification threshold. This indicates that the model was classifying a huge number of samples as positive, even the negative ones. However, when using a threshold of 0.9, the metrics' values normalized but remained very poor. Table 6 shows the values for various evaluation measures with MLBCE. The loss function was tested on 5 different batch sizes with an input size of 224 by 224 and was discarded for the rest of the research as the results were not encouraging.

Conventional binary cross entropy loss produced better results as compared to the custom loss. AUC did not improve, however, other metrics had higher values. Table 7 demonstrates the results of classification while using BCE loss. A similar approach was used to test BCE i.e. results were examined with 5 batch sizes with an input size of 224 by 224. We also tried a higher number of epochs to see if the results improved or not.

6) BATCH SIZE

Different batch sizes were also tested on this dataset. However, the results indicated that the lower batch sizes were producing better results. Results with batch sizes 8, 16, and 32 were a little better than batch sizes 64 or 128. Batch size of 16 was used in most of the experiments. Table 8 shows the results with respect to different batch sizes with 5, 10, and 15 epochs, respectively. The input image size was 224 * 224 and the loss used was BCE.

7) INPUT IMAGE RESOLUTION/SIZE

Image input sizes were also tweaked with the MobileNet model to increase image quality and detail. Though the model can benefit from higher resolution images, the same can be counterproductive in terms of a drastic increase in computational cost.

Table 9 shows MobileNet results of different input sizes and resolutions with a batch size of 16. Resolution of 500 by 500 again produces better results; however, the training time is twice the number of epochs. There is not a huge difference

TABLE 7. Results with BCE loss with 224 * 224 image resolution.

Epochs	**5**	**5**	**5**	**5**	**5**	**10**	**10**	**10**	**10**	**10**	**15**	**15**	**15**	**15**	**15**
Batch Size	4	8	16	32	64	4	8	16	32	**64**	4	8	16	32	64
Binary Accuracy	92	93	93	93	93	92	93	93	93	93	93	93	93	92.8	93.2
Recall	43	**56**	55	55	53	43	**56**	55	55	55	55	**56**	55	52.7	55.2
Precision	54	54	56	56	57	54	53	55	55	55	55	53	51	52	53.6
F1 Score	48	**55**	**55**	**55**	**55**	48	54	**55**	**55**	**55**	**55**	54	53	52.4	54.4
Specificity	97	98	98	**99**	**99**	97	98	98	98	98	98	98	98	97.5	98
Average AUC	54	**80**	**80**	**80**	79	53	**80**	**80**	**80**	79	**80**	79	79	76.6	79.5

TABLE 8. Results of batch sizes with 5, 10 & 15 Epochs with 224 * 224 image resolution.

Epochs	5	5	5	5	5	10	10	10	10	10	15	15	15	15	15
Batch Size	4	8	16	32	64	4	8	16	32	64	4	8	16	32	64
Binary Accuracy	92	93	93	93	93	92	93	93	93	93	93	93	93	92.8	93.2
Recall	43	**56**	55	55	53	43	**56**	55	55	55	55	**56**	55	52.7	55.2
Precision	54	54	56	56	**57**	54	53	55	55	**55**	**55**	53	51	52	53.6
F1 Score	48	55	55	**55**	55	48	54	**55**	**55**	**55**	55	54	53	52.4	54.4
Specificity	97	98	98	99	99	97	98	98	98	98	98	98	98	97.5	98
Average AUC	54	**80**	80	80	79	53	80	**80**	**80**	79	**80**	79	79	76.6	79.5

TABLE 9. Results of resolution 224, 300, 400 and 500 with batch sizes 8 and 16.

Time (Hours)	4.1	5.5	7.5	10	8.2	10	14	19.4	4.1	5.2	7.6	9.5	8	10.6	14.9	20.1
Batch Size	8	8	8	8	8	8	8	8	16	16	16	16	16	16	16	16
Epochs	5	5	5	5	10	10	10	10	5	5	5	5	10	10	10	10
Resolution	224	300	400	500	224	300	400	500	224	300	400	500	224	300	400	500
Binary Accuracy	93	93	93	93	93	93	93	93.3	93	93	93	94	93	93.3	93.3	93.4
Recall	56	55	57	56	56	**57**	55	56.6	55	56	56	**57**	55	56.1	55.6	55.2
Precision	54	55	54	**55**	53	54	55	54.2	**56**	55	**56**	55	55	53.4	55.1	54.6
F1 Score	55	55	55	**56**	54	55	55	55.4	55	55	56	**56**	55	54.7	55.2	54.9
Specificity	98	99	99	**99**	98	98	98	98	98	99	98	99	98	98	98	98.5
Average AUC	80	81	**81**	81	80	80	81	80.8	80	81	81	**81**	80	80	80.5	81

TABLE 10. Results of image normalization techniques without sharpening (300 * 300 image resolution and 16 batch size).

Epochs	5	5	5	10	10	10
Type of Normalization	One-Minus-One	Min-Max	Zero One	One-Minus-One	Min-Max	Zero One
Binary Accuracy	93.4	93.4	93.4	93.2	93.3	93.2
Recall	56.8	56	57	55.3	56	55.3
Precision	54.6	55.4	**54.5**	53.6	53	54
F1 Score	55.7	55.7	55.7	54.5	54.5	54.6
Specificity	98.4	98.4	98.2	98.1	98	97.7
Average AUC	81.8	81.2	80.7	80	80.3	80

between average AUCs of 300 by 300 and 500, but we do save time with the former one.

Observation 2: We can obtain the best overall results by choosing a small batch size i.e. 16, and by keeping the resolution around 300 by 300, and using BCE as a loss function. **Results after Image Preprocessing"** This experiment involves image normalization/preprocessing along with Threshold segmentation. The image resolution was increased which resulted in increased computational cost. However, 224 by 224 seems to work better most of the time. The following 3 types of scaling were tried:

- One-Minus-One scaling: The pixel values are normalized between -1 and 1
- Zero-One scaling: The values are between 0 and 1
- Min-Max scaling: The values are set considering the minimum and maximum values of the particular image.

Table 10 presents the results of MobileNet with BCE loss after applying three types of mentioned normalization techniques. The batch size was 16 and the input image size was 300 by 300.

8) IMAGE NORMALIZATION (WITH IMAGE SHARPENING)

Table 11 presents the results of MobileNet with BCE loss after applying three types of mentioned normalization techniques with an added filter that applies sharpening on the images. The batch size was 16 and the input image size was 300 by 300.

9) THRESHOLD SEGMENTATION WITH BINARY OTSU-THRESHOLDING

Image Segmentation was also applied using OTSU thresholding. This technique determines the optimum threshold to separate the foreground and background. However, it did not produce encouraging results, as it is hard to calculate the thresholds for lung tissues. Table 12 presents the results of the MobileNet model after applying OTSU. The batch size was 16 and the input image size was 300 by 300. The results were documented with three image normalization techniques specified earlier. We also tried OTSU without any normalization as well as only image sharpening.

TABLE 11. Results of image normalization techniques with sharpening (300 * 300 image resolution and 16 batch size).

Epochs	5	5	5	10	10	10
Type of Normalization	One-Minus-One	Min-Max	Zero One	One-Minus-One	Min-Max	Zero One
Binary Accuracy	93.4	93.4	93.4	93.2	93.3	93.3
Recall	56.7	55.1	56.8	55.4	55.7	56.7
Precision	55	56	**54.3**	53.4	53.2	53.6
F1 Score	55.8	55.5	55.5	54.4	54.4	55.1
Specificity	98.5	98.5	98.5	98	98.1	98.1
Average AUC	80.7	81	80.6	80	80.4	80.7

TABLE 12. Results after applying OTSU Thresholding with 300 * 300 image resolution and 16 batch size.

Epochs	5	5	5	5	5	10	10	10	10	10
Sharpening	-	-	-	-	Yes	-	-	-	-	Yes
One Zero	-	Yes	-	-	-	-	Yes	-	-	-
Min-Max	-	-	Yes	-	-	-	-	Yes	-	-
One Minus One	-	-	-	Yes	-	-	-	-	Yes	-
Binary Accuracy	93.2	93.2	93.2	93.4	93.1	93.2	93.2	93.2	93.2	93
Recall	51.3	52	52.8	**56.6**	52.6	53	52.7	52.2	**57**	53.7
Precision	54	53.6	52.5	54.3	52.8	53	53.5	53.6	51.2	50.6
F1 Score	52.5	52.8	52.6	**55.4**	52.6	53	53.1	53	54	52.1
Specificity	98.5	98.7	98.7	98.4	99	98.6	98.5	98.4	98	98.5
Average AUC	76.8	77.8	77.4	**81**	76.4	77.7	78	78	80.6	77.3

Observation 3: We can conclude from this experiment that the best normalization technique for this project is to keep pixel values between -1 and 1 or 0 and 1. Threshold segmentation does not aid lung disease classification significantly. Hence, semantic segmentation might be a better option.

10) RESULTS AFTER FINE TUNING

In addition, we've investigated the process of further optimizing the top-performing model through fine-tuning as part of our research. The following section shows the results after we froze the convolutional layers of the model. MobileNet's convolutional part which is used for feature extraction, was frozen and only the custom classifier was trained on different epochs. The batch size was kept at 16. It seems an increasing number of epochs do increase the values of evaluation measures. The increase in the results was minor and the higher epochs were taking too long to be continued in the research's limited time frame. Table 13 shows a comparative summary of the results. MobileNet model has trained with a batch of 16 on 10, 20, 30, 40, and 70 epochs. Different values were tried for the batch size but it did not seem to have any significant effect.

TABLE 13. MobileNetV1 results after fine tuning (frozen CNN & trained classifier) with 300 * 300 image resolution and 16 batch size.

Time (Hours)	8	17	39.6	53	90
Epochs	10	20	30	40	70
Binary Accuracy	93.1	93.1	93.1	93.2	93.2
Recall	49.7	49.8	50	51	**51.5**
Precision	52.7	52.4	52.3	52	**53.2**
F1 Score	51.2	51.1	51.1	51.5	**52.3**
Specificity	98.7	98.8	98.8	98.7	98.7
Average AUC	75.7	75.7	76.1	76.8	**78**

Observation 4: Fine-tuning the fully connected layer and freezing convolutional layers of the model slows down the convergence of the algorithm significantly.

11) RESULTS AFTER USING SYNTHETIC SAMPLES PRODUCED BY GAN

Table 14 presents the results of different models, trained on original as well as synthetic samples. Specifically, MobileNetV1 model was utilized. Images of approximately 224 * 224 in size were fed, and the results are documented.

TABLE 14. MobileNetV1 results after using GAN samples with 224 * 224 image resolution.

Epochs	5	10
Binary Accuracy	93.1	93
Recall	50	49.8
Precision	58.2	56.2
F1-Score	54.1	53
Specificity	99.1	98.7
Average AUC	78.4	78.8

Observation 5: Synthetic samples generated with GAN have not proved to be fruitful. Recall and AUC decreased after training the model with merged samples. However, if the quality of synthetic samples can be improved, the results might get better.

E. FINAL COMPARISON OF RESULTS OF EXISTING BEST MODEL (MOBILENETV2) AND SUGGESTED MODEL (MOBILENETV1)

Table 15 shows a summary of base papers and our project's results including Recall, Precision, F1-Score, average AUC, etc.

V. DISCUSSION

Lung diseases are one of the most common causes of death in the world. The diagnosis and management of lung diseases are a challenge for radiologists, especially in environments with scarce resources. The use of deep learning techniques in the fields of medical imaging, on large datasets, has allowed computer algorithms to produce as effective results as medical professionals. In this research work, we have

TABLE 15. Final comparison of results of existing and proposed approach.

	Existing Approach	Proposed Approach	
	MobileNetV2	MobileNetV1	MobileNetV1
Synthetic Sample	No	No	Yes
Binary Accuracy	90.2	93.4	93.1
Recall	45.3	57	50
Precision	-	54	58.2
F1-Score	55.6	55.3	54.1
Specificity	97.3	98.4	99.1
Average AUC	81	81	78.4

proposed a deep learning-based system for the classification of lung diseases from chest X-ray images. The system uses a MobileNetV1 and neural network classifier with geometric image augmentation. We have experimented with various aspects related to deep learning for the multi-label classification of lung disease. The results of our experiments show that deep learning models can be used to effectively classify lung diseases from chest X-ray images.

Following are some of the observations from this research:

- Our experiments have shown that MobileNet (version 1 and 2) models produce slightly better results than other CNN-based models such as VGG-16, DenseNet, InceptionNet, etc. MobileNet models are lightweight and efficient, which makes them well-suited for mobile and embedded devices. They also have a relatively small number of parameters, which makes them easier to train and deploy.
- Using a small batch size of 16, keeping the resolution around 300 by 300, and using BCE as a loss function produced the best overall results. The mentioned resolution provides a good balance between classification accuracy and computational cost. Furthermore, the best normalization technique for this project is to keep pixel values between -1 and 1 or 0 and 1.
- Threshold segmentation is a simple but effective technique for segmenting images. However, it may not be sufficient for lung disease classification, as it does not take into account the spatial relationships between pixels. Semantic segmentation is a more advanced technique that can take into account these relationships, and it may be a better option for lung X-ray segmentation.
- Fine-tuning the fully connected layer and freezing convolutional layers of the model slows down the convergence of the algorithm significantly.

VI. CONCLUSION

MobileNetV1 model with geometric image augmentation produced the best results, with a recall of 57%, a binary accuracy of 93.4%, an F1-score of 55.3%, and an AUC of 81%. Synthetic samples generated with GAN can help to improve the diversity of the training data and help with class imbalance in large datasets. We have implemented a DCGAN model to generate synthetic X-ray images, but we found that these images did not improve the performance of the MobileNetV1 model. After the inclusion of generated synthetic samples, the values for Recall, Precision, F1-Score, and AUC were 50, 58.2, 54.1, and 78.4, respectively. There are a few possible explanations for why the DCGAN model did not improve the performance of the MobileNetV1 model. First, the DCGAN model was trained on a relatively small number of samples for each class, which may not have been enough to capture the diversity of lung diseases in the NIH Chest X-ray 14 dataset. Second, the DCGAN model may have been overfitting to the training data, and so did we notice, which would have led to poor classification accuracy. Overall, the results of our experiments suggest that deep learning models can be used to classify lung diseases from chest X-ray images effectively. We believe that our system has the potential to be used as a clinical decision support tool for the early detection of lung diseases. However, more research is needed to improve the performance of such models, especially on large and imbalanced datasets. In future work, we plan to improve the performance of our system by using a larger and more diverse dataset of chest X-ray images.

ACKNOWLEDGMENT

The authors would like to thank the feedback provided by Rijja Masood, B.S. (CS) Student with the Computer Science Department, National University of Computer and Emerging Sciences, Lahore, Pakistan.

REFERENCES

[1] A. Souid, N. Sakli, and H. Sakli, "Classification and predictions of lung diseases from chest X-rays using MobileNet V2," *Appl. Sci.*, vol. 11, no. 6, p. 2751, Mar. 2021.

[2] A. T. Sahlol, M. A. Elaziz, A. T. Jamal, R. Damaševičius, and O. F. Hassan, "A novel method for detection of tuberculosis in chest radiographs using artificial ecosystem-based optimisation of deep neural network features," *Symmetry*, vol. 12, no. 7, p. 1146, Jul. 2020.

[3] S. N. H. S. Abdullah, F. A. Bohani, B. H. Nayef, S. Sahran, O. Al Akash, R. I. Hussain, and F. Ismail, "Round randomized learning vector quantization for brain tumor imaging," *Comput. Math. Methods Med.*, vol. 2016, pp. 1–19, Jan. 2016.

[4] O. Er, N. Yumusak, and F. Temurtas, "Chest diseases diagnosis using artificial neural networks," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7648–7655, Dec. 2010.

[5] H. Hu, Q. Li, Y. Zhao, and Y. Zhang, "Parallel deep learning algorithms with hybrid attention mechanism for image segmentation of lung tumors," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2880–2889, Apr. 2021.

[6] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, p. 2402, Dec. 2016.

[7] M. Hu, H. Lin, Z. Fan, W. Gao, L. Yang, C. Liu, and Q. Song, "Learning to recognize chest-Xray images faster and more efficiently based on multi-kernel depthwise convolution," *IEEE Access*, vol. 8, pp. 37265–37274, 2020.

[8] S. Guendel, S. Grbic, B. Georgescu, S. Liu, A. Maier, and D. Comaniciu, "Learning to recognize abnormalities in chest X-rays with location-aware dense networks," in *Proc. 23rd Iberoamerican Congr. Pattern Recognit.*, Madrid, Spain. Cham, Switzerland: Springer, Nov. 2018, pp. 757–765.

[9] P. Rajpurkar, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists," *PLOS Med.*, vol. 15, no. 11, Nov. 2018, Art. no. e1002686.

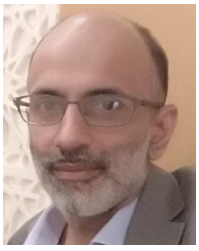
[10] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach, "Comparison of deep learning approaches for multi-label chest X-ray classification," *Sci. Rep.*, vol. 9, no. 1, p. 6381, Apr. 2019.

[11] T. Malygina, E. Ericheva, and I. Drokin, "Data augmentation with gan: Improving chest X-ray pathologies prediction on class-imbalanced cases," in *Proc. Int. Conf. Anal. Images, Social Netw. Texts*. Cham, Switzerland: Springer, 2019, pp. 321–334.
[12] F. Demir, A. Sengur, and V. Bajaj, "Convolutional neural networks based efficient approach for classification of lung diseases," *Health Inf. Sci. Syst.*, vol. 8, no. 1, pp. 1–8, Dec. 2020.
[13] G. B. Kim, K.-H. Jung, Y. Lee, H.-J. Kim, N. Kim, S. Jun, J. B. Seo, and D. A. Lynch, "Comparison of shallow and deep learning methods on classifying the regional pattern of diffuse lung disease," *J. Digit. Imag.*, vol. 31, no. 4, pp. 415–424, Aug. 2018.
[14] A. S. Pillai, "Multi-label chest X-ray classification via deep learning," 2022, *arXiv:2211.14929*.
[15] S. Motamed, P. Rogalla, and F. Khalvati, "Data augmentation using generative adversarial networks (GANs) for GAN-based detection of pneumonia and COVID-19 in chest X-ray images," *Informat. Med. Unlocked*, vol. 27, Jan. 2021, Art. no. 100779.
[16] F. Munawar, S. Azmat, T. Iqbal, C. Grönlund, and H. Ali, "Segmentation of lungs in chest X-ray image using generative adversarial networks," *IEEE Access*, vol. 8, pp. 153535–153545, 2020.
[17] M. Zak and A. Krzyżak, "Classification of lung diseases using deep learning models," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2020, pp. 621–634.
[18] Z. Tariq, S. K. Shah, and Y. Lee, "Lung disease classification using deep convolutional neural network," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2019, pp. 732–735.
[19] K. V. Priya and J. D. Peter, "A federated approach for detecting the chest diseases using DenseNet for multi-label classification," *Complex Intell. Syst.*, vol. 8, no. 4, pp. 3121–3129, Aug. 2022.
[20] R. Imran, N. Hassan, R. Tariq, L. Amjad, and A. Wali, "Intracranial brain Haemorrhage segmentation and classification," *iKSP J. Comput. Sci. Eng.*, vol. 1, no. 2, pp. 52–56, 2021.
[21] A. Wali, A. Naseer, M. Tamoor, and S. A. M. Gilani, "Recent progress in digital image restoration techniques: A review," *Digit. Signal Process.*, vol. 141, Sep. 2023, Art. no. 104187.
[22] S. Bharati, P. Podder, and M. R. H. Mondal, "Hybrid deep learning for detecting lung diseases from X-ray images," *Informat. Med. Unlocked*, vol. 20, Jan. 2020, Art. no. 100391.
[23] F. J. M. Shamrat, S. Azam, A. Karim, K. Ahmed, F. M. Bui, and F. De Boer, "High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images," *Comput. Biol. Med.*, vol. 155, Mar. 2023, Art. no. 106646.
[24] M. H. Al-Sheikh, O. Al Dandan, A. S. Al-Shamayleh, H. A. Jalab, and R. W. Ibrahim, "Multi-class deep learning architecture for classifying lung diseases from chest X-ray and CT images," *Sci. Rep.*, vol. 13, no. 1, p. 19373, Nov. 2023.
[25] Z. Jiang, W. Zaheer, A. Wali, and S. A. M. Gilani, "Visual sentiment analysis using data-augmented deep transfer learning techniques," *Multimedia Tools Appl.*, vol. 83, no. 6, pp. 17233–17249, Jul. 2023.
[26] A. Wali and M. Saeed, "Biologically inspired cellular automata learning and prediction model for handwritten pattern recognition," *Biologically Inspired Cognit. Archit.*, vol. 24, pp. 77–86, Apr. 2018.
[27] A. Fawaz, M. B. Ali, M. Adan, M. Mujtaba, and A. Wali, "A deep learning framework for efficient high-fidelity speech synthesis: Styletts," *iKSP J. Comput. Sci. Eng.*, vol. 1, no. 1, pp. 1–11, 2021.
[28] H. M. Hamza and A. Wali, "Pakistan sign language recognition: Leveraging deep learning models with limited dataset," *Mach. Vis. Appl.*, vol. 34, no. 5, p. 71, Sep. 2023.
[29] A. Wali, M. Ahmad, A. Naseer, M. Tamoor, and S. A. M. Gilani, "StynMedGAN: Medical images augmentation using a new GAN model for improved diagnosis of diseases," *J. Intell. Fuzzy Syst.*, vol. 44, no. 6, pp. 10027–10044, Jun. 2023.
[30] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1406.2661*.
[31] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

MUHAMMAD IRTAZA received the master's degree in computer science from the FAST National University of Computer and Emerging Sciences, Lahore. His expertise lies in the fields of computer vision and image processing, with a specific focus on medical and biomedical imaging. He specializes in the application of advanced image processing techniques to medical diagnostics, treatment methodologies, and the development of cutting-edge imaging technologies for biological and medical applications, including MRI, CT scans, and ultrasound.

ARSHAD ALI received the M.Sc. degree in computer science from Punjab University, Lahore, in 2003, the master's degree in information technology from the University of Avignon, France, in 2009, and the joint Ph.D. degree in telecommunication and computer science from the Institute of Telecom SudParis and UPMC (Paris VI), in 2012. He is currently an Associate Professor and the Head of the Cyber Security Department, National University of Computer Science and Emerging Sciences, Lahore Campus, Pakistan. He was a Postdoctoral Researcher with the Orange Laboratories, Paris. His research interests include mobile ad-hoc networks, AI with cyber security, NLP, and AI in healthcare and agriculture.

MARYAM GULZAR received the B.S. degree in computer science from the CS Department, University of Lahore (UOL), Pakistan, in 2016, and the M.S. (SPM) degree from the School of Computing, National University of Computer and Emerging Sciences (NUCES), Lahore Campus, Pakistan, in 2018. She is currently pursuing the Ph.D. degree with the Department of Software Engineering, LUT University, Finland. She is also a Junior Researcher with the Department of Software Engineering, LUT University. Earlier, she was a Lecturer with the Department of Software Engineering, University of Lahore, Pakistan. Her research interests include digital transmission, information systems, and machine learning. She won a Research Grant from the Higher Education Commission of Pakistan for four years to complete her Ph.D. studies.



AAMIR WALI received the Ph.D. degree in computer science from the FAST National University of Computer and Emerging Sciences. He has been teaching with the Department of Computer Science, FAST National University of Computer and Emerging Sciences, since 2004. His research interests include font development, writing systems, machine learning, image processing, human–computer interaction, and virtual/augmented reality.