Detecting Pneumonia with a Deep Learning Model and Random Data Augmentation Techniques

Tawfik Guesmi
Department of Electrical Engineering, College of Engineering University of Ha'il
Ha'il 2240, Saudi Arabia

***Abstract*—This research paper presents an investigation into the detection of pneumonia using deep learning models and data augmentation techniques. The study compares and evaluates the performance of different models based on experimental results. The proposed model consists of multiple convolutional layers and maxpooling layers. Extensive experiments were conducted on a dataset, and the results demonstrate the efficiency and accuracy of our approach. The findings highlight the potential of deep learning in pneumonia detection and contribute to the existing body of knowledge in this field. The implications of this research can have a significant impact on improving diagnostic accuracy and patient outcomes. Future research directions could explore further enhancements in the model architecture, investigate additional data augmentation techniques, and consider larger datasets for more comprehensive evaluations.**

***Keywords*—Deep learning; pneumonia detection; convolutional neural network; random data augmentation**

I. INTRODUCTION

The capabilities of e-health tools have been recently boosted and improved by advances in artificial intelligence (AI) that allows the detection and diagnosis of diseases. Artificial intelligence is not a newly-invented technique. In fact, it was prefigured in a chess computer program that was generated by Alan Turing in 1950 [1]. The health sector has not been deprived of these technological advances. Indeed, there has been considerable and growing interest in this health sector and especially in the automatic detection of diseases from medical images. As a subdomain of AI, machine learning makes use of algorithms so as to parse date, gain an understanding from the results, and apply the learning to make decisions and predictions. Thanks to the rise in computing power and the availability of huge datasets, researchers have proposed many new approaches of smart healthcare disease diagnosis and patient management by using machine learning and especially deep learning algorithms. We note that machine learning algorithms have been developed to detect objects or faces, to assist in healthcare, traffic prediction, natural disasters prediction, etc. In our research work, we are focused on the healthcare services by incorporating AI through disease detection and prediction using machine learning and deep learning. The proposed approach harnesses the benefits of AI-based systems in the medical diagnosis field by replicating human brain function for simple tasks and providing innovative solutions for more complex ones. Towards reaching our objective, we recommend implementing AI-based solutions. Our primary algorithmic approach includes machine learning, particularly deep learning algorithms, which provide computational models for learning data representations. We note that these algorithms have greatly improved previous disease detection and recognition efforts [2].

Deep learning reveals a complex structure in high-dimensional data such as images and videos by using the back-propagation algorithm. The goal is to help a machine regulate its internal parameters to enable it to generate a configuration in each layer from the previous one. Being the most representative model of deep learning, CNN has been broadly put in application in many research areas, such as image classification, face recognition and object detection. It's composed of the input layer, hidden layers (at least one), and an output layer. Constructing a typical CNN takes some steps. The starting phase involves two types of layers: convolutional layers and pooling layers. In the proposed approach, a convolutional layer contains filters characterized with parameters that should be learned. Thus, the filters' height and weight tend to be inferior to those of the input volume. Then an activation map based on neurons is computed by convolving each filter with the input volume. The concluding phase consists in obtaining the convolutional layer's output volume. This is achieved by stacking all filters' activation maps along the depth dimension [3].

The proposed CNN architecture, along with the well-known pre-trained models DenseNet and MobileNet, is evaluated and compared in terms of their matching performance and computational cost. Furthermore, the incorporation of random data augmentation techniques enhances the model's ability to generalize to new and unseen images, improving its robustness and reducing the risk of overfitting. The experimental results demonstrate that the proposed CNN model outperforms the existing models in terms of accuracy and provides a promising solution for accurate and efficient pneumonia detection. By highlighting the value of this paper, we contribute to the advancement of AI-based systems in medical diagnosis, specifically in the detection and diagnosis of pneumonia, which can lead to improved healthcare outcomes, more timely treatments, and ultimately, saving lives.

The main objective of the proposed approach is to detect pneumonia from chest X-ray datasets. Convolutional neural networks (CNNs) are effective tools for image understanding and are widely used in medical image analysis. For these reasons, we have used two very well known and very successful CNNs which are: DenseNet and MobileNet in order to test them to detect pneumonia disease. The results obtained were compared to our CNN model. For this, we used four different datasets to validate the results. Note that we have used different data augmentation techniques to overcome the problem of limited datasets [4]. In many computer vision tasks, the original dataset may be limited in size and may not reflect the variability of the real-world scenarios. For example, in an image classification task, the model may only see pictures of dogs taken from one angle and with a specific resolution. This lack of diversity in the training data can lead to poor performance when the model encounters new, unseen data. The main contribution and results are:

- The proposed CNN model, DenseNet and MobileNet showed performance improvements on augmented datasets.
- Our CNN model performed better than the other models.
- The results were validated on four different datasets. A comparative table has been drawn up for this purpose.
- We used random data augmentation techniques such as randomly flipping, zooming, shifting, and rotating images which can be highly beneficial for training image processing and computer vision models. These techniques can help to artificially increase the size of the training dataset and expose the model to a wider variety of image variations. It improves the model's ability to generalize to new images, making it more robust and less prone to overfitting.

In fact, using the classic techniques of data augmentation can be less beneficial as the model will be exposed only to the flipped version and it may not generalize well to the original version of the image in the case of the flipping technique. However, randomly flipping images can be more beneficial. It is important to expose the model to a diverse set of training data. Randomly flipping images horizontally or vertically can be a way to artificially increase the diversity of the training dataset by creating new images from existing ones. By randomly flipping images, the model is exposed to both the original image and its flipped version, which can help it learn to recognize objects regardless of their orientation.

Previous research in pneumonia detection has primarily focused on traditional machine learning algorithms and a limited set of image features. These approaches often struggle to capture the complex patterns and variations present in chest X-ray images, resulting in suboptimal performance and limited generalization capabilities. Furthermore, the use of pre-trained convolutional neural networks (CNNs) in this domain has been limited, and their potential for pneumonia detection remains underexplored. In this paper, we aim to address the gap between the existing approaches and the potential for leveraging deep learning techniques, specifically CNNs, for improved pneumonia detection. Our proposed work presents a detailed review of various CNN architectures, including well-known models such as DenseNet and MobileNet, and their characteristics. We then introduce an efficient CNN architecture for pneumonia detection using X-ray images, incorporating random data augmentation techniques. By leveraging the power of deep learning and exploring the potential of CNN models, we aim to overcome the limitations of existing approaches and achieve enhanced performance in pneumonia detection.

This paper is organized as follows: Section II explores the related research done in the same field. In Section III, there is a brief description of the two deep convolutional neural networks: DenseNet and MobileNet. In Section IV, the description of the applied methodology and the proposed CNN architecture. Section V presents the experimental result and performance analysis. Finally, Section VI shows the results and discussion and Section VII concludes this paper.

II. Related Works

In recent years, deep learning has opened up horizons for researchers in the field of medical sciences. Published research is promising. These studies were done to test the detection, prediction and diagnosis of disease. Today, the enormous progress and advances of CNNs have attracted the attention of researchers to apply them in many fields. Medical research is one of the most sought-after fields. All the details and features in a medical image are of high importance in the machine learning pipeline. The problem is that most known ML algorithms used classical features to develop detection and recognition systems [5]–[7]. In contrast, the use of deep learning (DL) models, in particular convolutional neural networks (CNN), has demonstrated a strong ability to extract relevant features in the image classification framework [8], [9]. Image classification can be significantly improved if we have a very rich set of extracted features. Indeed, the availability of pre-trained CNN models like MobileNet [10], AlexNet [11], ResNet [12] and DenseNet [13] speeds up and improves the relevant feature extraction procedure. Several interesting research papers on the disease of pneumonia have been published with the aim of classifying chest X-ray images [14]–[19]. In [20], authors implemented a deep convolution neural network on more than 100 thousand x-ray images of approximately 32,000 in order to analyze and recognize pulmonary infection and its subtypes. In [21], Amit Kumar et al. implemented a Mask-RCNN which performed a combination of pulmonary image segmentation and an image augmentation. They started by testing known detection techniques such as YOLO 3 and U-Net but the results were not motivating. They then proposed their own model based on Mask-RCNN and showed in the experimental results that the proposed identification model achieves better performance.

In order to take advantage of the characteristics of the Inception V3 model, authors, in [22], have implemented a CNN model based on Inception V3. The authors were able to successfully classify various types of pneumonia infections on pediatric patients. They developed a new CNN model not only to classify images into class of sick people and non-sick people but also to classify images showing pneumonia disease into two categories: pneumonia caused by bacteria and pneumonia caused by a virus.

A novel approach for automatic detection of pneumonia was proposed by Anuja Kumar et al. in [23]. In fact, they proposed a deep Siamese neural network by analyzing the amount of white substance presence on both the right and the left chest of X-ray image. In their approach, Paras et al. [24] were inspired by pre-trained AlexNet and GoogleNet data models as well as data augmentation. The authors in [25], developed numerous models in order to validate an accurate result in detecting pneumonia. They trained AlexNet, LeNet, GoogleNet, ResNet, and VGGNet on a dataset of over 26 thousand images of a resolution of 1024x1024. Vikash et al. proposed a novel approach for detection of pneumonia based on transfer learning and ImageNet model [26]. In [27], pneumonia was one of 14 different diseases that were detected using a 121-layer CNN on chest x-rays. In [28], authors developed an automated diagnosis of pneumonia by classifying X-ray images using deep CNN. They showed that the proposed model reached 91% of accuracy. In [29], authors proposed a novel deep convolutional neural network architecture to extract relevant features from chest X-ray images and classify them into two classes. They also studied in their paper, the influence of the size of the dataset on the performance of the model. They used the original dataset as well as its augmented version.

III. Deep Convolutional Neural Networks

A Convolutional Neural Network (CNN) is composed of neurons with varying weights and biases. These neurons receive inputs from preceding layers, producing a fast and precise algorithm [30], [31]. CNNs have proven to outperform traditional neural networks in detection and classification tasks, as seen in their successful classification of well-known image databases such as MNIST [32], [33] and CIFAR 10 [34], [35].

A. Convolution Layer

A convolution layer in deep learning is a layer in a neural network that performs a mathematical operation called convolution on the input data. The convolution operation involves sliding a small matrix (the "filter" or "kernel") over the input data and computing a dot product at each position, producing a feature map that represents important information from the input. This operation is repeated with multiple filters, effectively learning different features at different scales, and allowing the network to learn complex representations of the data. Convolution layers are commonly used in computer vision tasks, such as image classification and object detection. The formula for computing a single output element in a convolution operation is given as follows.

$$O_{i,j} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I_{i+m,j+n} \cdot F_{m,n} \tag{1}$$

where $O_{i,j}$ is the (i,j)th element of the output feature map, $I_{i,j}$ is the (i,j)th element of the input feature map, $F_{m,n}$ is the (m,n)th element of the filter (also called kernel) matrix and k is the size of the filter. This formula is applied element-wise for each position of the filter over the input feature map, with the result being a new output feature map that represents the filtered version of the input.

B. Activation Function

An activation function in deep learning is a non-linear function applied to the output of each neuron in a neural network. The activation function is used to introduce non-linearity into the model, allowing it to model complex relationships in the data. There are several commonly used activation functions, including (Fig. 1):

- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$
- Tanh: $f(x) = \tanh(x)$
- ReLU (Rectified Linear Unit): $f(x) = \max(0, x)$
- Leaky ReLU: $f(x) = \max(0.01x, x)$
- Softmax: used for multiclass classification, maps inputs to a probability distribution over the classes.

C. DenseNet

DenseNet is a network architecture characterized by the fact that each layer is directly connected to all the others. Feature maps from all layers that precede another are treated as separate inputs. On the other hand, the layers following any layer, are fed by its own feature maps. This connectivity model gives state-of-the-art accuracies on CIFAR10/100 (with or without data augmentation) [13], [36]. It's architecture is detailed in Table I.

TABLE I. DenseNet Architecture

Layers	Output Size
Convolution	112x112
Pooling	56x56
DenseBlock (1)	56x56
Transition Layer (1)	56x56—28x28
Dense Block (2)	28x28
Transition Layer (2)	28x28—14x14
Dense Block (3)	14x14
Transition Layer (3)	14x14—7x7
Dense Block (4)	7x7
Classification Layer	1x1

D. MobileNet

MobileNet is a CNN architecture that is among the first CNN models that aims to be deployed on mobile applications. The main innovation is that the convolutions are separable according to the depth. A separable convolution transforms a classical convolution kernel into two separate kernels. For example, a 4x4 kernel turns into a 4x1 kernel and a 1x4 kernel. The objective behind this separation is to minimize the number of operations needed to perform the convolution. Therefore, the model becomes more efficient. This model is, today, a reference for object detection, face detection, and for object classification. MobileNet model has 27 Convolution layers which includes 13 depthwise Convolution, 1 Average Pool layer, 1 Fully Connected layer and 1 Softmax Layer. This model was developed by Andrew G. Howard and other researchers from Google [10]. It's architecture is detailed in Fig. 2.

IV. The Proposed CNN Architecture

A. Layers Description

a new approach of drawing a CNN model has been proposed in order to classify chest X-ray images. The goal is to classify images into two classes: Normal X-ray image and X-ray image with pneumonia. The CNN architecture is based on:

- Convolutional layers
- Maxpooling layers

The resulting image after the last convolution/maxpooling layer is first flattened and then inserted into a dense layer.

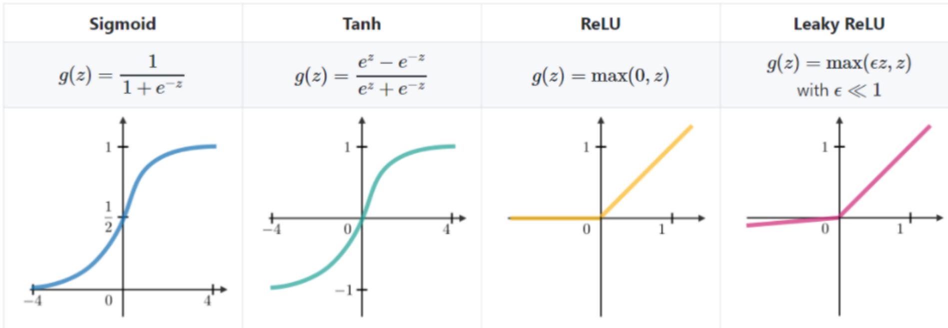

Fig. 1. Activation functions.

Type (Stride)	Filter Shape	Input Size
Conv (s2)	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw(s1)	3 x 3 x 32 dw	112 x 112 x 32
Conv (s1)	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw(s1)	3 x 3 x 64 dw	112 x 112 x 64
Conv (s1)	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw (s1)	3 x 3 x 128 dw	56 x 56 x 128
Conv (s1)	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw (s2)	3 x 3 x 128 dw	56 x 56 x 128
Conv (s1)	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw (s1)	3 x 3 x 256 dw	28 x 28 x 256
Conv (s1)	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw (s2)	3 x 3 x 256 dw	28 x 28 x 256
Conv (s1)	1 x 1 x 256 x 512	14 x 14 x 256
5X	3 x 3 x 512 dw	14 x 14 x 512
	1 x 1 x 512 x 512	14 x 14 x 512
Conv dw (s2)	3 x 3 x 512 dw	14 x 14 x 512
Conv (s1)	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw (s2)	3 x 3 x 1024 dw	7 x 7 x 1024
Conv (s1)	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool (s1)	Pool 7 x 7	7 x 7 x 1024
FC (s1)	1024 x 1000	1 x 1 x 1024
Softmax (s1)	Classifier	1 x 1 x 1000

Fig. 2. The detailed MobileNet architecture.

The sigmoid function is used to activate the layer where the output was introduced. Note also that the sigmoid function was used in the last layer since the classification is binary. Fig. 3 illustrates the architecture of the proposed CNN. The architecture of the proposed CNN is carefully designed to learn and classify pneumonia patterns in chest X-ray images effectively. It utilizes the hierarchical structure of convolutional and pooling layers to capture both local and global features, enabling accurate and robust predictions. The combination of convolutional and dense layers allows the network to learn complex relationships and make informed decisions based on the extracted features. Overall, the proposed CNN architecture offers a powerful tool for pneumonia detection and showcases promising potential in medical image analysis.

Initially, the image is resized to the size of (150x150). It is then integrated into a first layer (3x3x16) of sixteen filters and dimensions (3x3). The convolutional layer is then used to decompose the image to have new dimensions of (75x75x32). The latter is integrated into the Maxpooling layer having a window size of (2x2). We finally have an image with a new size. Different layers are listed below.

- The image crosses the second dimension convolutional layer (3x3x32). We will have as output of this layer an image (38x38x64). Shape Maxpooling layer (2x2) is introduced and gives as output an image with a new shape.
- The resulting image is passed through another convolutional layer of the same dimension and which has the same shape as the previous one. In order to detect more relevant details of the image, the latter is again processed by the convolutional layer. Thus, the image reaches a new shape of (19x19x128) and it is introduced through a Maxpooling layer of dimension (2x2).
- 64 filters make up the final layer which is of the shape of (3x3x64). The resulting image has the shape (5x5x256) and will once again be introduced into the maxpooling layer. The end result is a set of finer instances of the image which will help in better classification.

In the following, we move on to the description of the second phase: the deep neural network. After passing through the last layer, the output is flattened and inserted into the Deep Neural Network (DNN). Then, it is introduced into a 128 neurons layer in order to detect the key data of the image and its relevant characteristics. The ReLU function is used as an activation function. The last dense layer of the DNN is a single output neuron. The aim is to classify the chest X-ray images into two classes: images with pneumonia and images without pneumonia.

B. Summary of the CNN Model

The proposed CNN model is summarized as follows:

- Four convolutional layers.

TABLE II. DESCRIPTION OF THE CNN MODEL.

Layer	Filter	Kernel Size	Strid	Size of feautre Maps
Input	-	3 x 3	-	150 x150 x 3
Conv(1)	64	3 x 3	1 x 1	150 x150 x 3
Conv (2)	64	3 x 3	3 x 3	150 x150 x 3
Batch normalization	-	-	-	150 x150 x 3
Pooling		2 x 2	2 x2	75 x 75 x 3
Conv(3)	64	3 x 3	1	75 x 75 x 3
Dropout	-	-	-	75 x 75 x 3
Batch normalization	-	-	-	75 x 75 x 3
pooling	-	2 x 2	2 x 2	38 x 38 x 3
Conv(4)	128	3 x 3	1 x 1	38 x 38 x 3
Batch normalization	-	-	-	38 x 38 x 3
pooling	-	2 x 2	2 x 2	19 x 19 x 3
Conv(5)	128	3 x 3	3 x 3	19 x 19 x 3
dropout	-	-	-	19 x 19 x 3
Batch normalization	-	-	-	19 x 19 x 3
pooling	-	2 x 2	2 x 2	10 x 10 x 3
Conv(6)	256	3 x 3	3 x 3	10 x 10 x 3
Dropout	-	-	-	10 x 10 x 3
Batch normalization	-	-	-	10 x 10 x 3
pooling	-	2 x 2	2 x 2	5x 5 x 3

- Four Maxpooling layers.
- flattened layer of zero parameters.
- Dense layers of about 819328 parameters.
- The total number of parameters that can be trained in the network is 1,246,401 parameters.

Our model is based on the CNN model described in Table II and its architecture is detailed in Fig. 3.

C. Random Data Augmentation

In order to properly implement a CNN, a large dataset is required. In case we have a limited amount of data, we can use random data augmentation techniques which are a solution to artificially increase the amount of existing data. In the case of medical image datasets, the data is not available in large quantities. Random data augmentation is often considered to be better than classic data augmentation because it can increase the diversity of the training data in a more controlled manner. In classic data augmentation, the same transformation is applied to all instances of the data, which can lead to overfitting to the augmented data and decreased performance on the original data. On the other hand, in random data augmentation, different transformations are randomly applied to each instance of the data. This increases the diversity of the training data in a more controlled manner and can help prevent overfitting to the augmented data. By applying different transformations to different instances of the data, random data augmentation can help the model learn to recognize objects regardless of their orientation, scale, and deformation. This can improve the model's generalization ability and increase its robustness to changes in the input data. This is why we applied the data augmentation technique on our training data set. An example of random data augmentation is shown in Fig. 4, 5, 6 and 7. In fact, we added changes to our images by making minor changes, such as:

- Random rotating data: it consists of applying random rotations to images in a dataset in order to increase the diversity of the training data and reduce overfitting. This can be done by specifying a range of rotation

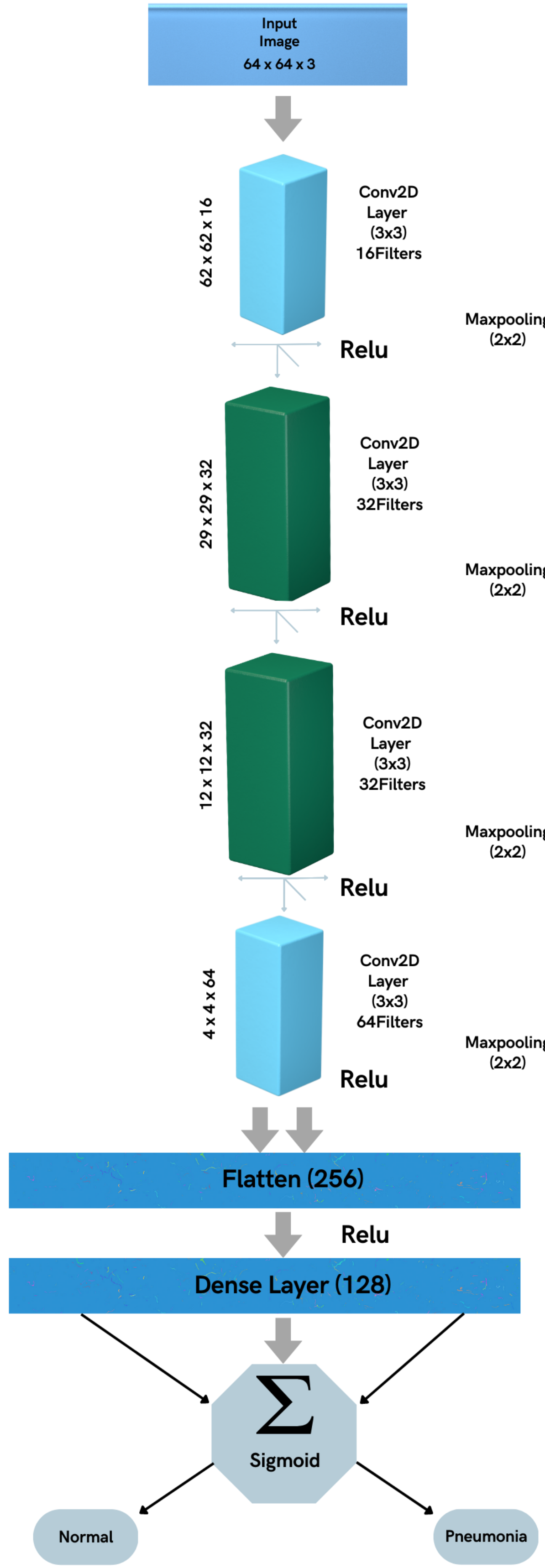

Fig. 3. Architecture of the CNN model.

angles, and then applying a random rotation within that range to each image in the dataset before it is used for training. This technique can be useful for image classification tasks, as it allows the model to learn to recognize objects in different orientations.

- Random Zoom: it involves randomly zooming in or out of an image by a certain percentage, while maintaining the aspect ratio of the original image. This technique can help increase the diversity of the training data and improve the robustness of the model by exposing it to different scales and perspectives of the same object. It can also help to prevent overfitting by making the model more generalizable to new images.
- Random shifting images horizontally or vertically: it consists of randomly shifting the position of an image by a certain number of pixels horizontally or vertically. This technique is used to simulate the effect of objects being slightly misaligned or translated in real-world scenarios. In such scenarios, a model trained on perfectly aligned images may not perform well when presented with images that are not perfectly aligned. However, by training the model on images that have been randomly shifted, the model can learn to be more robust to small changes in position and handle misalignment better.
- Random flipping images horizontally: the left and right sides of the image are switched. This can be done by reflecting the image across a vertical axis. The idea behind this technique is to artificially increase the diversity of the training data by exposing the model to both the original image and its flipped version.Flipping images horizontally can also be useful when the model needs to recognize objects that are symmetric across a vertical axis. For example, in object detection tasks, flipping the image horizontally and training the model on both the original and the flipped images can make the model more robust to detect the object in the image regardless of its orientation.
- Random cropping: the basic idea is to randomly select a rectangular region of an image, and then crop the image to that region. The cropped region is then resized to the original size of the image. We randomly select the starting and ending coordinates of the cropped region within the original image's dimensions. The cropped region is then resized to the original size of the image using interpolation to avoid distorting the image. the cropping parameters such as the size of the cropped region and the aspect ratio can be adjusted. For example, if the model is trained for object detection, the cropping area should be adjusted to keep the object of interest within the crop area. Also, when using random cropping, it is important to make sure that the entire image is covered by the crop area, otherwise important information may be lost.

We note that some data augmentation techniques are domain-specific, for example, in medical images rotating images can be harmful. In general, using a combination of different data augmentation techniques can be more effective than using a single technique, as it can provide the model with a more diverse set of training data. This will increase the size of our training data and our model will consider each of these small changes as a separate picture. In our work we applied the data augmentation Algorithm 1.

Algorithm 1: Random Data Augmentation

Input : Training dataset
Output: Augmented dataset
Procedure
  DataAugmentation
end
for *each image x in the training dataset* **do**
  $r \leftarrow \text{Random}(0, 1)$;
  if $r < p_{rotate}$ **then**
    $x \leftarrow \text{Rotate}(x, \text{angle})$;
  end
  if $r < p_{zoom}$ **then**
    $x \leftarrow \text{Zoom}(x, \text{zoom})$;
  end
  if $r < p_{h\_shift}$ **then**
    $x \leftarrow \text{Shift}(x, \text{h\_shift} \times \text{width})$;
  end
  if $r < p_{v\_shift}$ **then**
    $x \leftarrow \text{Shift}(x, \text{v\_shift} \times \text{height})$;
  end
  if $r < p_{flip}$ **then**
    $x \leftarrow \text{Flip}(x)$;
  end
end
Model.fit(Dataset);

In this work, we selected four different datasets where in each one we find a train folder, test folder and validation folder. Table III lists the number of images in each dataset. The datasets were selected from Guangzhou Women and Children's Medical Center [37] and they are images from pediatric patients of one to five years old.

V. Experimental Results and Performance of the CNN Model

A. Hyperparameter Optimization (HPO)

To search for optimal hyperparameters of the model, random search was performed [38]. Itis a method for searching for optimal hyperparameters of a training model that involves randomly sampling hyperparameter combinations from a predefined search space. The search space is defined by specifying a range or distribution for each hyperparameter. The outline of the algorithm for random search is as follows.

- Define the hyperparameter search space: it consists of specifying the range or distribution of possible values

TABLE III. Different Datasets of Chest X-ray Images

Data	Number of images	Training	Testing	Validation
Dataset 1	5872	4770	551	551
Dataset 2	5856	4762	612	482
Dataset 3	6896	5532	720	644
Dataset 4	4665	3672	474	519

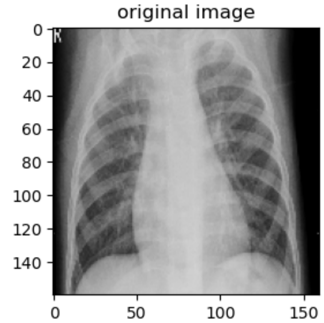

Fig. 4. Images without augmentation.

Fig. 5. Image augmented by random translation technique.

Fig. 6. Image augmented by random flipping technique.

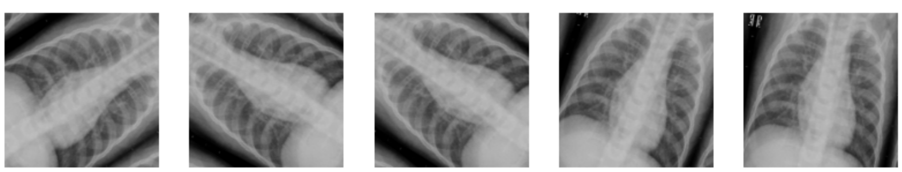

Fig. 7. Image augmented by random rotation technique.

for each hyperparameter.

- Initialize the random search object by creating an instance of the random search function, such as RandomizedSearchCV in scikit-learn, and specifying the model, the hyperparameter search space, the number of iterations, and other parameters such as the number of cross-validation folds.
- Generate random samples of hyperparameters byrandomly sampling hyperparameter combinations from the defined search space. The number of samples is controlled by the number of iterations specified in step 2.
- Train the model with each sample of hyperparameters: For each generated sample of hyperparameters, train the model using the corresponding sample of hyperparameters and evaluate its performance using a performance metric such as accuracy or F1-score on the validation set.
- Select the best set of hyperparameters by selecting the set of hyperparameters that results in the best performance on the validation set as the best set of hyperparameters.
- Validate the model on unseen data by using the best set of hyperparameters to train a final model and evaluate its performance on unseen data.

Hyperparameters obtained by the random search of the model were as follows. The model was trained using a batch size of 32, where 32 data samples are used to update the model's parameters in each iteration. The training process continued for 12 iterations. The early stopping technique was used to avoid overfitting, where the training process is stopped if the validation loss does not decrease for 7 consecutive iterations. This is done to ensure that the model generalizes well on unseen data.

B. Evaluation and Results

The training accuracy of a model is a measure of how well the model is able to predict the correct labels for the training data. The validation accuracy is a measure of how well the model is able to predict the correct labels for the validation data, which is a subset of the data that is held out from the training process and used to evaluate the model's performance. In general, the training accuracy of a model will be higher than the validation accuracy, because the model has seen the training data during the training process and has learned to predict the labels for those data points accurately. The validation accuracy is a more realistic measure of the model's performance, because it reflects the model's ability to generalize to unseen data. In order to determine the performance of the proposed method, we examined the accuracy, precision, recall, and F1 score. Accuracy is the proportion of correctly classified instances (True Positives and True Negatives) out of all instances. It is computed as follows:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (2)$$

Where True Positives (TP) are the number of instances where the model correctly predicted the positive class, True Negatives (TN) are the number of instances where the model correctly predicted the negative class, False Positives (FP) are the number of instances where the model incorrectly predicted the positive class and False Negatives (FN) are the number of instances where the model incorrectly predicted the negative class. Precision is the proportion of correctly classified positive instances out of all positive instances predicted by the model. It is calculated as:

$$Precision = TP/(TP + FP) \quad (3)$$

Recall is the proportion of correctly classified positive instances out of all actual positive instances. It is calculated as follows:

$$Recall = TP/(TP + FN) \quad (4)$$

F1 Score is the harmonic mean of precision and recall. It is calculated as follows:

$$F1\text{-}score = \frac{2 \cdot TP}{2 \cdot TP + (FP + FN)}. \quad (5)$$

All values of accuracy, precision, recall an F1 score are listed in Table IV.

In order to assess the influence of the size of the dataset on the performance of the CNN model, we plotted the curves of Training accuracy/Validation accuracy and Training loss/Validation loss. The tests were carried out, first, on four original datasets. Fig. 8 show that the training and validation accuracy varies with the epoch count. In fact, the model is learning efficiently, the training and validation accuracy reach a plateau after the end of the 20^{th} epoch indicating that the model has reached its maximum performance. The Training accuracy curve demonstrates the progression of the model's accuracy on the training set over successive epochs. As the training progresses, the accuracy steadily increases, indicating that the model is effectively learning the patterns and features of the pneumonia dataset. The upward trend in the curve signifies the successful optimization of the model's parameters, leading to improved classification accuracy. Similarly, the Training loss curve illustrates the decline in the loss function during the training phase. The loss function measures the discrepancy between the predicted and actual values, and the decreasing trend of the curve indicates that the model is converging towards a better approximation of the ground truth labels. A lower loss value indicates that the model is becoming more proficient at minimizing errors and making more precise predictions. The Validation accuracy and loss curves provide insights into the model's generalization performance on unseen data. The Validation accuracy curve tracks the accuracy of the model on a separate validation dataset that was not used for training. A rising validation accuracy curve indicates that the model is not overfitting and can generalize well to new data.

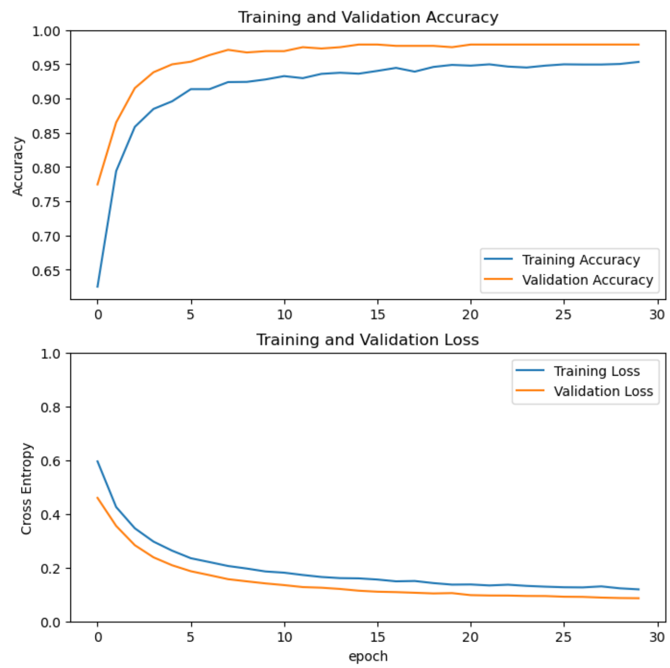

Fig. 8. Training and validation accuracy/loss curves of our CNN model applied on dataset 1.

A comparison of the obtained metrics our solution with those of known models has been summarized in the Table V.

VI. Results and Discussion

Note that the proposed model comprises four convolutional layers accompanied by four additional Maxpooling layers. The flattened layer of zero parameters and dense layers of about 819328 parameters come next. We thus have a total number of parameters which amounts to 1,246,401 parameters. We used two well-known and very powerful models to test the effectiveness of the proposed model. The CNN model was able to accomplish good values of Accuracy. After the augmentation of data, there was a clear improvement in the performance of all models and especially our CNN model (Table VI).

Four Chest X-ray images datasets served as the basis for our experiment.The datasets is publicly available on Kaggle, a shared data platform, and consists of 23289 real images

TABLE IV. PERFORMANCE OF THE PROPOSED METHOD

Model	Accuracy	Precision	Recall	F1-score
Our model	0.92	0.92	0.91	0.91

TABLE V. COMPARISON TABLE BETWEEN DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1-score
VGG16	0.96	0.93	1.0	0.97
ResNet-50	0.89	0.87	0.93	0.90
VGG-19	0.93	0.94	0.93	0.93
Inception-V3	0.75	0.77	0.71	0.74
ResNet-101	0.74	0.74	0.74	0.73
DenseNet121	0.49	0.50	0.48	0.49
Our model	0.92	0.92	0.91	0.91

TABLE VI. COMPARISON OF ACCURACY VALUES BETWEEN DIFFERENT MODELS

Data/Model	CNN		MobileNet		DenseNet	
	With aug	Without aug	With aug	Without aug	With aug	Without aug
Data1	97.88	80.64	99.99	62.50	80.76	80.76
Data2	95.34	80.41	97.11	62.50	87.78	78.39
Data3	95.04	81.02	96.05	61.18	87.78	79.89
Data4	94.99	80.70	98.26	61.18	87.78	79.89

developed by radiologists using data from real affected patients. We split our data into training, validation, and testing. First, data augmentation is done to enhance our dataset by doing minor changes in our images. We trained the models for 15 epochs with a batch size of 32 and a learning rate equal to 0.01. Our model showed a considerable improvement of performance between the basic and augmented images. We evaluated also the MobileNet and DenseNet models on data without and with augmentation and we find that augmenting data gives good results and enhanced the accuracy of different models. By applying data augmentation techniques, such as random cropping, flipping, rotation, and scaling, a larger and more diverse training dataset can be created from the original data. This increased diversity in the training data can lead to improved performance of the model on the test set, as it has seen similar variations during training. Additionally, data augmentation can also act as a regularization technique, preventing the model from overfitting to the training data. However, it's important to note that too much data augmentation can lead to overfitting to the augmented data and decreased performance on the original data. It's also important to find the right balance between the degree of augmentation and the quality of the augmented data to prevent information loss or degradation. In general, it's a good practice to experiment with different augmentation techniques and evaluate their effect on the model's performance. The results indicated that random data augmentation can be considered better than classic data augmentation because it increases the diversity of the training data in a more controlled manner, preventing overfitting to the augmented data and increasing the generalization ability of the model.

VII. CONCLUSIONS

This paper presents a comparative study of Deep Learning Models for the detection of Pneumonia, incorporating data augmentation techniques. The experimental results validate the effectiveness and accuracy of our proposed model. It showed good results compared to MobileNet and DenseNet. A remarkable improvement was noticed after applying the data augmentation techniques on the different datasets. Data augmentation can be a powerful technique for improving the performance of deep learning models. By artificially increasing the size and diversity of the training data, data augmentation can prevent overfitting and increase the generalization ability of the model. This can lead to enhanced results on the test set, and improved performance in real-world scenarios.

While our research has presented valuable insights into the detection of pneumonia using deep learning models and data augmentation techniques, there are a few aspects that merit further consideration. Firstly, it is important to acknowledge the limitations of our study, such as the reliance on a specific dataset and the need for further validation on larger and more diverse datasets. Additionally, exploring the impact of different data augmentation strategies and their effects on model performance could be an interesting avenue for future investigation. Moreover, investigating the generalizability of our proposed model to other medical imaging tasks and assessing its performance in real-world clinical settings could provide valuable insights. Lastly, incorporating interpretability techniques to understand the model's decision-making process and exploring ways to address any potential biases are important directions for future research. By addressing these questions and focusing on these areas, we believe that further advancements can be made in the field of pneumonia detection using deep learning techniques.

REFERENCES

[1] A. Turing, "Faster than thought," *Pitman, New York*, vol. 4, no. 1, pp. 286–310, 1953.

[2] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, no. 7, p. 939, 2019.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 545–563, 2021.

[5] D. K. Das, M. Ghosh, M. Pal, A. K. Maiti, and C. Chakraborty, "Machine learning approach for automated screening of malaria parasite using light microscopic images," *Micron*, vol. 45, pp. 97–106, 2013.

[6] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, and G. Thoma, "Image analysis and machine learning for detecting malaria," *Translational Research*, vol. 194, pp. 36–55, 2018.

[7] N. E. Ross, C. J. Pritchard, D. M. Rubin, and A. G. Duse, "Automated image processing method for the diagnosis and classification of malaria on thin blood smears," *Medical and Biological Engineering and Computing*, vol. 44, no. 5, pp. 427–436, 2006.

[8] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[9] R. Nijhawan, R. Verma, S. Bhushan, R. Dua, A. Mittal *et al.*, "An integrated deep learning framework approach for nail disease identification," in *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2017, pp. 197–202.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[15] M. Woźniak, D. Połap, G. Capizzi, G. L. Sciuto, L. Kośmider, and K. Frankiewicz, "Small lung nodules detection based on local variance analysis and probabilistic neural network," *Computer methods and programs in biomedicine*, vol. 161, pp. 173–180, 2018.

[16] Y. Gu, X. Lu, L. Yang, B. Zhang, D. Yu, Y. Zhao, L. Gao, L. Wu, and T. Zhou, "Automatic lung nodule detection using a 3d deep convolutional neural network combined with a multi-scale prediction strategy in chest cts," *Computers in biology and medicine*, vol. 103, pp. 220–231, 2018.

[17] T. K. Khanh Ho and J. Gwak, "Multiple feature integration for classification of thoracic disease in chest radiography," *Applied Sciences*, vol. 9, no. 19, p. 4130, 2019.

[18] G. Liang and L. Zheng, "A transfer learning method with deep residual network for pediatric pneumonia diagnosis," *Computer methods and programs in biomedicine*, vol. 187, p. 104964, 2020.

[19] A. A. Saraiva, D. Santos, N. J. C. Costa, J. V. M. Sousa, N. M. F. Ferreira, A. Valente, and S. Soares, "Models of learning to classify x-ray images for the detection of pneumonia using neural networks." in *Bioimaging*, 2019, pp. 76–83.

[20] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2097–2106.

[21] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. Rodrigues, "Identifying pneumonia in chest x-rays: A deep learning approach," *Measurement*, vol. 145, pp. 511–518, 2019.

[22] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[23] A. K. Acharya and R. Satapathy, "A deep learning based approach towards the automatic diagnosis of pneumonia from chest radio-graphs," *Biomedical and Pharmacology Journal*, vol. 13, no. 1, pp. 449–455, 2020.

[24] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.

[25] S. V. Militante, N. V. Dionisio, and B. G. Sibbaluca, "Pneumonia detection through adaptive deep learning models of convolutional neural networks," in *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*. IEEE, 2020, pp. 88–93.

[26] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. De Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest x-ray images," *Applied Sciences*, vol. 10, no. 2, p. 559, 2020.

[27] P. Rajpurkar, J. Irvin, R. L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. P. Langlotz *et al.*, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists," *PLoS medicine*, vol. 15, no. 11, p. e1002686, 2018.

[28] S. Bangare, H. Rajankar, P. Patil, K. Nakum, and G. Paraskar, "Pneumonia detection and classification using cnn and vgg16," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 12, pp. 771–779, 2022.

[29] H. Sharma, J. S. Jain, P. Bansal, and S. Gupta, "Feature extraction and classification of chest x-ray images using cnn to detect pneumonia," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 227–231.

[30] N. Wang, Y. Li, and H. Liu, "Reinforced neighbour feature fusion object detection with deep learning," *Symmetry*, vol. 13, no. 9, p. 1623, 2021.

[31] J. Zhang, J. Liu, and Z. Wang, "Convolutional neural network for crowd counting on metro platforms," *Symmetry*, vol. 13, no. 4, p. 703, 2021.

[32] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, no. 276, p. 2, 1995.

[33] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[34] X. Zhu and M. Bain, "B-cnn: branch convolutional neural network for hierarchical classification," *arXiv preprint arXiv:1709.09890*, 2017.

[35] K. Kanwal, K. T. Ahmad, R. Khan, A. T. Abbasi, and J. Li, "Deep learning using symmetry, fast scores, shape-based filtering and spatial mapping integrated with cnn for large scale image retrieval," *Symmetry*, vol. 12, no. 4, p. 612, 2020.

[36] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Weinberger, "Convolutional networks with dense connectivity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[37] D. Kermany, K. Zhang, M. Goldbaum *et al.*, "Labeled optical coherence tomography (oct) and chest x-ray images for classification," *Mendeley data*, vol. 2, no. 2, 2018.

[38] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.