An effective model for the detection of pneumonia from chest X-ray images using inner residual inception

Mohammed M. Nasef (Mnasef81@yahoo.com)
Menoufia University
Aya El-Sayed Shehata
Menoufia University
Amr M. Sauber
Menoufia University

Research Article

Keywords: Pneumonia, Convolution Neural Network, Inception, Residual

Posted Date: January 13th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2457904/v1

Additional Declarations: No competing interests reported.

An effective model for the detection of pneumonia from chest X-ray images using inner residual inception

Mohammed M. Nasef [1, a], Aya El-Sayed Shehata [1, b], Amr M. Sauber [1, c]

[1] *Mathematics and Computer Science Department, Faculty of Science, Menoufia University, 32511, Egypt*

[a] *Mnasef81@yahoo.com , Mohammed_nasef@science.menofia.edu.eg*

[b] *ayamarzouk83@gmail.com, ayashehata22@science.menofia.edu.eg*

[c] *amrmausad@computalityit.com, amr@science.menofia.edu.eg*

Abstract

Pneumonia is a serious disease that can lead to death if it is not diagnosed in an accurate manner. This paper presents three models for diagnosing pneumonia based on Chest X-Ray images. The first proposed model depends on the combination of inception, residual, and dropout. The second model is based on adding a batch normalization layer to the first model. The third model adds inner residual inception. The inner residual inception block has four branches, each of which has a significantly deeper root than any other known inception block, necessitating the use of residual connections between each branch. Inner residual inception blocks eventually consist of 4 distinct ResNet architectures. Each branch has a building block that is repeated three times with residuals, and then a dropout layer is added on top of that. These models used logistic regression and the Adam optimizer. The metrics used to evaluate the models are accuracy, precision, recall, F1-score, AUC, and balanced accuracy. From the results, the third proposed model has achieved the highest accuracy of 96.76%, and the best balance accuracy of 95.08%.

Keywords:

Pneumonia, Convolution Neural Network, Inception, Residual.

1. Introduction

Pneumonia is responsible for 4 million deaths per year, over 150 million infections each year, and 15% of all death in children under the age of five. Identifying pneumonia in its early stages can save many lives. Pneumonia is an infection that causes inflammation in one or both lungs. It can be caused by a virus, bacteria, fungi, or other germs [1]. Infection usually occurs when a person breathes air that carries germs. Chest x-ray images are one way to diagnose pneumonia [2-5].

Machine learning techniques can help increase medical diagnosis accuracy. When it comes to medical diagnosis, the overall accuracy is not a good indication of the model's performance, however, we need to consider two scenarios in which we examine the two types of errors a model can make. the first type of error is known as false positive, in that error, the model predicts that a patient has pneumonia when he does not have pneumonia. In the second type of error which is known as false negative, the model predicts that a patient does not have pneumonia when he does. Both errors are very dangerous as the first type will be responsible for delaying the diagnosis and hence causing deterioration of patient health and increasing the percentage of death. the second type will cause a healthy person to go through wrong and unnecessary treatment which can have side effects and affects his life. The relationship between two types of errors is a trade, and some models choose to decrease the first type at the expense of increasing the second type.

Several studies have been shown to detect Pneumonia from chest x-ray images. In [6] proposed an ensemble model that used transfer learning from five pre-trained models namely AlexNet, DenseNet121, Inception V3, GoogLeNet, and ResNet18. The pre-trained models are combined and used to extract features which were then passed to a classifier. The model used data

augmentation for better generalization and scored an overall accuracy of 96.39%. In [7] studied the impact of data augmentation and dropout by designing four convolution neural network (CNN) models. One with both data augmentation and dropout, one with only data augmentation, one with only dropout, and one without them both. It was found that the model having both augmentation and dropout scored the highest accuracy of 90.68%. In [8] introduced a CNN model trained from scratch and doesn't rely on transfer learning. The model employed many data augmentation methods to increase the size of the dataset. The model consists of two parts, a feature extractor, and a sigmoid classifier. The feature extractor is composed of a stack of convolution and max-pooling layers. The model used different data sizes ranging from 100 to 300. The highest accuracy of 93.73% was reached at a data size of 200. In [9] proposed a CNN model to classify pneumonia. The CNN consists of 10 layers, the first 7 are convolution layers, and then 3 dense layers. They scored an accuracy of 95.30%. In [5] introduced a computer-aided classification of pneumonia, called Ensemble Learning (EL), to simplify the diagnosis process on chest X-ray images. Three well-known CNNs (DenseNet169, MobileNetV2, and Vision Transformer) pretrained using the ImageNet database. The proposed EL achieve an accuracy of 93.91% and a F1-score of 93.88% on the testing phase.

All previous studies ignored the study of the balance between classes, and this may affect the quality of the results. Therefore, this paper has studied the balance between the classes under study for the three proposed models.

The rest of the paper is organized as follows, in section 2 we discuss proposed models, in section 3 introduce experimental results, in section 4 discuss the advances and limitations of proposed models, and finally, in section 5 give a conclusion and some future works.

2. Proposed Models

This paper presents two baseline models built from different versions of inception models and three new models that attempt to overcome the issues faced by baselines. These models consist of two phases. The feature extraction is the first phase. In the second phase, a simple logistic regression algorithm was used to classify the x-ray image as either normal or pneumonia.

2.1 Feature extraction phase

when detecting pneumonia, it can be in the upper left corner of the image or in the lower left corner and so on. This huge variation in the location of the information causes a lot of problems while learning and choosing the right filter sizes for convolution neural networks become difficult. Inception Blocks have been first proposed as part of GoogLeNet model [10] to solve problems caused by the large variations in object size that can be found in the image. The first inception block proposed in [10] applied four branches of filter on the input at the same time. The first branch consisted of $\mathbf{1 \times 1}$ convolution filter, the second branch consisted of $\mathbf{1 \times 1}$ convolution filter followed by $\mathbf{3 \times 3}$ convolution filter, the third branch consisted of $\mathbf{1 \times 1}$ convolution filter followed by $\mathbf{5 \times 5}$ convolution filter and the last branch consisted of $\mathbf{3 \times 3}$ max pooling filter followed by $\mathbf{1 \times 1}$ convolution filter. GoogLeNet has a stack of nine inception blocks stacked linearly. It consists of an overall 27 layers including the pooling layers. It uses global average pooling at the end of the last inception block. Ever since the introduction of inception blocks and GoogLeNet model (InceptionV1), researchers have been trying to upgrade and modify inception blocks to improve accuracy. InceptionV2 and InceptionV3 were proposed in [11]. In inceptionV2, the authors factorized the $\mathbf{5 \times 5}$ convolution filter in the inception block into two $\mathbf{3 \times 3}$ convolution filter and staking two $\mathbf{3 \times 3}$ convolution filter is 2.78 faster than using a $\mathbf{5 \times 5}$ convolution filter. In inceptionV2 rather than repeating the same inception block for specific number of times, they also introduced three main inception blocks where each is repeated for number of times. InceptionV3 did not include any changes in the inception block from InceptionV2, however the model itself had some

improvements such as using RMSProp optimizer, using label smoothing, using Batch Normalization in the classifier, and using factorized $\mathbf{7 \times 7}$ convolution filters. InceptionV4 and Inception-ResNet was proposed in [12]. InceptionV4 added a bunch of initial set of convolutions called stem before introducing inception blocks. In inceptionV4, they introduced reduction blocks that is responsible for changing the height and the width of the input. Inspired by ResNets, a hybrid Inception-ResNet model was proposed. The model has the same structure as InceptionV4 but adds residual connections between inception blocks.

This paper introduces some modifications to the inception block that can be applied to any model and help increase accuracy for any inception-based model. We start with two baseline models: Inception model and Inception-Residual model the first uses the same inception block in InceptionV1 and the second used the same idea proposed in the last Inception-ResNet model. The models are a stack of five inception blocks, followed by a classifier.

2.1.1 Baseline 1: Pure Inception

As a baseline, we used the original inception blocks proposed in GoogLeNet model, stacked 5 of them on top of each other, and then a classifier as shown in Fig.1. Although this model looks nothing like InceptionV1, it will serve as a benchmark to see whether the modifications to the inception block improves the accuracy or not. The input image of size 64×64 goes through a 3×3 convolution with 64 filters and is then followed by 5 inception blocks. Each inception block has 4 branches. Having an input image X the model can be summarized in algorithm 1.

Algorithm 1: Pure Inception Model

In: $\boldsymbol{Dataset\ X}$ *//Chest X-Ray images*

Out: $\boldsymbol{Prediction\ \hat{Y}}$

$X = X/255.0$ *//Normalization*

$X = Conv(X,\ kernal = (3,3),\ filters = 64,\ padding = same,\ stride = 1,\ activation = relu)$

while $\boldsymbol{i \neq 5}$ ***do***: *//Apply inception block for 5 times*

  $b_1 = Conv(X,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

  $b_{21} = Conv(X,\ kernal = (1,1),\ filters = 96,\ pad = same,\ stride = 1,\ activation = relu)$

  $b_{22} = Conv(b_{21},\ kernal = (3,3),\ filters = 128,\ pad = same,\ stride = 1,\ activation = relu)$

  $b_{31} = Conv(X,\ kernal = (1,1),\ filters = 16,\ pad = same,\ stride = 1,\ activation = relu)$

  $b_{32} = Conv(b_{31},\ kernal = (5,5),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

  $b_{41} = MaxPool(X,\ kernal = (3,3),\ stride = 1)$

  $b_{42} = Conv(b_{41},\ kernal = (1,1),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

  $X = Concatenate(b_1, b_{22}, b_{32}, b_{42})$

end while

$X = Flatten(X)$

$\hat{Y} = Dense(1,\ activation = sigmoid)$

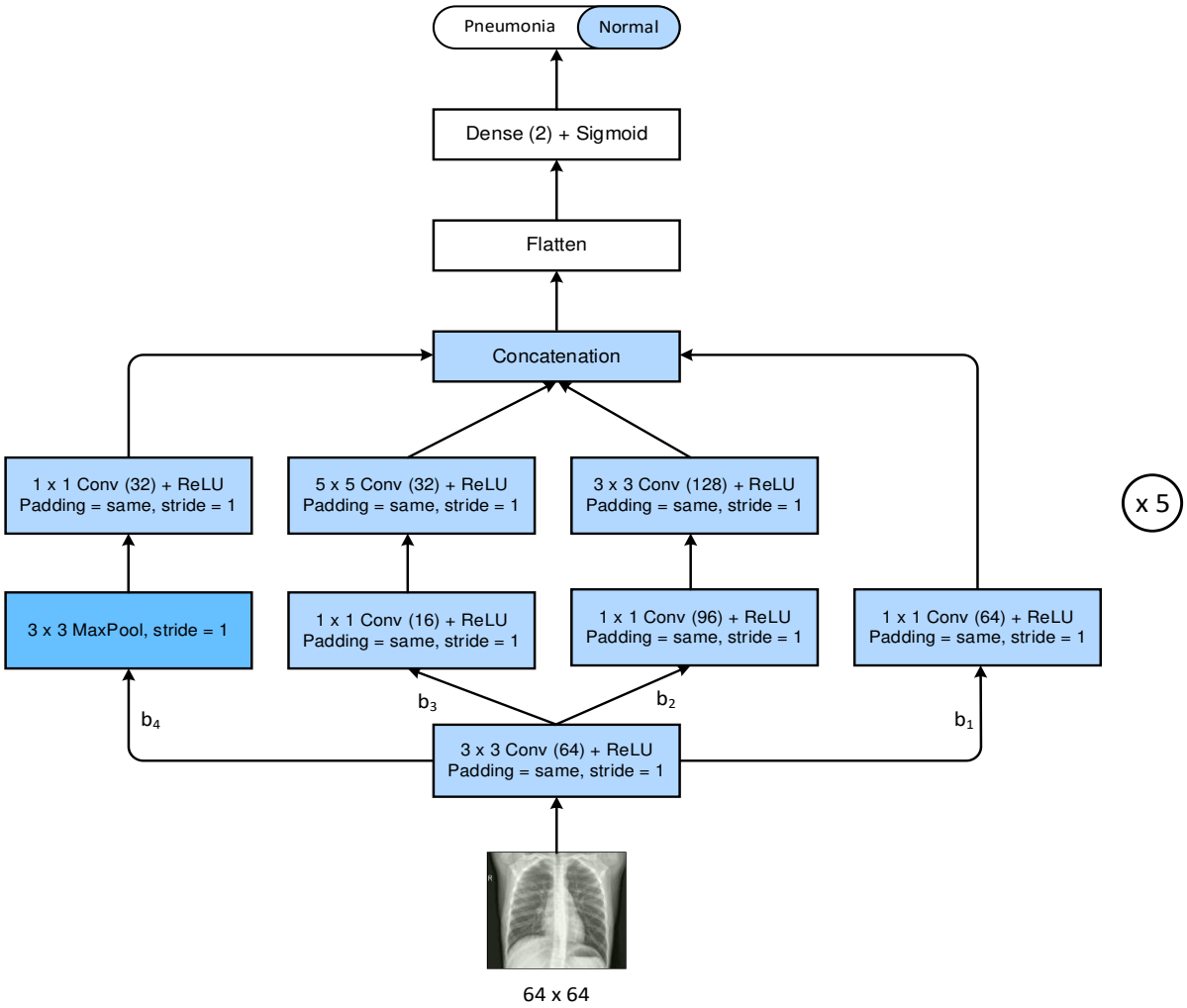

Fig 1: Pure Inception Model

2.1.2 Baseline 2: Inception + Residual

The second baseline adapted the last upgrade used in inception blocks, which is using residual connections [13]. Stack 5 of the inception-residual blocks on top of each other followed by a classifier as shown in Fig.2. Also, the model looks nothing like the Inception-ResNet model, but it will also serve as a benchmark for the latest update to inception blocks and see whether the modifications improve the accuracy or not. The input image of size 64×64 goes through a 3×3 convolution with 64 filters and is then followed by 5 inception blocks. Each inception block has 4 branches. The difference here, after concatenation, the output goes through a 1×1 convolution with 64 filters to enable adding the output of the first 3×3 convolution to it. Algorithm 2 summarizes the model.

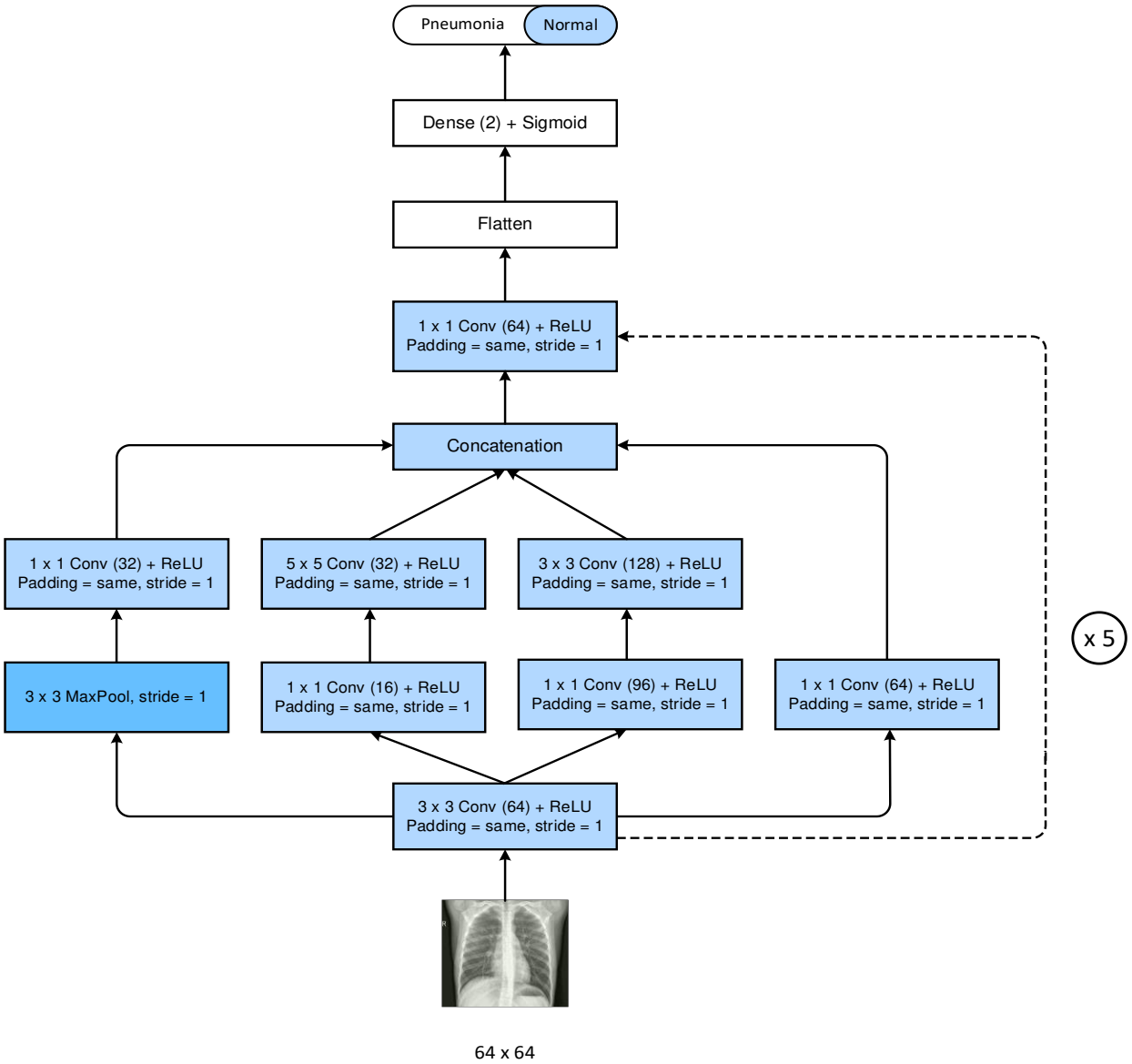

Fig 2: Inception + Residual Model

Algorithm 2: Inception + Residual Model

In: $\boldsymbol{Dataset\ X}$ *//Chest X-Ray images*

Out: $\boldsymbol{Prediction\ \hat{Y}}$

$X = X/255.0$ *//Normalization*

$X = Conv(X,\ kernal = (3,3),\ filters = 64,\ pad = same,\ stride = 1,\ activation = relu)$

while $\boldsymbol{i \neq 5}$ ***do***: *//Apply inception-residual block for 5 times*

$b_1 = Conv(X,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

$b_{21} = Conv(X,\ kernal = (1,1),\ filters = 96,\ pad = same,\ stride = 1,\ activation = relu)$

$b_{22} = Conv(b_{21},\ kernal = (3,3),\ filters = 128,\ pad = same,\ stride = 1,\ activation = relu)$

$b_{31} = Conv(X,\ kernal = (1,1),\ filters = 16,\ pad = same,\ stride = 1,\ activation = relu)$

$b_{32} = Conv(b_{31},\ kernal = (5,5),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

$b_{41} = MaxPool(X,\ kernal = (3,3),\ stride = 1)$

$b_{42} = Conv(b_{41},\ kernal = (1,1),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

$Z = Concatenate(b_1,\ b_{22},\ b_{32},\ b_{42})$

$Z = Conv(Z,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

$X = Z + X$ *//Residuals*

end while

$X = Flatten(X)$

$\hat{Y} = Dense(1,\ activation = sigmoid)$

2.1.3 Proposed Model 1: Inception + Residual + Dropout

Dropout [14] has been widely used as a regularization technique to fight overfitting. Although dropout has become a main component in every model, it never got used inside inception blocks. The first proposed modification to the inception block is adding dropout after the deeper 3 branches as illustrated in **Fig.3**. Although inception model goes wider than deeper, but some branched could learn to much and cause overfitting. The model can be described in algorithm 3.

Algorithm 3: Inception + Residual + Dropout Model

In: ***Dataset X*** *//Chest X-Ray images*

Out: ***Prediction $\hat{Y}$***

$X = X/255.0$ *//Normalization*

$X = Conv(X,\ kernal = (3,3),\ filters = 64,\ pad = same,\ stride = 1,\ activation = relu)$

while i $\neq$ **5** ***do***: *//Apply the block for 5 times*

$b_1 = Conv(X,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

$b_{21} = Conv(X,\ kernal = (1,1),\ filters = 96,\ pad = same,\ stride = 1,\ activation = relu)$

$b_{22} = Conv(b_{21},\ kernal = (3,3),\ filters = 128,\ pad = same,\ stride = 1,\ activation = relu)$

$d_1 = Dropout(b_{22},\ drop_{rate} = 0.3)$

$b_{31} = Conv(X,\ kernal = (1,1),\ filters = 16,\ pad = same,\ stride = 1,\ activation = relu)$

$b_{32} = Conv(b_{31},\ kernal = (5,5),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

$d_2 = Dropout(b_{32},\ drop_{rate} = 0.1)$

$b_{41} = MaxPool(X,\ kernal = (3,3),\ stride = 1)$

$b_{42} = Conv(b_{41},\ kernal = (1,1),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

$d_3 = Dropout(b_{42},\ drop_{rate} = 0.2)$

$Z = Concatenate(b_1,\ d_1,\ d_2,\ d_3)$

$Z = Conv(Z,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

16 $X = Z + X$ *//Residuals*

16 ***end while***

18 $X = Flatten(X)$

19 $\hat{Y} = Dense(1,\ activation = sigmoid)$

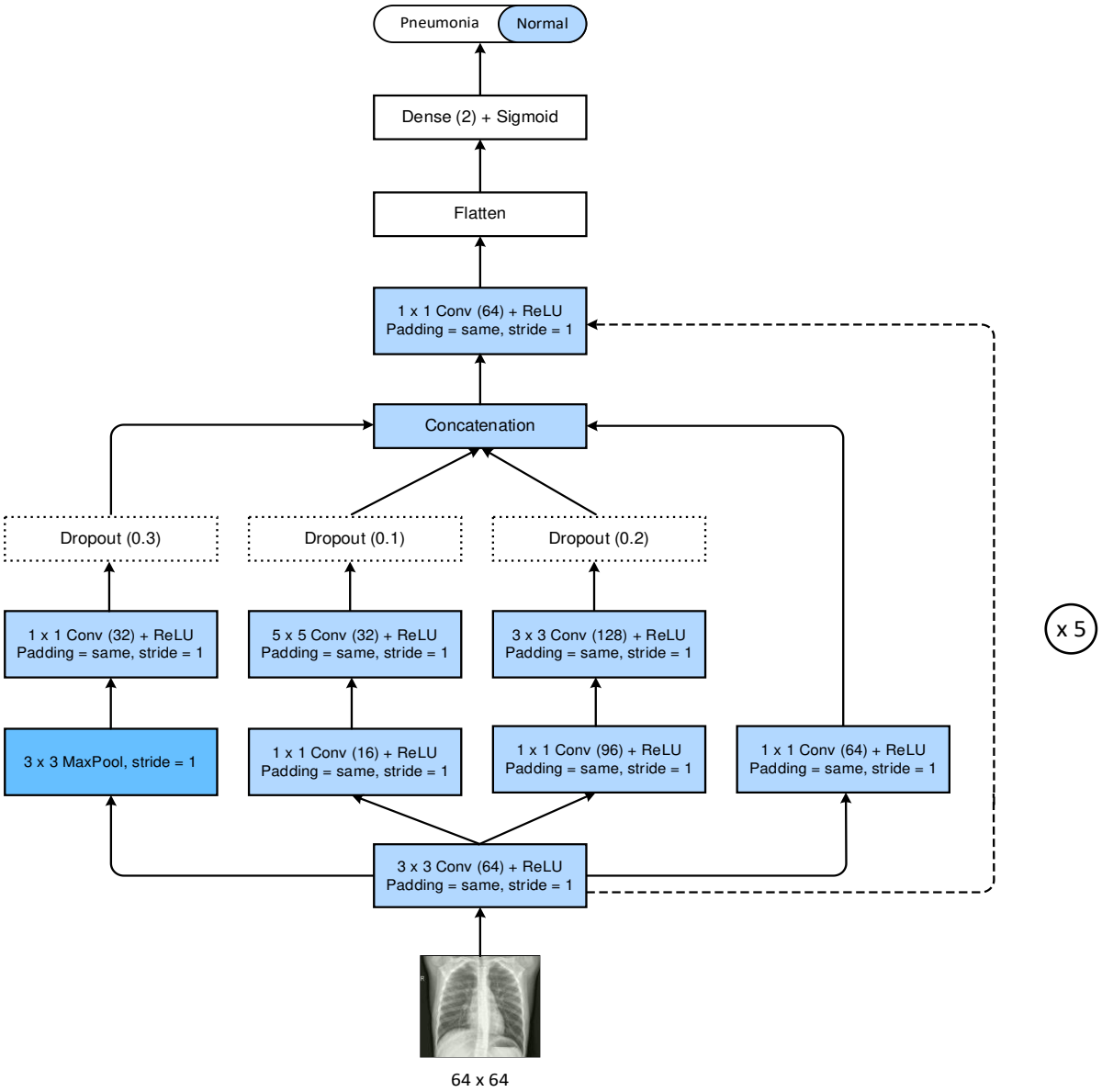

Fig 3: Inception + Residual Model + Dropout Model

2.1.4 Proposed Model 2: Inception + Residual + Dropout + Batch Normalization

In addition to adding dropout in the first proposed model, we added a batch normalization [15] layer after each branch as shown in **Fig.4**. Batch Normalization is an effective technique that consistently accelerates the convergence of deep networks and improves accuracy. Although InceptionV3 model used batch normalization, it was used in the classifier and never used as part of the inception block itself. Algorithm 4 summarizes the model.

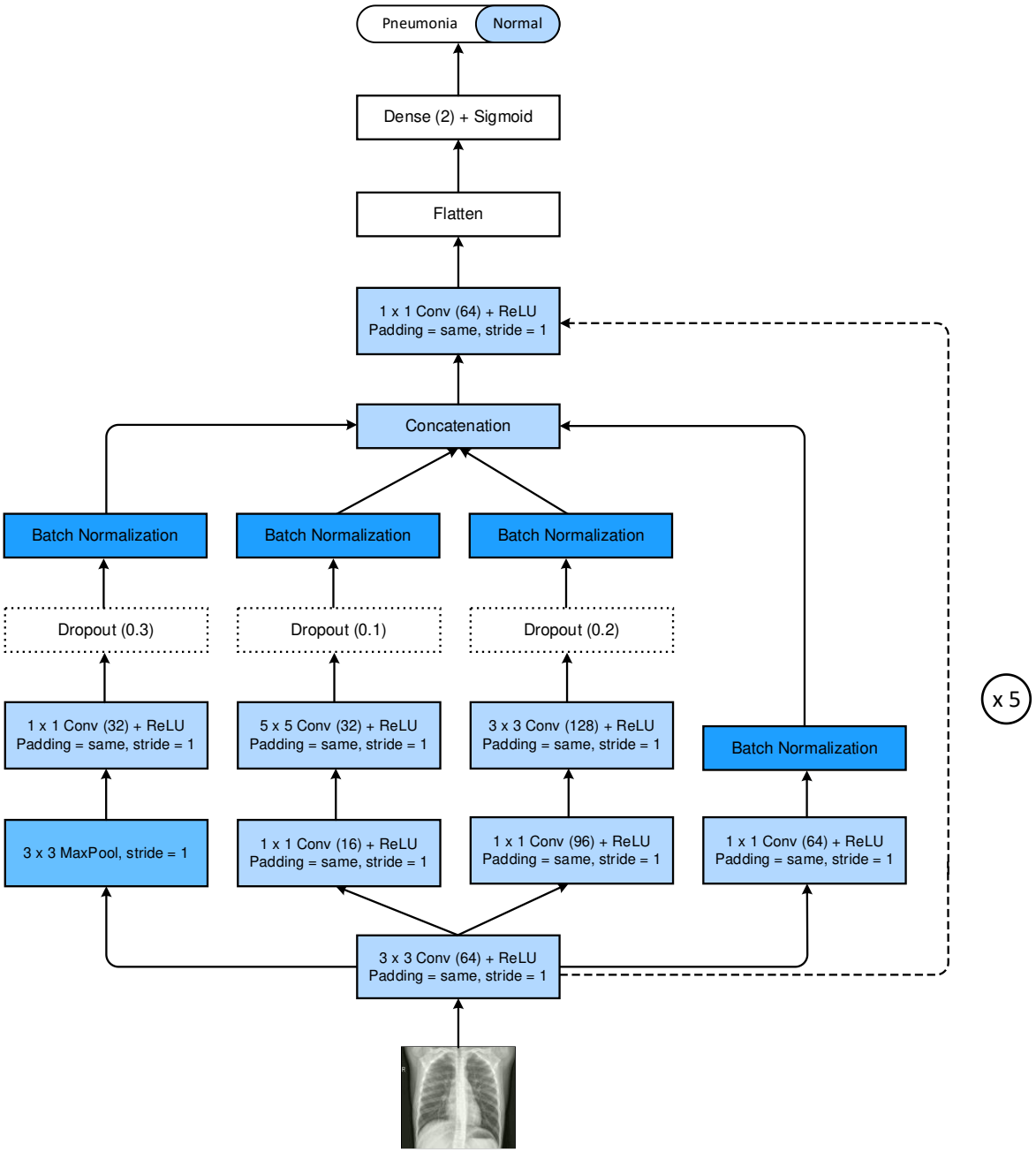

Fig 4: Inception + Residual + Dropout + Batch Normalization model

Algorithm 4: Inception + Residual + Dropout + Batch Normalization Model

In: $\boldsymbol{Dataset\ X}$ *//Chest X-Ray images*

Out: $\boldsymbol{Prediction\ \hat{Y}}$

1 $X = X/255.0$ *//Normalization*

2 $X = Conv(X,\ kernal = (3,3),\ filters = 64,\ pad = same,\ stride = 1,\ activation = relu)$

3 $\boldsymbol{while\ i \neq 5\ do}$: *//Apply the block for 5 times*

4 $b_1 = Conv(X,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

5 $bn_1 = BatchNorm(X)$

6 $b_{21} = Conv(X,\ kernal = (1,1),\ filters = 96,\ pad = same,\ stride = 1,\ activation = relu)$

7 $b_{22} = Conv(b_{21},\ kernal = (3,3),\ filters = 128,\ pad = same,\ stride = 1,\ activation = relu)$

8 $bn_2 = BatchNorm\big(Dropout(b_{22},\ drop_{rate} = 0.3)\big)$

9 $b_{31} = Conv(X,\ kernal = (1,1),\ filters = 16,\ pad = same,\ stride = 1,\ activation = relu)$

10 $b_{32} = Conv(b_{31},\ kernal = (5,5),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

11 $bn_3 = BatchNorm\big(Dropout(b_{32},\ drop_{rate} = 0.1)\big)$

12 $b_{41} = MaxPool(X,\ kernal = (3,3),\ stride = 1)$

13 $b_{42} = Conv(b_{41},\ kernal = (1,1),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

14 $bn_4 = BatchNorm\big(Dropout(b_{42},\ drop_{rate} = 0.2)\big)$

15 $Z = Concatenate(bn_1,\ bn_2,\ bn_3,\ bn_4)$

16 $Z = Conv(Z,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

16 $X = Z + X$ *//Residuals*

18 ***end while***

19 $X = Flatten(X)$

20 $\hat{Y} = Dense(1,\ activation = sigmoid)$

2.1.5 Proposed Model 3: Inner Residual Inception

The first inception model tends to go wider rather than deeper. But what if the model can go wider and deeper at the same time so that it can learn more complex structure and deal with information variation across the images at the same time? That was the idea behind inner residual inception blocks. The inner residual inception block consists of 4 branches, each branch is much deeper that all known inception blocks which make it essential to apply residual connections between every branch. Eventually, inner residual inception blocks are an ensemble of 4 unique ResNet architecture. Each branch has a building block that is repeated 3 times with residuals and then followed by a dropout layer. The first branch building block consists of a $\mathbf{3 \times 3}$ convolution with 96 filters, followed by a $\mathbf{2 \times 2}$ max pooling filter followed by a batch normalization layer. The second branch building block is a $\mathbf{5 \times 5}$ convolution with 32 filters, followed by a $\mathbf{2 \times 2}$ max pooling filter and a batch normalization layer. The third branch building block is a $\mathbf{3 \times 3}$ convolution with 128 filters, followed by a $\mathbf{2 \times 2}$ max pooling filter and a batch normalization layer. The fourth branch is a factorized $\mathbf{3 \times 3}$ convolution into $\mathbf{1 \times 3}$ convolution and $\mathbf{3 \times 1}$ convolution with 64 filters each, followed by a $\mathbf{2 \times 2}$ max pooling filter and a batch normalization layer. Before applying dropout, the first 3 branches go through a $\mathbf{1 \times 1}$ convolution with 64 filters. The output of the four branches gets concatenated and then go through a $\times \mathbf{1}$ convolution with 64 filters to match the channels of the input to apply residual connections between Inner Residual Inception blocks. The model is shown in **Fig.5** and algorithm 5 summarizes it.

Algorithm 5: Inner Residual Inception

In: $\boldsymbol{Dataset\ X}$ *//Chest X-Ray images*

Out: $\boldsymbol{Prediction\ \widehat{Y}}$

$X = X/255.0$ *//Normalization*

$X = Conv(X,\ kernal = (3,3),\ filters = 64,\ pad = same,\ stride = 1,\ activation = relu)$

Function $FirstBranch_{block}(X)$:

    $l_1 = Conv(X,\ kernal = (3,3),\ filters = 96,\ pad = same,\ stride = 1,\ activation = relu)$

    $l_2 = BatchNorm\big(MaxPool(l_1,\ kernal = (2,2),\ stride = 1)\big)$

    $return\ (X + l_2)$

Function $SecondBranch_{block}(X)$:

    $l_1 = Conv(X,\ kernal = (5,5),\ filters = 32,\ pad = same,\ stride = 1,\ activation = relu)$

    $l_2 = BatchNorm\big(MaxPool(l_1,\ kernal = (2,2),\ stride = 1)\big)$

    $return\ (X + l_2)$

Function $ThirdBranch_{block}(X)$:

    $l_1 = Conv(X,\ kernal = (3,3),\ filters = 128,\ pad = same,\ stride = 1,\ activation = relu)$

    $l_2 = BatchNorm\big(MaxPool(l_1,\ kernal = (2,2),\ stride = 1)\big)$

    $return\ (X + l_2)$

Function $ForthBranch_{block}(X)$:

    $l_1 = Conv(X,\ kernal = (3,1),\ filters = 64,\ pad = same,\ stride = 1,\ activation = relu)$

    $l_2 = Conv(X,\ kernal = (1,3),\ filters = 64,\ pad = same,\ stride = 1,\ activation = relu)$

    $l_3 = BatchNorm\big(MaxPool(l_1,\ kernal = (2,2),\ stride = 1)\big)$

    $return\ (X + l_3)$

while $\boldsymbol{i \neq 5}$ ***do***: *//Apply inner residual inception block for 5 times*

  $\boldsymbol{b_1 = X,\ b_2 = X,\ b_3 = X,\ b_4 = X}$

  while $\boldsymbol{j \neq 3}$ ***do***:

    $b_1 = FirstBranch_{block}(b_1)$

    $b_2 = SecondBranch_{block}(b_2)$

    $b_3 = ThirdBranch_{block}(b_3)$

    $b_4 = ForthBranch_{block}(b_4)$

  end while

  $b_1 = Conv(b_1,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

  $b_2 = Conv(b_2,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

  $b_3 = Conv(b_3,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

$d_1 = Dropout(b_1,\ drop_{rate} = 0.3)$

$d_2 = Dropout(b_2,\ drop_{rate} = 0.3)$

$d_3 = Dropout(b_3,\ drop_{rate} = 0.3)$

$d_4 = Dropout(b_4,\ drop_{rate} = 0.3)$

$Z = Concatenate(d_1,\ d_2,\ d_3,\ d_4)$

$Z = Conv(Z,\ kernal = (1,1),\ filters = 64,\ pad = same, stride = 1,\ activation = relu)$

$X = Z + X$ *//Residuals*

end while

$X = Flatten(X)$

$\hat{Y} = Dense(1,\ activation = sigmoid)$

2.2 Classification phase

For each case in the dataset, each of the suggested models generates a feature vector that may be used to train a classifier. For binary classification used logistic regression [16], which estimates the probability that a given example belongs to one of the two classes using the sigmoid function Equation (1).

$$sigmoid(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

$$\hat{y}^{(i)} = prediction(x) = \begin{cases} 0\ (normal) & if \quad \sigma\left(x^{(i)}\right) \geq 0.5 \\ 1\ (pneumonia\ ) & if \quad \sigma\left(x^{(i)}\right) < 0.5 \end{cases} \tag{2}$$

Where $\hat{y}^{(i)}$ is the prediction of the i^{th} training example $x^{(i)}$.

We used an *Adam* optimizer [17] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ and a learning rate $= 3e^{-4}$ to minimize the binary cross-entropy loss function (log loss) **Equation (3)**.

$$loss = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left(\hat{y}^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - \hat{y}^{(i)}\right) \tag{3}$$

Where $y^{(i)}$ is the true label of an example $x^{(i)}$, and m is the number of training samples.

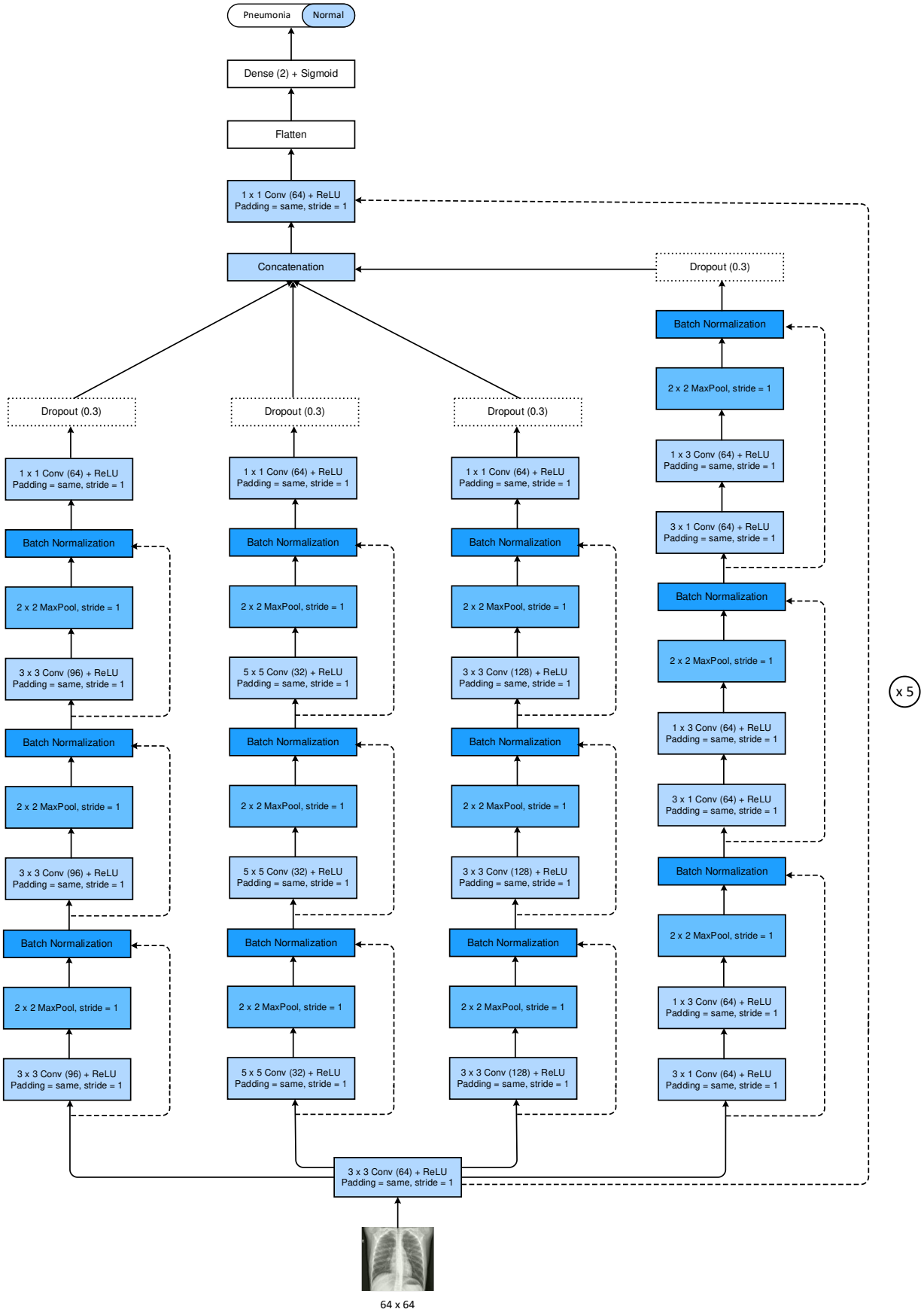

Fig 5: Inner Residual Inception Model

3. Experimental Results

3.1 Dataset

In this paper, the Chest X-Ray images dataset collected from [18]. The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (Pneumonia / Normal). For this study, the data was grouped and reshuffled and split into two folders (train, test). The data contains 5,863 X-Ray images split as shown in Table 1.

Table 1: Dataset.

Split	Size	Normal	Pneumonia
Training	4684	1736	2948
Testing	1172	317	855

The dataset has two types of pneumonia and **Fig 6** shows examples of Chest X-Rays in patients with pneumonia. The normal chest X-ray (left) depicts clear lungs without any areas of abnormal opacification in the image. Bacterial pneumonia (middle) typically exhibits a focal lobar consolidation, in this case in the right upper lobe (white arrows), whereas viral pneumonia (right) manifests with a more diffuse ‘‘interstitial’’ pattern in both lungs.

Fig 6: Chest X-Ray samples [13]

All models trained on a machine with 8 cores Intel Xeon E5-2620 v4 processor with 128 GB of RAM installed.

3.2 Results and Discussion

In this section introduce the results for the five models (two baselines and three proposed modifications). All models have run for 50 epochs and the best fitted model during those epochs was chosen. For evaluating the models, used 6 metrics to measure the performance defined in **Equations 4-8**. The metrics are Accuracy, Precision, Recall, F1-score [2, 19], and AUC [20].

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (7)$$

$$Balanced_{Accuracy} = \frac{1}{2}\left(\frac{TP}{TP + FP} + \frac{TN}{TN + FN}\right) \quad (8)$$

Where:
True Positives **(TP)**: the number of correct predictions made by the model that a patient suffers pneumonia.

False Positives **(FP)**: the number of wrong predictions made by the model that a patient does not suffer from pneumonia.

True Negatives **(TN)**: the number of correct predictions made by the model that a healthy person does not suffer from pneumonia.

False Negatives **(FN)**: the number of wrong predictions made by the model that a healthy person suffers from pneumonia.

Balanced accuracy is a good measure for unbalanced datasets like the one being used since one class (Normal) has many fewer training samples than the other class (Pneumonia). A balance accuracy defined as **equation 9** [21].

$$Balanced\ Accuracy = \frac{1}{K}\sum_{k=1}^{K}\frac{r_k}{m_k} \quad (9)$$

Where K is the number of classes, m_k is the number of samples belonging to class k and r_k is the number of samples accurately predicted belonging to class k.

3.2.1 Baseline 1: Pure Inception results

Pure inception model has been trained for 50 epochs and it reached its heights accuracy at the 10th epoch and did not improve after that as shown in **Fig 7**.

Pure inception model reached a test accuracy of 95.05% and a near 100% training accuracy which means that the model is overfitting.

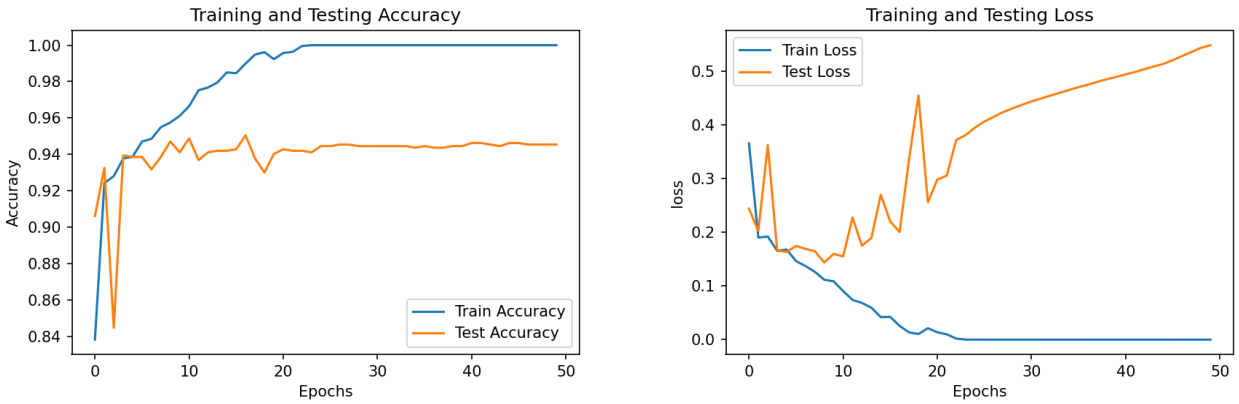

Fig 7: Pure Inception model loss and accuracy while training

3.2.2 Baseline 2: Inception + Residual results

The inception-residual model has also been trained for 50 epochs and it reached its heights accuracy at the 15th epoch and did not improve after that as shown in **Fig.8**.

Inception-residual model did not improve from the pure inception model as it reached a test accuracy of 94.96% and a 100% training accuracy which means that the model is also overfitting the data. However, the inception-residual model scored higher balanced accuracy that pure inception model as it scored 94.27% compared to 94.13% for pure inception.

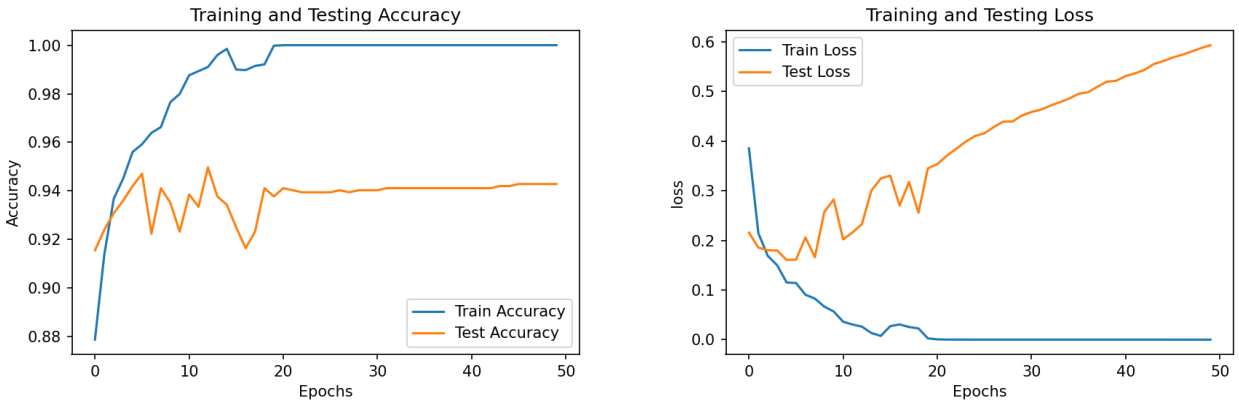

Fig 8: Inception + Residual model loss and accuracy while training

3.2.3 Proposed Model 1: Inception + Residual + Dropout results

As seen in the two baseline models, the biggest problem facing them is overfitting. In this model we use dropout to try to reduce overfitting and increase accuracy. The model is trained for 50 epochs and reached its highest test accuracy of 94.54% at the 37th epoch as shown in **Fig.9**.

The model reached a training accuracy of 99.6% and a balanced accuracy of 92.39%. Compared to our two baselines, adding dropout did not neither improve the accuracy nor fighting overfitting.

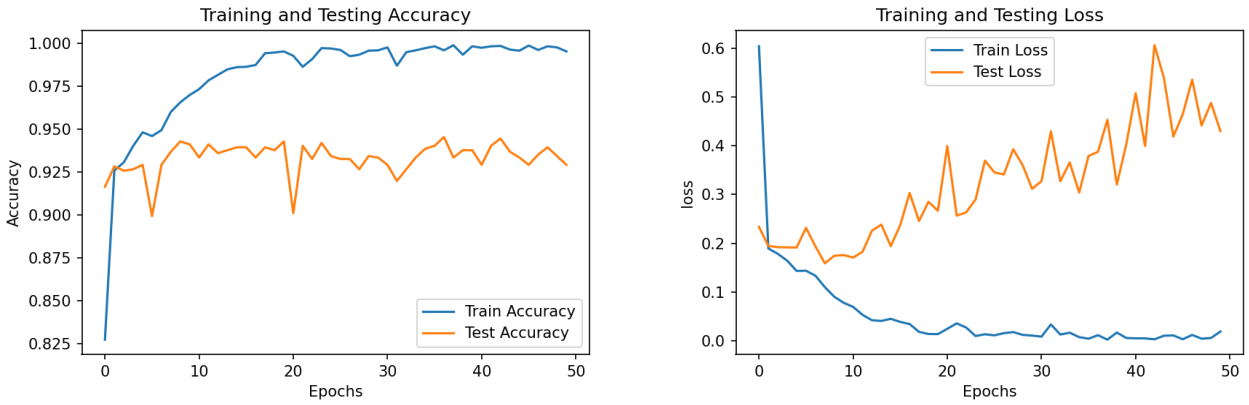

Fig 9: Inception + Residual + Dropout model loss and accuracy while training

3.2.4 Proposed Model 2: Inception + Residual + Dropout + Batch Normalization results

Unfortunately, adding dropout to inception blocks did not help improving performance. In this model we add another piece that can help reduce overfitting and increase performance which is Batch Normalization (BN). The model is trained for 50 epochs and reached its highest test accuracy of 94.8% at the 26th epoch as shown in **Fig.10**. The model reached a training accuracy of 99.5% and a balanced accuracy of 92.26%.

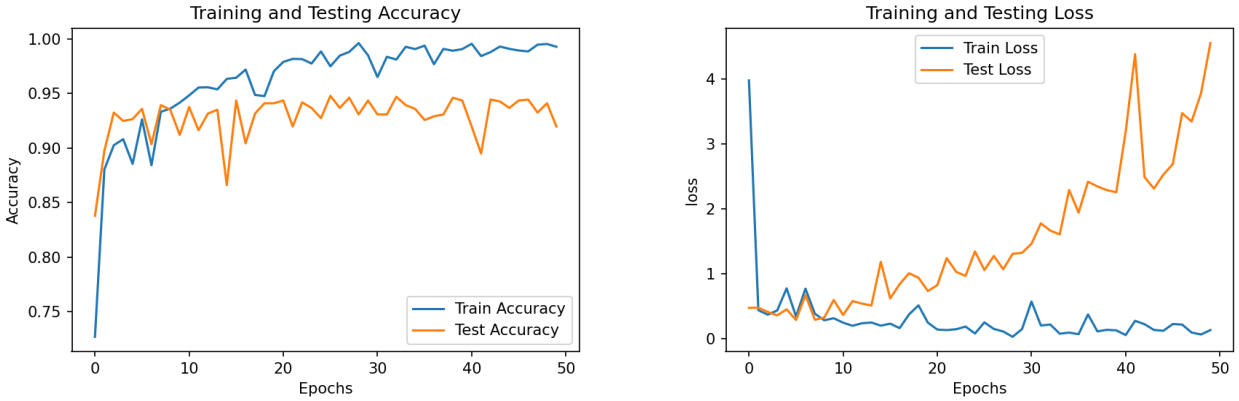

Fig 10: Inception + Residual + Dropout + BN model loss and accuracy while training

3.2.2.5 Proposed Model 3: Inner Residual Inception results

Although adding BN to inception block helped reduce the gap between training and testing curve as shown in **Fig.10** and reduce overfitting a little, it did not improve the accuracy compared our two baselines.

In this model we are taking the inception block to the next level by going both deeper and wider and learn more complex structure.

The model its best test accuracy of 96.76% at the 40th epoch as shown in **Fig.11**. The model reached a training accuracy of 99.3% and the model has a balanced accuracy of 95.08%. The model succeeded at both reducing overfitting and improve accuracy.

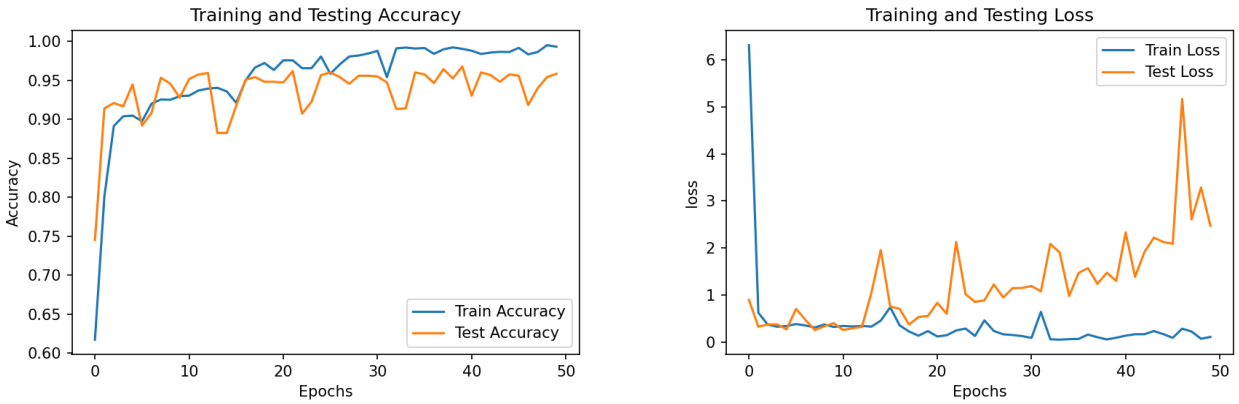

Fig 11: Inner Residual Inception model loss and accuracy while training

Fig.12 shown the confusion matrix for the five models.

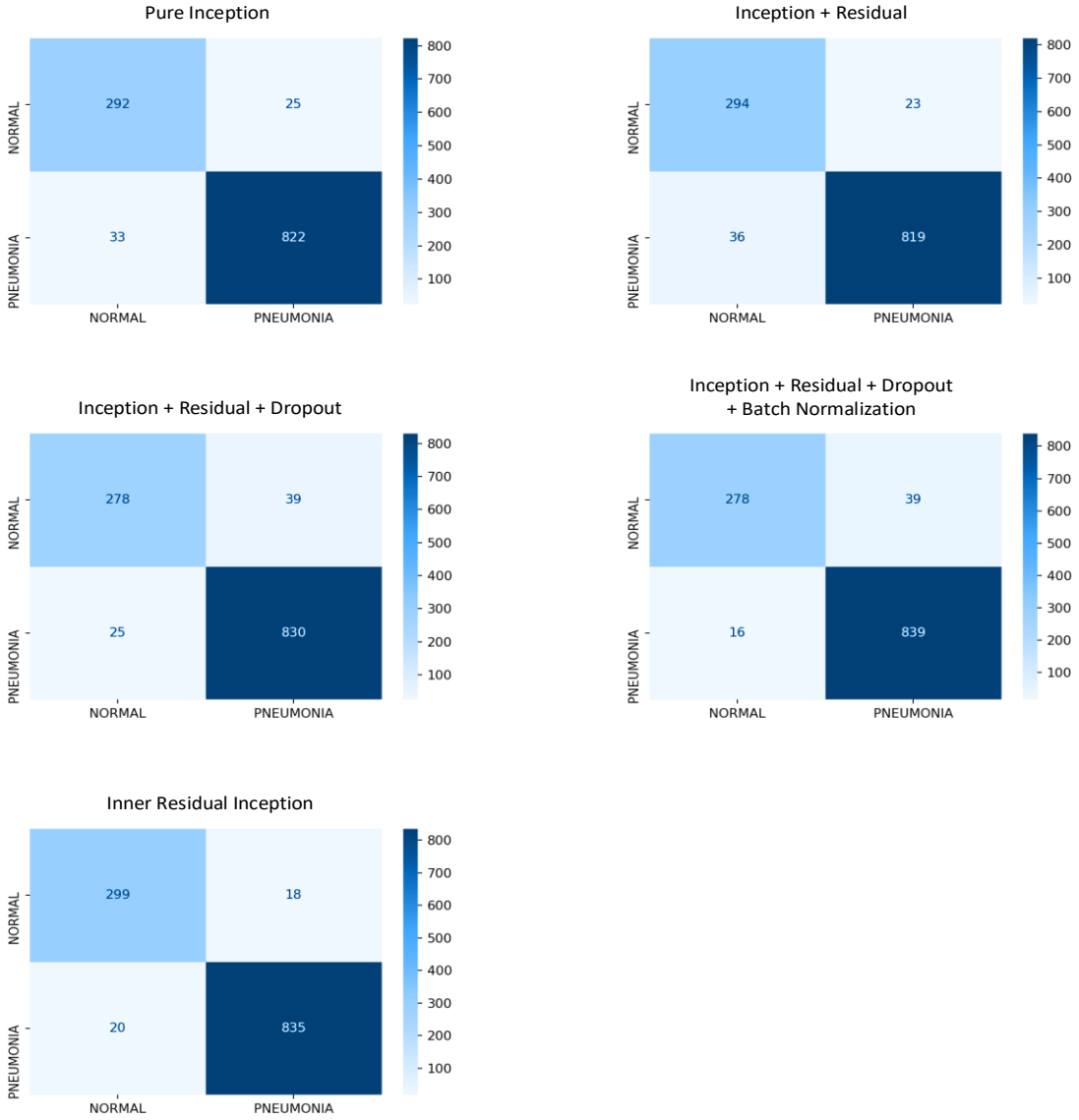

Fig 12: Confusion Matrix for the five models

Table 2 introduce the numerical results of three proposed models. The third model achieve the higher accuracy value equal 96.76% with higher balance accuracy equal 95.08%.

Table2: The results of the proposed models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (%)	Balanced Accuracy (%)
Proposed Model 1	94.54	95.51	97.08	96.29	95.92	92.39
Proposed Model 2	94.8	95.22	97.78	96.48	93.93	92.26
Proposed Model 3	**96.76**	**97.74**	**97.99**	**97.86**	**97.07**	**95.08**

Fig. 13 shows the comparison between the numerical results of precision. The proposed model 3 achieves a higher value of 97.74% As it has a higher success rate than the proposed model 2 with 2.52%, but ResNet 50 [25] achieves a lower value equal to 82.33%.

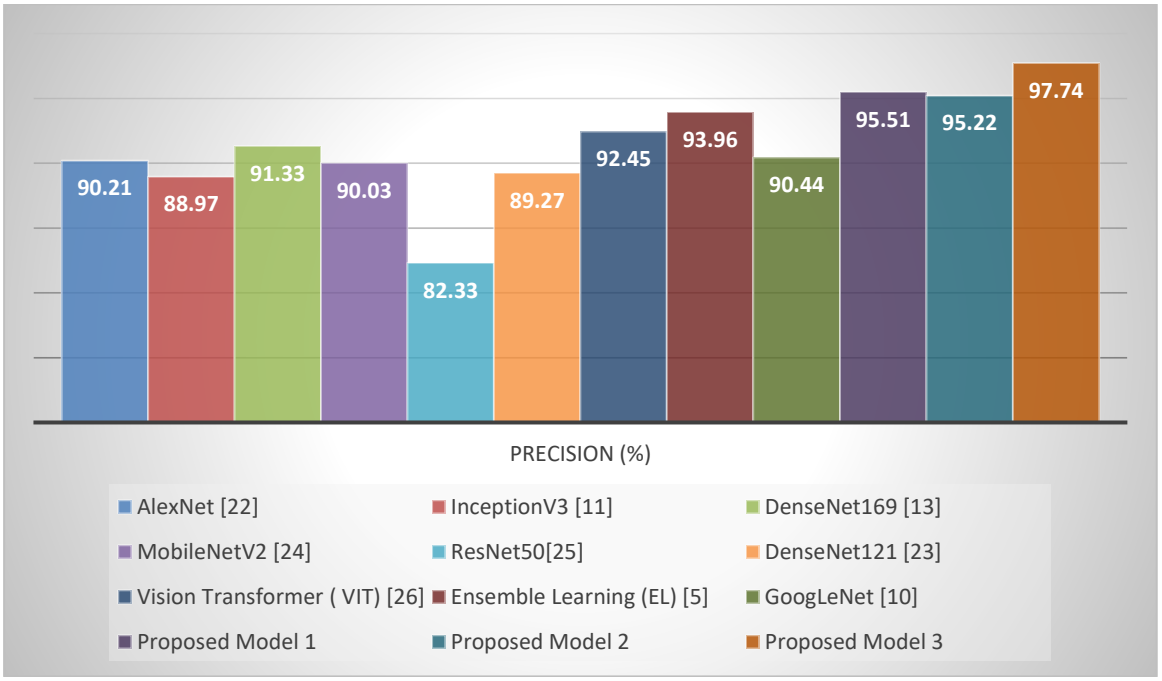

Fig 13: The precision value for the proposed models and well-known methods

Fig. 14 shows the comparison between the values of recall. The AlexNet[22] achieves a higher value equal to 98.97%, which is higher than the proposed model 3 with 1.02%. The proposed model 3 achieves 97.99% and has a higher success rate than the proposed model 2 with 0.21%, but ResNet 50 [25] achieves a lower value equal to 82.22%.

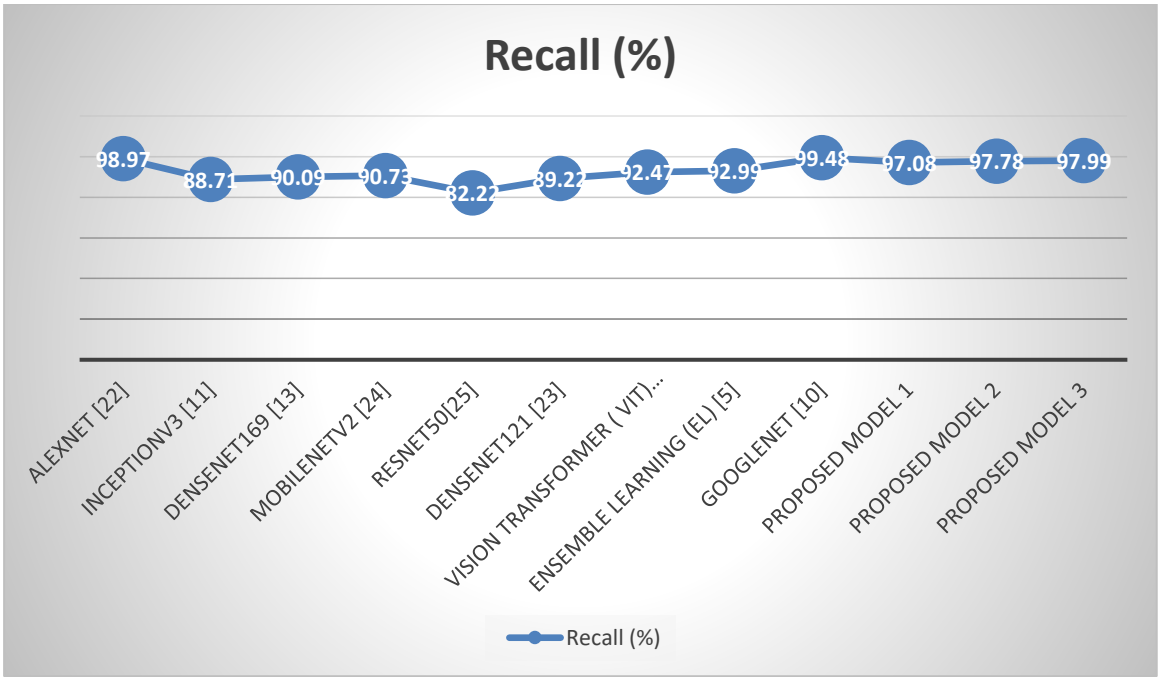

Fig 14: The recall value for the proposed models and well-known methods

Fig. 15 shows the comparison between the values of F1-score. The proposed model 3 achieves a higher value equal to 97.86%, which is higher than the proposed model 2 with 1.42%. MobileNetV2[24] achieves a lower value equal to 82.26%.

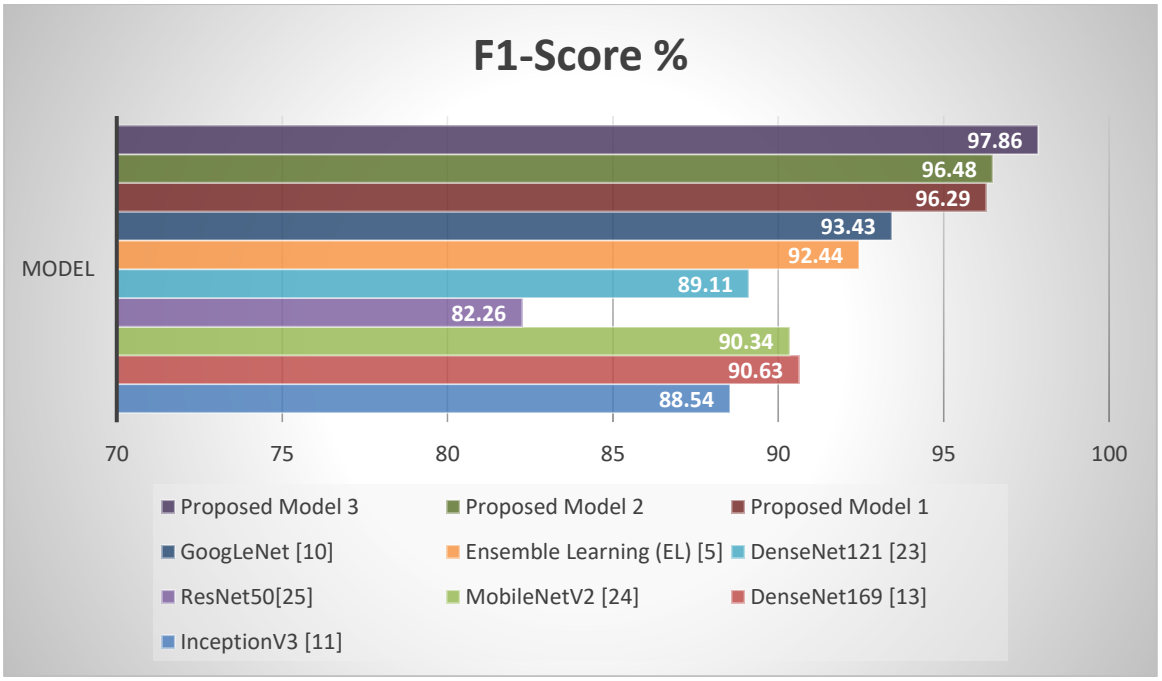

Fig 15: The F1-score value for the proposed models and well-known methods

Table 3 introduces a numerical comparison between the three proposed models and well-known algorithms using accuracy values and publish year. The proposed model 3 achieves a higher value of accuracy equal to 96.76%, which is higher than the proposed model 2 with 1.94%. Ensemble learning (NL) [5] achieves 93.91% but is still lower than the proposed model 3 with 2.85%. DCGAN [27] achieves a lower value of accuracy equal to 84.19%

Table 3. Comparative accuracy results for the well-known methods and the proposed models

Method/Ref.	Accuracy (%)	Year
DCGAN / Moradi, A.; et al. [27]	84.19	2018
Kermany et al. [18]	92.80	2018
VGG16/ Ayan, E.; et al. [28]	87.00	2019
Stephen, O., et al., [8]	93.73	2019
Liang, G.; et al.[29]	90.50	2020
DenseNet121/ Salehi, M; et al. [30]	86.80	2021
Ensemble Learning (EL)/ Alhassan Mabrouk et al. [5]	93.91	2022
Proposed Model 1	94.54	2022
Proposed Model 2	94.80	2022
Proposed Model 3	**96.76**	**2022**

4. Discussion

The proposed model (inner residual inception) improved the inception block and can replace inception blocks in any model to boost its accuracy. From the results, we highlight some of its advantages and limitations:

4.1 Advantages

1. The proposed model surpassed current state-of-the-art models on pneumonia using Chest X-Ray
2. The proposed model introduced a new inception block that is more efficient and powerful than current inception blocks.
3. The proposed model found a good balance in reducing both Type 1 & 2 errors.
4. The proposed model used 64 size image which is very small compared to other models that used 224 images and bigger.
5. The proposed model is 14.5 faster than AlexNet, 1.67 faster than DenseNet121, 6.2 faster than InceptionV3, 1.67 faster than GoogLeNet and 2.6 faster than ResNet18.
6. The proposed model can deal with unbalanced datasets.

4.2 Limitations

1. The proposed model wasn't tested for real data that can suffer from noisy chest x-ray.
2. The proposed model can 't fully replaces doctors as the total accuracy didn't surpass human-level accuracy.

5. Conclusion

Using machine learning to diagnose pneumonia from chest X-rays increases the accuracy of the diagnosis that a doctor may rely on in treating patients. The main contribution of this paper is the development of a model for classifying chest X-ray images according to whether they have pneumonia or not. This paper has suggested three models relying on machine learning techniques. The first model relied

on inception, residual, and dropout, and it achieved a 94.54% accuracy rate. The second model depends on adding a batch normalization layer to the first model, which achieved a 94.8% accuracy rate. The third model relied on introducing a new inception block that is more efficient and powerful than current inception blocks. It achieved a 96.76% accuracy rate. The main conclusion of this paper is the extent to which it is possible to rely on the third model, as it has achieved the highest accuracy and balance accuracy compared to other models in the literature. So, an automated model for diagnosing pneumonia with a high degree of accuracy can be relied upon by the doctor, but we cannot ignore the doctor's experience in diagnosis. in the future, suggests trying this model on actual patients' images that contain some image problems such as noise or missing parts of the image, as well as adding some improvements to this model for increasing accuracy, efficiency, and effectiveness.

Funding: This study was not funded by any organization.

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication: All author gave their consent.

Data availability: Available

References

[1] Gilani, Z., Kwong, Y. D., Levine, O. S., Deloria-Knoll, M., Scott, J. A., O'Brien, K. L., & Feikin, D. R. (2012). A literature review and Survey of Childhood Pneumonia Etiology Studies: 2000–2010. Clinical Infectious Diseases, 54(suppl_2). https://doi.org/10.1093/cid/cir1053

[2] Ayan, E., Karabulut, B., & Ünver, H. M. (2021). Diagnosis of pediatric pneumonia with ensemble of deep convolutional neural networks in chest X-ray images. Arabian Journal for Science and Engineering, 47(2), 2123–2139. https://doi.org/10.1007/s13369-021-06127-z

[3] Suryaa, V. S., R, A. X., & S, A. M. (2021). Efficient DNN ensemble for pneumonia detection in chest X-ray images. International Journal of Advanced Computer Science and Applications, 12(10). https://doi.org/10.14569/ijacsa.2021.0121084

[4] Zhang, D., Ren, F., Li, Y., Na, L., & Ma, Y. (2021). Pneumonia detection from chest x-ray images based on Convolutional Neural Network. Electronics, 10(13), 1512. https://doi.org/10.3390/electronics10131512

[5] Mabrouk, A., Díaz Redondo, R. P., Dahou, A., Abd Elaziz, M., & Kayed, M. (2022). Pneumonia detection on chest X-ray images using ensemble of deep convolutional Neural Networks. Applied Sciences, 12(13), 6448. https://doi.org/10.3390/app12136448

[6] Chouhan, V., Singh, S. K., Khamparia, A., Gupta, D., Tiwari, P., Moreira, C., Damaševičius, R., & de Albuquerque, V. H. (2020). A novel transfer learning based approach for pneumonia detection in chest X-ray images. Applied Sciences, 10(2), 559. https://doi.org/10.3390/app10020559

[7] Sharma, H., Jain, J. S., Bansal, P., & Gupta, S. (2020). Feature extraction and classification of chest X-ray images using CNN to detect pneumonia. 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence). https://doi.org/10.1109/confluence47617.2020.9057809

[8] Stephen, O., Sain, M., Maduh, U. J., & Jeong, D.-U. (2019). An efficient deep learning approach to pneumonia classification in Healthcare. Journal of Healthcare Engineering, 2019, 1–7. https://doi.org/10.1155/2019/4180949

[9] Saraiva, A., Ferreira, N., Lopes de Sousa, L., Costa, N., Sousa, J., Santos, D., Valente, A., & Soares, S. (2019). Classification of images of childhood pneumonia using convolutional neural networks. Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies. https://doi.org/10.5220/0007404301120119

[10] Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2015.7298594
[11] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.308
[12] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-V4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1). https://doi.org/10.1609/aaai.v31i1.11231
[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.90
[14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(56), 1929–1958.
[15] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on International Conference on Machine Learning , 37, 448–456.
[16] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. Wiley.
[17] Kingma, D.P., & Ba, J. (2017). Adam: A method for stochastic optimization. arXiv preprint https://arxiv.org/abs/1412.6980v9
[18] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., … Zhang, K. (2018). Identifying medical diagnoses and treatable diseases by image-based Deep Learning. Cell, 172(5). https://doi.org/10.1016/j.cell.2018.02.010
[19] Kassani, S.H.; Kassani, P.H.; Wesolowski, M.J.; Schneider, K.A.; Deters, R. Classification of histopathological biopsy images using ensemble of deep learning networks. https://doi.org/10.48550/arXiv.1909.11870
[20] Ling, C. X., Huang, J., & Zhang, H. (2003). AUC: A better measure than accuracy in comparing learning algorithms. Advances in Artificial Intelligence, 329–341. https://doi.org/10.1007/3-540-44886-1_25
[21] Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. 2010 20th International Conference on Pattern Recognition. https://doi.org/10.1109/icpr.2010.764
[22] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional Neural Networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386
[23] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.243
[24] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2018.00474
[25] Targ S., Almeida D., & Lyman K.(2016). Resnet in resnet: Generalizing residual architectures. https://doi.org/10.48550/arXiv.1603.08029
[26] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X.; Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit A., & Houlsby N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. https://doi.org/10.48550/arXiv.2010.11929
[27] Moradi, M., Madani, A., Karargyris, A., & Syeda-Mahmood, T. F. (2018). Chest X-ray generation and data augmentation for cardiovascular abnormality classification. Medical Imaging 2018: Image Processing. https://doi.org/10.1117/12.2293971
[28] Ayan, E., & Unver, H. M. (2019). Diagnosis of pneumonia from chest X-ray images using Deep Learning. 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT). https://doi.org/10.1109/ebbt.2019.8741582
[29] Liang, G., & Zheng, L. (2020). A transfer learning method with deep residual network for pediatric pneumonia diagnosis. Computer Methods and Programs in Biomedicine, 187, 104964. https://doi.org/10.1016/j.cmpb.2019.06.023

[30] Salehi, M., Mohammadi, R., Ghaffari, H., Sadighi, N., & Reiazi, R. (2021). Automated detection of pneumonia cases using deep transfer learning with paediatric chest X-ray images. The British Journal of Radiology, 94(1121), 20201263. https://doi.org/10.1259/bjr.20201263