Scope
Volume 14 Number 02 June 2024

A Deep Analysis Study on Leverage Convolutional Neural Networks Method for Pneumonia Detection System using Machine Learning

Shrestha Majumder
Assistant Professor, Department of Computational Sciences, Brainware University, Kolkata, India

Dr R.Naveenkumar
Associate Professor, Department of Computer Science and Engineering, Brainware University, Kolkata, India

Abstract: The use of data mining and machine learning has become essential for the detection and prevention of various diseases. For children under five, interstitial lung diseases like pneumonia are the main cause of death. Children under five every year are affected by each day for various reasons. This includes around the maximum number of newborn babies. Almost all the maximum deaths are preventable. According to a UNICEF report, there are more than 1,400 instances of pneumonia per 100,000 children worldwide or one case for every 71 children annually. The majority of the affected kids were less than two. The healing process for children with pneumonia can be accelerated with prompt diagnosis. To effectively identify pneumonic lungs from chest X-rays, we have applied convolutional neural network models in this study for a better result. Medical professionals can use these models to treat pneumonia in the actual world. The first, second, third and fourth model consists of two convolutional layers. The first model achieves an accuracy of 89.74%, the second one reaches an accuracy of 85.26%. Furthermore, recall and F1 scores are calculated from the confusion matrix of each model for better evaluation.
Keywords: Convolutional neural networks (CNNs), Pneumonia detection, ReLU, Max-pooling, and Forward propagation theory.

Introduction

This paper presents convolutional neural network models to accurately detect pneumonic lungs from chest X-rays, which can be utilized in the real world by medical practitioners to treat pneumonia [1]. Early detection of pneumonia is facilitated by these models, which can quickly classify chest X-ray pictures into two categories: normal and pneumonia. Although transfer learning models based on convolutional neural networks like VGG16 and VGG19 are some of the

most successful ImageNet dataset models with pre-trained weights, they were not trained on this dataset as the size of dataset taken for our research is not as extensive compared to ones which generally employ transfer learning [2] . Two classification models were built using CNN to detect pneumonia from chest X-ray images to help control this deadly infection in children and other age groups [3]. There is no direct association between the number of convolutional layers and the accuracy of the model, however the accuracy of the model is strongly correlated with the size of the dataset, meaning that using large datasets helps improve the accuracy of the model [4]. By analyzing the models after each execution, a specific amount of combinations of convolution layers, dense layers, dropouts, and learning rates must be trained in order to get the optimum outcomes. A model is considered underperforming, ineffective, and even dangerous if it achieves high accuracy but low recall values [5]. This is because larger false-negative values indicate a higher percentage of cases in which the patient is correctly predicted by the model to be normal when, in fact, they are ill. Hence, it would risk the patient's life. The research process is divided into four key parts: formulating the problem, gathering datasets, experimenting, and summarizing the outcomes. These phases determine the methodologies used in this study[8].
This research started with formulating the research problem that is reviewing the literature and formulating the research problem [10]. Following the creation of the research topic, this study determined the objectives, research scope, and research process limits. The gathering of datasets is the second stage of the study [11].The dataset items were collected from Kaggle [12] . The data preparation stage constituted the third stage of the investigation and comprised the following: Transforming Data into the Correct Format Preprocessing Data - Manipulating Data using Machine Learning Several experiments were carried out and data was gathered during the experimentation phase [13,14].

Creating Deep Learning Model for Pneumonia

In this paper, we will address treating pneumonia, which is a potentially fatal illness that can affect one or both lungs and is typically brought on by bacteria, fungi, or viruses. Using the x-rays we have, we will identify this lung condition. The dataset for chest X-rays was obtained from Kaggle and includes a variety of x-ray pictures that are categorized as "Normal" or "Pneumonia." We will be creating a deep-learning model which will tell us whether the person has pneumonia disease or not having pneumonia.

CNN Architecture using Tensor flow back:

CNN models were built from the ground up and trained on Kaggle's Chest X-ray images (Pneumonia) dataset. Keras neural network library with TensorFlow backend has been used to implement the models. The dataset consists of 5218 training images,624 testing images and 18 validation images. The technique of data augmentation has been used to improve the dataset's performance. The four models have been trained on the training dataset, each with a different number of convolutional layers. This model was trained for 5 epochs, with training and testing batch sizes of 4 and 1, respectively.

Activation functions: All four of the models in this research use the Softmax activation function. All four models use this widely used activation function in their last dense layer. Inputs are normalized into a probability distribution using this activation function. With this kind of activation function, the most common cost function is the categorical cross-entropy cost function.

Pooling layer: Convolutional layers are followed by pooling layers. The type of pooling layer used in models is max-pooling layers. The 2 × 2 max-pooling layer chooses the highest pixel intensity values from the image window that the kernel is currently covering. Max-pooling is used to down sample images, hence reducing the dimensionality and complexity of the image . Two other types of pooling layers can also be used which are general pooling and overlapping pooling. The models presented in this paper use the max-pooling technique as it helps recognize salient features in the image.

Flattening layer and fully connected layers: Convolutional layers are followed by pooling layers. The type of pooling layer used in models is max-pooling layers. The 2 × 2 max-pooling layer chooses the highest pixel intensity values from the image window that the kernel is currently covering. Max-pooling is used to down sample images, hence reducing the dimensionality and complexity of the image. Two other types of pooling layers can also be used which are general pooling and overlapping pooling. The max-pooling technique is used in the models provided in this paper because it facilitates the recognition of important elements in the image.

```
flatten_layer =Flatten()(vgg_model.output)
prediction =Dense(len(classes), activation='softmax')(flatten_layer)
```

Based on the above code the flattened part of the image is calculated

Reducing overfitting: Since the first model shows significant overfitting, the later models used the dropout technique. The dropout technique helps to reduce overfitting and tackles the problem of vanishing gradients. By using the dropout strategy, each neuron is encouraged to create a unique representation of the incoming data. During the training phase, this approach randomly breaks connections between neurons in successive layers. Additionally, the models' learning rate was adjusted to lessen overfitting. Data augmentation techniques can also be employed to reduce overfitting.

Algorithm of CNN Classifiers: The algorithms used in the convolutional neural network classifiers have been explained in Figs. 1 and 2. Figure 3 shows the flowchart of the overall schema of research. The number of epochs for all the classifier models presented in this paper was fixed at 20 after training and testing several CNN models over the course of the research. Classifier models trained for more number of epochs have shown overfitting. Several optimizer functions were also trained and studied. Adam optimizer function was finalized to be used for all classifiers after it gave the best results. Initially, a simple classifier model with a convolutional layer of image size set to 64 * 64, 32 feature maps and employing the ReLU activation function was trained. A fully connected dense layer with 128 perceptrons was utilized. To improve the result, the second classifier model was trained with one more convolutional layer of 64 feature maps for better feature extraction. The number of perceptrons in the dense layer was also doubled to 256 so that better learning could be achieved. The third model was trained for three convolutional layers with 128 feature maps in third convolutional layer for more detailed feature extraction. Dense layer was kept unchanged. Dropout layer was introduced at 0.3, and learning rate of optimizer was lowered to 0.0001 to reduce the overfitting. The final fourth classifier model was trained for four convolutional layers with 256 feature maps in fourth convolutional layer. Dense layer, dropout layer and learning rate were kept same as third classifier model. The results have been summarized in the subsequent section of this paper.

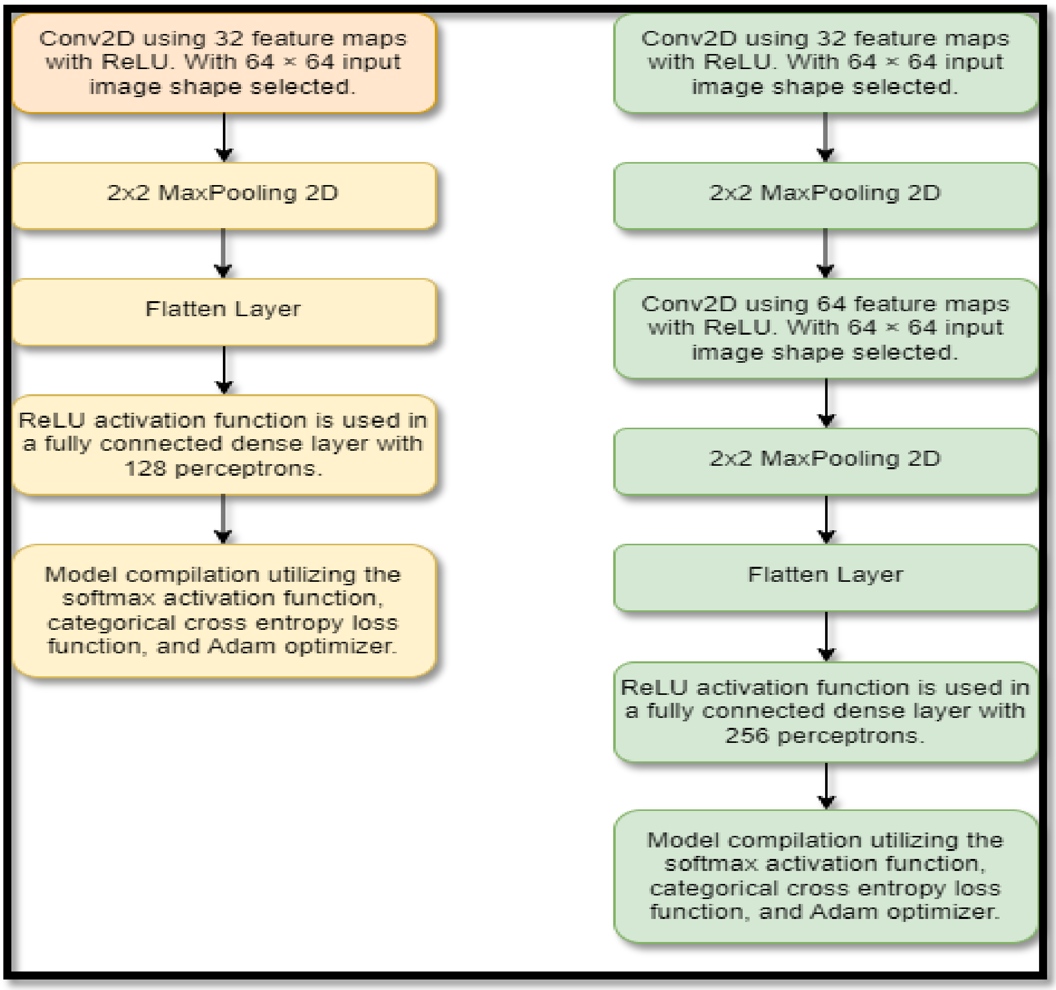


Figure 1: Algorithms of CNN classifier model 1 (left) and model 2 (right)

Algorithm of CNN classifiers:

Vgg16: Convolution neural net (CNN) architecture is represented by VGG16.It is regarded as one of the best vision model designs available at this time. The most distinctive feature of VGG16 is that its convolution layers of a 3x3 filter with stride 1 are its main focus, rather than having a lot of hyper-parameters. The padding and maxpool layer of a 2x2 filter with stride 2 are also always employed. Throughout the whole architecture, the convolution and max pool layer arrangements are maintained. Ultimately, it consists of two FC (completely connected layers) with a softmax as the output. The 16 in VGG16 stands for the 16 weighted layers it contains. This network has 138 million (approximately) parameters and is really huge.

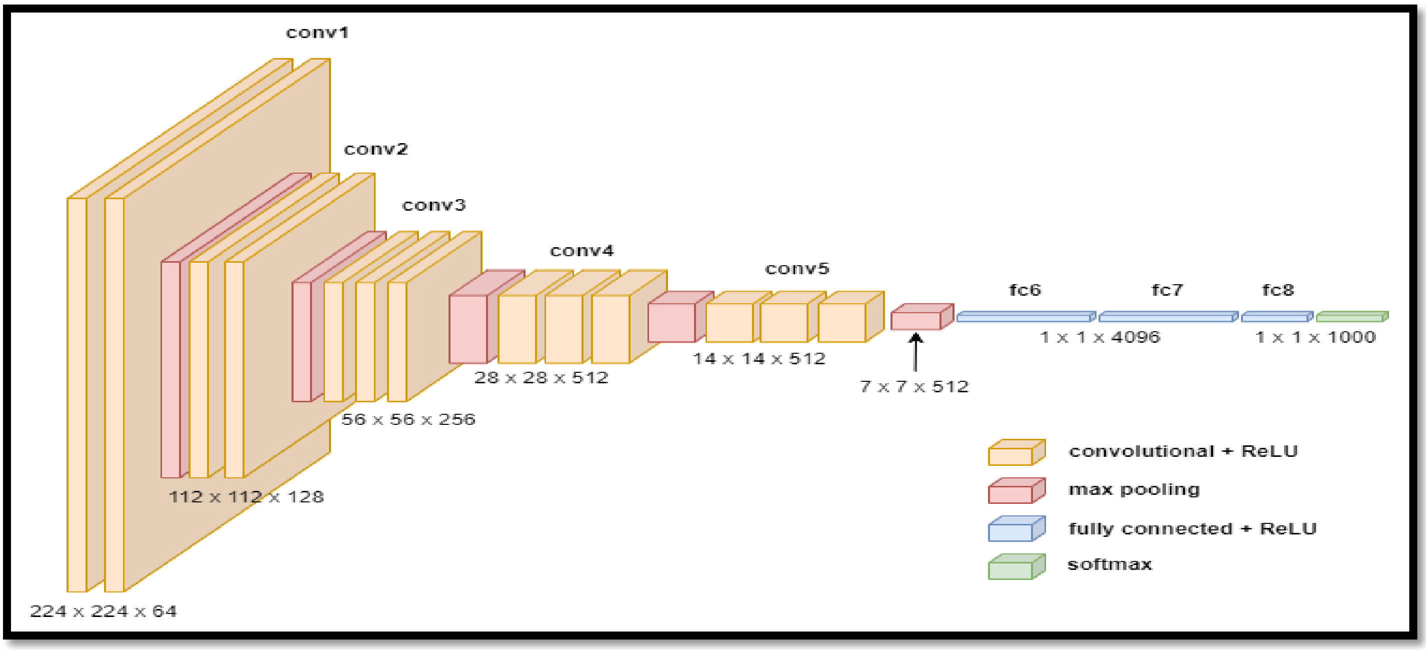


Figure 2: VGG16 Architecture

Vgg19: In the context of large-scale image recognition, we examine how the convolutional network's depth affects accuracy in this work. Our primary contribution is a comprehensive analysis of networks with progressively deeper layers utilizing an architecture with extremely small (3x3) convolution filters. The results demonstrate that expanding the depth to 16–19 weight layers can lead to a notable improvement over the state-of-the-art setups. Based on these results, our team entered the 2014 ImageNet Challenge and placed first and second in the localization and classification tracks, respectively. We further demonstrate that our representations attain state-of-the-art outcomes when they are well-generalized to additional datasets. To encourage more study on the application of deep visual representations in computer vision, we have made our two top-performing ConvNet models publicly available.

Dataset: The dataset is arranged into three subfolders, one for each image category (Pneumonia/Normal), and three folders (train, test, and val). There are two categories (Normal/Pneumonia) and 5,863 X-ray images (JPEG). Anterior-posterior chest X-ray images were chosen from retrospective cohorts of pediatric patients from Guangzhou Women and Children's Medical Center, Guangzhou, aged one to five. Every chest X-ray image was taken as a standard clinical procedure for the patients. In order to ensure quality control for the interpretation of chest x-ray pictures, all chest radiographs were first screened to eliminate any low-quality or unreadable scans. Before the photos' diagnoses could be used to train the AI system, they were evaluated by two board-certified medical professionals. A third expert verified the evaluation set to make sure there were no grading problems.

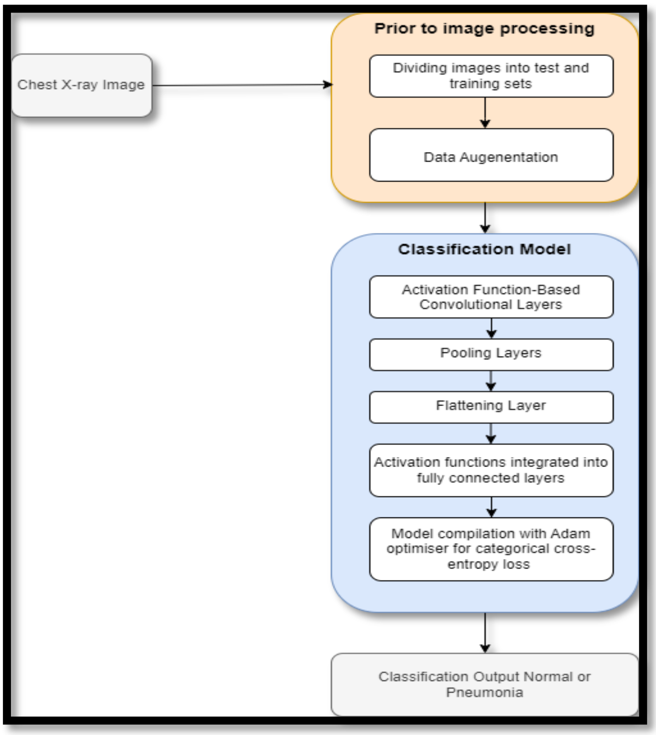
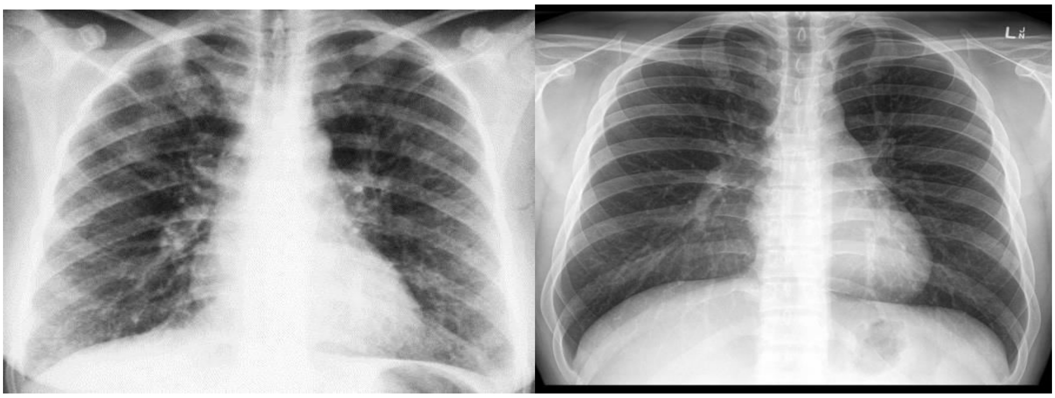


Figure 3: Detailed schema of the experiment conducted



Normal Lung **Affected Lung**

Figure 4: The sample images from the dataset used during the research

Experimental Setup and Result Analysis

As we have developed models for both pneumonia and diabetes we will discuss the setup for the model and the corresponding results individually without any confusion, as we did earlier.

First, let's explore the methodology we used to develop pneumonia :

Tools and Technologies:

VGG16:

This Convolutional Neural Network (CNN) architecture is simple and widely used for ImageNet, a large visible database mission used in research on visual object detection software.

Transfer learning (TL):

This deep learning approach involves pre-training a neural network, storing the knowledge it gains from addressing a particular problem, and then applying that knowledge to new datasets.

In this article suggests that the knowledge acquired from identifying 1000 distinct classes in ImageNet could be useful in identifying the illness.

Model architecture:

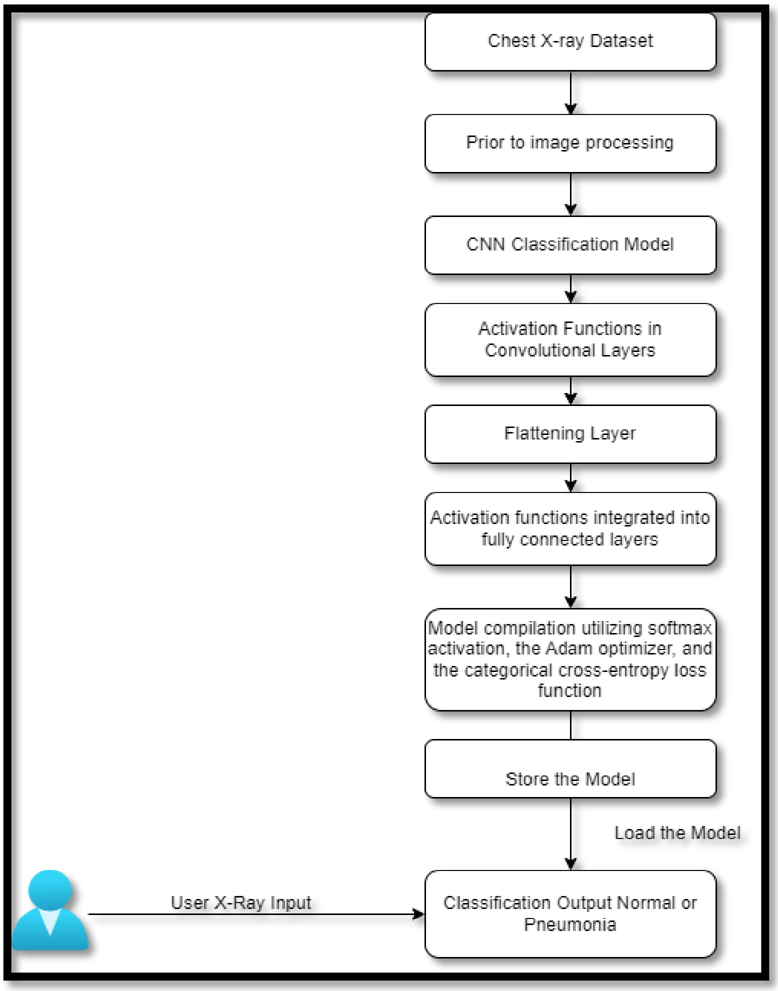


Figure 5: X-ray Input Model

Modules Required for Implementation :

Keras:

It is a deep learning module for Python that utilizes the TensorFlow framework. It was designed to make deep learning model implementation for research and development as simple and quick as feasible.Being the fact that Keras runs on top of Keras we have to install TensorFlow first. To install this library, type the following commands in IDE/terminal.

pip install tensorflow

pip install keras

SciPy: A free and open-source Python module for technical and scientific computing is called SciPy. We must install the SciPy module since this topic requires Image Transformations. Enter the following command in the IDE or terminal to install this library.

pip install scipy

glob:

The glob module in Python is used to obtain path names and files that match a given pattern. This module is used in this post to determine the number of classes that are present in our train dataset folder.
pip install glob2

Results and Discussion:

The training and validation accuracies throughout five epochs, as well as the training and validation losses, are displayed in the provided data. This is a thorough breakdown of the findings.

Epoch	training loss	validation loss
1	0.2829	0.7651
2	0.2256	0.9566
3	0.2114	0.9347
4	0.2297	0.6316
5	0.2478	1.033

Figure 6: Training and Validation Loss

Epoch	Training Accuracy	validation accuracy
1	0.9199	0.8654
2	0.9511	0.8013
3	0.9553	0.883
4	0.9557	0.9215
5	0.9553	0.8958

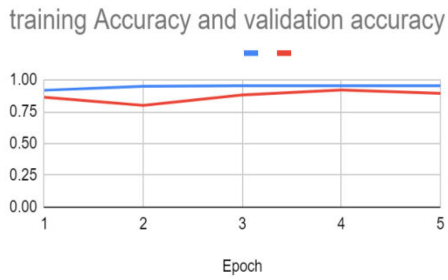


Figure 7: Training and Validation Accuracy

The VGG16 model successfully collects and learns pertinent features for pneumonia detection from chest X-ray images thanks to its deep architecture. Its capacity to classify pneumonia is demonstrated by its high training and validation accuracy. Nonetheless, variations in validation loss and sporadic indications of overfitting imply that although the model is robust, cautious management is necessary to guarantee the best possible generalization.

Initial Performance: With a high training accuracy and a respectable validation accuracy, the VGG16 model performs well in the early stages of pneumonia detection.

Overfitting Signs: Starting with epoch 2, overfitting is indicated by a marked increase in validation loss at the expense of a high training accuracy.

Improvement and Fluctuations: Around epoch 4, the model's generalization gets better, as seen by decreased validation loss and increased accuracy. Variations in validation loss, however, point to a possible inconsistent model performance on unobserved data.

Final Analysis: The model exhibits great training and validation accuracy by epoch 5, although overfitting is evident due to substantial validation loss. Although the training accuracy is 95.53%, the validation accuracy stabilizes at 89.58%, suggesting possible problems with generalization.

Recommendations for Improvement:

Regularization: To avoid overfitting, use strategies like dropout or L2 regularization.

Early Stopping: When validation performance no longer improves, use early stopping to end training.

Data Augmentation: To boost generalization, add supplemented data to the training dataset.
Hyperparameter tuning: To determine the best training configuration, modify batch sizes, learning rates, and other hyperparameters.
These issues can be resolved in order to improve VGG16's performance in pneumonia identification and guarantee strong and accurate forecasts.
In conclusion, even if the VGG16 model detects pneumonia with encouraging results, careful attention is required to reduce overfitting and enhance generalization to new data.

Comparison of Performance of Models

Classifier Model	Recall(%)	F1 Score(%)
Model 1(one conv. layer)	96	92
Model 2(two conv. layers)	94	89

Figure 8: Performance of classifier model Recall and F1 score

1. Classifier Model

The models that are being compared are listed in this column. This table includes two models:

- **Model 1**: There is only one convolutional (conv.) layer in this model.
- **Model 2**: There are two convolutional (conv.) layers in this model.

2. Recall (%)

Recall is the percentage of real positives that the model properly identifies; it is sometimes referred to as sensitivity or true positive rate. It is computed as follows:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Negatives (FN)}}$$

A model's recall quantifies its capacity to locate each pertinent instance within a dataset. This table contains:

- With a 96% recall rate, Model 1 accurately detects 96% of the positive cases.

- With a 94% recall rate, Model 2 accurately detects 94% of the positive cases.

3. F1 Score (%)

The F1 score is a statistic that provides a balance between precision and recall, calculated as the harmonic mean of the two. When there is an imbalance in the distribution of classes, it is very helpful. One can compute the F1 score as follows:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where precision is: $$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

This table contains:

- With an F1 score of 92%, Model 1 exhibits a strong balance between recall and precision.
- Although it still performs well, Model 2's F1 score of 89% is a little lower than Model 1's, suggesting that it is not as balanced.

Comparative Analysis:

Model 1 (one conv. layer)

- **Recall: 96%**
    - When it comes to recognizing the positive cases, Model 1 is very successful. Merely 4% of the true benefits are overlooked.
- **F1 Score: 92%**
    - Because of its high F1 score, Model 1 is a dependable model for classification tasks where both precision and recall are crucial. It does this by maintaining a good balance between the two measures.

Model 2 (two conv. layers)

- **Recall:94%**
    - Although Model 2 misses 6% of the true positives, it still does a good job of detecting positive cases—just not as well as Model 1.
- **F1 Score: 89%**
    - Model 2 may have a somewhat lower precision or a higher rate of false positives than Model 1 since the F1 score is lower than that of Model 1. This suggests a less balanced recall and precision performance.

Although Model 2 (which has two convolutional layers) outperforms Model 1 (which has one convolutional layer) in terms of recall and F1 score. This implies that, in this situation, the more straightforward model (Model 1) outperforms the more intricate model (Model 2) in accurately recognizing positive cases and keeping a decent balance between precision and recall. This could be because of other complications the additional layer introduces, or it could be because of overfitting in Model 2, where the additional layer does not generalize as well to the dataset.

Classifier Model	Validation Accuracy(%)	Validation Loss(%)
Model 1(one conv. layer)	89.74	27.31
Model 2(two conv. layers)	85.26	38.36

Figure 9: Performance of classifier model 1 and model 2

Two classifier models are compared in the table: Model 1, which has a single convolutional layer, and Model 2, which has two convolutional layers. Validation loss and validation accuracy are the measures taken into account.

Model 1 (one convolutional layer)

- Validation Accuracy: 89.74%
- Validation Loss: 27.31%

Model 1 exhibits improved generalization and predictions that are more in line with actual values thanks to its higher validation accuracy and reduced validation loss. This shows that the model is not overfitting the training set and is appropriate for the job at hand.

Model 2 (two convolutional layers)

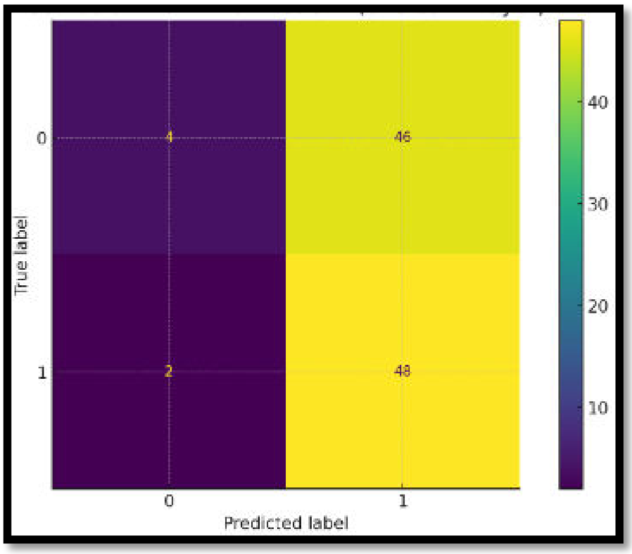
- Validation Accuracy: 85.26%
- Validation Loss: 38.36%

On both metrics, Model 2, with its extra convolutional layer, performs worse. The increased loss and decreased accuracy imply that the complexity is not beneficial to performance and may even cause overfitting.

Performance Comparison Summary

- **Generalization**: Model 1's higher validation accuracy suggests that it generalizes to new data more effectively.
- **Prediction Accuracy**: Based on its reduced validation loss, Model 1 predicts values that are more in line with reality.
- **Model Complexity**: Model 1's (one convolutional layer) more straightforward architecture works better than Model 2's (two convolutional layers).

Models 1 and 2 do better in this comparison. More generalization and accuracy are achieved on the validation set by the simpler, single-convolutional model than by the more intricate, two-convolutional model. The significance of selecting the right model complexity to prevent overfitting and guarantee good performance on unknown data is highlighted by this.
Below figures show the confusion matrices, accuracy graphs and loss graphs of all CNN classifier models.The loss and accuracy plots both exhibit overfitting. Low F1 scores, recall, and accuracy are also present. It obtained the highest accuracy and recall with the least amount of overfitting. Numerous attempts were made to improve the performance by varying the settings and adding additional convolutional layers.



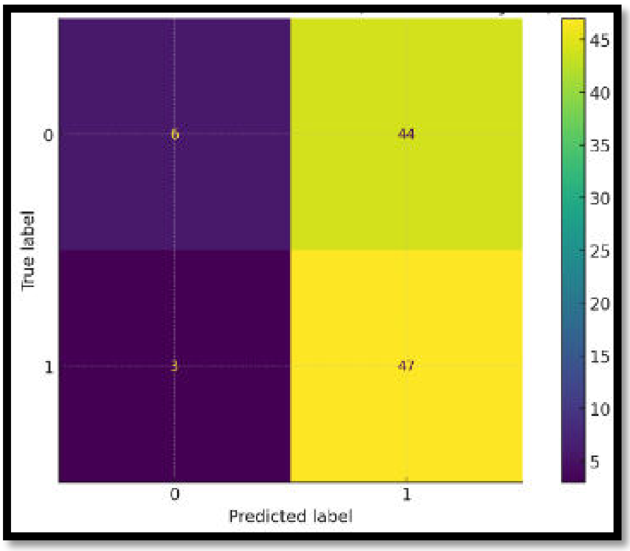
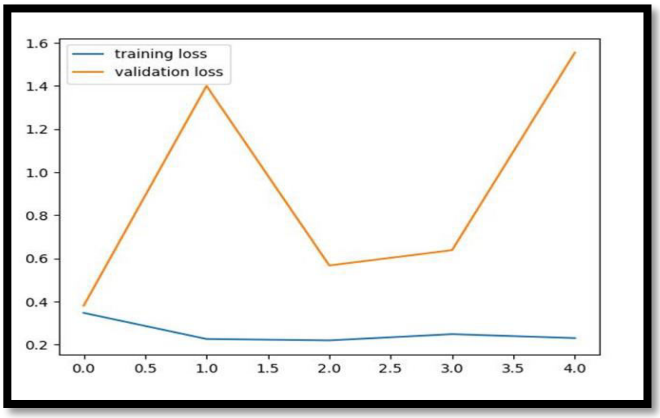


Figure 10: Confusion matrix for model 1 **Figure 11: Confusion matrix for model 2**



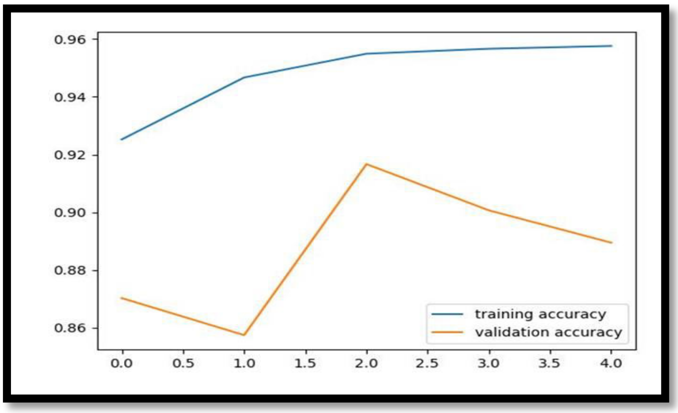


Figure 12: Figure 13:
Training loss vs validation loss using VGG16 **Training accuracy vs validation accuracy using VGG16**

Conclusion

The text highlights the critical role of data mining and machine learning in healthcare, specifically in detecting and preventing diseases like pneumonia and diabetes. Convolutional neural networks (CNNs) have been successfully applied to accurately detect pneumonia from chest X-rays, demonstrating high accuracy rates and potential for real-world application in medical settings. These CNN models offer quick and precise image processing, aiding in early diagnosis and treatment, thus potentially reducing mortality rates, especially among children. Moreover, evaluating various performance measures, including accuracy, precision, recall, and F-measure, these methods have shown promising results in early prediction and intervention, ultimately contributing to saving lives. In conclusion, the paper presented underscore the significance of leveraging machine learning and data mining techniques in healthcare. By accurately detecting diseases like pneumonia, nowadays researchers started working more on CNN techniques using machine learning algorithms, these approaches facilitate timely interventions, thereby improving patient outcomes and potentially reducing the burden of preventable diseases worldwide.

Reference

1. 1.Mitushi Soni, Dr. Sunita Varma:"Diabetes Prediction using Machine Learning Techniques" International Journal of Engineering Research & Technology (IJERT): ISSN: 2278-0181 IJERTV9IS090496 (This work is licensed under a Creative Commons Attribution 4.0 International License.) Published by : www.ijert.org Vol. 9 Issue 09, September-2020
2. Jaiswal, A.K., Tiwari, P., Kumar, S., Gupta, D., Khanna, A., Rodrigues, J.J.: Identifying pneumonia in chest x-rays: a deep learning approach. Measurement 145, 511–518 (2019)
3. Kim, D.H., MacKinnon, T.: Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. Clin. Radiol. 73(5), 439–445 (2018)
4. Bernal, J., Kushibar, K., Asfaw, D.S., Valverde, S., Oliver, A., Martí, R., Lladó, X.: Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. Artif. Intell. Med. 95, 64–81 (2019)
5. Arthur, F., Hossein, K.R.: Deep learning in medical image analysis: a third eye for doctors. J. Stomatology Oral Maxillofac. Surg.
6. Krizhevsky, A., Sutskever, I., Hinton, G.E:. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
7. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2014). arXiv preprint arXiv:1409.1556
8. Xu, Y., Jia, Z., Ai, Y., Zhang, F., Lai, M., Eric, I., Chang, C.: Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation. In: 2015 international conference on acoustics, speech and signal processing (ICASSP), pp. 947– 951 (2015)
9. ALzubi, J.A., Bharathikannan, B., Tanwar, S., Manikandan, R., Khanna, A., Thaventhiran, C.: Boosted neural network ensemble classification for lung cancer disease diagnosis. Appl. Soft Comput. 80, 579–591 (2019)
10. Vora, J., Tanwar, S., Polkowski, Z., Tyagi, S., Singh, P.K., Singh, Y.: Machine learning-based software effort estimation: an analysis. In: 11th International Conference on Electronics, computers and Artificial Intelligence (ECAI 2019), pp. 1–6, University of Pitesti, Pitesti, Romania, 27–29 June 2019
11. www.familjadheshendeti.com Bo He, Kuang-i Shu and Heng Zhang, Machine Learning and Data Mining in Diabetes Diagnosis and Treatment, IOP Conference Series: Materials Science and Engineering, Volume 490, Issue 4, IOP Conf. Series: Materials Science and Engineering 490 (2019) 042049 IOP
12. 13.Bisandu, Desmond & Datiri, Dorcas & Onokpasa, Eva & Thomas, Godwin & Haruna, Musa & Aliyu, Aminu. (2019). Diabetes Prediction using Data Mining Techniques. International Journal of Innovation Science. 4. 103-111.
13. Pang-Ning Tan; Michael Steinbach; Anuj Karpatne; Vipin Kuma Introduction to Data Mining 2nd ed, Publisher: Pearson, 2019, Print ISBN: 9780133128901, 0133128903 eText ISBN: 9780134080284, 013408028.