STREAMLIT APPLICATION FOR PNEUMONIA DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

M. SHASHIDHAR, [2]C. RATNA PRABHA, [3]Y. R. JANARDHAN REDDY, [4]P. RAMA RAO

[1,2,3,4]Assistant Professor, Department of Computer Science and Engineering, G Pulla Reddy Engineering College, Kurnool, India

ABSTRACT: Pneumonia was a severe lung infection that viruses or bacteria could cause and could range in severity. Streptococcus pneumoniae was a common bacterium that could cause life-threatening pneumonia if left untreated. This infection caused inflammation in the lungs, making it difficult to breathe and reducing the amount of oxygen the body received. Pneumonia could affect one or both lungs, and it occurs when fluid or pus-filled the air sacs (alveoli) in the lungs. In remote areas, an automated system for detecting pneumonia would have been very helpful in treating patients quickly. Convolutional Neural Networks (CNNs) were a popular deep learning algorithm used to classify diseases by analyzing medical images, and they had successfully detected pneumonia. Pre-trained CNN models that had learned features from large datasets could benefit image classification tasks. By analyzing chest X-ray images, a deep learning algorithm could identify whether a patient had pneumonia, with an accuracy rate of almost 90%. Users could upload an X-ray image to the Streamlit web application and view the results.

KEYWORDS: Pneumonia, X-ray images, CNN, Streamlit web application**.**

I. INTRODUCTION

Pneumonia, characterized by lung inflammation resulting from viral or bacterial infections, can manifest in varying degrees of severity. This inflammation hampers the ability of patients to obtain sufficient oxygen, as it involves filling air sacs (alveoli) in the lungs with fluid or pus, potentially affecting one or both lungs [1].
Tragically, pneumonia claims the lives of over 800,000 children under five annually, with approximately 2,200 deaths occurring daily. The incidence of pneumonia stands at over 1,400 infected children per 100,000 children [2]. Despite the expertise of seasoned medical professionals, accurately diagnosing pneumonia solely based on X-ray images has remained an enormous challenge. This difficulty arises from X-ray images exhibiting similar region characteristics for different diseases, including lung cancer. Consequently, the traditional diagnostic methods for pneumonia are extremely time and energy-consuming, and establishing a standardized diagnostic process for determining whether a patient has pneumonia has been infeasible. This is where Convolutional Neural Networks (CNNs) prove invaluable, as they possess automatic feature extraction capabilities [3].

Artificial intelligence (AI) has a machine learning subfield that allows systems to learn from experience and improve performance without explicit programming. It entails the development of computer programs that can use the information to learn independently [4]. On the other hand, deep learning is a branch of machine learning that takes its cues, particularly from artificial neural networks, from the structure and operation of the human brain. As shown in Figure, deep learning technology powers a variety of everyday goods and services, including digital assistants, voice-activated TV remotes, credit card fraud detection systems, and self-driving cars [5].

In Machine Learning, feature extraction is a labor-intensive and time-consuming endeavor that demands significant expertise. However, in Deep Learning, the neural network performs feature extraction and classification tasks, constantly improving performance by comparing predicted and actual outputs [6]. Artificial neural networks (ANNs) or simulated neural networks (SNNs), often known as neural networks, are a subset of machine learning techniques that serve as the basis for deep learning algorithms. They were created precisely to mimic the organization and operation of the human brain, notably the patterns of communication between organic neurons.

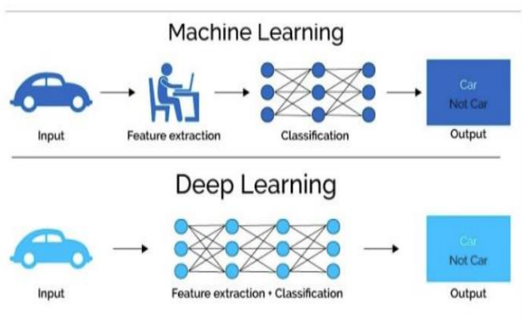

Fig. 1: MACHINE LEARNING VS DEEP LEARNING

Artificial neural networks consist of different layers, comprising an input layer, one or more hidden layers, and an output layer. Each unit within these layers, known as an artificial neuron, is interconnected with other neurons and is associated with a weight and a threshold [7]. As illustrated in the provided diagram, the input data is fed into the input layer and propagated through subsequent hidden layers. Each input is multiplied by its corresponding weight within the hidden layer and then aggregated. Now, let's focus on Convolutional Neural Networks (CNNs). These specialized artificial neural networks are extensively employed for image recognition and processing tasks. CNNs are specifically designed to process pixel data, utilizing a system similar to a multilayer perceptron but with reduced processing requirements, as shown in Fig. 2.

Architecture of CNN:

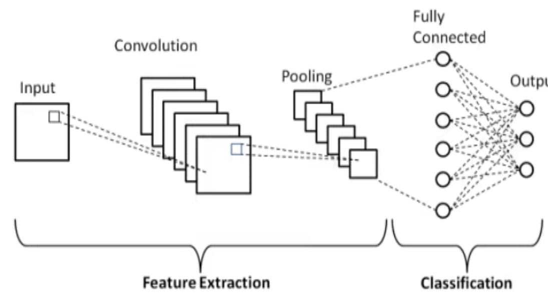

Fig. 2: ARCHITECTURE CONVOLUTIONAL NEURAL NETWORK

The primary objective of a convolutional layer is to identify visual features in images, such as edges, lines, and color variations. This property is particularly intriguing because once the network has learned a specific feature at one location in an image, it can recognize it elsewhere as well.

Convolutional Neural Networks (CNNs) utilize filters (kernels or feature detectors) to detect these features throughout an image. The filter scans over different parts of the image, checking for the presence of the target feature. The results from the filter's operations are summed and then passed through an activation function.

Pooling layers play a role in reducing the dimensions of the generated feature maps. This reduction helps decrease the number of parameters that need to be learned and the overall computational load on the network. The feature map created by the convolutional layer is summarised by the pooling layer's characteristics in a particular area [8].

To get the data ready for input into the following layer, it is necessary to flatten it into a onedimensional array. The output of the convolutional layers can be flattened to form a single, lengthy feature vector, which the network can then analyse further, as seen in Fig. 3.

Fig. 3: FLATTENING

Contribution of the Paper

- Our proposed model utilizes Convolutional Neural Networks (CNNs) and a publicly available dataset.
- We performed various image operations, such as rotations and contrast adjustments, to enhance the training process. The objective of our model is to predict the likelihood of pneumonia if a case is detected.
- To provide an accessible interface for users, we developed a Web Application using Streamlit.
- This application allows users to input relevant information and displays the corresponding output, including the image and the prediction of the pneumonia case.

Highlights

1. The proposed model gives more accuracy than existing models. It's almost 90%.
2. It gives better results even the input image is tilted.
3. If the case is pneumonia, it will predict the percentage of pneumonia too.

II. LITERATURE SURVEY

The article relied on data from the "Chest X-Ray8" dataset, which, although significant, lacks diversity in X-ray capturing setup and human subjects. The proposed system was only tested on Chest X-ray (CXR) images and must be validated for different imaging modalities such as CT scans and MRI. The article also emphasizes the importance of verifying that the AI/ML model is learning relevant features for prediction and incorporating explainability in the prediction process [9].The article focuses on building a deep-learning architecture to classify plant diseases based on images. Several state-of-the-art architectures were compared, and DenseNet201 performed the best with a testing accuracy of 95%. The goal was to develop a model with minimal computational resources to provide faster and more accurate results in the healthcare industry, particularly in imaging [10].

An AI-based method for automatically detecting pneumonia from chest X-rays is suggested in the article. ResNet50, a pre-trained deep CNN learner, is used, and the suggested AAPSO is used for feature selection. Extensive experiments show that the suggested method is more reliable than state-of-the-art methods. [11].Using CT scan pictures, the article suggests a twostage methodology for detecting COVID-19 and CAP. For the first step, it employs a finetuned DenseNet architecture, and for the second stage, it uses a fine-tuned EfficientNet design for fine-grained differential classification. The framework efficiently performs binary and precise three-class classification tasks using CT scan images[12].

In order to identify pneumonia from chest X-rays, the article introduces a unique framework that integrates radiomic characteristics and contrastive learning. The model highlights the regions of interest (ROIs) in chest X-rays as a result of the attention process, and experimental results show greater performance compared to baselines. [13].The article highlights how AI can shape the future of pneumonia detection, presenting a model with a high prediction accuracy of 91.04%. It emphasizes that AI is a multiplier for human ingenuity, not a replacement [14].

PneumoXttention achieves a diagnostic accuracy of 92%, outperforming human radiologists with an accuracy of 72%. The article emphasizes the importance of tools like PneumoXttention in reducing the number of deaths caused by pneumonia [15].The article proposes a simple and effective algorithm for localizing lung opacities using a single-shot detector RetinaNet with Se-ResNext101 encoders. Multiple improvements are implemented to increase the accuracy, including global classification output, data augmentations, and ensemble techniques [16].In order to categorise chest X-ray images into three categories— normal, pneumonia, and COVID-19—the article offers a concatenated neural network built on Xception and ResNet50V2. The model achieves good accuracy, sensitivity, and total accuracy on two datasets. [17].

DenseNet201 effectively trains on a smaller set of complex data and performs admirably for classifying pneumonia. According to the report, radiologists can use this computer-aided diagnostic tool to quickly determine the forms of pneumonia and get clinically helpful images. [18]. Several deep learning models show promising performances in pneumonia detection, exceeding 84% average accuracy on the Pneumonia reorganized dataset. These models are highly likely to detect pneumonia cases during the epidemic [19]. The article creates an algorithm that can identify pneumonia from chest X-ray images better than practising radiologists. The algorithm's expansion to detect numerous diseases works better than earlier cutting-edge techniques. The technique seeks to enhance medical imaging expertise accessibility and healthcare delivery [20]. The article's ensemble approach enhances prediction consistency and accuracy [21].

III. SYSTEM DESIGN

Each year, more than 150 million individuals, particularly children under 5, are affected by pneumonia. This global health issue is particularly challenging in regions with limited medical resources and personnel, such as the 57 nations in Africa, where a shortage of 2.3 million doctors and nurses exists. In these areas, the accurate and prompt diagnosis of pneumonia is crucial. It ensures timely access to treatment and can significantly save time and money for those already facing poverty.

Our primary challenge was to develop a robust image classifier that automatically identifies whether a patient is suffering from pneumonia by analyzing chest X-ray images. The algorithm’s accuracy was of utmost importance, as it directly impacts people's lives. We needed to design a neural network that could effectively generalize on unseen data to achieve optimal performance. This required careful consideration and attention to detail in crafting the neural network’s architecture to ensure its ability to classify pneumonia cases accurately.

Methodology:

In this project, we adopted a specific methodology to address the problem. The first step involved treating each image as a 2-dimensional matrix, where the matrix values represented the features and ranged from 0 to 255. The images were standardized to a size of 150x150 pixels. To ensure consistency and compatibility with the chosen neural network architecture, we normalized the matrix of image features to the range of 0 to 1. This step helps improve the model's convergence and stability during training.

To mitigate overfitting, we incorporated a Dropout layer with a rate of 0.1 in the fully connected layers of the neural network. Dropout randomly deactivates a fraction of neurons during each training step, reducing the network's reliance on specific features and improving generalization. We employed the binary cross entropy loss function for binary classification (pneumonia or non-pneumonia). This loss function measures the dissimilarity between the predicted probabilities and the actual labels, effectively measuring the model's performance.

To obtain the probability distribution over the output classes, we utilized a sigmoid layer in the final stage of our neural network. This layer maps the network's outputs to a range of 0 to 1, representing the likelihood of each class. Considering the CNN architecture's ability to extract relevant features, we leveraged CNNs as feature extractors in our methodology. The CNN component of our model learned and extracted discriminative features from the input images, which were then utilized in subsequent layers for classification. By adopting these techniques and utilizing CNNs as feature generators, we aimed to develop an effective model for pneumonia detection using chest X-ray images.

The Dataset:

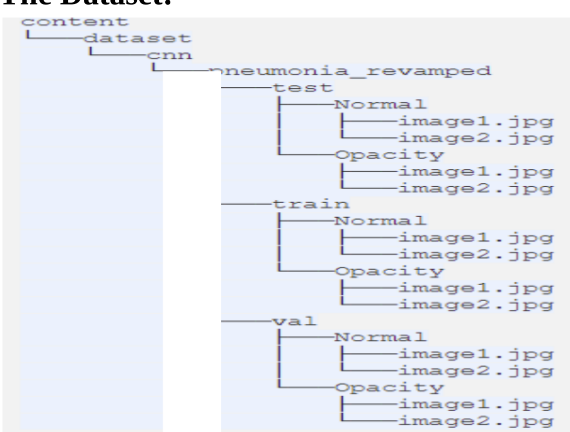

Fig. 4: DIRECTORY TREE

Chest X-Ray images from Kaggle will be the dataset we use for the image classification. It has two categories, Pneumonia and Normal. The data set has subfolders for each image category, Opacity (such as Pneumonia) & Normal, and is divided into 3 folders (train, test, validation).Total Images: 5856

Class/ folder	Train (%)	Test (%)	Validation (%)
Normal	1082	234	267
Pnuemonia	3110	390	773

Fig. 4 the dataset is organized into three folders i.e., Train, Test, and Val. Each folder contains two subfolders which are the respective classes i.e., Normal and Opacity. Example of an image belonging to Pneumonia and Normal shown in Fig. 5:

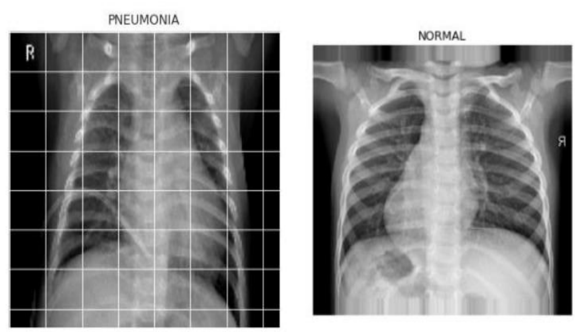

Fig. 5: DATASET EXAMPLE

Data Augmentation:

To address imbalanced data and prevent overfitting, it is often beneficial to expand our dataset artificially. This can be achieved by introducing small transformations to the training data and replicating variations in the process. These techniques, known as data augmentation techniques, involve modifying the array representation of the training data while keeping the labels unchanged. By applying such techniques, we can develop a more robust model. Some commonly used data augmentation techniques include width shifting, horizontal flipping, vertical flipping, etc. These transformations introduce slight variations to the original data, increasing its diversity and reducing the risk of overfitting. By augmenting the dataset with these techniques, we create a more comprehensive training set for our model.

Once the data has undergone augmentation, it is processed through an activation function. If the output of this function exceeds a specified threshold, the corresponding node is activated, and the data is passed to the next layer of the neural network. This activation step helps determine the relevance or importance of the information in the network's decision-making process. By employing data augmentation techniques and incorporating activation functions, we can enhance our model's ability to handle imbalanced data, improve generalization, and make more accurate predictions shown in Fig. 6.

The output is calculated by using the following formula:

$$Z = Bias + W_1X_1 + W_2X_2 + \ldots + W_nX_n$$

$$\text{Output}=f(x)=\begin{cases} 1 \ if \ \sum w1x1 + b \geq 0 \\ 0 \ if \ w1x1 + b < 0 \end{cases}$$

Example of some images from a newly created training set after applying Data Augmentation:

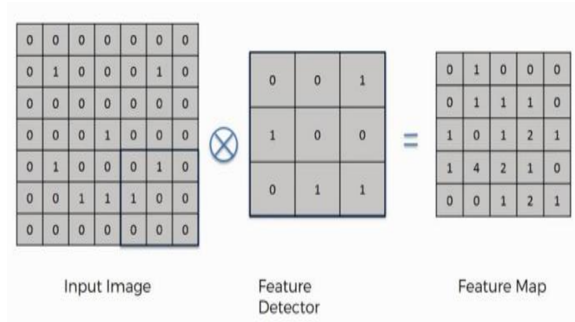

Fig. 6: DATA AUGMENTATION

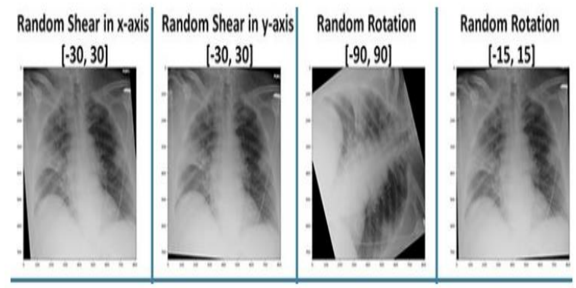

Fig. 7: CONVOLUTION

Convolution:
Here are the three elements that enter into the convolution operation:

- Input image
- Feature detector
- Feature map

We have used 5 Convolution layers in our network with number of filters as 32,64,64,128,256 of size (3,3). It's crucial to separate the feature map from the other two components. Any digit, not just 1s and 0s, can be found in the feature map's cells. We would arrive at the following outcomes after inspecting each pixel in the input image shown in Fig. 7: The feature detector's primary function is to sort through the data in the input image, remove any elements that are not essential to it, and filter out the remaining information.

Generation of Feature Map:

- You overlay the input image with it, starting in the upper-left corner and enclosing it in the borders you see above.

Then, you count the number of cells where the feature detector matches the input image.

- The feature map's top-left cell is then updated with the number of matched cells.
- The feature detector is then moved one cell to the right, and the identical procedure is repeated. This motion is referred to as a stride.
- You can move it to the following row and repeat the process once you have completed the first row.

Pooling:

• The input's height and width are reduced by the pooling (POOL) layer. It facilitates computation reduction and increases the input invariance of feature detectors The "spatial variance" capability of the convolutional neural network is achieved through this technique. Additionally, pooling reduces the number of parameters and the size of the images, which helps to avoid the "overfitting" problem.In this research, we apply the feature maps with the maxpooling of size (2,2) with strides as 1 and 2.

ReLU:

We employed ReLU, or rectified linear activation function, is a piecewise linear function that, if the input is positive, outputs the input directly; otherwise, it outputs zero. The excellent outcomes achieved by this activation function, as illustrated in Figure 8, are simpler to train for.

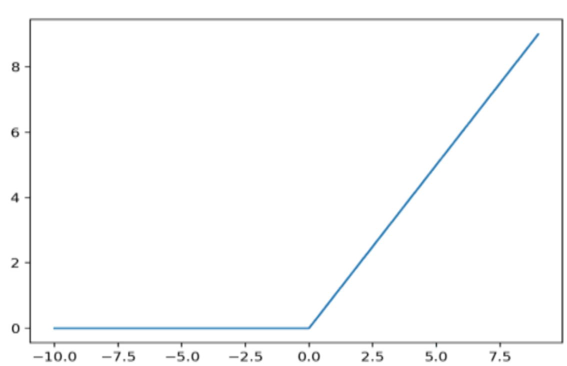

Fig. 8: RELU

As seen in Figure 8 the activation function gives no output until it is less than 0 and activates the node when the output exceeds the threshold value that is 0.

Dropout:

The dropout layer drops neurons randomly based on the given dropout ratio. Our model used a dropout ratio of 0.1 after every convolutional layer.

Example of Dropout:

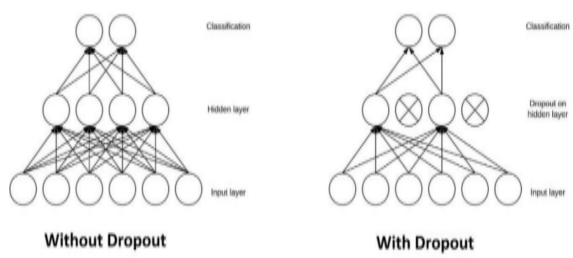

Fig. 9: DROPOUT

Fig. 9 shows that a dropout ratio of 0.5 is used to randomly drop nodes in the hidden layer.

Output layer:

In the output layer, we use sigmoid activation function as our problem involves binary classification, and the activation function gives probability output between 0-1.In our problem, if the output lies below 0.5 then it will belong to normal class and if the output exceeds 0.5 then it belongs to opacity i.e., pneumonia.

Streamlit:

We usedstreamlit framework present in python to represent our project as a web application.The design of the webapp is shown below in Fig. 10.

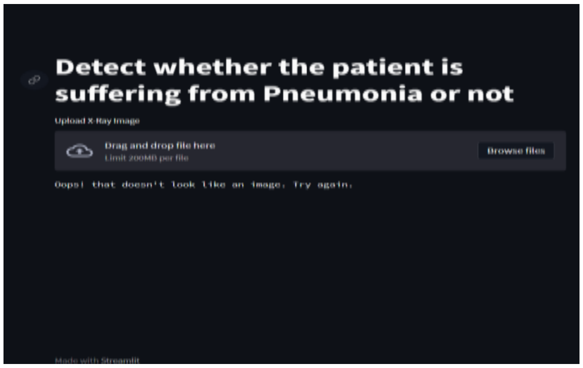

Fig. 10: WEB APPLICATION

Flowchart:

Fig. 11: FLOW CHART OF THE MODEL

IV. RESULT AND DISCUSSION

For the implementation part of our project, we have used two IDE's, which are:

- Google Collab

We have used Google Collab for building the CNN model because image processing requires high computational power and may take hours for training. In Google Colab there is an internal GPU called Nvidia Tesla K80, which makes the training faster.

- Spyder IDE

We have used Spyder IDE inside Anaconda Navigator to deploy the model on streamlit.

After uploading an X-Ray image we'll get the below result

Fig. 12: NORMAL CASE EXAMPLE

Testing the model by uploading an X-Ray image of normal class. Similarly testing the model by uploading an image of Pneumonia class.

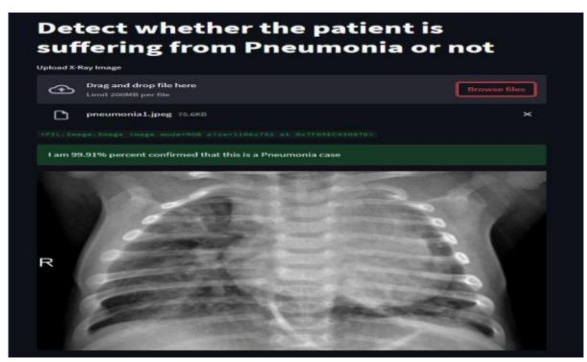

Fig. 13: PNEUMONIA CASE EXAMPLE

Our model successfully predicted that the input image belongs to the pneumonia class.

Visualization:

Plotting model accuracy over training and validation set during training

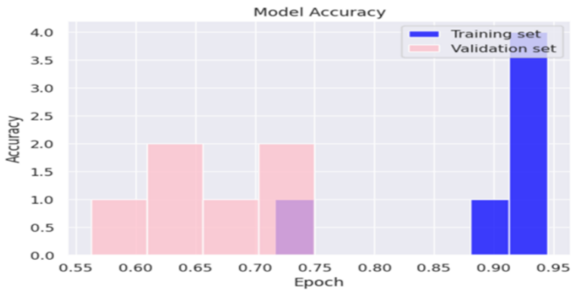

Fig. 14: VISUALISING MODAL ACCURACY DURING TRAINING

Plotting model loss over training and validation set during training

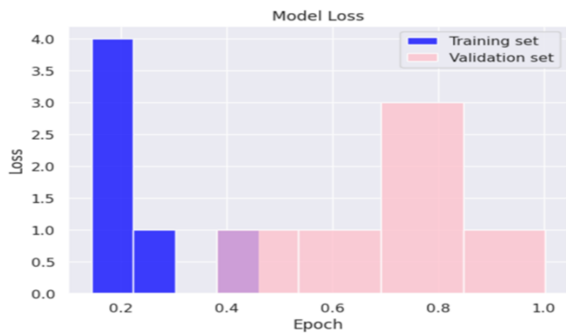

Fig. 15: VISUALISING MODEL LOSS DURING TRAINING

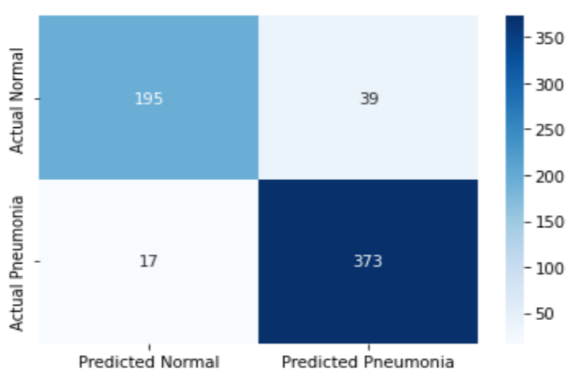

Fig. 16: PLOTTING CONFUSION MATRIX

Summary:

	precision	recall	f1-score	support
NORMAL	0.92	0.83	0.87	234
PNEUMONIA	0.91	0.96	0.93	390
accuracy			0.91	624
macro avg	0.91	0.89	0.90	624
weighted avg	0.91	0.91	0.91	624

Fig. 17: CLASSIFICATION REPORT

V. CONCLUSION

In this project, we conducted experiments to detect pneumonia from X-Ray images. We explored various dimensions for the images and designed different network architectures to optimize our model's performance.To evaluate our model, we utilized a publicly available Chest X-Ray dataset and achieved a good accuracy of 91% on the test set. Our model demonstrated excellent performance in classifying pneumonia images, achieving an accuracy of 95%. However, the accuracy for normal images was relatively lower at 85%, mainly due to the imbalanced nature of the training data. We recognize the need for additional X-Ray image datasets to further improve our model. Early detection of pneumonia is critical for determining appropriate treatment and preserving patients' lives. Therefore, incorporating more diverse and extensive X-Ray datasets will enhance the robustness and accuracy of our model. We conclude by emphasizing that Deep Learning in the medical field is still in its early stages but has already shown remarkable results. New ideas and advancements will emerge as research progresses, making Deep Learning a potential game-changer. Future research can go beyond the simple diagnosis of pneumonia and integrate a wider and more diversified dataset to categories various lung illnesses. Other medical imaging techniques, such MRIs or mammograms, can also be analyzed to make predictions about the risk that cancer will develop in a patient. Overall, by harnessing the power of Deep Learning and leveraging diverse datasets, we can significantly advance medical diagnostics and improve patient outcomes in the future.

VI. REFERENCES

[1]. Hammoudi, Karim et al. “Deep Learning on Chest X-ray Images to Detect and Evaluate Pneumonia Cases at the Era of COVID-19.” Journal of Medical Systems 45 (2020): n. pag.

[2]. Rajpurkar, Pranav et al. “CheXNet: Radiologist-Level Pneumonia Detection on Chest XRays with Deep Learning.” ArXiv abs/1711.05225 (2017): n. pag.

[3]. Team, The DeepRadiology. “Pneumonia Detection in Chest Radiographs.” ArXiv abs/1811.08939 (2018): n. pag.

[4]. Shenoy, Vishal P and Sachin B. Malik. “CovXr: Automated Detection of COVID-19 Pneumonia in Chest X-Rays through Machine Learning.” 2021 IEEE Symposium Series on Computational Intelligence (SSCI) (2021): 1-6.

[5]. Zhang, Jianpeng et al. “COVID-19 Screening on Chest X-ray Images Using Deep Learning based Anomaly Detection.” ArXiv abs/2003.12338 (2020): n. pag.

[6]. Li, Hongjia et al. “Deep Compressed Pneumonia Detection for Low-Power Embedded Devices.” LABELS/HAL-MICCAI/CuRIOUS@MICCAI (2019).

[7]. Asnaoui, Khalid El et al. “Automated Methods for Detection and Classification Pneumonia based on X-Ray Images Using Deep Learning.” ArXiv abs/2003.14363 (2020): n. pag.

[8]. Ali, Wardah et al. “Pneumonia Detection in Chest X-Ray Images: Handling Class Imbalance.” (2023).

[9]. Porag, Al Mohidur Rahman et al. “A Comparison Study of Deep CNN Architecture in Detecting of Pneumonia.” ArXiv abs/2212.14744 (2022): n. pag.

[10]. Pramanik, Rishav et al. “An adaptive and altruistic PSO-based deep feature selection method for Pneumonia detection from Chest X-rays.” Applied Soft Computing 128 (2022): 109464 - 109464.

[11]. Darapaneni, Narayana et al. “Pneumonia Detection in Chest X-Rays using Neural Networks.” ArXiv abs/2204.03618 (2022): n. pag.

[12]. Kulkarni, Viraj et al. “A Classical-Quantum Convolutional Neural Network for Detecting Pneumonia from Chest Radiographs.” ArXiv abs/2202.10452 (2022): n. pag.

[13]. Chaudhary, Shubham et al. “Detecting Covid-19 and Community Acquired Pneumonia Using Chest CT Scan Images with Deep Learning.” ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2021): 8583-8587.

[14]. Han, Yan et al. “Pneumonia Detection on Chest X-Ray Using Radiomic Features And Contrastive Learning.” 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI) (2021): 247-251.

[15]. Dossou, Bonaventure F. P. et al. “An Approach to Intelligent Pneumonia Detection and Integration.” ArXiv abs/2012.03487 (2020): n. pag.

[16]. Singh, Sanskriti. “PneumoXttention: A CNN compensating for Human Fallibility when Detecting Pneumonia through CXR images with Attention.” 2021 12th International Symposium on Image and

Signal Processing and Analysis (ISPA) (2020): 61-66.

[17]. Tam, Leo K. et al. "Weakly supervised one-stage vision and language disease detection using large scale pneumonia and pneumothorax studies." International Conference on Medical Image Computing and Computer-Assisted Intervention (2020).

[18]. Motamed, Saman and Farzad Khalvati. "Inception Augmentation Generative Adversarial Network." ArXiv abs/2006.03622 (2020): n. pag.

[19]. Gabruseva, Tatiana et al. "Deep Learning for Automatic Pneumonia Detection." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2020): 1436-1443.

[20]. Rahimzadeh, Mohammad and Abolfazl Attar. "A New Modified Deep Convolutional Neural Network for Detecting COVID-19 from X-ray Images." ArXiv abs/2004.08052 (2020): n. pag.

[21]. Rahman, Tawsifur et al. "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray." ArXiv abs/2004.06578 (2020)