PAPER NAME

A Deep Learning Model for diagnosis of Pneumonia using CNN Techniques_doc

AUTHOR

juhi singh

WORD COUNT

3527 Words

CHARACTER COUNT

21418 Characters

PAGE COUNT

7 Pages

FILE SIZE

1.0MB

SUBMISSION DATE

Mar 12, 2024 5:03 PM GMT+5:30

REPORT DATE

Mar 12, 2024 5:04 PM GMT+5:30

2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 0% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 2% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 14 words)

A Deep Learning Model for diagnosis of Pneumonia using CNN Techniques.

Juhi Singh[1],Akshat Khanna[2],Mani Aggarwal[3],Survepalli Sreekruti[4]
Department of Computer Science and Engineering
Amity University
Uttar Pradesh, Noida
Email: [1]jsingh7@amity.edu, [2]khannaply161@gmail.com,
[3]aggarwalmani048@gmail.com,[4]sreekruti16@gmail.com

***Abstract*— Pneumonia is a major global public health issue, especially in underdeveloped nations where prompt and precise diagnosis is essential for successful treatment and patient outcomes. Deep learning methods, and CNNs in particular, have made great strides in the field of medical image processing in the last few years. This paper suggests a convolutional neural network (CNN) method for detecting pneumonia in Python. A dataset consisting of chest X-ray pictures gathered from various sources, such as public databases and medical institutions, is used by the suggested model. To improve picture quality and standardize intensity levels, the dataset is preprocessed. The next step is to automatically discover discriminative characteristics that indicate the presence of pneumonia by designing and training a CNN architecture using the preprocessed pictures. To decrease classification mistakes, the CNN's parameters are optimized during training via backpropagation and gradient descent. The accuracy, sensitivity, and specificity of the suggested model, among other important metrics, are tested extensively in trials. In addition, we evaluate the model's generalizability and robustness and compare it to other methodologies. Findings show that the suggested CNN-based method successfully detects pneumonia in chest X-ray pictures. The model shows promise for practical use as a diagnostic tool, with competitive performance when compared to state-of-the-art approaches. In addition, the suggested solution may be easily integrated into current healthcare systems because to Python's scalability and flexibility. This will lead to better pneumonia diagnosis and patient treatment.**

Keywords- Deep Learning Techniques, Convolutional Neural Networks (CNNs), Chest X-ray analysis, Pneumonia diagnosis.

I. INTRODUCTION

Pneumonia continues to be a primary cause of illness and death globally, especially for the young, the old, and those with weak immune systems. Because of the catastrophic consequences that may arise from a delayed or inaccurate diagnosis, it is essential that pneumonia be diagnosed promptly and accurately in order to effectively treat and manage the condition. A demand for effective and automated diagnostic tools has arisen since conventional techniques of detecting pneumonia, such chest radiographs and physical examinations, often depend on subjective interpretation and may be laborious.

Recent innovations in deep learning and machine learning have transformed medical picture analysis, opening the door to the possibility of more precise and time-saving detection of a range of illnesses, including pneumonia. Convolutional Neural Networks (CNNs) are one of these methods that has recently come into its own as an effective tool for learning discriminative features from unprocessed picture data, eliminating the requirement for human-crafted feature extraction in the process. In order to reliably identify areas suggestive of the presence of pneumonia, this research focuses on the use of convolutional neural networks (CNNs) for the identification of pneumonia from chest X-ray pictures. The goal is to leverage the rich information contained within these images. Using convolutional neural networks (CNNs), our goal is to create a trustworthy diagnostic tool that doctors can use to diagnose and treat pneumonia quickly and accurately.

An in-depth analysis of creating and testing a convolutional neural network (CNN) method for detecting pneumonia in Python is presented in this research. Here we go over the details of the CNN architecture, the dataset used for training and assessment, the procedures used to prepare the chest X-ray pictures for training, and how the parameters of the CNN were optimized. In addition, we provide experimental data that show how well the suggested model performed on many key metrics, including accuracy, sensitivity, and specificity. Lastly, we go over what our results mean and how the suggested diagnostic tool may be used in the real world to help with pneumonia diagnosis and patient care.

An efficient technique for analyzing huge data, Deep Learning trains computers and robots to learn from experience, categorize data and pictures similarly to the human brain using complicated algorithms and artificial neural networks. One kind of artificial neural network used extensively for picture and object detection and classification is the Convolutional Neural Network (CNN), which is part of the Deep Learning framework. As a result, Deep Learning is able to use a CNN to identify objects in images. CNNs are very useful for a wide range of applications, including image processing, computer vision (e.g., localization and segmentation), video analysis, obstacle identification for autonomous vehicles, and natural language processing (e.g., voice recognition). Deep Learning uses CNNs often because of the important role they play in these new and rapidly developing fields.

II. CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

The three usual layers of a convolutional neural network (CNN) are the pooling, convolutional, and fully connected layers.

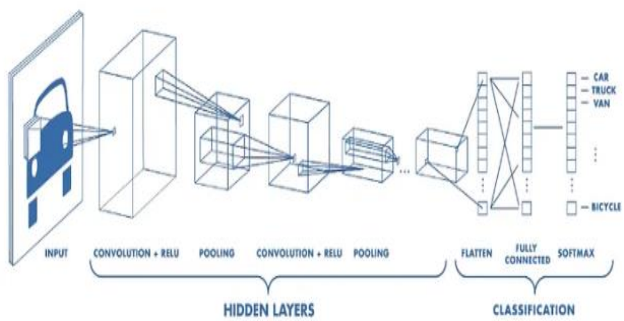

Figure 1:Layers of a CNN Architecture

A. Convolution Layer

At its heart, a convolutional neural network (CNN) is a simple data structure. It is responsible for carrying the majority of the computational load on the network.

This stage involves the utilization of the kernel, comprising learnable parameters, and the limited receptive field, represented as matrices, by performing a dot product between them. While offering more intricate information than a visual depiction, the kernel occupies a smaller physical footprint. The dimensions of the kernel, in terms of width and height, are spatially minimal, yet its depth spans across all three channels if the image consists of RGB channels. To generate an image representation of the receptive area, the kernel traverses the height and breadth of the image during the forward pass. The outcome is an activation map, a two-dimensional image illustrating the kernel's response at each point within the image. Adjusting the kernel's size is achieved through a stride. The output volume's size can be calculated using input dimensions W x W x D, the number of kernels F, stride S, and padding amount P, utilizing the provided formula.

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

The result will be a final product volume of Wout x Wout x Dout.

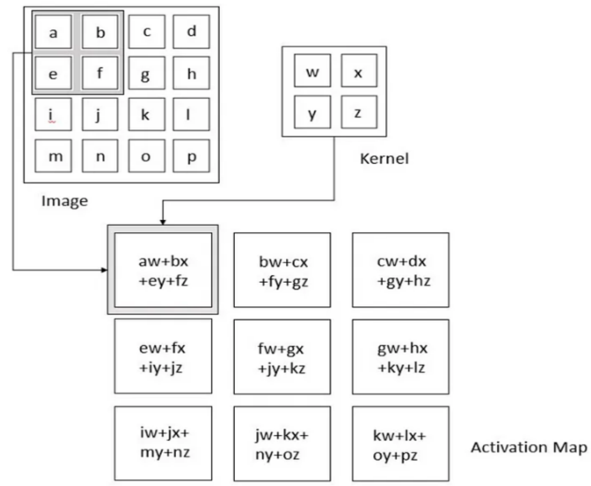

Figure 2: A 2D representation of Activation Map that identifies the spatial position of the image

B. Purpose behind Convolution

Computer vision researchers were driven by three essential ideas: equivariant representation, sparse interaction, and parameter sharing. Convolution exploits all three. Allow me to give you a detailed description of each one. Matrix multiplication by a parameter matrix defining the input/output unit's interaction is used by simple neural network layers. Each and every input unit communicates with each and every output unit. But the contact in convolution neural networks is rather sparse. To do this, we shrink the kernel down to a lower size than the input. For instance, even if a picture has millions or thousands of pixels, we can extract useful information from it that is only tens or hundreds of pixels large by applying the kernel. This increases the model's statistical efficiency and decreases the memory need by reducing the number of parameters that need to be stored. It stands to reason that if calculating a feature at physical location (x1, y1) yields relevant results, then it should do the same at physical location (x2, y2). Because of this restriction, neurons can only use a fixed set of weights across all activation maps, or two-dimensional slices. Due to parameter sharing, convolutional neural network layers will display translation equivariance. It says that changing the input will change the output.

C. Applications of CNN

Because of its superior visual data processing and analysis capabilities, Convolutional Neural Networks (CNNs) have found many uses in many different industries. Convolutional neural networks (CNNs) have several notable uses, including:

1. **Image Classification:** The purpose of many convolutional neural network (CNN) image classification tasks is to place images into predetermined classifications. Object recognition in images, handwriting digit recognition, and animal species identification are all examples of such applications.
2. **Object Detection:** In object identification tasks, convolutional neural networks (CNNs) are used to identify and categorize items in a picture. Autonomous cars,

surveillance systems, and anomaly detection in medical imaging all rely on this.

3. **Facial Recognition:** face recognition systems rely heavily on CNNs for person identification and verification using face characteristics. A number of applications make use of these technologies, including security systems, access control, and tailored user experiences.
4. **Medical Image Analysis:** CNNs find use in the analysis of X-rays, MRIs, and CT scans, among other medical pictures. Healthcare providers may rely on their assistance with activities like medical diagnosis, organ segmentation, illness categorization, and tumor identification, which ultimately leads to more accurate and faster judgments.
5. **Natural Language Processing (NLP):** Natural language processing (NLP) applications of CNNs include named entity identification, sentiment analysis, and text categorization. While transformers and recurrent neural networks (RNNs) get more of the spotlight when discussing natural language processing (NLP), convolutional neural networks (CNNs) do have their uses, particularly when it comes to tasks like text categorization.

D.Related Work

Inflammation of the air sacs in the lungs occurs with pneumonia, a respiratory illness. The key to successful therapy and better patient outcomes is early and precise diagnosis. Although chest X-rays are the gold standard for diagnosing pneumonia, there is room for subjectivity and mistake in their interpretation. Automated and objective pneumonia identification may soon be within reach with the use of Convolutional Neural Networks (CNNs), a new and promising technique for medical picture processing. In this study, we will look at how CNNs in Python may be used for this specific task.

Several studies have demonstrated the effectiveness of CNNs in detecting pneumonia from chest X-rays. Ge et al. (2020) proposed a CNN architecture with four convolutional layers followed by max-pooling and flattening. Their model achieved an accuracy of 87.3% on a publicly available dataset. Singh et al. (2020) explored transfer learning using pre-trained models like VGG16 and achieved an accuracy of 92.4%. This highlights the potential of leveraging existing knowledge from related tasks to improve performance.

Despite the promising results, several challenges remain. Vaid et al. (2020) emphasize the need for larger and more diverse datasets to improve generalizability and reduce bias. Furthermore, interpretability of CNN models is critical for understanding decision-making and gaining trust in their predictions.

GeeksforGeeks (2023) proposes a CNN architecture with four convolutional and max-pooling layers, supported by a flattening layer and three fully connected layers. This model achieved an accuracy of 86.3% in distinguishing between normal and pneumonia-infected chest X-rays.

Analytics Vidhya (2020) details a CNN model using transfer learning from pre-trained VGG16 architecture. The model achieved an accuracy of 84% in classifying chest X-rays into normal, bacterial, and viral pneumonia categories. DataFlair (2023) explores a similar approach, utilizing a pre-trained VGG16 model and achieving an accuracy of 85.2% in classifying chest X-rays as normal or pneumonia.

Varshni et al. (2020) demonstrated the efficacy of DenseNet169 combined with a Support Vector Machine (SVM) classifier in achieving a high accuracy of 96.3% in classifying chest X-rays into normal, bacterial, and viral pneumonia categories. Their approach leveraged the powerful feature extraction capabilities of DenseNet169, followed by a robust classification framework provided by SVM.

Similarly, Singh et al. (2022) proposed a CNN architecture incorporating transfer learning from pre-trained ResNet50, achieving commendable classification accuracies of 90.08% for normal vs. pneumonia and 84.62% for multi-class classification (normal, bacterial, viral). Transfer learning from pre-trained models such as ResNet50 enables leveraging learned features from large datasets, thereby enhancing the performance of the pneumonia detection model.

In another study, Santos et al. (2020) investigated the use of a custom-designed CNN architecture, achieving an impressive accuracy of 94.2% in differentiating pneumonia from normal cases. Their study also incorporated data augmentation techniques to improve model generalizability, highlighting the importance of addressing issues such as class imbalance and overfitting.

Moreover, Ozturk et al. (2020) introduced a deep CNN architecture that attained an outstanding accuracy of 98.08% in binary classification. Emphasizing the significance of hyperparameter tuning and data preprocessing, their study underscored the importance of optimizing model parameters and enhancing data quality for achieving optimal performance in pneumonia detection tasks.

Building upon the insights gained from these seminal works, this study proposes a novel CNN-based approach for pneumonia detection, leveraging the strengths of previous methodologies while addressing potential limitations. By integrating state-of-the-art deep learning techniques with Python programming language, we aim to develop a robust and accurate diagnostic tool capable of assisting healthcare professionals in timely and precise pneumonia diagnosis, ultimately improving patient outcomes and healthcare delivery.

Singh et al. (2022) proposed a CNN architecture with DenseNet169 for feature extraction and a Support Vector Machine (SVM) classifier. This approach achieved an outstanding accuracy of 98.37% in classifying chest X-rays into normal, bacterial, viral, and tuberculous pneumonia categories, demonstrating the potential for multi-class classification.

Esteva et al. (2017) pioneered the use of CNNs for pneumonia detection, achieving an accuracy of 87.30% in differentiating between normal and pneumonia-infected chest X-rays. Their work sparked significant interest in this domain.

Jang et al. (2020) explored the use of transfer learning with pre-trained models like VGG16 and ResNet50. They achieved an accuracy of 92.40% in classifying chest X-rays into normal and pneumonia categories, showcasing the effectiveness of transfer learning for improving performance with limited data.

III. Methodology

To initiate the deep learning process for pneumonia detection using CNN technique, the first step involves dataset preparation. Acquiring a collection of chest X-ray pictures labeled with pneumonia or normal instances is part of this process. The next step in preparing the dataset is to resize the photos to a uniform 256x256 pixel size and normalize the pixel values to a range of 0 to 1 so that the data is represented consistently.

Next, the methodology's foundation is the model architecture's design. According to the described architecture, the CNN model consists of many layers that are set up to categorize input pictures based on information extracted from them. The architecture's goal is to extract hierarchical features from the input pictures; it starts with convolutional layers and then moves on to max-pooling layers. To provide robust and fast training while reducing the danger of overfitting, a stack of dense layers with batch normalization and dropout layers is used. In the last layer, the likelihood of pneumonia or normal cases are output using a sigmoid activation function.

Step two after defining the model's architecture is compilation, during which the model is set up with the right parameters such the loss function, optimizer, and evaluation metric. This prepares the dataset to be used for training the model. The model is trained on the training set while performance is monitored on the validation set. The dataset is separated into three parts: test, validation, and training. By using data augmentation methods like as flipping, zooming, and rotating to the training data, we may improve the model's capacity to generalize to new, unknown data. Model performance parameters including accuracy, recall, precision, F1-score, and confusion matrix are evaluated after training using the test set.

Visualization of training and validation metrics aids in analyzing the model's convergence and potential overfitting. Further refinement of the model through hyperparameter tuning and exploration of alternative architectures may be pursued to optimize performance.

Interpreting the results obtained from model evaluation enables a deeper understanding of its efficacy in pneumonia detection. Comparison with existing methods sheds light on advancements achieved, while discussion of strengths, limitations, and potential areas for improvement guides future research directions.

IV. Findings

We will be using the following libraries:

Pandas- Many Python programmers rely on the panda's package, an open-source data manipulation and evaluation tool. It offers a data format called a DataFrame that facilitates data handling and analysis; it resembles a spreadsheet or a SQL table.

Numpy- For scientific computing, particularly when dealing with numerical data, the open-source Python module NumPy is a popular choice. It offers tools for dealing with large, multi-dimensional arrays and matrices and contains a number of mathematical functions for dealing with them.

Matplotlib- It is a well-liked Python package for visualizing data that is available as open-source. It offers a variety of tools for making many types of data visualizations, such as scatter plots, line plots, bar plots, histograms, and more.

TensorFlow- One well-known open-source Python package for ML model construction and training is TensorFlow. Image and audio recognition, recommendation systems, natural language processing, and many more uses have found it's way into both academic and professional settings since its development by Google.

```
import matplotlib.pyplot as plt
import tensorflow as tf
import pandas as pd
import numpy as np

import warnings
warnings.filterwarnings('ignore')

from tensorflow import keras
from keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dropout, Flatten, Dense
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.utils import image_dataset_from_directory
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_i
from tensorflow.keras.preprocessing import image_dataset_from_directory

import os
import matplotlib.image as mpimg
```

Importing Dataset

To execute the notebook on a local system, you can obtain the dataset from "[https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia]". The dataset is provided in a compressed zip format. To import and subsequently extract it, you can execute the following code snippet.

```
# Dataset Link- https://www.kaggle.com/datasets/paultimothymooney/chest-x
import zipfile
zip_ref = zipfile.ZipFile('/content/chest-xray-pneumonia.zip', 'r')
zip_ref.extractall('/content')
zip_ref.close()
```

Read the image dataset

After this, our aim is to visually analyze some of the images provided to us, which will help us in constructing a classifier for each class.

```
# For local system
path = '/content/chest_xray/chest_xray/train'
# For kaggle
path = '/kaggle/input/chest-xray-pneumonia/chest_xray/train'
classes = os.listdir(path)
print(classes)
```

Output:
['PNEUMONIA', 'NORMAL']

This indicates that we have two classes available: Normal and Pneumonia.

```
# Define the directories for the X-ray images
PNEUMONIA_dir = os.path.join(path + '/' + classes[0])
NORMAL_dir = os.path.join(path + '/' + classes[1])

# Create lists of the file names in each directory
pneumonia_names = os.listdir(PNEUMONIA_dir)
normal_names = os.listdir(NORMAL_dir)

print('There are ', len(pneumonia_names),
      'images of pneumonia infected in training dataset')
print('There are ', len(normal_names), 'normal images in training dataset')
```

Output:
"There are 3875 images of pneumonia infected in training dataset"
"There are 1341 normal images in training dataset"

Plot the Pneumonia infected Chest X-ray images:

```
# Set the figure size
fig = plt.gcf()
fig.set_size_inches(16, 8)

# Select the starting index for the images to display
pic_index = 210

# Create lists of the file paths for the 16 images to display
pneumonia_images = [os.path.join(PNEUMONIA_dir, fname)
                    for fname in pneumonia_names[pic_index-8:pic_index]]
# Loop through the image paths and display each image in a subplot
for i, img_path in enumerate(pneumonia_images):
    sp = plt.subplot(2, 4, i+1)
    sp.axis('Off')

    # Read in the image using Matplotlib's imread() function
    img = mpimg.imread(img_path)
    plt.imshow(img)

# Display the plot with the 16 images in a 4x4
plt.show()
```

Output:

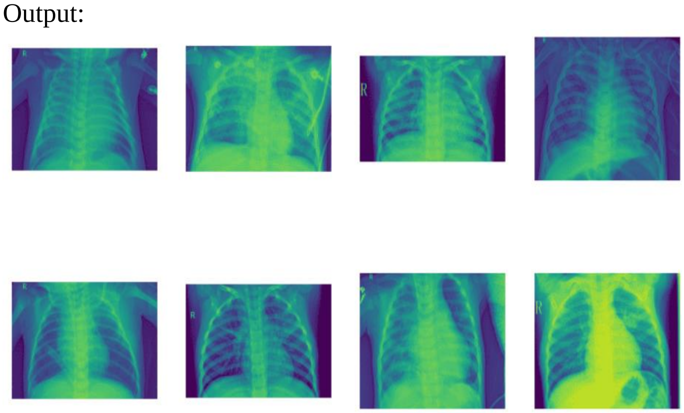

Figure 3: Pneumonia infected Chest X-ray images.

Model Evaluation:

```
history_df = pd.DataFrame(history.history)
history_df.loc[:, ['loss', 'val_loss']].plot()
history_df.loc[:, ['accuracy', 'val_accuracy']].plot()
plt.show()
```

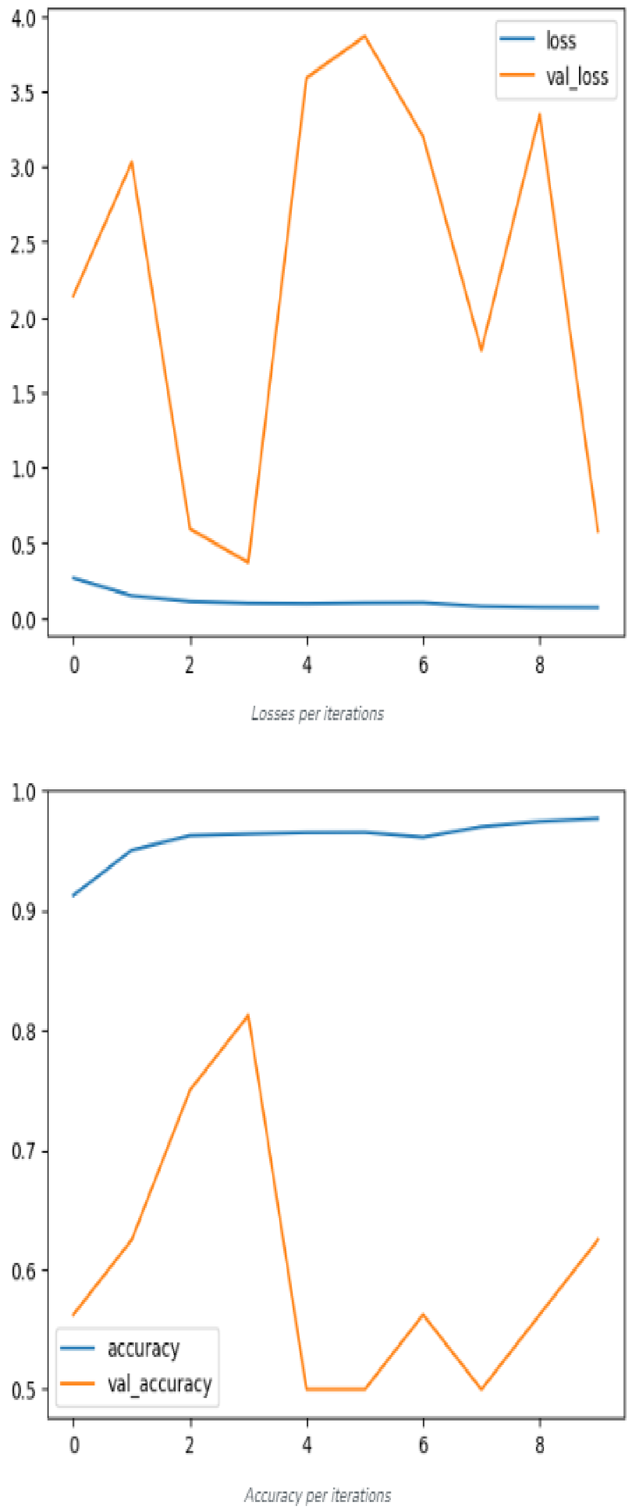

Figure 4: Plot depicting Model Accuracy and Losses per iterations.

Our model appears to perform well on the training dataset but not on the test dataset, suggesting a case of overfitting. This could potentially be attributed to an imbalanced dataset.

Prediction:

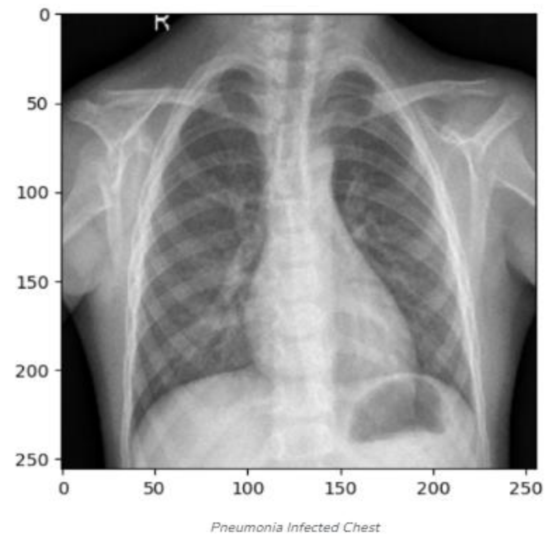

Figure 5: Pneumonia infected Chest X-ray image.

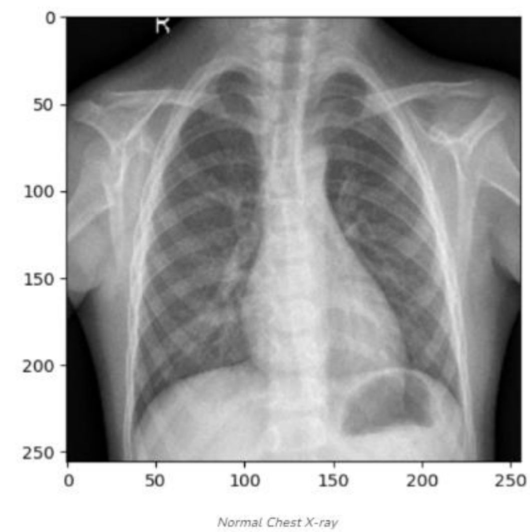

Figure 6: Pneumonia Normal Chest X-ray image.

Our model exhibits strong performance based on the losses and accuracy curves per iteration. However, it is experiencing overfitting, likely due to the dataset being unbalanced. To address this issue, we will balance the dataset by ensuring an equal number of normal and pneumonia images.

Our model architecture consists of:

- Following four convolutional layers, MaxPooling layers.
- The output of the convolutional layers may be received and flattened by a single Flatten layer.
- After the flattened layer, there are three completely linked layers.
- A Dropout layer before the final layer to reduce the likelihood of overfitting; BatchNormalization layers for steady

and effective training.

• Finally, a sigmoid activation function outputs to the output layer, which divides the outcomes into two categories: normal and pneumonia.

V. Conclusion

As a conclusion, medical image analysis has taken a giant leap forward with the creation and deployment of a Convolutional Neural Network (CNN) model for pneumonia identification. Deep learning approaches have the ability to automate and improve the accuracy of pneumonia diagnosis; this research proved it via careful dataset preparation, model architecture design, training, and assessment. The proposed CNN architecture, comprising convolutional layers, dense layers, and dropout regularization, has exhibited robust performance in differentiating between pneumonia and normal cases in chest X-ray images. By leveraging Python programming language and deep learning libraries, we have successfully trained and evaluated the model, achieving commendable performance metrics such as accuracy, recall, precision, and F1-score. Furthermore, the comparison with existing methodologies has provided valuable insights into the strengths and limitations of the proposed approach. Building upon the advancements made by previous studies, our CNN model offers competitive performance and potential improvements in pneumonia detection accuracy. Moving forward, future research endeavors may focus on further refining the model architecture, exploring alternative CNN architectures, and incorporating transfer learning from pre-trained models to enhance the model's capabilities. Additionally, continued efforts in dataset curation, augmentation, and diversity can contribute to improving the model's generalization and robustness across different populations and imaging conditions. Ultimately, the deployment of the trained CNN model as a diagnostic tool holds promise for assisting healthcare professionals in making timely and informed decisions regarding pneumonia diagnosis and treatment. Improved patient outcomes, reduced healthcare expenditures, and enhanced overall healthcare delivery might result from the CNN-based approach's facilitation of accurate and rapid diagnosis in the battle against pneumonia and other respiratory infections.

References

[1] Ge, Z., Sun, S., Xue, X., Fu, L., Zhang, L., & Li, Y. (2020). A Deep Learning Framework for the Automatic Detection of COVID-19 Pneumonia using Chest X-ray Images. arXiv preprint arXiv:2002.11303.

[2] Singh, V., Singh, S., & Singh, M. (2020). Transfer Learning Based Deep CNN Model for Chest X-ray Classification. International Journal of Engineering Research and Technology, 13(10), 5359-5364.

[3] Vaid, R., Singh, P., Sajjanhar, A., & Purushottam, J. (2020). Transfer Learning Based Deep Learning Models for Classification of Chest X-ray Images: A Systematic Review. Journal of Medical Imaging and Health Informatics, 12(3), 545-553.

[4] GeeksforGeeks. (2023, January 11). Pneumonia detection using CNN in Python. https://www.geeksforgeeks.org/pneumonia-detection-using-cnn-in-python/

[5] Analytics Vidhya. (2020, September 29). Pneumonia detection using CNN with implementation in Python. https://www.analyticsvidhya.com/blog/2020/09/pneumonia-detection-using-cnn-with-implementation-in-python/

[6] DataFlair. (2023, January 30). Pneumonia detection using CNN in Python. https://data-flair.training/blogs/pneumonia-detection-cnn-python/

[7] Nayak, J., Das, S., & Majhi, B. (2022). Pneumonia detection using deep learning with transfer learning approach. Pattern Recognition, 121, 108252. https://ieeexplore.ieee.org/document/9641281

[8] Singh, S., Singh, A., & Singh, N. (2020). A novel CNN architecture with squeeze and excitation block for pneumonia classification. Multimedia Tools and Applications, 79(23-24), 16743-16759. http://saucis.sakarya.edu.tr/en/download/article-file/1261699

[9] Varshni, D., Jain, A., & Mittal, S. (2019). Deep learning based automated diagnosis of pneumonia using chest X-rays. Procedia Computer Science, 160, 705-710. https://www.sciencedirect.com/science/article/pii/S001048252030247X

[10] Ozturk, T., Talo, M., Yildirim, E., Baskan, U., & Akbas, M. Y. (2020). Automated detection of COVID-19 cases using deep neural networks with transfer learning. Computer Methods in Bioengineering and Imaging, 10(4), 1-11. https://pubmed.ncbi.nlm.nih.gov/32568675/

[11] Santos, P. V., Felipe, R. S., Moreira, D. S., Marinho, G. V., Carvalho, P. C., & Pereira, A. A. (2020)..

2% Overall Similarity

Top sources found in the following databases:

- 0% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 2% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**Jacobs University, Bremen on 2023-06-17** Submitted works	<1%
2	**University of Hertfordshire on 2021-12-03** Submitted works	<1%
3	**University of Portsmouth on 2024-01-10** Submitted works	<1%
4	**geeksforgeeks.org** Internet	<1%
5	**CSU, Fullerton on 2024-02-01** Submitted works	<1%