CHAPTER 3

Methodology

3.1 Introduction

Pneumonia detection through chest X-rays has seen significant advancements with the integration of machine learning (ML) techniques. This methodology outlines the steps involved in developing a web application that optimizes ML models for pneumonia detection using chest X-ray images. Throughout history, infectious illnesses have posed a significant threat to human health. Among infectious diseases, pneumonia ranks first(Akter & Shamsuzzaman, 2015). The definition of pneumonia is inflammation of the lung brought on by live organisms like bacteria and viruses acting on the tiny air sacs(McLuckie, 2009). An estimated 4 million individuals worldwide lose their lives to pneumonia each year, accounting for around 7% of all cases. In cases like these, early diagnosis is crucial(Summah & Qu, 2009). Common signs and symptoms include coughing, shortness of breath, and chest pain. Chest X-ray pictures and sputum culture are diagnostic tools. The goal of artificial intelligence, which has gained popularity recently, is to comprehend human intelligence in order to improve the use of computers. Experts in the early identification of diseases stand to benefit from the application of machine learning and deep learning, two subfields of artificial intelligence that are employed in many other sectors. Researchers have created CNN-based deep networks in diverse ways, and these networks have been used, particularly in computer vision, for classification, segmentation, object identification, and localization. In addition to computer vision issues, CNNs are utilized in the diagnosis of Alzheimer's disease, the categorization of skin lesions, the detection of breast cancer, the segmentation of brain tumors, and other applications. It has been applied to the resolution of medical issues with observed success. Pneumonia can be diagnosed using the tests listed below: Lung CT, lung ultrasonography, lung needle biopsy, and chest X-ray MRI of the chest (D. Berliner et al., 2016). As of right now, one of the best ways to identify pneumonia is using a chest X-ray (He et al., 2016). Because CT imaging usually takes much longer than X-ray imaging and because there aren't enough high-quality CT scanners in many developing countries, X-ray imaging is favored. On the other hand, the most accessible diagnostic imaging method is X-rays. Examining the literature reveals a large number of research, particularly in the area of pneumonia detection. Using Streamlit, this web application offers an interactive platform for fine-tuning machine learning models designed to identify pneumonia in chest X-rays. Researchers and healthcare professionals can now easily access advanced machine learning techniques with the help of this application, which streamlines the processes of model selection, hyperparameter tweaking, and performance evaluation.

3.2 Dataset

The corresponding clinicians label X-rays in datasets. The technique section contains the dimensions of the data set's tagged view counts. This work makes use of the Chest X-ray Images (Pneumonia) dataset, which is accessible to the public. Two subfolders holding normal chest X-ray images and one exhibiting pneumonia comprise the collected dataset. There are 4273 cases of pneumonia and 1583 normal cases out of the total 5856 data points. Pictures of pneumonia and typical classifications taken from the database are shown in Figure 1.

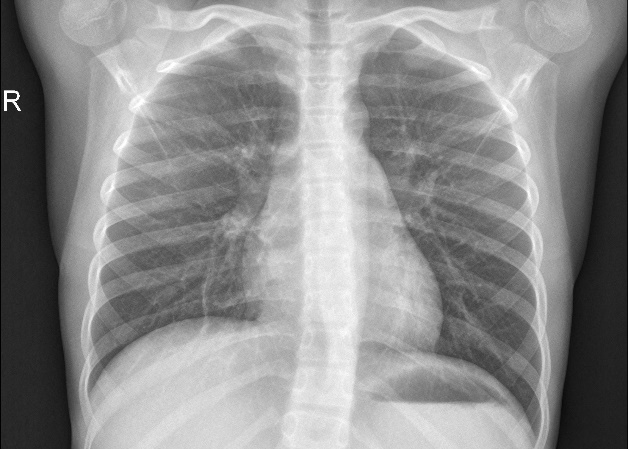


FIG. 3.1

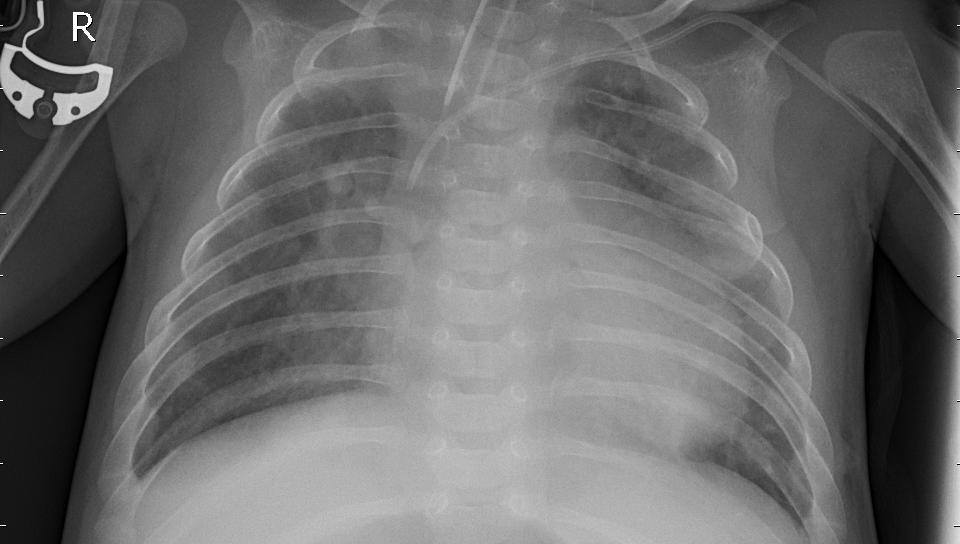


FIG. 3.2

A dataset of thoracic X-ray pictures that was made available on the Kaggle platform is used for this investigation. To be used in artificial intelligence research, these photos are first sorted by three experts based on their expert ratings; low-quality or unintelligible images are then removed. Figure 3.1 and 3.2 display sample X-ray pictures of pneumonia illness.

3.3 Deep Learning

Artificial neural networks (ANNs) with multiple layers are referred to as deep learning. Because deep learning can handle massive volumes of data, it has emerged as one of the most potent tools in the literature (Albawi et al., 2017a). Among deep learning networks, convolutional neural networks (CNNs) are among the most well-known. Considerable progress has been made in areas of research such pattern recognition, image processing, and voice recognition with CNN. CNN is a deep neural network that uses the convolution function on at least one of its layers to extract features. CNN manages the image's properties regardless of location, which is one of the reasons it's widely utilized in studies.

3.3.1 Convolutional Neural Network

To compare the study's findings, a few of the most well-known CNN models that have been utilized in comparable investigations are covered in this part.

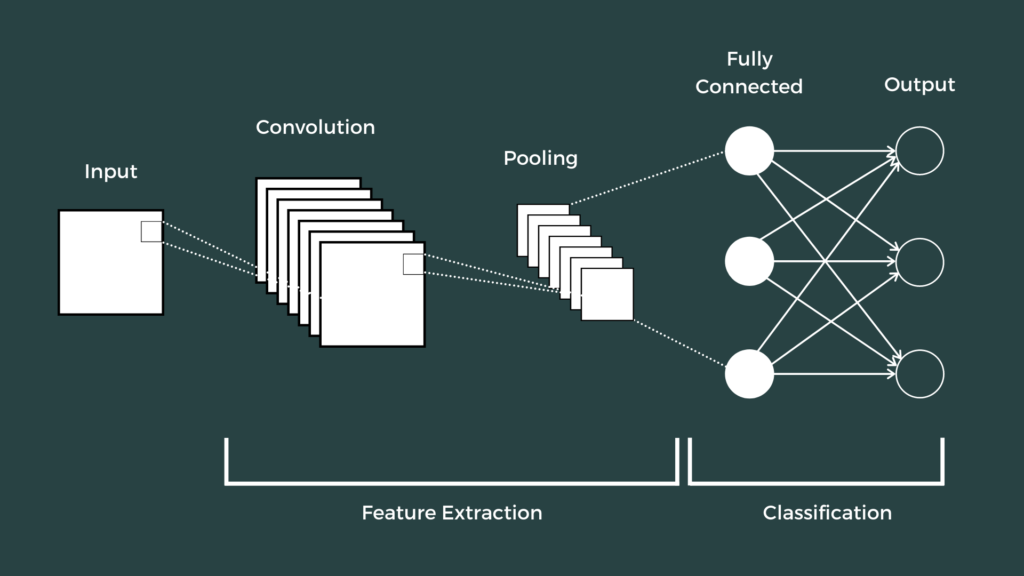


FIG 3.3

3.3.2 Convolutional Neural Network Layers

a. Input Layer

The input layer is the CNN's initial layer. The network receives the input data in its raw form at the input layer. The pixels in the input are translated into numerical data so that the computer can interpret the input (picture). It matters how big a picture is in the input layer of a network that is being used. Large input sizes may slow down the network since they require more memory and have more parameters, but they may also produce better results because more properties can be extracted. Furthermore, when the number of parameters decreases with small input sizes, the network may produce faster results; but, as feature extraction decreases, the network's performance may also decline (Özkan & Ülker, 2017).

b. Convolution Layer

As a result of its role in extracting features from the image that is fed into the network, the convolution layer is sometimes referred to as the feature extraction layer. Using different filters (kernel) in the convolution layer, the feature map of the picture is produced [3]. You can use filters that are 2x2, 3x3, or 5x5 in different sizes. Applying the convolution procedure to the image is done with these filters. In the process of training CNNs to identify which parts of the data are crucial for identifying the network's characteristics, the coefficients of the filters vary with each learning iteration in the training set (Özkan & Ülker, 2017).

c. The Pooling Layer

In CNNs, the pooling or pooling layer is typically utilized to lower the computational load and parameter count. Without changing the depth, it decreases the input size for the subsequent convolution layer. In other words, after pooling, the input data's width and height values decrease, but the depth—that is, the number of channels—remains unchanged. Broadly speaking, two distinct pooling levels can be identified.

d. Activation Layer

The convolution layer is followed by the activation layer. This non-linear layer is typically employed to restrict and modify the output generated (Albawi et al., 2017b). This layer's functions include reducing the negative values in the linear network that is created after the convolution layer to zero, converting the network into a non-linear form, and accelerating learning. Another name for it is the activation function. ReLU is the most often used activation function, however there are several others, including tanh, sigmoid, and ELU.

e. The Fully Connected Layer

The full link layer in a CNN comes after the layers of pooling, ReLU, and convolution. Flattening is the process of converting the feature map for the entire link layer into a 1D feature vector (Özkan & Ülker, 2017). This layer, which is depicted in Figure 3.4, is known as the full link layer because it is linked to every part of the preceding layer. Several categories are created for photographs by using the entire link layer.

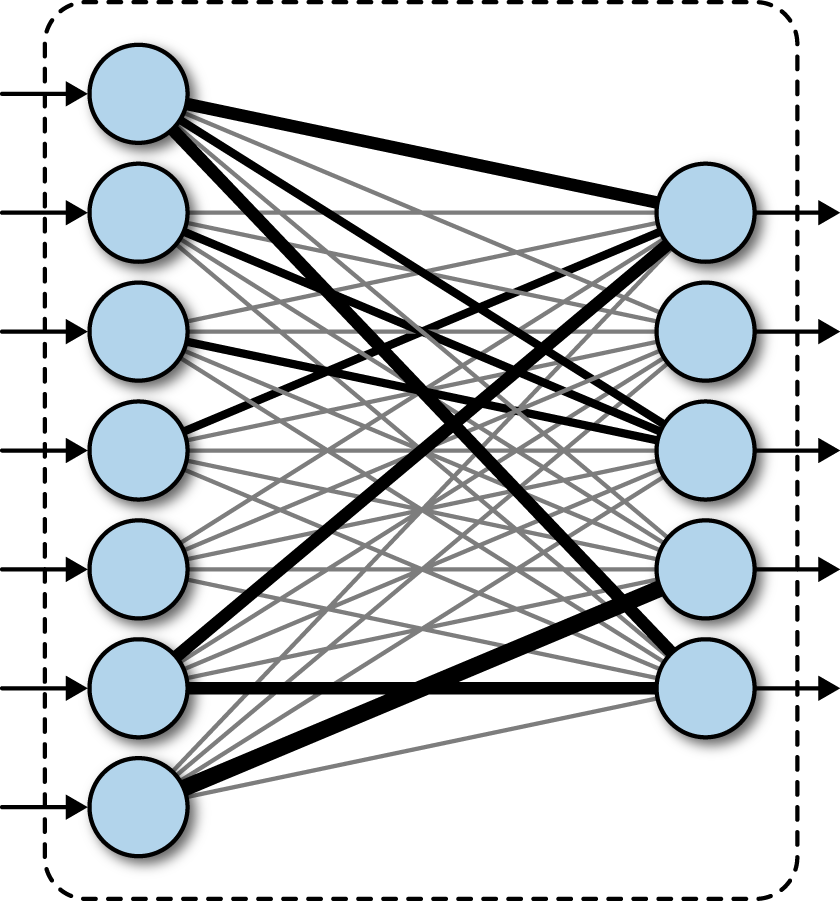


FIG 3.4

3.3.3 Convolutional Neural Network Models

To compare the study outcomes, a few of the most well-known CNN models that have been employed in related studies are covered in this part. Pre-trained weights from the ImageNet dataset are provided with these deep learning models. These models can be applied to tasks including categorization, feature extraction, and prediction. To begin learning a new task, one can take a pre-trained image classification model that has been trained to extract potent and informative features from photos. Three pre-trained models are used to the task of image classification. This section makes use of the ResNet50, MobileNet, and Alexnet architectures.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Network | Deep | Parameter (million) | Input  Image  Size |
| 1 | MobileNetv2 | 53 | 3.5 | 224 x 224 |
| 2 | ResNet50 | 50 | 25.6 | 224 x 224 |
| 3 | AlexNet | 8 | 61.0 | 224 x 224 |

Table 3.1

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Predicted** | |
| **Negative** | **Positive** |
| **Actual Cases** | **Negative** | True Negatives (tn) | True Positives (tp) |
| **Positive** | False Negatives (fn) | False Positives (fp) |

Table 3.2

The accuracy given in Equation 1 is a measure of how well the learning model is (Burkov, 2019);

Accuracy, 𝐴𝐶𝐶=𝑡𝑛+𝑡𝑝𝑡𝑛+𝑡𝑝+𝑓𝑛+𝑓𝑝 (1)

Equation 2, True Positive Rate (Recall, (TPR)) is a measure that defines how many of the positive predictions are true (Burkov, 2019).

Recall, 𝑇𝑃𝑅=𝑡𝑝𝑡𝑝+𝑓𝑝 (2)

Equation 3, Precision positive predictive value (PPV) is a measure that defines how many of the positive predictions are correct (Burkov, 2019).

Precision, 𝑃𝑃𝑉=𝑡𝑝𝑡𝑝+𝑓𝑛 (3)

3.4 System Architecture

The application's architecture revolves around Streamlit, a Python framework that facilitates the quick creation of interactive data science applications (Streamlit, 2020). Real-time interaction with machine learning models is made possible by the system's integration of the frontend, backend, and machine learning components into a streamlined interface.

1. Frontend: Designed using Streamlit, the frontend gives users an easy-to-use interface to upload photos, choose models, and see the outcomes. Machine learning applications can be easily deployed using Streamlit thanks to its real-time capabilities and ease of use.
2. Backend: TensorFlow is the Python library that is used by the backend to train and optimize models. They also handle image processing and model inference.
3. Machine Learning Pipeline: TensorFlow is used to construct the machine learning models, which are then integrated into the backend. Pre-trained models are loaded into this pipeline, which then refines them using the X-ray dataset and produces predictions (Abadi et al., 2012).

3.5 Data Preprocessing

3.5.1 Image Acquisition

The NIH Chest X-ray collection, among other publicly accessible datasets, provided the chest X-ray images used in this investigation (Wang et al., 2017). With more than 5,000 photos, this dataset is a good option for deep learning model training. Radiologists with 14 various thoracic illnesses, including pneumonia, labeled the photos.

3.5.2 Image Augmentation

During preprocessing, picture augmentation techniques were used to improve the generalization of the model. These consist of arbitrary flips, rotations, scaling, and brightness changes. According to Shorten and Khoshgoftaar (2019), image augmentation reduces overfitting by diversifying the training data.

3.5.3 Normalization

The photos were scaled to 224 by 224 pixels and their pixel values were set to a range of [0, 1] in order to normalize them. As a result, input dimension consistency is guaranteed, and neural network convergence is enhanced during training (LeCun et al., 2015).

3.6 Model Selection

3.6.1 Pre-trained Models

Transfer learning was used by using pre-trained models such VGG16, ResNet50, and InceptionV3, given the complexity of medical picture categorization (Simonyan & Zisserman, 2015; He et al., 2016; Szegedy et al., 2016). These models offer a solid foundation and are refined on the chest X-ray dataset to identify pneumonia. They were trained on the ImageNet dataset.

3.6.2 Custom Models

Custom Convolutional Neural Networks (CNNs) were created to investigate the possible advantages of architectures especially suited to the pneumonia detection challenge, in addition to pre-trained models. The unique models are designed to better capture the subtleties of the medical imagery by incorporating extra layers and distinct activation capabilities.

3.7 Model Optimization

3.7.1 Hyperparameter Tuning

The program uses random and grid search techniques to automatically adjust hyperparameters. In order to improve model performance, this procedure optimizes variables including learning rate, batch size, and number of epochs (Bergstra & Bengio, 2012).

3.7.2 Performance Metrics

Metrics including accuracy, precision, recall, F1 score, and AUC-ROC are used to assess the performance of the model. These measures offer a thorough understanding of how well the model differentiates between instances with and without pneumonia (Sokolova & Lapalme, 2009).

3.7.3 Cross-Validation

To evaluate the generalization and robustness of the model, K-fold cross-validation was used. By ensuring that the model operates consistently across several data subsets, this strategy lowers the possibility of overfitting (Kohavi, 1995).

3.8 Model Deployment

3.8.1 Model Export

Models are saved in a TensorFlow-compatible format (such as a.h5 file) after optimization. These models can be used in a clinical setting for real-time pneumonia identification, or they can be reloaded into the program for additional testing.

3.8.2 Model Serving

Real-time pneumonia prediction from uploaded X-ray pictures is made possible by the application using the Streamlit interface to deliver the models. Streamlit manages the combination of model inference and visualization, giving the user instantaneous feedback.

3.8.2 User Interface

Users can submit chest X-ray images, choose from a variety of machine learning models, and modify hyperparameters instantly thanks to the Streamlit-built user interface. Using methods like Grad-CAM, which emphasizes areas of interest in the X-ray pictures, the program also visualizes the results, giving prediction probabilities and model explanations (Selvaraju et al., 2017).

3.9 Validation and Testing

3.9.1 Validation Dataset

Using an independent test set of chest X-rays that were not seen during training, the application was verified. A vital component of using machine learning models in therapeutic contexts is ensuring that they can generalize to new, unseen data, which is what this validation process assures of.

3.9.2 Usability Testing

To make sure the application is user-friendly and fulfills the needs of its intended audience, usability testing was carried out on a sample of data scientists and healthcare professionals. During this testing phase, customer feedback was utilized to improve the overall user experience and refine the interface (Brooke, 1996).

3.9.3 Security and Compliance

This application places a high premium on data security. The frontend and backend communicate via encryption, and all patient data is anonymized. Patient privacy is safeguarded by the application's compliance with pertinent healthcare legislation, including HIPAA (Office for Civil Rights, 2013).

3.10 Conclusion

The most crucial prerequisite for making an accurate diagnosis of pneumonia disease is the availability of specialized radiologists. The main goals of this research are to increase medical proficiency in fields where radiologists are still underrepresented and to enable early detection of pneumonia to avert adverse outcomes (death, etc.). Improving healthcare can be greatly benefited from the development of algorithms in this field. Currently, several models are utilized to assess chest pictures of pneumonia patients, including the proposed CNN model and pre-trained MobileNet, ResNet, and AlexNet. In particular, applying these models to X-ray images has produced encouraging outcomes. This discovery is especially significant because X-ray scans are reasonably priced and readily available.

This methodology describes how to create a web application using Streamlit to optimize machine learning models for the diagnosis of pneumonia from chest X-rays. The program is user-friendly and has strong machine learning capabilities, which makes it suitable for a variety of users. Potential future developments could involve incorporating more sophisticated deep learning methods and broadening the application's detection capabilities.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Zheng, X. (2012). TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation, 265–283.

Akter, S., & Shamsuzzaman, J. F. (2015). Community acquired pneumonia. Int J Respir Pulm Med, 2–2.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017a). Understanding of a convolutional neural network. International Conference on Engineering and Technology, 1–6.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017b). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology, 1–6.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 281–305.

Brooke, J. (1996). A “quick and dirty” usability scale. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. Weerdmeester (Eds.). Usability Evaluation in Industry, 189–194.

Burkov, A. (2019). The hundred-page machine learning book. 3–5.

D. Berliner, N. Schneider, T. Welte, & J. Bauersachs. (2016). The differential diagnosis of dyspnoea. Dtsch. Arztebl., 113, 834–844.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2 , 1137–1143.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature.

McLuckie, A. (Ed. ). (2009). Respiratory disease and its management. Springer Science & Business Media.

Office for Civil Rights. (2013). U.S. Department of Health and Human Services. Health Insurance Portability and Accountability Act of 1996 (HIPAA).

Özkan, İ. N. İ. K., & Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. Gaziosmanpaşa Bilimsel Araştırma Dergisi,. 85-104.

Selvaraju, R. R., Batra, D., Cogswell, M., Das, A., Vedantam, R., & Parikh, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, 618–626.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 427–437.

Summah, H., & Qu, J. M. (2009). Biomarkers: a definite plus in pneumonia. Mediators of Inflammation.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 2097–2106.