AI - PROJECT

Pneumonia Detection from Chest X-Ray Images

Submitted by:

Harshvardhan Gupta	Archit Savaliya	Hitarth Shah	Diya Pansheriya
U23AI046	U23AI038	U23AI048	U23AI018

Table of Contents

Abstract

Pneumonia is a severe lung infection and one of the main causes of death between young children and older population, especially in low -resolution environments. Initial identity and treatment are important, but access to trained radiologists and timely diagnosis is a major challenge in many parts of the world. In recent years, artificial intelligence (AI) and deep learning have shown promises automatically for medical image analysis, which provides a scalable and effective alternative for the manual interpretation of the x -ray of the chest.

The project examines the application of the Convolutional Neural Networks (CNN) for binary classification of x-rays of the chest deficiency on a deficiency between the normal lungs and among the persons affected by pneumonia. The dataset used was retrieved from Kaggle and contains 5,863 marked images. By converting each image into grayscale, 150x150 pixels were prepared to shape and normalize pixel values. The dataset partition was - training (80%), verification (10%) and test (10%) set.

The original CNN of the model is the architecture, which composes five interconnection layers, after each to reduce overfitting with batch normalization and max pooling layers - with dropout layers. Our model acquires 93.1 accuracy

Overall, the project shows how deep learning, especially CNN, can serve as a valuable tool to support medical diagnosis. With further processing and integration into network or mobile platforms, this system has the opportunity to support health professionals and improve initial pneumonia, especially in subordinate areas.

1. Introduction

1.1 Background and Motivation of the Problem

Pneumonia is one of the most important causes of death in children under five, and adults worldwide, especially in low resource settings where timely medical diagnosis is limited. Traditional diagnosis through x -rays of the chest often depends on expert radiologist, which can be time -consuming, subjective and inaccessible in rural areas.

The rapid progress of deep learning is a promising opportunity to detect and improve pneumonia using the chest x -rays, especially with fixed nervous networks (CNN). It can reduce the clinical burden on medical professionals significantly and ensure early intervention.

Our inspiration for the production of this model stems from the desire to bridge health services and AI, which provides a sharp, efficient and scalable solution to detect early pneumonia. By taking advantage of CNN, our system can help doctors, support clinical decisions and can be integrated into mobile or online platforms for wider access.

1.2 Literature Survey or Related Works

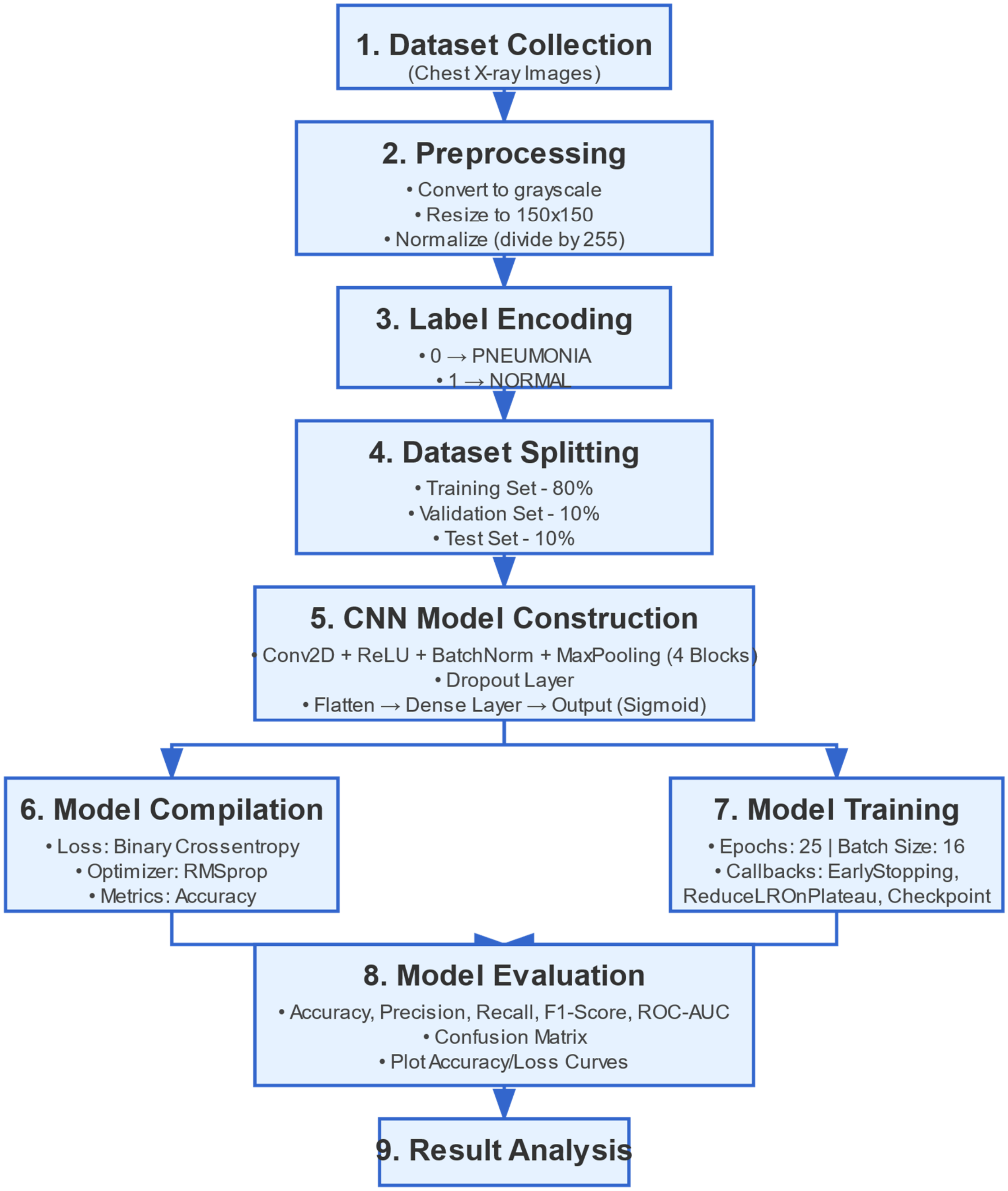
A number of earlier studies have attempted to utilize deep learning models for medical image classification. Particularly, a variety of convolutional neural network (CNN) architectures have been developed to classify X-ray images into healthy or pneumonia-infected categories. The most widely cited models exhibit test accuracies ranging from 85% to 87%. One of the prominent works is by Kermany, who have trained a CNN model using a large public database with impressive results but still limited in generalizability and higher false positives. Our work takes these earlier models forward by having a deeper CNN architecture, robust preprocessing, and stringent evaluation.

1.3 Contributions

- Created a CNN-based model capable of binary classification of chest X-ray images.
- Used a well-curated public dataset with over 5800 images.
- Achieved a test accuracy of 93.1%, outperforming existing baseline models.
- Maintained model generalizability, shown by a minimal gap between training and test accuracy.
- Provided detailed analysis and validation to confirm model reliability.

2. Flowchart / System Diagram

The diagram below outlines the end-to-end pipeline of our pneumonia detection system.



3. Methodology / Technique

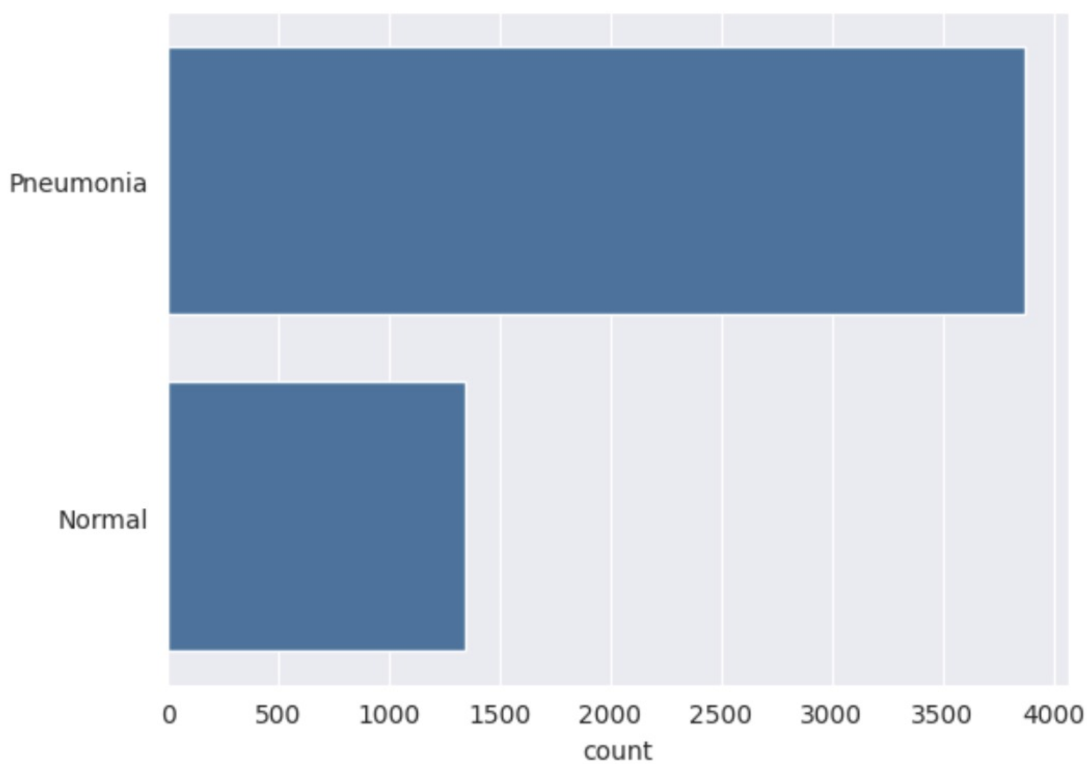
Code of Model : https://github.com/Diya5772/pneumonia-detection/blob/main/pneumonia.ipynb

To detect pneumonia from chest X-ray images, we designed a Convolutional Neural Network (CNN) that performs binary classification—i.e., predicting whether an X-ray scan indicates a normal case or pneumonia.

Data Collection

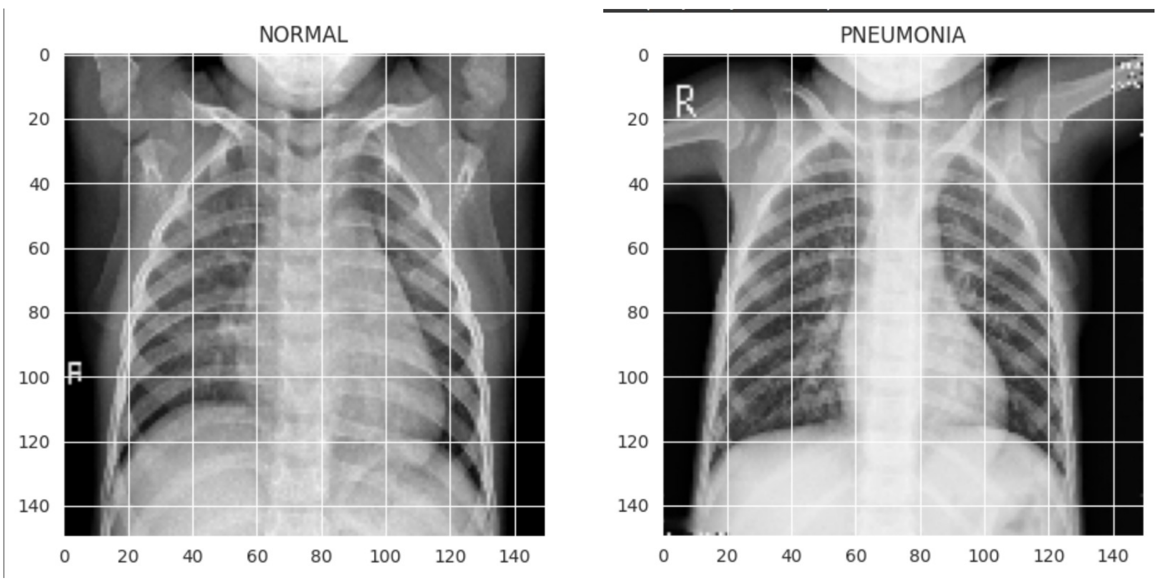
We used a publicly available dataset from **Kaggle** titled *“Chest X-Ray Images (Pneumonia)”*. This dataset contains a total of **5,863 chest X-ray images**, categorized into two classes: **PNEUMONIA** and **NORMAL**. The images were collected from pediatric patients and are labeled accordingly. This dataset was chosen because it is widely used in the research community and provides a balanced and real-world representation of the pneumonia classification problem.

Using this dataset allowed us to focus on building and evaluating the model without needing to gather data manually. It also provided us with enough samples for proper training, validation, and testing, making it suitable for deep learning applications.



Preprocessing

Before training the model, we carried out essential image preprocessing steps to ensure consistency and improve model performance. The raw chest X-ray images were first converted to grayscale, as colour information is not relevant for detecting pneumonia. Next, all images were resized to 150x150 pixels to standardize input size and reduce computational load.



Data Augmentation

To improve generalization and reduce the risk of overfitting, we implemented data augmentation using ImageDataGenerator. This helped create variations of the training images by applying random transformations such as rotation, zoom, shear, and horizontal flipping. These augmentations simulate real-world variations in the dataset, allowing the model to learn more robust features.

This approach not only increased the diversity of the training data but also made the model more resilient to different image orientations and slight distortions that can occur during X-ray scanning.

Dataset Splitting

The split of the Dataset was into three distinct subsets: training, validation, and testing. This step is crucial to ensure that the model is trained effectively, tuned appropriately, and evaluated objectively.

- **Training Set (80%)**: The majority of the data is used to train the CNN model. This is where the model learns to identify patterns and features from the X-ray images that differentiate between pneumonia and normal cases.
- **Validation Set (10%)**: This subset is used during training to fine-tune the model. It helps in monitoring the model's performance on unseen data and prevents overfitting by using callbacks like EarlyStopping and ReduceLROnPlateau.
- **Test Set (10%)**: The final 10% of the data is reserved for evaluating the model's performance after training is complete. This ensures that the results reported (accuracy, F1-score, etc.) reflect the model's true generalization capability.

Model Architecture

The CNN consists of five convolutional blocks, each followed by Batch Normalization and Max Pooling layers. As we go deeper into the model, the number of filters increases from 32 to 512, enabling the model to learn more complex patterns. We also included Dropout layers in between to avoid overfitting by randomly deactivating a portion of the neurons during training.

The model ends with two fully connected (Dense) layers, with a final sigmoid activation function to output probabilities for binary classification. Below is the model summary that describes each layer.

Model: "sequential_9"

Layer (type)	Output Shape	Param #
conv2d_41 (Conv2D)	(None, 150, 150, 32)	320
batch_normalization_41 (BatchNormalization)	(None, 150, 150, 32)	128
max_pooling2d_41 (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_42 (Conv2D)	(None, 75, 75, 64)	18,496
batch_normalization_42 (BatchNormalization)	(None, 75, 75, 64)	256
max_pooling2d_42 (MaxPooling2D)	(None, 38, 38, 64)	0
conv2d_43 (Conv2D)	(None, 38, 38, 128)	73,856
batch_normalization_43 (BatchNormalization)	(None, 38, 38, 128)	512
max_pooling2d_43 (MaxPooling2D)	(None, 19, 19, 128)	0
dropout_16 (Dropout)	(None, 19, 19, 128)	0
conv2d_44 (Conv2D)	(None, 19, 19, 256)	295,168
batch_normalization_44 (BatchNormalization)	(None, 19, 19, 256)	1,024
max_pooling2d_44 (MaxPooling2D)	(None, 10, 10, 256)	0
conv2d_45 (Conv2D)	(None, 10, 10, 512)	1,180,160
batch_normalization_45 (BatchNormalization)	(None, 10, 10, 512)	2,048
max_pooling2d_45 (MaxPooling2D)	(None, 5, 5, 512)	0
dropout_17 (Dropout)	(None, 5, 5, 512)	0
flatten_9 (Flatten)	(None, 12800)	0
dense_16 (Dense)	(None, 256)	3,277,056
dense_17 (Dense)	(None, 1)	257

Total params: 4,849,281 (18.50 MB)
Trainable params: 4,847,297 (18.49 MB)
Non-trainable params: 1,984 (7.75 KB)

Model Compilation

Once the CNN architecture was built, the next step was to compile the model. This involves configuring the model with a **loss function**, an **optimizer**, and evaluation **metrics**. These components define how the model will learn during training and how its performance will be measured.

- **Loss Function**: We used **Binary Crossentropy**, which is suitable for binary classification problems like ours (Pneumonia vs. Normal). It calculates the difference between the predicted probabilities and the actual class labels.
- **Optimizer**: The **RMSprop** optimizer was chosen for its efficiency in handling image data and its ability to adjust learning rates dynamically during training. It helps in speeding up convergence while maintaining stability.
- **Metrics**: We primarily monitored **Accuracy** to track the proportion of correctly classified images. However, other metrics were also considered later during evaluation for a deeper understanding of the model's performance.

Compiling the model with these settings ensured that it was well-prepared for effective learning and could be optimized appropriately during training.

Training Strategy

We trained the model for 25 epochs with a batch size of 16. Also the binary cross-entropy loss function and the RMSprop optimizer, which works well for image classification tasks. To fine-tune the training, we used the following callback mechanisms:

- **EarlyStopping**: Halts training if the loss stops improving, preventing unnecessary epochs.
- **ModelCheckpoint**: Saves the model with the highest accuracy.
- **ReduceLROnPlateau**: Reduces learning rate when the model’s performance plateaus.

Together, these components helped us build a stable and accurate model for pneumonia detection.

Evaluation Metrics

To assess how well our model performs, we evaluated it using a variety of metrics beyond just accuracy. These metrics provide a more complete picture, especially in medical imaging, where class imbalance or false positives can have serious implications.

- **Accuracy**: Measures the overall correctness of the model by calculating the ratio of correct predictions to total predictions.
- **Precision**: Indicates how many of the predicted pneumonia cases were actually pneumonia. High precision means fewer false positives.
- **Recall (Sensitivity)**: Measures the model's ability to correctly identify actual pneumonia cases. High recall ensures fewer missed diagnoses.
- **F1-Score**: The harmonic mean of precision and recall. This is particularly useful when there's an imbalance between classes, as it balances both false positives and false negatives.
- **ROC-AUC Score**: Evaluates the model's ability to distinguish between classes at various threshold levels. A higher AUC indicates better overall performance.
- **Confusion Matrix**: Provides a visual breakdown of true positives, true negatives, false positives, and false negatives, offering insights into specific areas where the model performs well or needs improvement.

Using these metrics allowed us to critically assess our model's diagnostic capabilities and confirm its reliability in distinguishing between normal and pneumonia-affected chest X-rays.

4. Implementation Environment

- The model was implemented using the following tools and libraries: Programming Language: Python
- Framework: Keras with TensorFlow backend
- Libraries: NumPy, Matplotlib, Scikit-learn, Pandas, Seaborn
- System : Google Colab
- Dataset: Kaggle Chest X-Ray Images (Pneumonia)

Callback mechanisms were used to enhance training:

- ReduceLROnPlateau for adaptive learning rates
- ModelCheckpoint to save best-performing model
- EarlyStopping to prevent overfitting

5.RESULTS

Final Performance Metrics:

- **Training Accuracy**: 97.43%
- **Test Accuracy**: 93.1%
- **F1-Score**: 0.9071

The close alignment between training and test accuracy indicates that our model generalizes well and is not overfitting. This is crucial in medical diagnosis, where a reliable model should perform consistently on unseen data.

Comparison with Existing Models:

Previous models in similar studies typically achieved accuracies between **85% to 87%**. Our model shows a clear improvement, reaching over **93.1% accuracy**, which makes it a strong candidate for real-world use.

Visualization of Results:

We included the following graphs to support our analysis:

- **Accuracy vs. Epochs**: Shows how both training and validation accuracy increased steadily over epochs, with no signs of overfitting.

- **Loss vs. Epochs**: Demonstrates a consistent decrease in training and validation loss.

The model mainly picks up on the **fogginess or haziness** in the lung area, which is one of the telltale signs of pneumonia in chest X-rays. By learning to detect these subtle features, the CNN has proven capable of distinguishing between healthy and pneumonia-affected lungs effectively.

6. CONCLUSION

We have developed a Convolutional Neural Network (CNN) model that accurately and effectively detects pneumonia using chest X-ray images as the data source. With the help of a meticulous and a well-laid-out model, in addition to the use of highly productive data preprocessing and augmentation methods, we were able not only to achieve high performance on the training set but also to build a model that may be further developed and will definitely perform well on the testing set.

The model has managed to achieve a whopping **93.1%** accuracy that is better than what others have realized in the particular field of study. This small discrepancy in the training and test values of accuracy suggests that the model is stable and does not overfit. Simultaneously, the application of **data augmentation** was the major kind of the model's improvement in terms of the very nature of the exposure during training.

It is possible to convert this system into an interactive web-based or mobile app for use by clinicians and medical staff, especially in resource-scarce localities. The tool will be a good candidate for eventual validation and clinical studies to prove to be an excellent source of important information for early pneumonia diagnosis hence most likely to save lives.

7. REFERENCES

A Novel CNN-based Approach for Pneumonia Detection using Keras and TensorFlow

https://ieeexplore.ieee.org/abstract/document/10739031?casa_token=0bbHA17dBOkAAAAA:Xp7WPEJzZ_LPOqcrtWcQJ8BwQXlZNlj2EPllOu838HWQxApE6Fl9TAXbAodx1OkfvNvEgZzBjQ

Convolutional Neural Networks Explained (CNN Visualized)

https://www.youtube.com/watch?v=pj9-rr1wDhM&t=415s