Pediatric Chest X-ray Image Classification for Pneumonia Detection

Introduction

The project aims to develop an AI system that accurately classifies pediatric chest X-ray images into Pneumonia and Normal categories. This classification task is crucial for assisting healthcare professionals in diagnosing respiratory conditions in pediatric patients. The dataset used for this project was obtained from Kaggle and X-ray images sourced from retrospective cohorts of pediatric patients aged one to five years at the Guangzhou Women and Children's Medical Center.

Dataset Description

The dataset from Kaggle consists of **3,322 chest X-ray images** categorized into Pneumonia and Normal classes. These images, stored in JPEG format, underwent resizing to 224x224 pixels to match the input size expected by the ResNet model. Pixel values were scaled between 0 and 1 and then normalized using mean and standard deviation. Additionally, the dataset was split into training (with 1,341 instances each of Pneumonia and Normal), validation (624 instances), and test sets (16 instances). These preprocessing steps ensure compatibility with the ResNet architecture and enhance model performance.

Model Architecture

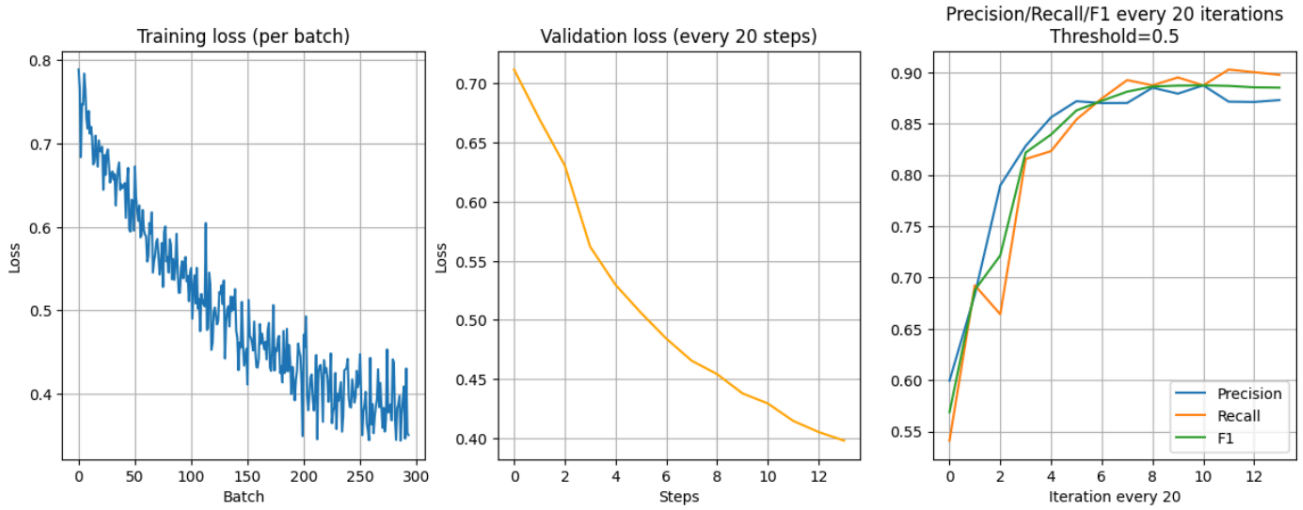
I used a pre-trained ResNet-18 model for the architecture model and fine-tuned it with my dataset.

Training Process

The training process involved fine-tuning a pre-trained ResNet-18 model for binary classification of pediatric chest X-ray images. The model's fully connected layer was adapted to output a single value representing the probability of pneumonia presence. A stochastic gradient descent **(SGD) optimizer** with a **learning rate of 0.001** was used, and the **BCEWithLogitsLoss** function served as the loss criterion. Key metrics like **accuracy, precision, recall, and F1 score** were monitored during training. Data augmentation techniques were not applied. Training continued until no improvement in the F1 score was observed for two consecutive iterations, with a periodic saving of model checkpoints.

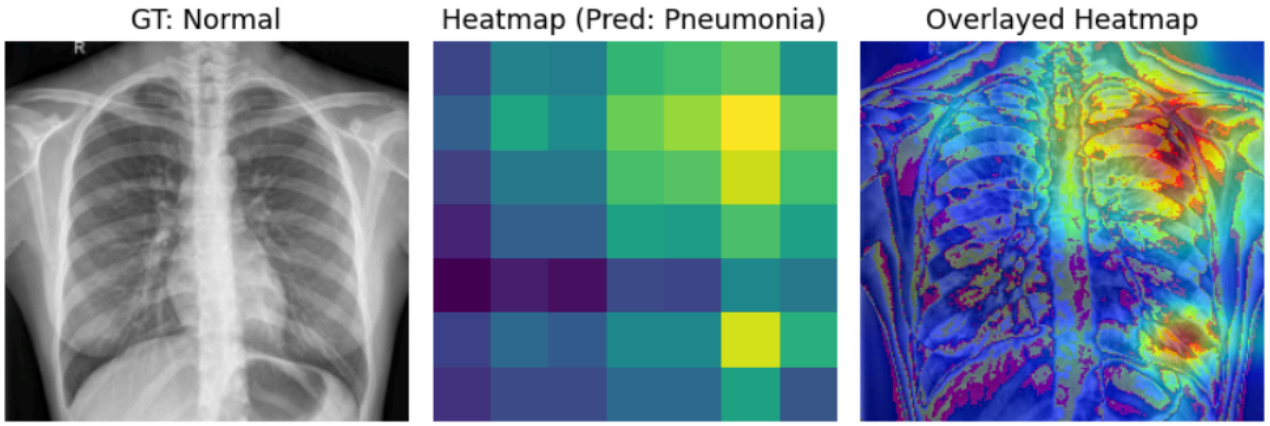
Evaluation Results

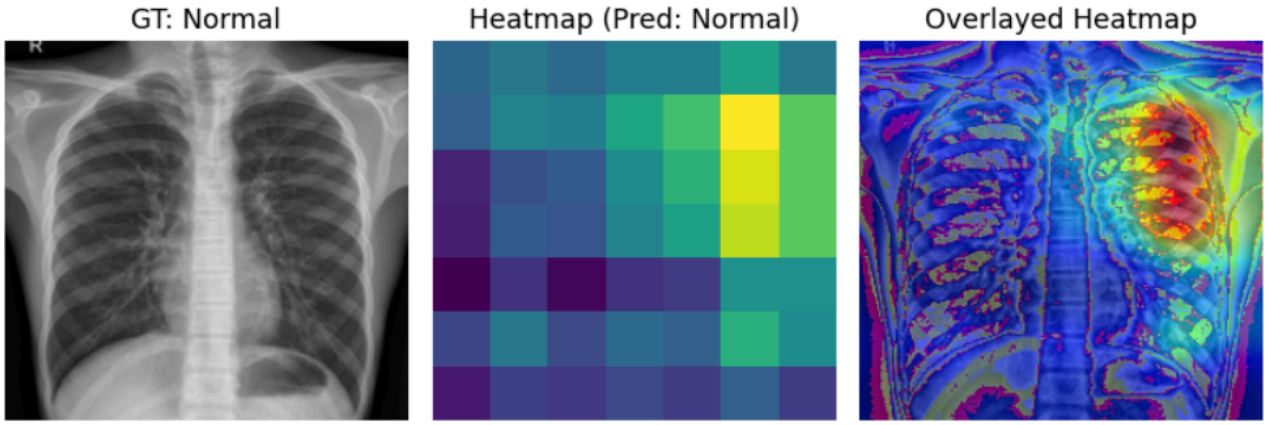
At the end of training, after 280 iterations, the model achieved a training accuracy of 89.30% and a validation accuracy of 85.42%. The corresponding training and validation losses were 0.507 and 0.398 respectively. Moreover, the precision, recall, and F1 score of the model were 87.28%, 89.74%, and 88.50% respectively. These evaluation metrics provide a comprehensive understanding of the model's performance and effectiveness in classification tasks after 280 training iterations.

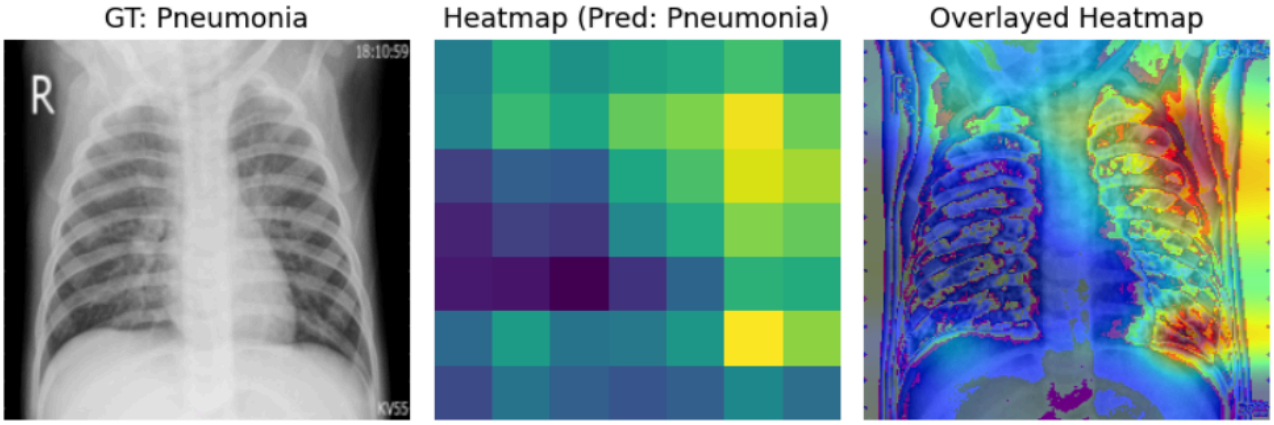


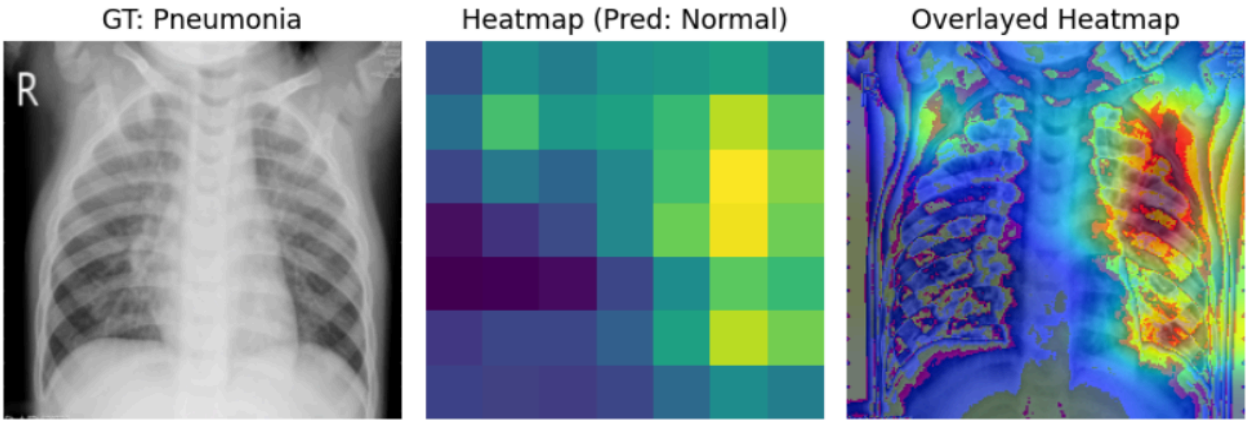

```
Iteration: 220

Training Accuracy: 0.8799, Validation Accuracy: 0.8590
Training loss: 0.5370718240737915       Validation loss: 0.4295364737510681
Precision: 0.8872       Recall: 0.8872  F1 Score: 0.8872

Iteration: 240

Training Accuracy: 0.8852, Validation Accuracy: 0.8558
Training loss: 0.5259713530540466       Validation loss: 0.41478567123413085
Precision: 0.8713       Recall: 0.9026  F1 Score: 0.8866

Iteration: 260

Training Accuracy: 0.8870, Validation Accuracy: 0.8542
Training loss: 0.5158567428588867       Validation loss: 0.40541707575321195
Precision: 0.8710       Recall: 0.9000  F1 Score: 0.8852

Iteration: 280

Training Accuracy: 0.8930, Validation Accuracy: 0.8542
Training loss: 0.5066614151000977       Validation loss: 0.3981399953365326
Precision: 0.8728       Recall: 0.8974  F1 Score: 0.8850

Training complete!
```

In addition to the evaluation metrics, GradCAM images were generated to show which parts of the input images the model focused on to make predictions.









Insights Gained

Based on the evaluation results, our model showed clear strengths and areas for improvement. It performed well overall, with strong performance metrics indicating its effectiveness. However, we also noticed some variability in its performance, especially when comparing validation results to training results. One interesting insight we gained from this project is how the model focuses on specific areas of the lungs to make predictions. We found that it tends to look at the left side of the lungs, particularly the upper left area, to determine if a patient has pneumonia.

However, this approach led to some misclassifications. For example, the model sometimes classified normal cases as pneumonia when focusing on the entire left lung. Conversely, it also missed a pneumonia case although it sees a full part of the left lung. This inconsistency might be due to overfitting, which became apparent after around 240 iterations when validation results started to decline while training results remained strong.