Optimizing a Convolutional Neural Network for Pneumonia Classification in Chest Radiographs

Nizar Altawam - 2023

Executive Summary

Goal: Detecting Pneumonia in chest X-ray with Machine learning

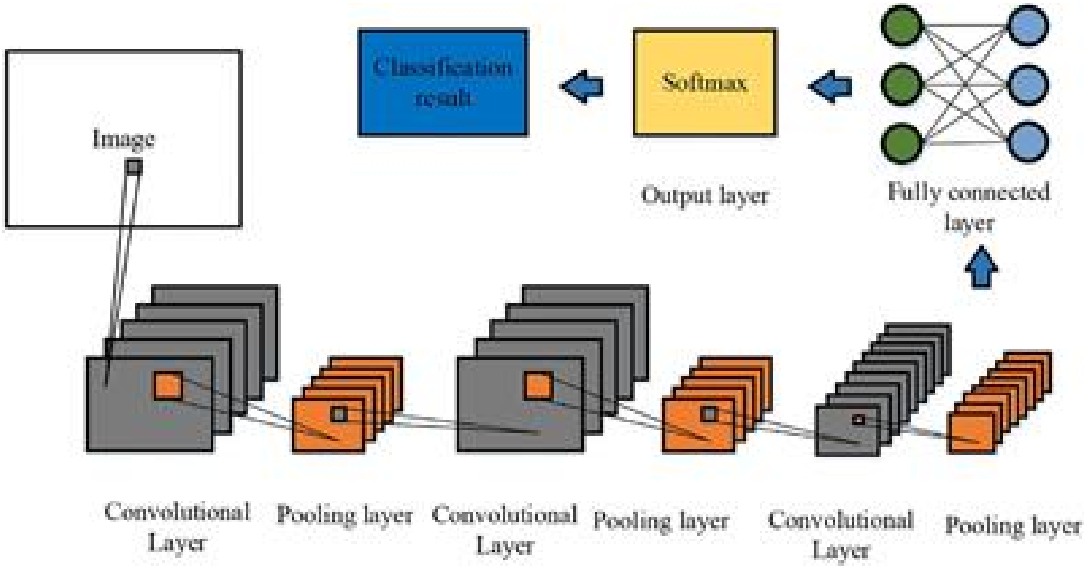
Key challenge: Improve accuracy while managing costs

Result: Built a CNN model with an accuracy of 83% on a test set

Background

- Computer vision techniques like convolutional neural networks (CNNs) can provide an **automated approach** to analyzing medical images and extracting diagnostic information

- CNNs apply a series of *convolutional*, *pooling*, and *fully-connected layers* to automatically learn hierarchical features directly from image data.



Project as a Data Science problem

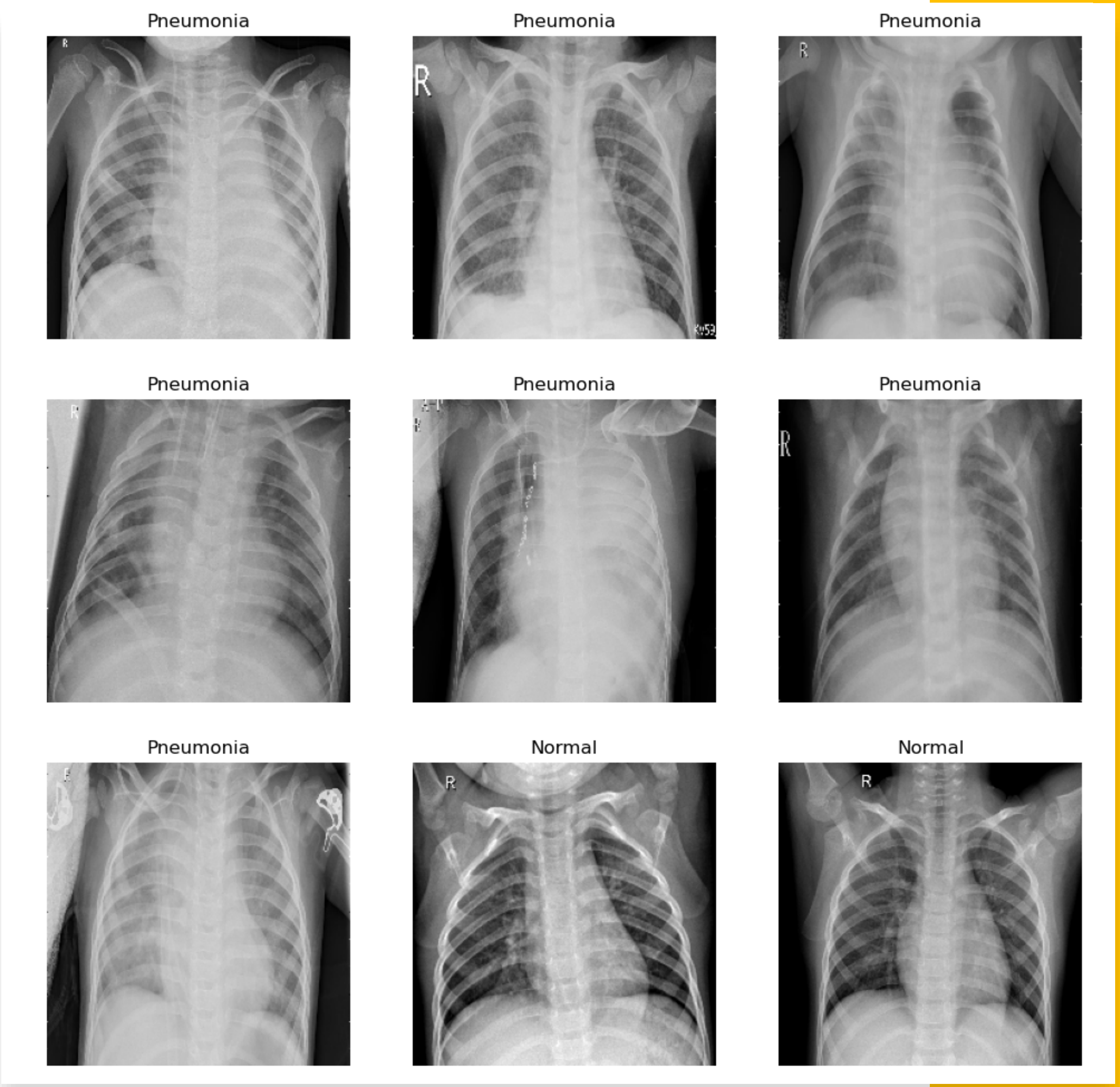
1. Dataset:
   - Source: Kaggle | Chest X-Ray Images (Pneumonia) - License: CC BY 4.0
   - 5863 chest XR images divided into 2 categories: <u>Pneumonia</u> and <u>Normal</u>
2. Pre-process the data
3. Design a CNN architecture and train the model
4. <u>Success</u>: Model accurately classifies the radiographs with emphasis on balancing precision and recall.

Exploratory Analysis

- Class imbalance in original dataset:

Normal	1347
Pneumonia	3875

- Variability in pneumonia patters



Data Processing

Train/Validation/Test split (5216:16:624)

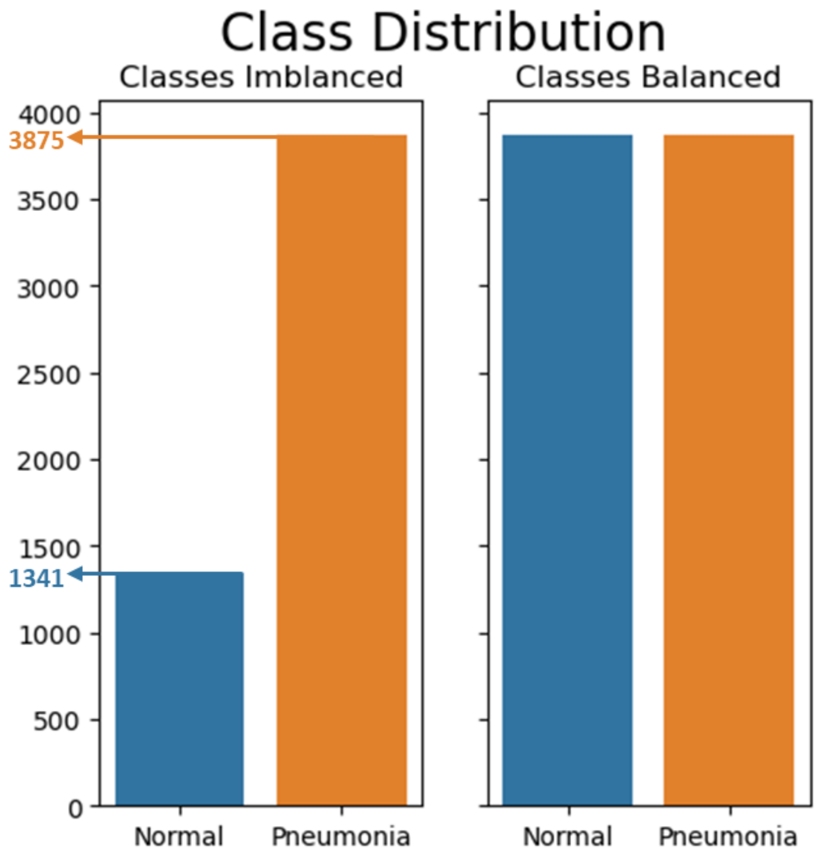
Image resizing to 224x224 pixels

Class balancing with SMOTE

Shuffling Training data

Normalization

Encoding with LabelEncoder()



CNN Architecture

- Convolutional layers for feature extraction
- Pooling layers downsample *(reduce the dimensions of the feature maps)*
- Flatten features into a 1D vector
- Fully connected layers classify
- Dropout regularization to prevent overfitting
- Binary classification output layer

Layer type	Output Shape	Param #
Conv2D	(None, 222, 222, 32)	320
Conv2D	(None, 220, 220, 64)	18496
MaxPooling2D	(None, 110, 110, 64)	0
Conv2D	(None, 108, 108, 32)	18464
MaxPooling2D	(None, 54, 54, 32)	0
Conv2D	(None, 52, 52, 64)	18496
MaxPooling2D	(None, 26, 26, 32)	0
Flatten	(None, 43264)	0
Dense	(None, 64)	2768960
Dropout	(None, 64)	0
Dense	(None, 1)	65

Total params: 2,824,801

Trainable params: 2,824,801

Non-trainable params: 0

Model Compiling & Training

Early stopping: Creates a callback to stop training early if the validation loss does not improve after 3 epochs (patience=3) to prevent overfitting.

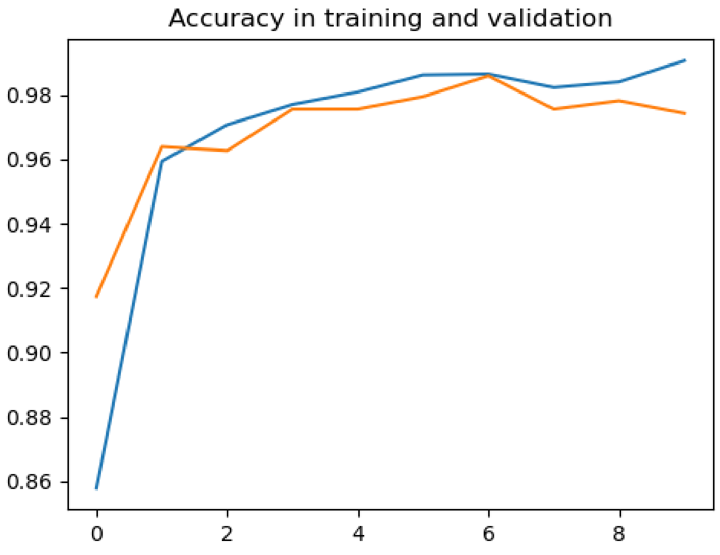
Adam Optimizer: adaptive optimization algorithm. It computes the adaptive learning rates for each parameter. This helps accelerate convergence.

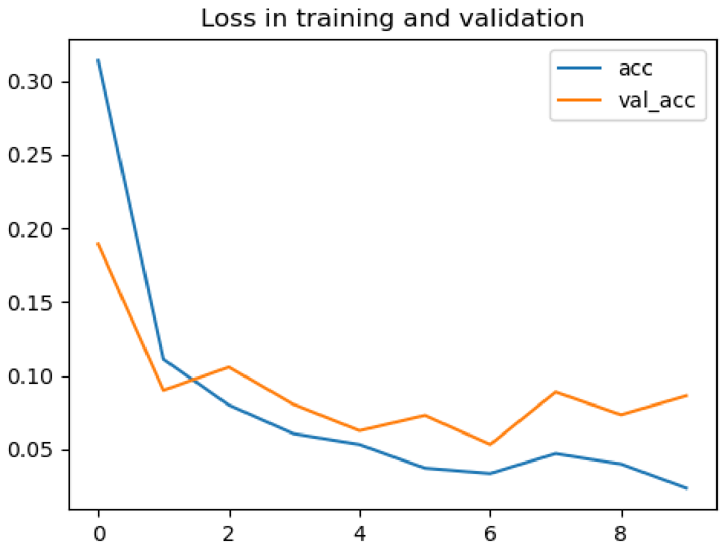
Loss: binary_crossentory used for binary classification problems.

Training Results

EarlyStopping worked as expected to stop training after the model started overfitting, while still retaining good performance on the validation set after 10 epochs.

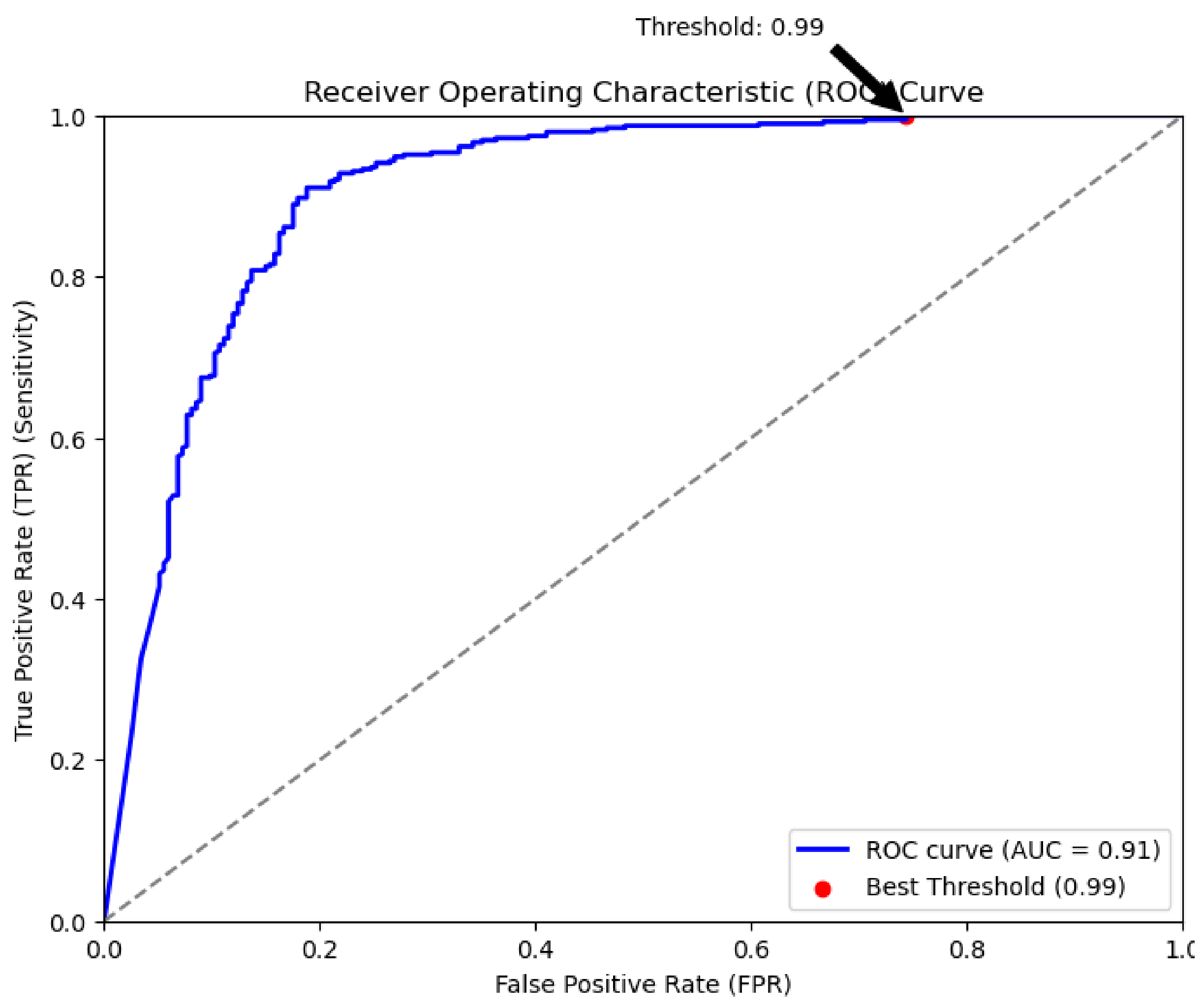
- 99% training accuracy
- 97% validation accuracy

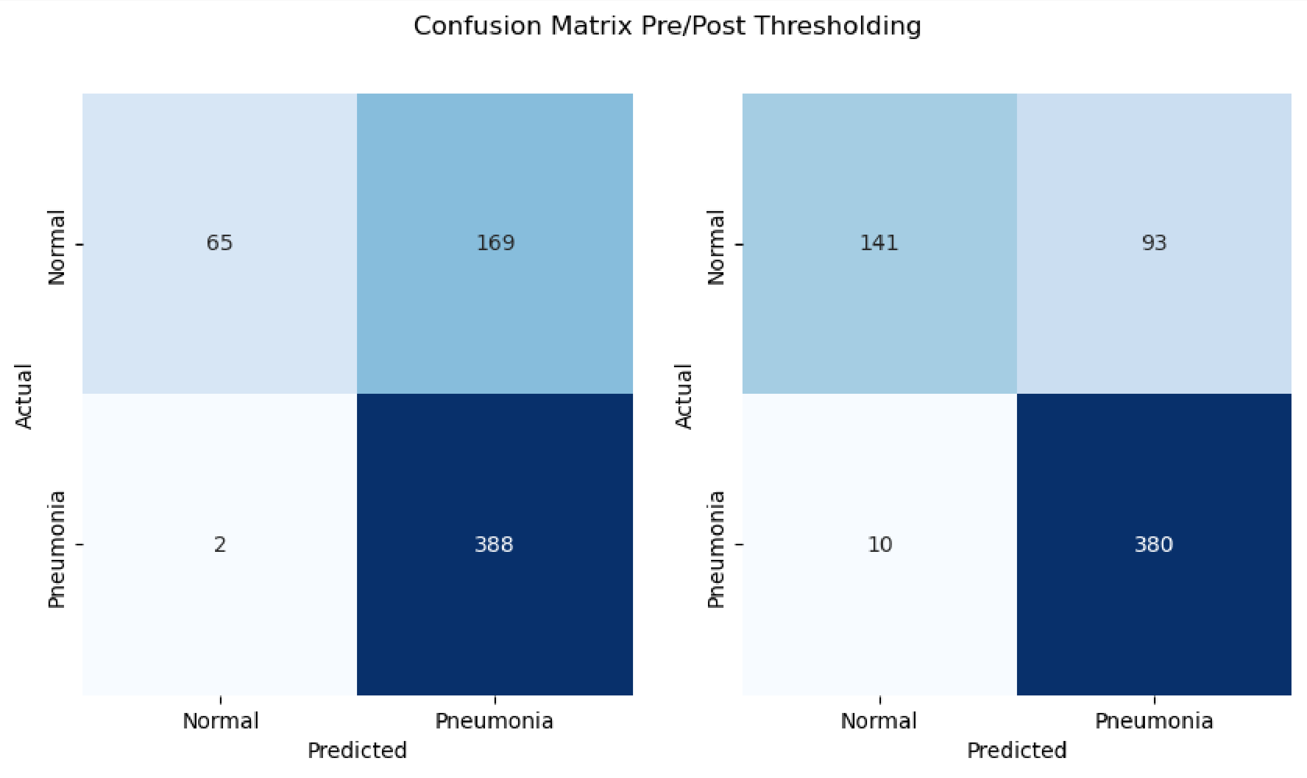




Model Evaluation & Thresholding

- Adjusted probability threshold using F1-score
  - Minimize combination of FPs and FNs
- Optimized balance of precision/recall with F1: **0.88**
  - <u>Best threshold</u> at this F1: **0.99**





Evaluation Results

- Improved accuracy: 72% to 83%
- Improved precision: 70% to 80%

With a threshold increase from 0.5 to 0.99, we are able prevent **76** incorrect diagnoses but at the cost of **8** missed ones.

This is the best threshold to balance the precision/recall of this model.

Assumptions & Limitations

- Test data matches the patterns in the training data *(pediatric patients, same facility, etc.)* With more variability in real-world x-rays, the model performance could potentially degrade.
- Relatively small dataset also limits the robustness of the model.
- Training classes balanced artificially with SMOTE oversampling may impact generalizability.
- Biases inherent in the data could affect generalizability
- The goal of this project was to balance Precision and Recall, other scenarios might call for maximizing precision to avoid missed diagnoses or optimizing the recall to cut costs on unnecessary workups.

Future Work
Additional work on this model could help improve its viability for clinical use

- More varied training data could make the model more robust
- Further tuning the CNN architecture and training hyperparameters could potentially boost accuracy further
- Employing ensemble techniques with multiple models could lead to more stable predictions

- As discussed in the previous slide, model can be optimized via thresholding for different goals.

Conclusion

- This project demonstrated successful development and application of a CNN model for classifying chest x-rays as normal or depicting pneumonia
- The model was trained on over 5,000 images and optimized using techniques like early stopping and threshold adjustment
- The final CNN model achieved an accuracy of 83% on the test set
- Thresholding the classification probabilities improved the precision from 69% to 80%.