Capstone Project

Chest X-Ray Image Classification for Respiratory Diseases

Using CNN (Convolutional Neural Network)

Sameer Ansari

21/08/2023

Table of Contents

1) Introduction

Diagnosis of Respiratory Diseases through Chest X-Ray Image Classification

Respiratory diseases, including COVID-19, viral pneumonia, and normal lung conditions, have significant implications for public health. The accurate and early detection of these conditions is crucial for effective medical intervention. This project focuses on leveraging Convolutional Neural Networks (CNNs) to classify Chest X-Ray images into three categories: COVID-19, normal, and viral pneumonia. The goal is to develop a robust model that aids medical professionals in making accurate and timely diagnoses.

Respiratory diseases can be very serious and it is important to diagnose them quickly and accurately. Traditional methods of diagnosis can be slow and sometimes inaccurate. This project uses machine learning to develop a model that can automatically analyse chest X-ray images and classify them into three categories: COVID-19, normal, and viral pneumonia. The goal is to create a reliable model that can help doctors make accurate diagnoses more quickly. The project will also explore how different image formats affect the model's performance and how it could be used in real-world clinical settings.

2) Client

This project holds several potential benefits for clients, particularly healthcare institutions, medical professionals, and researchers. Here's how this project can be helpful for clients: -

1) Accurate and timely diagnosis: The CNN model can accurately classify chest X-ray images into three categories: COVID-19, normal, and viral pneumonia. This can lead to faster and more reliable diagnoses, so that doctors can start treatment quickly.

2) Reduced workload for radiologists: The CNN model can help radiologists by pre-classifying X-ray images. This can help radiologists focus on the most critical and complex cases, so they can be more efficient.

3) Early disease detection: The model can help identify potential cases of respiratory diseases early on. This is important for managing and containing diseases like COVID-19 and viral pneumonia.

4) Objective and consistent analysis: The CNN model provides an objective and consistent approach to image analysis. This means that the results are not influenced by factors like fatigue or bias, so they are more reliable.

5) Resource optimization: The model can help healthcare facilities optimize the allocation of resources, such as isolation rooms and medical staff. This can help ensure that resources are used efficiently.

6) Research and insights: The dataset and model can be valuable for researchers studying respiratory diseases. The insights gained from the model's classifications can help us understand these diseases better and develop new treatments.

7) Remote diagnostics: The model can be used for remote diagnostics, which is important in situations where on-site expertise is limited.

8) Continual learning and improvement: The model's performance can be improved over time by updating it with new data. This means that the model can adapt to new cases and emerging patterns, so it can become even more accurate.

3) Dataset

The dataset sourced from Kaggle (https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset) consists of Chest X-Ray images categorized into three classes: COVID-19, normal individuals, and viral pneumonia cases. The dataset is divided into two main directories, namely "train" and "test," each containing three subfolders corresponding to the three classes: "covid," "normal," and "viral pneumonia." In total, the dataset contains 317 image files, with 238 being in JPEG format, 68 in JPG format, and 11 in PNG format. This dataset serves as a valuable resource for training and evaluating Convolutional Neural Networks (CNNs) to accurately classify Chest X-Ray images into these three clinically relevant

categories, which could contribute to the detection and diagnosis of respiratory diseases such as COVID-19 and viral pneumonia.

4) Connecting Google Colab with Kaggle through API

To streamline the data acquisition process, Google Colab was connected with Kaggle through the API. This allowed for the direct download of the dataset into Colab, reducing the download time significantly.

5) Extracting Files from the Zip Archive

The downloaded dataset was in a compressed zip archive format. Extracting the files was necessary to access and process the data effectively within the Colab environment.

6) Checking Number of Images in Each Category for Train and Test Data

Understanding the distribution of images across categories in both the training and test datasets provided insights into the dataset's balance and helped in preparing for subsequent data processing steps.

7) Importing Required Libraries

Essential libraries for data manipulation, visualization, and machine learning were imported to facilitate various stages of the project.

8) Data Augmentation

Data augmentation, a technique that involves applying transformations to the training data, was employed to increase the diversity and size of the dataset. This approach helps the model generalize better by exposing it to various perspectives of the same images.

9) Creating Training and Test Datasets Using Data Generators

Data generators were created using the flow_from_directory() method, which allowed the efficient loading of images from directories while applying data augmentation and shuffling. This step facilitated the division of data into training and validation sets.

10) Building CNN Architecture

The CNN model architecture was constructed using the Keras Sequential API. The architecture included multiple convolutional layers, max-pooling layers, and dense layers for classification. The model was compiled with appropriate loss and optimization functions.

11) Defining Callbacks for Early Stopping and Model Checkpoint

Callbacks were defined to prevent overfitting and to save the best model weights during training. Early stopping aimed to halt training when validation loss stopped improving, while the model checkpoint saved the best weights based on validation performance.

12) Fitting the Model on Training and Validation Data

The CNN model was trained on the training dataset while monitoring its performance on the validation dataset. The early stopping and model checkpoint callbacks were used during this stage.

13) Plotting Loss and Accuracy Curves

Visualizing the training and validation loss and accuracy curves helped assess the model's learning process and generalization capabilities. Overfitting and convergence could be observed through these curves.

14) Model Evaluation

The model's performance was evaluated using various metrics, including accuracy, precision, recall, and F1-score. Classification reports provided insights into class-wise performance on both the training and validation datasets.

(I) Train and Validation Data (Accuracy & Loss)

Data	Accuracy	Loss
Training	0.8845	0.3104
Validation	0.8636	0.3314

(II) Train and Validation Data (Accuracy, Precision, Recall, F1 Score)

Dataset	Accuracy	Precision	Recall	F1-Score
Train	0.322709	0.321385	0.322709	0.321778
Validation	0.484848	0.484734	0.484848	0.484069

(III) Train and Validation Data (Classification Report)

Train Classification Report:

	precision	recall	f1-score	support
0	0.45	0.45	0.45	111
1	0.17	0.16	0.16	70
2	0.26	0.29	0.27	70
accuracy			0.32	251
macro avg	0.30	0.30	0.30	251
weighted avg	0.32	0.32	0.32	251

Validation Classification Report:

	precision	recall	f1-score	support
0	0.54	0.50	0.52	26
1	0.35	0.35	0.35	20
2	0.55	0.60	0.57	20
accuracy			0.48	66
macro avg	0.48	0.48	0.48	66
weighted avg	0.48	0.48	0.48	66

15) Predictions on Images

The trained model was used to make predictions on sample images from each category. These predictions demonstrated the model's ability to classify images into COVID-19, normal, and viral pneumonia categories.

16) Out-of-Time Validation

Out-of-time validation was performed to assess the model's performance on new data collected after the model was trained. Predictions were made on a set of new images to gauge the model's generalization capabilities.

17) Conclusion

This project trained a CNN model to classify chest X-ray images into three categories: COVID-19, normal, and viral pneumonia. The project used a systematic approach, from collecting data to evaluating the model. The results showed that the model can accurately diagnose respiratory diseases. This is important because it can help doctors to intervene early and improve patient outcomes. This project shows how deep learning can be used to improve medical diagnostics and the importance of accurate classification in healthcare.

18) Model Performance & Summary

(I) In this dataset when I use a deeper CNN structure, the model overfits the data. This is because a deeper model has more parameters, which can make it more likely to memorize the training data rather than learning the underlying patterns.

(II) The model achieved a training loss of 0.3104 and a training accuracy of 0.8845. This means that the model made an average error of 0.3104 on the training data and correctly classified 88.45% of the training images.

(III) The model also achieved a validation loss of 0.3314 and a validation accuracy of 0.8636. This means that the model made an average error of 0.3314 on the validation data and correctly classified 86.36% of the validation images.

(IV) The model performed well on both the training and validation data. It correctly predicted the images of all 3 classes (Covid, Normal, and Viral Pneumonia). It also performed well on out-of-validation data.

19) Further Analysis

(I) The more data we provide to train the model, better its performance will be. This is because the model will have more examples to learn from.

(II) If we have images or data of more categories, we can build a model that can classify a wider variety of chest X-rays and identify more respiratory diseases.