50.039 Theory and Practice of Deep Learning

Project Report: Classification of COVID-19 Respiratory Sounds

https://github.com/ssiyer4/DL_COVID19-COSWARA

Shwetha Iyer (1006308), Javin Eng (1005978), Lim Cheng Ee (1004896)

I. Introduction + Motivation

The focus of this project is to develop a deep learning model with a medical application. The medical application our team has chosen is identifying and classifying respiratory sounds into two categories ("positive" and "negative") for diagnostic purposes, focusing on the COVID-19 respiratory illness. This project aims to leverage deep learning techniques to improve the accuracy and efficiency of medical sound classification, contributing to enhanced healthcare diagnostics. This topic is especially relevant in today's world, as we still face the aftermath of the 2020 COVID-19 global pandemic.

II. Methods

1. Dataset

The dataset chosen for this task is the Coswara dataset[1] (Bhattacharya et al., 2023) which contains respiratory sounds recorded between April-2020 and February-2022 from 2635 individuals (1819 SARS-CoV-2 negative, 674 positive, and 142 recovered subjects. The dataset was collected mainly in India and contains respiratory sounds caused by various strains of the COVID-19 virus which makes this dataset robust. The whole dataset was not used for this project, due to file compression errors faced when downloading the dataset. A subset of the Coswara dataset containing 814 samples from 814 participants from the "breathing - deep" category was utilised for this project.

The meta-data provided in the Coswara repository has these following labels for the audio samples:

a. Healthy
b. No_resp_illness_exposed
c. Resp_illness_not_identified
d. Recovered_full
e. Positive_asymp
f. Positive_mild
g. Positive_moderate

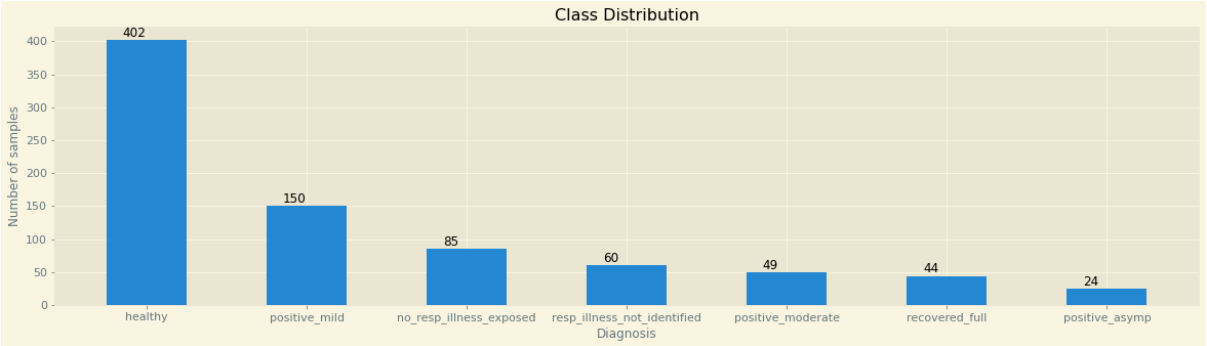


Fig 1 - original distribution of dataset

As one can note from the figure above, the distribution of classes is imbalanced in the subset of the Coswara dataset used for this project. This imbalance was alleviated by re-assigning the labels into two classes - “positive” and “negative”. “Positive” denotes positive for the COVID-19 illness, symptomatic or otherwise. “Negative” denotes negative for the COVID-19 illness, regardless of whether previously infected or not. After assigning the new labels, this is the distribution attained:

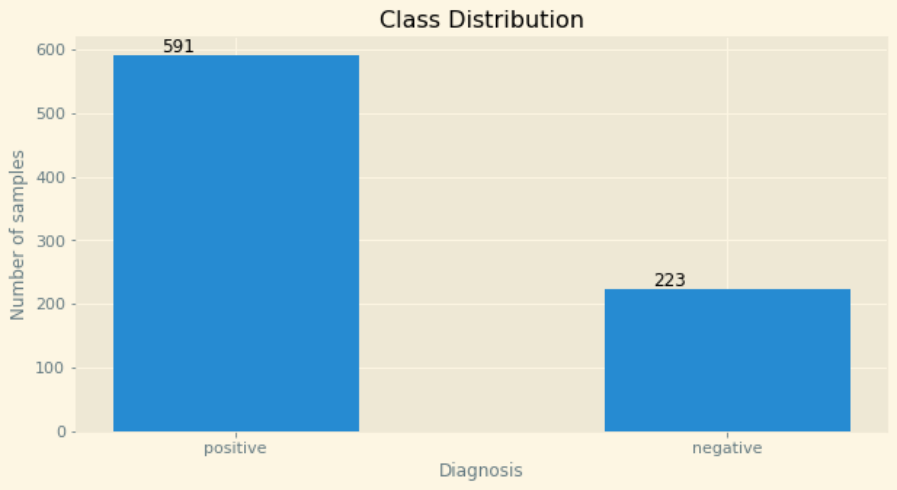


Fig 2 - binary distribution of data

2. Data Pre-processing

A set of features was extracted from the audio signals. The librosa library was utilised to extract the following features from the signal:

- Mel-Frequency Cepstral Coefficients (MFCCs): the distribution of energy across different frequency bands.
- Zero Crossing Rate (ZCR): the rate at which the audio signal changes sign.
- Root Mean Square (RMS) Energy: average power of the audio signal over time.
- Spectral Centroid: the "centre of mass" or average frequency of the power spectrum, to analyse into its tonal or harmonic content.

- Spectral Bandwidth (Entropy): the spread of the power spectrum around the spectral centroid, which provides information about its complexity.
- Autocorrelation: measures the similarity between a signal and a delayed version of itself. Used to detect repeating patterns within the audio signal.

3. Model Architecture

The CNN-LSTM architecture developed combines convolutional neural networks (CNNs) and long short-term memory networks (LSTMs). The model starts with two convolutional layers, each followed by batch normalisation and ReLU activation, to extract spatial features from the data. These layers have a kernel size of 3 and padding of 1, maintaining the original dimensionality, with a max pooling layer to reduce the feature map size. A dropout layer is also incorporated after the convolutional blocks to prevent overfitting.

The extracted features are then fed into an LSTM layer, and the output from the LSTM is directed to a fully connected layer that classifies the input into categories based on the learned features.

To optimise the model, a grid search was conducted over various configurations of dropout rates, LSTM dimensions, and numbers of LSTM layers to identify the setup that minimises validation loss.

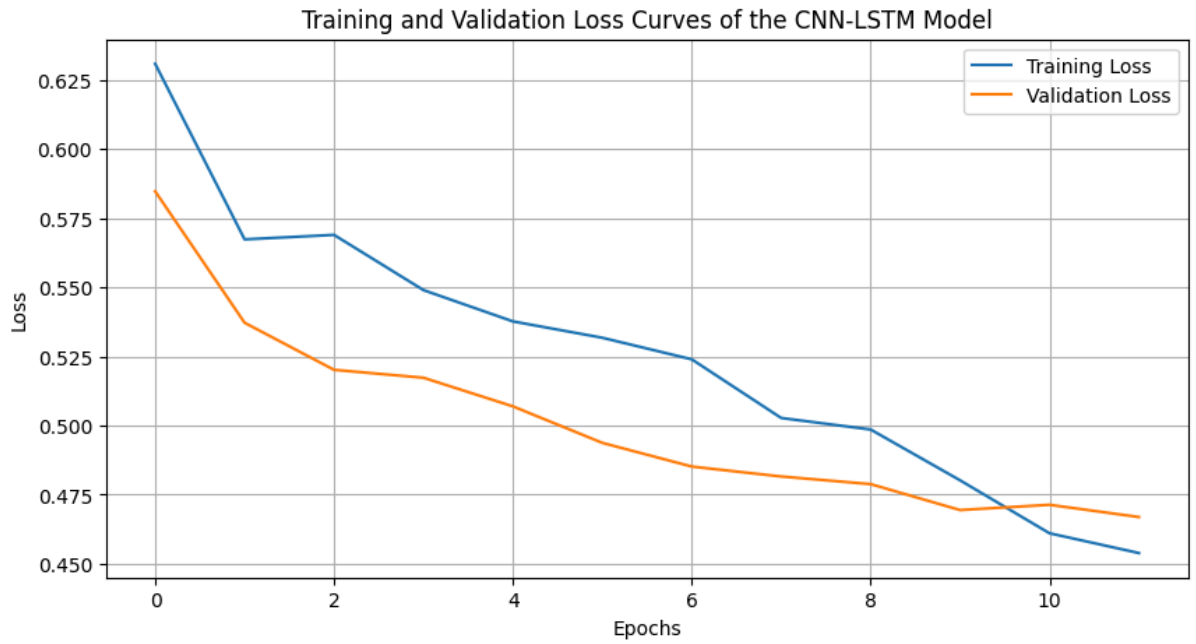


Fig 3 - loss curves

Test Loss: 0.4668, Test Accuracy: 79.75%

Best Model Settings: {'dropout': 0.3, 'hidden_dim': 128, 'num_layers': 1}

4. Training

The training process for the CNN-LSTM model involved using a grid search to explore various configurations by adjusting dropout rates, LSTM hidden dimensions, and the number of LSTM layers. Each configuration was trained for 10 epochs with the Adam optimizer at a learning rate of 0.001. Dropout regularisation was incorporated during training to prevent overfitting and enhance the model's ability to generalise across different datasets.

After each epoch, the model's performance was evaluated on a validation set by measuring the loss. The best model configuration with the lowest validation loss was saved for further evaluation.

5. Evaluation Metrics

The table below outlines the metrics of our final model:

Evaluation Metric	Value
Accuracy	0.7975
F1	0.5074
Precision	0.739
Recall	0.3863

Table 1 - evaluation metrics

III. Results + Discussion

A. Comparison with Published Model

The Coswara dataset has previously been analysed by Bhattacharya et al. (2023)[1]. They created a multi-modal model which produced a COVID-19 prediction score based on feeding the 9 different audio files available through bi-directional LSTMs, the symptoms through a decision tree, and then fusing the outputs together.

Their model reportedly obtained fairly successful results of >80% AUC on COVID-19 detection from sound samples.

Compared to our model, it is likely that the accuracy gap is caused by the greater amount of data which their model used, as our model only used a subset of the samples, and only used a single audio file per patient.

IV. Other Approaches

A. GRU based approach

This approach uses a hybrid CNN-GRU architecture. The main idea is to use the CNN layers to first extract audio features from the files, coupled with dropout layers to reduce overfitting. These extracted features are then passed to the GRU layer, which train to recognise patterns in the features across a time series.

Finally, the output from the GRU is piped to a series of Linear layers to consolidate and finally produce an output classification using Sigmoid activation and Softmax in the final output layer.

However, we did not end up using this model, as it suffered from modal collapse and seemed to over-fit even with several different hyperparameters, leading us to suspect that it was too complex for our limited dataset.

B. NN based approach

The second discarded approach is a neural network which consists of a sequence of fully connected layers. They are interconnected in a feedforward manner. The input layer of the model is defined by a Linear layer (nn.Linear) with 25 neurons, corresponding to the size of the input feature vectors. Each neuron in this layer represents a specific feature extracted from the respiratory data. These features correspond to the features outlined in the section above.

Following the input layer, the model is composed of hidden layers, each having a Linear layer followed by a rectified linear unit (ReLU) activation function. The choice of ReLU activation is because of its effectiveness in introducing non-linearity to the model, enabling it to learn complex patterns and relationships within the data. The number of neurons in each hidden layer progressively increases, allowing the model to capture hierarchical representations of the input features.

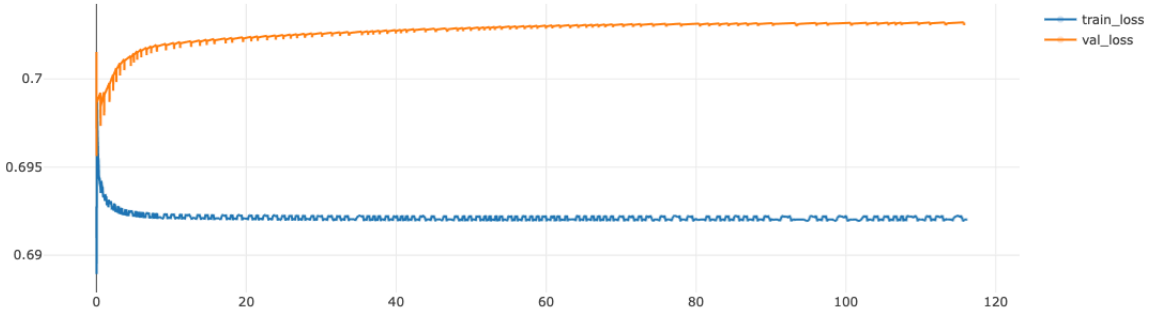
In the model architecture, the hidden layers consist of three successive Linear-ReLU blocks, with 64, 128, and 256 neurons, respectively. These layers serve as feature extractors, transforming the input features into higher-level representations. The final hidden layer of the model is followed by another Linear layer with 64 neurons. This layer acts as a bottleneck, reducing the dimensionality of the feature space and consolidating the learned representations before reaching the output layer. The output layer is composed of a Linear layer with 2 neurons and generates the final prediction label (“positive” or “negative”) for respiratory diagnoses. The output layer is activated using the sigmoid function, which squashes the raw scores to the range [0, 1]. This transforms the scores into probabilities, representing the likelihood of each diagnosis class.

The model was trained using PyTorch, with an Adam optimiser, and over 100 epochs with a batch size of 32, with a final accuracy of 82%. Grid search was implemented to tune the hyperparameters (learning rate, batch size, and epochs). Additionally, the mlflow library was used to keep track of loss and other metrics. The final model hyperparameters are as follows:

Learning rate: 0.008

Batch size: 32

Epochs: 100



As can be seen, the training loss and validation loss, while low, are diverging from each other before plateauing. This is suspected to happen because of the original imbalance in the dataset and the inability of the neural network to learn the pattern and generalise on unseen data.

V. Project Contribution

Lim Cheng Ee: CNN-LSTM approach

Javin Eng: GRU approach

Shwetha Iyer: NN approach, data pre-processing

VI. References

[1] Bhattacharya, Durgaprasad, et al. “Coswara: A Respiratory Sounds and Symptoms Dataset for Remote Screening of SARS-CoV-2 Infection.” Scientific Data, vol. 10, no. 1, June 2023, https://doi.org/10.1038/s41597-023-02266-0.