Automated COVID-19 Detection Using Deep Learning

Rodrigo Alarcon, Emma Conti, Lamine Deen,
Audrey Eley
Advisor: Dr. Zahra Nematzadeh

Goals and Motivation

The COVID-19 pandemic has underscored the need for effective and innovative diagnostic tools. A web-based application that analyzes cough audio to predict COVID-19 infection can offer a convenient and non-invasive screening method, potentially aiding early detection and reducing healthcare burdens. Although at-home testing for COVID-19 is effective, it can be expensive and inconvenient. Therefore, a convenient COVID-19 screening tool to assess the need for testing can help people make an informed decision.

Approach

- **Cough Recording**

The user can record their coughs and receive predictions on their COVID-19 infection status. This feature not only provides real-time feedback but also aids in maintaining a history of the user's infection status, making it a non-invasive and cost-effective tool for early screening. By tracking this data over time, the user can monitor their health status without the immediate need for a healthcare provider.

Approach (cont.)

- **Web App**

The web app design prioritizes ease of use, ensuring anyone can navigate it effortlessly and check their COVID-19 status at any time. Other aspects of the website will include details about the research and development of the ML model

Approach (cont.)

- **Week Long Progression Chart**

The user can view a week-long progress chart, making it easier to visualize changes in their infection status over time. This feature helps determine when a user is recovering and no longer symptomatic, or still infected. The user-friendly layout ensures that navigating the web app is straightforward.

Users can access their data at any time, providing continuous access to their COVID-19 status history. This eliminates the need to wait for a healthcare provider for early detection and offers users a convenient and effective way to monitor their health

Novel Features

- **Real time symptom checking:** Users can record an audio and check if they are symptomatic in seconds.

- **Week-long progress chart:** Users can track their infection status over the course of a week.

- **Symptom monitoring**: The chart helps users monitor changes in symptoms and status over time.

- **Early diagnostic tool:** It encourages users to get tested and seek professional medical care if necessary.

Technical Challenges

- **Web app:**: The team needs to acquire full-stack development skills to integrate the CNN model into a user-friendly web app with real-time functionality.

- **Feature engineering and data augmentation**: converting data into the right format for the CNN and applying data augmentation techniques to improve model robustness.

- **Model architecture selection**: Researching and choosing the optimal CNN architecture, such as ResNet or MobileNet, is crucial for balancing accuracy, speed, and computational efficiency in real-time detection.

- **Latency and real-time processing**: To meet real-time performance requirements, the CNN must be optimized for low-latency predictions through techniques like model compression or efficient inference frameworks.

- **Generalization and robustness:** The model needs to generalize well across diverse environments, audio qualities, and user conditions, which requires careful tuning to minimize false positives and negatives in real-world scenarios.

Tools Needed

Convolutional Neural Network (CNN)
- A neural network architecture with the potential for good performance for this classification task. This intakes cough audio as input and outputs a COVID-19 infection prediction.

ResNet50
- A 50 layer deep convolutional neural network used to classify images. For our purposes, this will be used as a benchmark model.

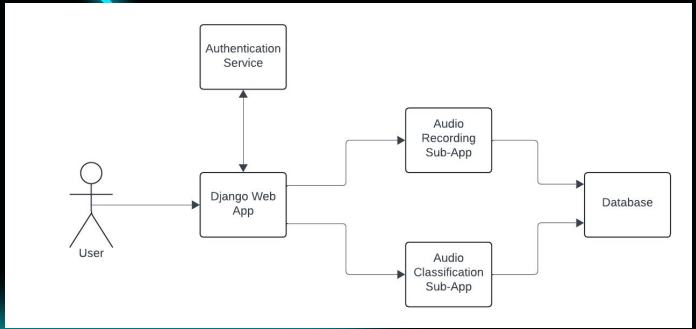
Django
- A web development framework. This handles the back end of the website and data transference.

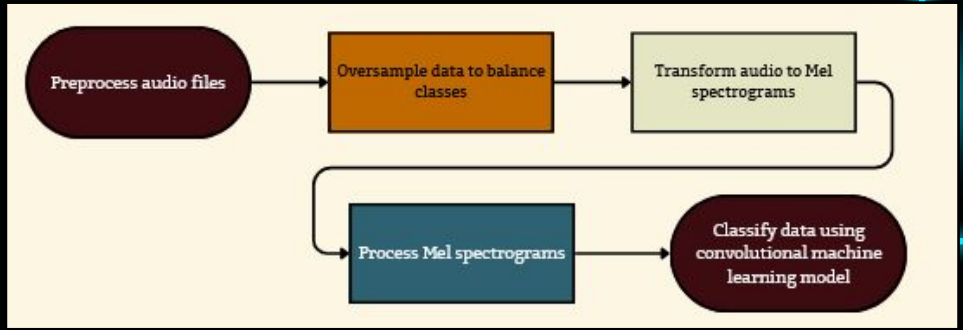
TensorFlow, Numpy, Pandas
- Common python frameworks and tools that help in manipulating data and creating the neural network model.

Librosa
- Python package used to manipulate audio samples of coughing.

Design

- WebApp will be integrated with the CNN in order to provide results based on a trained CNN model used to identify whether or not a cough shows signs of COVID-19.
- Users shall be able to access the web app to view their previous recordings and log their symptoms. These results will be graphed and shown in a usable format for the user to view their current health status.
- The WebApp shall have a recording feature implemented to allow users to record their own coughs for immediate testing.
- The WebApp shall have an audio recording uploading feature implemented to allow users to record their own coughs for retroactive testing.





Evaluation

Speed: The dataset has been pre-processed to include an equal amount of cases for each type of cough that will be present during testing.

Accuracy: The level of accuracy achieved in current CNN testing is not ideal, but should improve now that the dataset has been cleaned.

Reliability: The WebApp runs reliably, and as the CNN continues to be implemented

User Survey: Users are currently able to access our website and are able to see up to date information on the research we have done and will continue to do throughout the project. Users are able to create an account that will direct them to the screen that

Progress Summary

Module/feature	Completion %	To do
GUI	70%	allowing drop down menus, mouse-over help messages...
WebApp Home Screen	90%	Still finalizing design. Content has been completed.
WebApp Login Screen	80%	Looking for a more secure way to store usernames and passwords, otherwise the login and account creation is completely functional
WebApp sickness tracking	20%	Deciding on formatting for data to be displayed - Current status - Current recommendations - Previously logged symptoms
CNN	40%	Results are skewed due to issues with dataset, testing set to continue once dataset is cleaned
ResNet50 CNN model	40%	Results are skewed due to issues with dataset, testing set to continue once dataset is cleaned
WebApp recording feature	30%	Basic recording function works on local copy. Troubleshooting needed to record on live webapp

Milestone 4

FEB 24 — Finish cleaning data

FEB 24 — ML testing and refinement of framework

FEB 24 — Web testing

FEB 24 — Integration of CNN on WebApp

Milestone 5

MAR 26 — ML testing and refinement of framework

MAR 26 — Web testing

MAR 26 — Integration of CNN on WebApp

Milestone 6

APR 21

Finish ML testing

APR 21

Finish web testing

APR 21

Finalize integration of CNN and Webapp

Task Matrix

Task	Rodrigo	Emma	Lamine	Audrey
1. Finish cleaning data	25%	25%	25%	25%
1. ML testing	0%	50%	50%	0%
1. Continued refinement of ML workflow	10%	40%	40%	10%
1. Web testing	50%	0%	0%	50%
1. Continued integration of WebApp and CNN	40%	10%	10%	40%

Questions?