Automated COVID-19 Detection Using Deep Learning

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

Goals and Motivation

The COVID-19 pandemic has highlighted the demand for innovative diagnostic tools. A web-based app that analyzes cough audio to detect flu-like symptoms could provide a convenient, non-invasive screening method, helping with early detection and easing the strain on healthcare systems.

Approach

- **Early Diagnosis Tool**

The app enables users to record their coughs and receive real-time predictions regarding their COVID-19 infection status. By acting as an early diagnostic tool, it allows for non-invasive and cost-effective monitoring, helping users identify potential symptoms before needing immediate medical intervention. Although it provides real-time insights, it encourages users to take a COVID-19 rapid test and seek professional care when necessary.

Approach (cont.)

- **Progression Tracking**

A key feature of the app is the week-long progress chart, which allows users to easily track changes in their infection status over time. This visual representation helps users monitor their health, making it easier to see if they are recovering, still symptomatic, or possibly infected. This functionality provides a clear view of the user’s current and projected condition across several days, offering valuable insights into their health trajectory.

Approach (cont.)

- **Ease of Use and Security**

The app is designed with user-friendliness in mind, allowing users to effortlessly navigate and check their symptoms at any time. Ensuring smooth functionality for all users, the interface is simple and intuitive. Moreover, strict data security measures are implemented and user can download their data anytime they need to.

Approach (cont.)

- Stay Healthy and Avoid Severe Symptoms

By helping users monitor their symptoms and infection status, the app plays a crucial role in promoting health awareness. It enables users to stay proactive about their well-being, providing early warning signs that could prevent the development of severe symptoms. Through timely prompts to get tested and seek medical care, the app supports users in staying healthy.

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

Frontend:

- User interface (UI) design

- Web framework (e.g., React, Vue.js)

- Real-time audio recording functionality

- Progress tracking and visualization (charts, graphs)

Backend:

- Web server (e.g., Node.js, Django)

- API for real-time interaction between frontend and backend

- Integration with the CNN model for processing requests

- User authentication and session management

Tools Needed

Model:

- Convolutional Neural Network (CNN) for audio processing

- Feature engineering and preprocessing pipeline

- Data augmentation techniques

- Deployment framework (e.g., TensorFlow Lite, ONNX)

- Optimization for real-time inference (e.g., model pruning, quantization)

Database:

- User data storage (e.g., user profiles, history of predictions)

- Database for storing progress charts and infection status logs (e.g., SQL, NoSQL)

Tools Needed

Infrastructure:

- Load balancing and scalability tools to handle multiple requests in real-time

- Real-time performance monitoring (e.g., latency, model accuracy)

- Automated testing (for frontend, backend, and model accuracy)

Security:

- Data encryption (for audio data and user information)

- Secure authentication protocols (e.g., OAuth)

Milestone 1

SEPT 30	Pick web framework
SEPT 30	Pick ML framework
SEPT 30	Research Audio classification
SEPT 30	Design ML workflow

Milestone 1 (cont.)

SEPT 30 — Design web workflow and high level design

SEPT 30 — Develop a testing plan for ML

SEPT 30 — Develop a testing plan for web

SEPT 30 — Becoming familiar with CNN and RNN

Milestone 1 (cont.)

SEPT 30 — Becoming familiar with web-dev

Milestone 2

OCT 28	Refine ML workflow
OCT 28	Begin feature engineering on dataset
OCT 28	Begin working on web framework front end
OCT 28	Begin working on web framework back end

Milestone 3

NOV 25	Begin ML Testing
NOV 25	Begin web testing
NOV 25	Integrating base ML model with web using a Neural Network framework

Task Matrix

Task	***Rodrigo***	***Emma***	***Lamine***
Pick web framework	Pick web framework	Learn about selected framework	Learn about selected framework
Pick ML framework	Group decision for base framework	Group decision for base framework	Group decision for base framework
Become familiar with ML	Learn and familiarize with ML	Learn and familiarize with ML	Learn and familiarize with ML
Become familiar with web-dev	Research web-dev	Research web-dev	Research web-dev
Research sound classification	Research sound classification	Research sound classification	Research sound classification
Required research	Different alternatives for audio as input data	Different audio to image conversion techniques	Feature engineering
Design ML workflow (beginning to end)	Design 25%	Design 50%	Design 25%
User interaction (SSD)	Develop 25%	Develop 25%	Develop 50%
Develop a testing plan for ML and web	Develop 33%	Develop 33%	Develop 33%

Questions?