Automated COVID-19 Detection Using Deep Learning

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

Task Matrix

Task	Completion %	Rodrigo	Emma	Lamine	To Do
1. Pick web framework	100%	60%	20%	20%	Continue working with Django
2. Pick ML framework	100%	25%	25%	50%	
3. Become familiar with ML	90%	33%	33%	33%	Continue research
4. Become familiar with web-dev	90%	60%	20%	20%	Continue working with Django
5. Research sound classification	90%	40%	40%	20%	Continue research, test alternate methods
6. Required research	90%	20%	60%	20%	Continue research
7. Design ML workflow (beginning to end)	100%	15%	60%	25%	
8. User interaction (SSD)	100%	15%	15%	70%	
9. Develop a testing plan for ML	100%	33%	33%	33%	
10. Develop a testing plan for Web	90%	60%	20%	20%	Finalize testing plan
11. Requirements Gathering	90%	10%	10%	80%	

Task 1 - Web Framework

- Selected Django as the web framework
- Versatility in both frontend and backend development
- Rapid development process
- Extensive documentation

Task 2 - ML Framework

- Selected PyTorch as the ML framework for developing our custom CNN.

- Chose PyTorch for its dynamic computation graph and ease of building/modifying neural networks.

- PyTorch offers robust GPU acceleration, essential for efficient training of deep learning models.

- Overcame the challenge of choosing between PyTorch and other frameworks like TensorFlow/Keras; PyTorch's flexibility and documentation made it ideal for us.

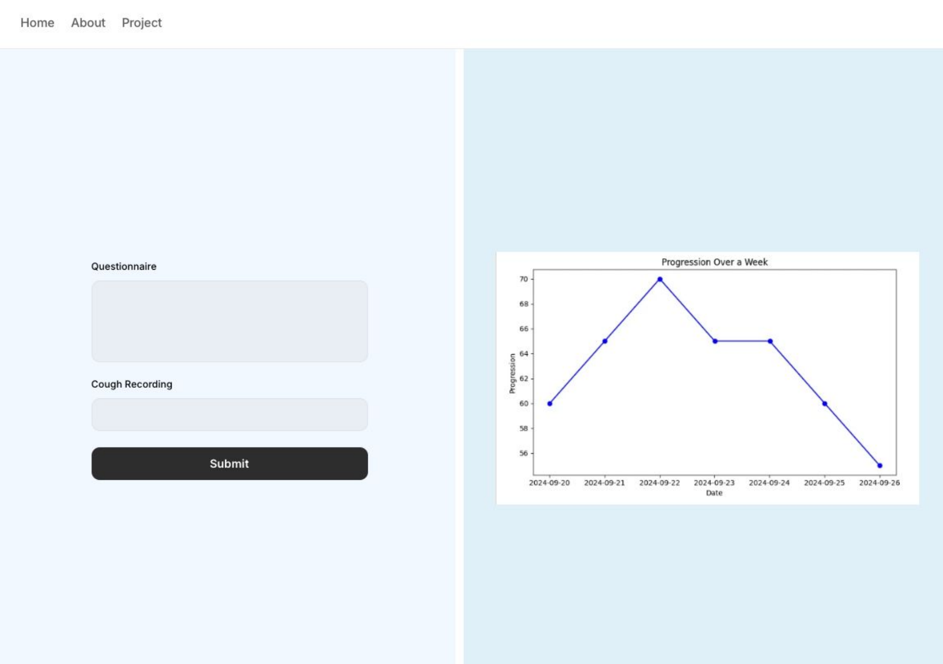
Task 3 - Familiarity with ML

- Learning about CNNs and how to develop a CNN workflow for our project
- Researching sound classification
- Researching other CNN models to see what works for sound classification

Task 4 - Familiarity with Web Development

- Focused on familiarizing with Django's structure and core features
- Reviewed documentation and tutorials to understand inner workings
- Building small demo projects to understand MVC and template rendering
- Nexts steps include working on initial web components and developing a basic prototype for further integration with ML model

Mock up


Home
About
Project
Questionnaire
Cough Recording
Submit
Progression Over a Week
Progression
70
68
66
64
62
60
58
56
2024-09-20
2024-09-21
2024-09-22
2024-09-23
2024-09-24
2024-09-25
2024-09-26
Date

Task 5 - Research Sound Classification

Determining how to analyze the sound data

Spectrograms is the best option for this project, especially in conjunction with a CNN

K. Zaman, M. Sah, C. Direkoglu and M. Unoki, "A Survey of Audio Classification Using Deep Learning," in IEEE Access, vol. 11, pp. 106620-106649, 2023, doi: 10.1109/ACCESS.2023.3318015.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10422379/

Task 6 - Research Alternatives for ML

An RNN model would also be suitable for sound classification, but would not be better than a CNN model for cough classification

Researching a potential RNN-CNN hybrid model to see if that would be ideal for this particular dataset, otherwise CNN is the best option

Task 7 - Design ML Framework

- Outlined the complete ML workflow from data collection to model evaluation.

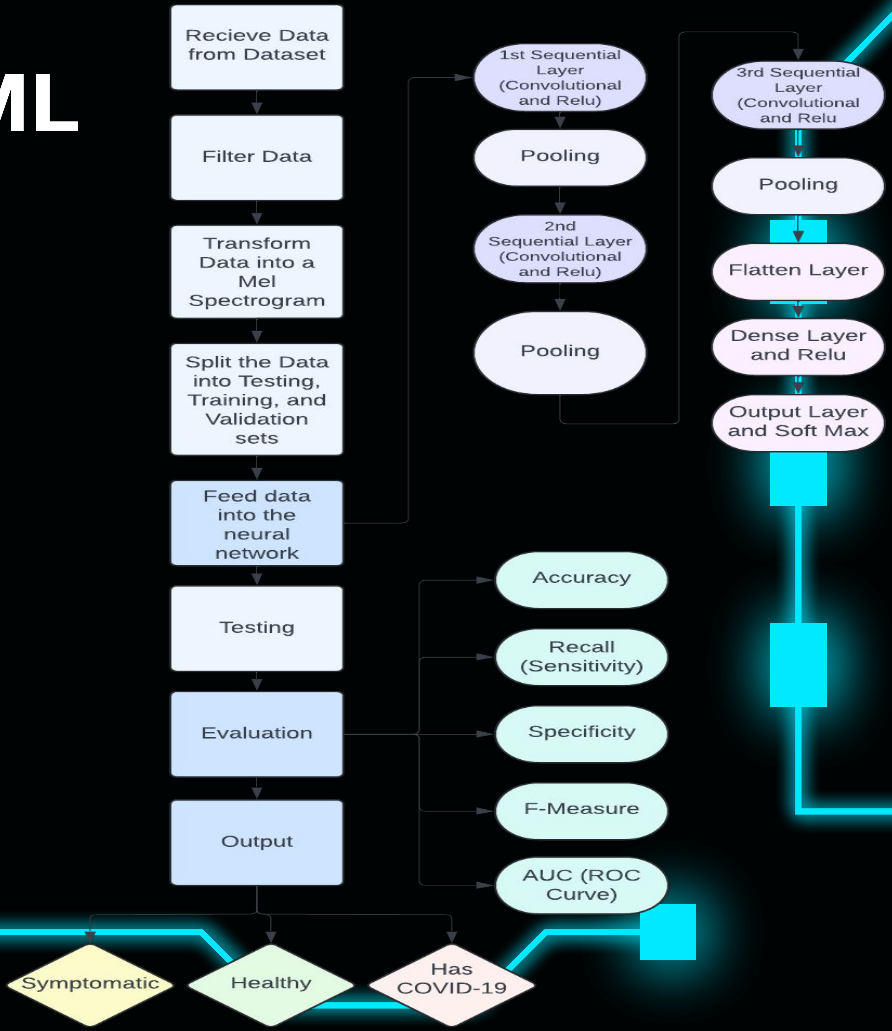
- Use LibROSA and Pandas to preprocess audio files into mel spectrograms for CNN input.

- Implement simple data augmentation techniques: noise addition, pitch shifting, and time-stretching.

- Plan to compare our custom PyTorch CNN against three benchmark pretrained models to evaluate performance.

Task 7 - Design ML Workflow

- 3 convolutional layers
- 5 primary evaluation points
- 3 possible outputs


Recieve Data from Dataset
Filter Data
Transform Data into a Mel Spectrogram
Split the Data into Testing, Training, and Validation sets
Feed data into the neural network
Testing
Evaluation
Output
1st Sequential Layer (Convolutional and Relu)
Pooling
2nd Sequential Layer (Convolutional and Relu)
Pooling
3rd Sequential Layer (Convolutional and Relu
Pooling
Flatten Layer
Dense Layer and Relu
Output Layer and Soft Max
Accuracy
Recall (Sensitivity)
Specificity
F-Measure
AUC (ROC Curve)
Symptomatic
Healthy
Has COVID-19

Task 8 - User Interactions

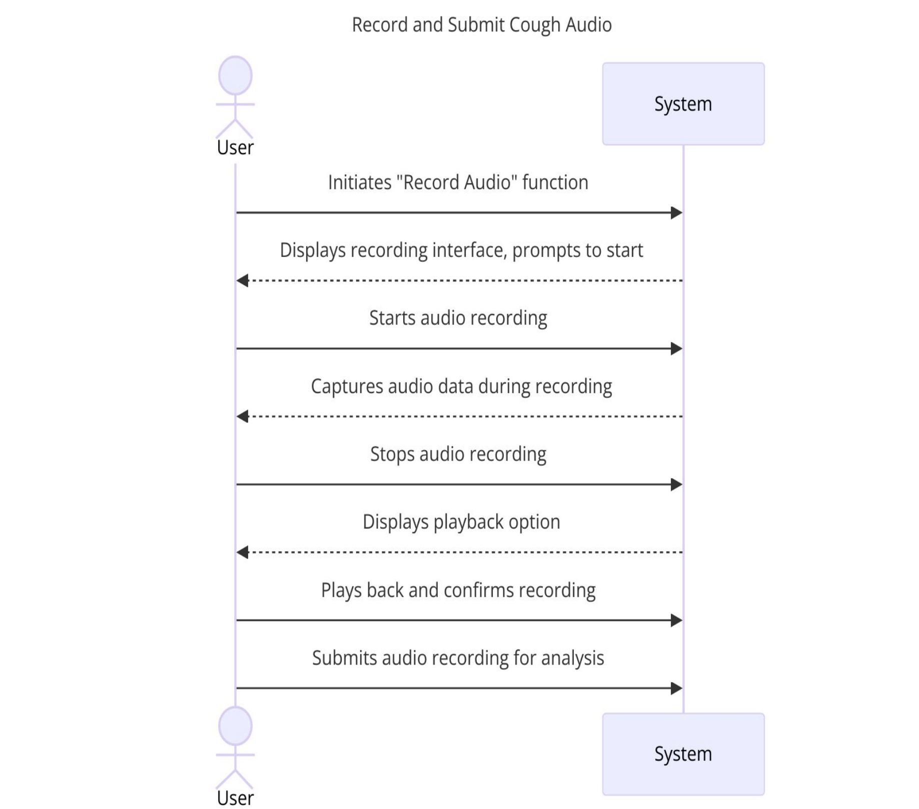
- Created a System Sequence Diagram (SSD) to map out user interactions with the system.

- Defined user actions: record cough, submit audio, receive health prediction, and view health progression.

- Addressed the obstacle of translating complex backend processes into an intuitive user interface.

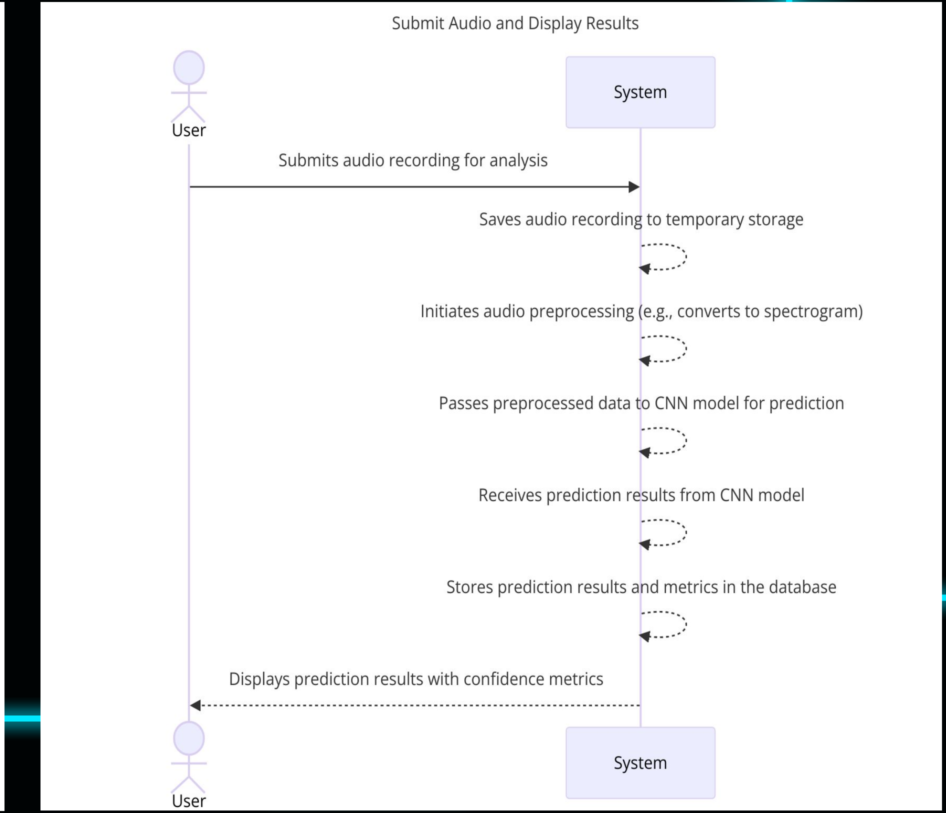
- Balanced displaying essential information (e.g., prediction confidence metrics) with maintaining a user-friendly design.

SSDs



Record and Submit Cough Audio

User
System
Initiates "Record Audio" function
Displays recording interface, prompts to start
Starts audio recording
Captures audio data during recording
Stops audio recording
Displays playback option
Plays back and confirms recording
Submits audio recording for analysis
User
System



Submit Audio and Display Results

User
System
Submits audio recording for analysis
Saves audio recording to temporary storage
Initiates audio preprocessing (e.g., converts to spectrogram)
Passes preprocessed data to CNN model for prediction
Receives prediction results from CNN model
Stores prediction results and metrics in the database
Displays prediction results with confidence metrics
User
System

Task 9 - ML Testing Plan

Of the 3000 usable test files we have in our dataset:

50% training

25% validation

25% testing

Task 10 - Web App Testing Plan

- Developed a preliminary testing plan for web functionality
- Will focus on compatibility testing across devices and browsers
- Also test user authentication and data retrieval

Task 11 - User Requirements Gathering

- Conducted comprehensive requirements gathering to define project scope and objectives.

- Identified key stakeholders: end-users, sponsoring professor, medical professionals, and the development team.

- Specified business, functional, and non-functional requirements (e.g., real-time predictions, security measures, scalability).

- Faced the challenge of ensuring compliance with health data privacy laws (HIPAA) adding complexity to requirements.

Task Matrix: Milestone 2

Task	Rodrigo	Emma	Lamine
1. Refine ML Workflow	Finalize research to determine best path forward	Finalize research to determine best path forward	Finalize research to determine best path forward
2. Begin Feature Engineering on Dataset	Learn about selected features	Learn about selected features	Determine feature engineering on dataset
3. Begin Working on WebFramework Frontend	Integrate and test necessary front end components	Integrate and test necessary front end components	Familiarize with web dev
4. Begin Working on Web Framework Back End	Design 50%	Design 30%	Design 20%
5. Pick 3 benchmark models	Familiarize and select benchmark CNN models	Familiarize and select benchmark CNN models	Familiarize and select benchmark CNN models

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

Questions?