1. Covid19 Audio Cough Classification
   Emma Conti - econti2020@my.fit.edu
   Lamine Deen - ldeen2016@my.fit.edu
   Rodrigo Alarcon - ralarcon2019@my.fit.edu
   Audrey Eley - aeley2020@my.fit.edu
2. Faculty Advisor: Dr. Nematzadeh Zahra znematzadeh@fit.edu
3. Client: Dr. Nematzadeh Zahra, professor at Florida Tech

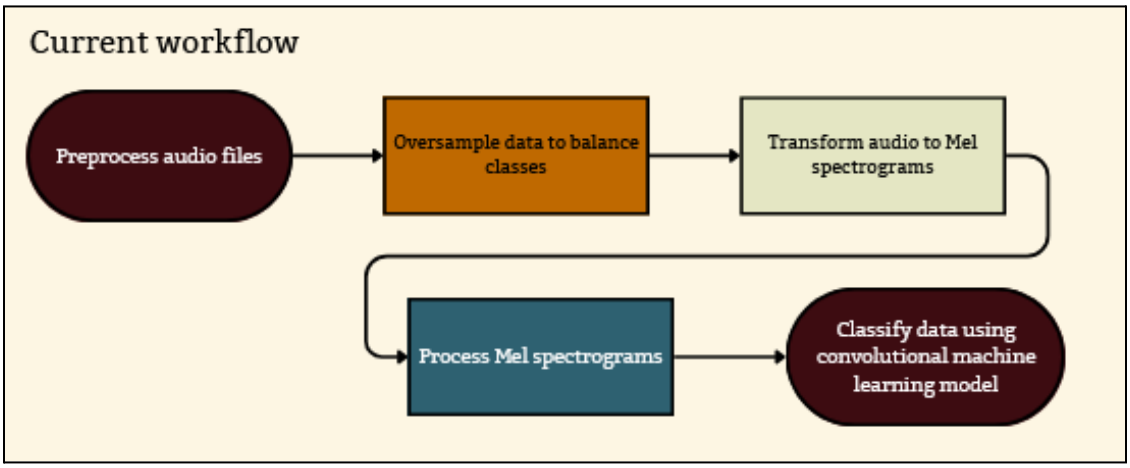
4. **Milestone 2 Matrix**

Task	Completion %	Rodrigo	Emma	Lamine	Audrey	To Do
1. Refine ML Workflow	100%	15%	15%	35%	35%	Nothing to refine as of yet due to issues with dataset
2. Begin Feature Engineering on Dataset	100%	5%	5%	85%	5%	First set of features selected for initial tests, accuracy of 39%
3. Begin Working on Web Framework Frontend	80%	85%	5%	5%	5%	Complete layout including home page. Add an additional page to present ML model
4. Begin Working on Web Framework Backend	80%	85%	5%	5%	5%	Add additional fields to User DB and incorporate with account. Change
5. Pick benchmark model	100	10%	70%	10%	10%	Testing to be completed in next section.

5. Discussion (at least a few sentences, ie a paragraph) of each accomplished task (and obstacles) for the current Milestone:

- Task 1:

Refining the machine learning workflow will be an iterative process that will continue until a sufficiently accurate model is found. For this stage of the project, refining the machine learning workflow entailed researching potential workflows, defining an initial workflow, and making minor adjustments to enhance the initial workflow. Looking at previous works that use ML to diagnose COVID-19 using cough audio, an effective strategy seems to be converting audio files to images and classifying using a convolutional neural network. In this strategy, results indicate that it is most effective to convert audio to mel spectrograms. For our initial model, we decided to use a workflow in which we convert audio data to mel spectrograms, and use a CNN to classify. This workflow was chosen because we believe that CNNs take advantage of the spatial information available to us, and have the potential to be highly accurate. This workflow was further refined by preprocessing the audio files, using oversampling techniques on the audio files, and processing the mel spectrograms once they are created.



Current machine learning workflow

- Task 2:

Choice of model:
The choice of a Convolutional Neural Network (CNN) for cough audio classification is driven by its effectiveness in capturing spatial hierarchies, particularly in image-like data representations like Mel spectrograms. Here, the audio data is transformed into Mel spectrograms, which effectively capture frequency and time features in a 2D format suitable for CNN processing. CNNs, with their ability to detect local patterns through convolutional filters, are well-suited to identify key cough sound features such as changes in frequency and amplitude. This simple CNN model architecture consists of a convolutional layer followed by max pooling, which helps reduce dimensionality while preserving critical features. A fully connected layer at the end allows classification across three cough-related classes.

1. Exploration: This analysis begins by loading the dataset, to analyze basic information such as missing data and inconsistencies. The exploration phase includes checking for discrepancies between the metadata and actual audio files within the dataset. Specifically, it identifies and counts extra files in the archive folder that aren't present in the metadata and vice versa. This process cleans the dataset by removing rows associated with missing audio files to ensure only valid entries are retained.
2. Augmentation: This file focuses on audio data augmentation to increase dataset variability, enhancing model generalization. Initially, the file converts .webm files to .wav format for compatibility with audio processing libraries like Librosa. Augmentation techniques applied include noise addition, pitch shifting (both up and down), and time stretching (both elongated and shortened). Each augmentation produces a new audio file, creating multiple variants for each original audio file.
3. Transformation: This file transforms the augmented audio files into Mel spectrogram images. It utilizes Torchaudio to load .wav files, resampling them if needed, and then generates Mel spectrograms with specified parameters (sample rate, FFT size, hop length, and Mel bins). Each spectrogram is converted to decibel scale for visualization clarity, plotted, and saved as an image in the output directory. The file concludes with cropping each spectrogram image, and saving them, preparing the data for CNN input.

CNN Architecture

Layer	Type	Configuration
Convolutional Layer 1	Conv2d	3 input channels, 10 filters, 3x3 kernel, padding=1
Activation Function	ReLU	Applied after conv1
Pooling Layer	MaxPool2d	2x2 kernel, stride=2
Flatten	Reshape	Converts feature map into a 1D vector
Fully Connected Layer 1	Linear	Input: 217280 neurons, Output: 3 classes

Model Hyperparameters

Hyperparameter	Value
Loss Function	CrossEntropyLoss
Optimizer	SGD
Learning Rate	0.01
Number of Classes	3
Input Size (Spectrogram)	224 x 97
Batch Size	64

Mel Spectrograms Hyperparameters

Parameter	Value
Sample Rate	16000
FFT Size (n_fft)	1024
Hop Length	512
Number of Mel Bands (n_mels)	128

Data Augmentation

Techniques	Description
Noise Addition	Adds realistic noise to the raw waveform, enhancing data variability without introducing unnatural patterns.
Pitch Shifting	Shifts the pitch of the raw waveform to simulate different vocal characteristics, preserving natural time-frequency structure.
Time-Stretching	Alters the speed of the raw waveform to create varied temporal representations, maintaining a realistic time axis in the spectrogram.

Table of Results

Experiment	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Best Epoch	Training Runtime (min)
Benchmark	1.03	48.06%	1.07	41.64%	4	0.06
He Initialization	1.00	51.99%	1.10	38.66%	4	0.05
Xavier Initialization	1.04	47.32%	1.07	38.51%	4	0.06
DeepNet	1.00	51.79%	1.05	44.18%	15	0.22
WideNet	0.95	53.93%	1.05	44.48%	6	0.17
ELU Activation	0.99	53.13%	1.07	42.54%	4	0.06
Swish Activation	1.03	46.97%	1.07	42.54%	3	0.06
Focal Loss	0.45	49.25%	0.47	41.34%	4	0.07
Label Smoothing	1.05	47.61%	1.08	42.39%	4	0.06
Dropout Regularization	1.03	48.61%	1.07	41.64%	4	0.05
L2 Regularization	1.03	48.06%	1.07	41.64%	4	0.05
Momentum Optimization	0.84	63.57%	1.20	42.09%	16	0.13

- Task 3:

An initial website has been created that presents general information as well as the project documentation and milestones to users. This includes a main layout, the nav bar, and placeholder sections for additional information and the ML model. The layout is so far responsive to different screen sizes with some minor tweaks to be made. With the Django structure, all pages of the website are consistent and follow the same design rules. The Django structure also helps with adding additional information easily without much work.

- Task 4:

The initial website also includes a user model on top of the general front-end work. This model extends directly from Django so it may be changed in the future to allow for additional user information to be saved. Currently, users can create new accounts, log in, and log out securely using Django's authentication protocol. This may also get updated in the future to allow for more secure user authentication. Additionally, as part of the backend, each link routes URL's accordingly to lead to the correct page.

- Task 5:

Initial tests with our CNN had an accuracy of approximately 40%. ResNet 50 is another CNN with test cases for Mel Spectrograms with much higher accuracy levels. Using ResNet 50 as a benchmark test will allow us to compare our results to a pre-made CNN's analysis of our dataset. This will better determine where our CNN is lacking, and help us to better choose the features needed for our CNN. ResNet 50 was especially successful in using Mel Spectrograms, the same format we will be using for our dataset, to analyze theirs. Their features were tailored to analyzing and finding patterns in the Mel Spectrograms of the Tuberculosis cough recordings. This makes ResNet 50 especially well-suited for extracting patterns from visual representations of audio. ResNet 50 was not the only benchmark CNN considered, the others include ResNet 14, VGG 16, an Inception v4. ResNet 14 was a CNN we did initial tests with for out dataset, and that also yielded low results with an accuracy below 50%.

References

Awan, Mazhar & Rahim, Mohd & Salim, Naomie & Mohammed, Mazin & Zapirain, Begoña & Abdulkareem, Karrar. (2021). Efficient Detection of Knee Anterior Cruciate Ligament from Magnetic Resonance Imaging Using Deep Learning Approach. Diagnostics. 11. 105. 10.3390/diagnostics11010105.

Troy, Mazerolle. "Using VGG16 to Classify Spectrograms." *Kaggle*, Kaggle, 22 Dec. 2023, www.kaggle.com/code/troymazerolle/using-vgg16-to-classify-spectrograms.

Yadav, Jyoti & Varde, Aparna & Liu, Hao & Antoniou, George & Xie, Lei. (2024). Audiovisual Multimodal Cough Data Analysis for Tuberculosis Detection.

6. Discussion (at least a few sentences, ie a paragraph) of contribution of each team member to the current Milestone:

Rodrigo Alarcon

I have worked on developing the front and back end of the website using Django which is primarily written in python with some template work done in HTML and CSS. This includes the page for the project documentation with working download links. The home page has a current placeholder template and the user pages allow for creating an account and signing in. The links actively route to the correct pages using Django's URL dispatcher. Additional template pages have also been created to begin working on a ML model integration. With the Django structure, different apps have been created within the primary project, with each app controlling a separate function of the website. The main app displays the general information and project documentation. The user app control user login/logout functions as well as creating accounts. This will be extended to contain additional user information and covid related information. The last app will be the ML app which will delegate ML model specific functions.

Emma Conti

I have been acquainting myself with both web design and Django specifically as I prepare to have a larger hand in the web design aspects of this project in future Milestones.

A majority of my research has been in determining which CNNs would be good benchmark tests for our project to help determine if our model is accurate, and what specific

areas our model is lacking. The determining factor for using VGG 16, besides it being a well known CNN, was the presence of research on using VGG 16 with mel spectrograms. Because we are testing audio files, but processing them as mel spectrograms, it is not the same as image classification, which is what VGG was made for. There is a dataset showing VGG 16 classifying mel spectrograms based on what genre of music they are, proving it can effectively be used for audio tests (accuracy of 90%). The determining factor for using ResNet 50 as opposed to 14 was its previous use cases in medical datasets. ResNet 14 has medical applications, but those were all specifically image based. Initial testing on the dataset with ResNet 14 yielding a result of approximately 40% accuracy. As opposed to ResNet 50 which has applications for mel spectrograms for coughs specifically which has accuracy results of 87%. Datasets for cough audio detection for tuberculosis have been developed, which should make tailoring our CNN much easier when comparing to test results from ResNet 50. The applications for our CNN will ultimately fall within the medical field, and therefore having a test set with a medical application to determine where we can strengthen our own CNN will be incredibly beneficial. Finally, Inception v4 was chosen because of the Soundception test set developed. Soundception was developed specifically to identify complex soundscapes, specifically bird noises with an accuracy of 71%. With minute differences between bird noises, there may be ways to relate the Inception v4 model to our CNN to help with differentiation between complex sounds in the form of coughs.

References

Awan, Mazhar & Rahim, Mohd & Salim, Naomie & Mohammed, Mazin & Zapirain, Begoña & Abdulkareem, Karrar. (2021). Efficient Detection of Knee Anterior Cruciate Ligament from Magnetic Resonance Imaging Using Deep Learning Approach. Diagnostics. 11. 105. 10.3390/diagnostics11010105.

Troy, Mazerolle. “Using VGG16 to Classify Spectrograms.” *Kaggle*, Kaggle, 22 Dec. 2023, www.kaggle.com/code/troymazerolle/using-vgg16-to-classify-spectrograms.

Yadav, Jyoti & Varde, Aparna & Liu, Hao & Antoniou, George & Xie, Lei. (2024). Audiovisual Multimodal Cough Data Analysis for Tuberculosis Detection.

Lamine Deen

In the report notebook, I explored the dataset, identified and filtered out incomplete entries, and augmented the audio files. Augmentation involved converting files to `.wav` format and applying transformations like noise addition, pitch shifts, and time stretching. The augmented data was then transformed into mel spectrogram images, which were cropped to remove unnecessary borders.

Next, I established a benchmark CNN model and tested variations, including He and Xavier initialization for gradient stability, deeper and wider architectures for complex feature extraction, and activation functions like ELU and Swish. I also experimented with Focal Loss and Label Smoothing to address class imbalances, Dropout and L2 regularization to prevent overfitting, and Momentum for faster convergence. Each approach offered insights into optimizing training, accuracy, and model generalization.

Audrey Eley

Being a recent addition to this team, I have been orienting myself in the project and researching machine learning techniques for audio classification. This has included reviewing project documentation that was written prior to me joining the team, and reviewing articles and scholarly works regarding this topic. I have found some previous works which used machine learning to diagnose COVID-19 using cough audio that achieved highly accurate results. I have been researching the techniques used in these successful studies. In future milestones, I would like to try some of these techniques for preprocessing the audio files.

7. Plan for the next Milestone 3 (Task Matrix)

Task	Rodrigo	Emma	Lamine	Audrey
1. Begin ML Testing	Test using 3 chosen benchmark models and initial testing from our model.			
2. Refine ML Workflow	Continue to improve the ML model. Determine which improvement strategies to implement based on testing results.			
3. Begin Web Testing	Begin implementing a framework for users to access the CNN and upload their coughs.			
4. Integrating Base ML Model with Web Using a Neural Network Framework	Determine how successfully and efficiently the two can be integrated, and what may need to change within the web framework to better accommodate and suit the CNN.			

8. Discussion (at least a few sentences, ie a paragraph) of each planned task for the next Milestone or

- Task 1:
  - From this point, machine learning testing will begin on the prepared cough audio data by first building a base CNN model and then iteratively enhancing it to improve accuracy. The process will involve experimenting with normalization and standardization techniques to ensure that the input data is properly scaled, allowing the model to converge faster and perform better. Various weight initialization strategies, such as Xavier or He initialization, will be tested to ensure appropriate starting values for the weights, avoiding problems like vanishing or exploding gradients. Different architectures will be explored, beginning with simple CNN layers and gradually increasing complexity, adding more convolutional layers, batch normalization, or residual connections to capture deeper patterns in the mel spectrograms. Experiments with different activation functions, such as ReLU, Leaky ReLU, or Swish, will identify the most suitable nonlinear transformations for the task. The loss function will be fine-tuned, starting with categorical cross-entropy for classification, with the potential to implement custom loss functions tailored to cough detection. To prevent overfitting, regularization methods like L2 weight decay will be introduced, along with dropout to randomly deactivate neurons during training, encouraging better generalization. By systematically testing these enhancements, the model's performance will be gradually optimized, aiming for improved accuracy in COVID-19 detection.
- Task 2:
  - The experimentation and testing taking place in Task 1 will serve as the basis for this milestone's iteration of ML refinement. This will involve evaluating the

performance preprocessing techniques, weight initializations, CNN architectures, and hyperparameters tested in Task 1. Based on the performance of these relative to their counterparts in the base model that we are currently working with, we will continue to revise our workflow to improve model precision.

        - Task 3:
    - As the project progresses, the website will continue to be developed to make it more user friendly, as well as to integrate the necessary components to allow users to record and upload their recordings to then be evaluated by the CNN. For the next milestone, the primary changes being made to update the website will be: user authentication, the main page, the addition of the research page, and the development of the actual user interface for when users access their cough information. The user authentication will be reviewed so it can be made more secure, especially because medical information is being used to determine if a cough is COVID-19. The main access page will also be added in order to allow for a user to more easily understand what the web app is intended for, and to better allow them to navigate. The research page will go more in depth about why the CNN we are using is important, and how it has been developed. This will be separate from the milestone pages related to the senior design project to allow for users to easily access and read through this information outside of the context of senior design. The next milestone is when the CNN will be first integrated with the website, so updating the user interface where they are uploading their coughs to be evaluated is a necessary component of the next milestone. Ensuring that it is both easy to navigate and can be used successfully are the two initial primary goals. Finally, a more detailed web testing plan will be developed and implemented over the course of Milestone 3.
        - Task 4:
    - During the next milestone, we will begin work to integrate the machine learning and web components of the project. This may require changes to existing version of the web framework. We may need to adopt frameworks or libraries, such as TensorFlow.js, to deploy our ML model to the web. For this task, we will research integration strategies and choose one to move forward with. This task will lead to the full integration of web and ML components in subsequent milestones. The integration begun here will be using our base ML model. This version of the model will be improved upon moving forward, so this task may be revisited to properly integrate new versions of the model.

9. Meeting Date: October 24th, 2024

10. See Faculty Advisor Feedback Below

11. Meeting Date: October 24th, 2024

12. Faculty Advisor feedback on each task for the current Milestone 2
    - Task 1:
    - Task 2:
    - Task 3:
    - Task 4:
    - Task 5:

Faculty Advisor Signature: _____ Date: _____

<u>Evaluation by Faculty Advisor</u>

Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu

Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Rodrigo	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Emma	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Lamine	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Audrey	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: _____ Date: _____