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 3 Matrix

Task	Rodrigo	Emma	Lamine	Audrey
1. Begin ML Testing	Test using benchmark model (ResNet50) 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.			

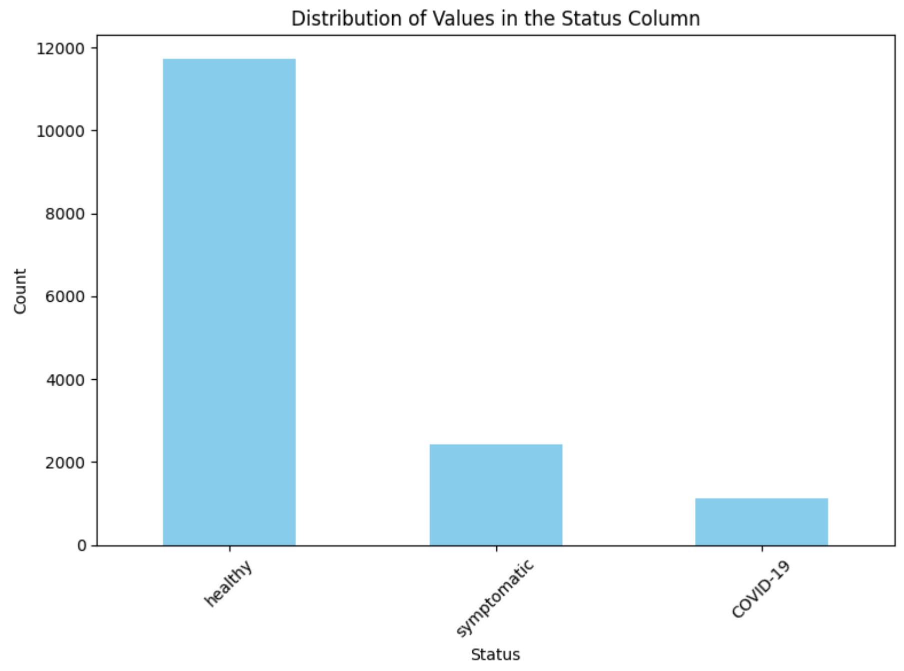
5. Discussion (at least a few sentences, ie a paragraph) of each accomplished task (and obstacles) for the current Milestone:

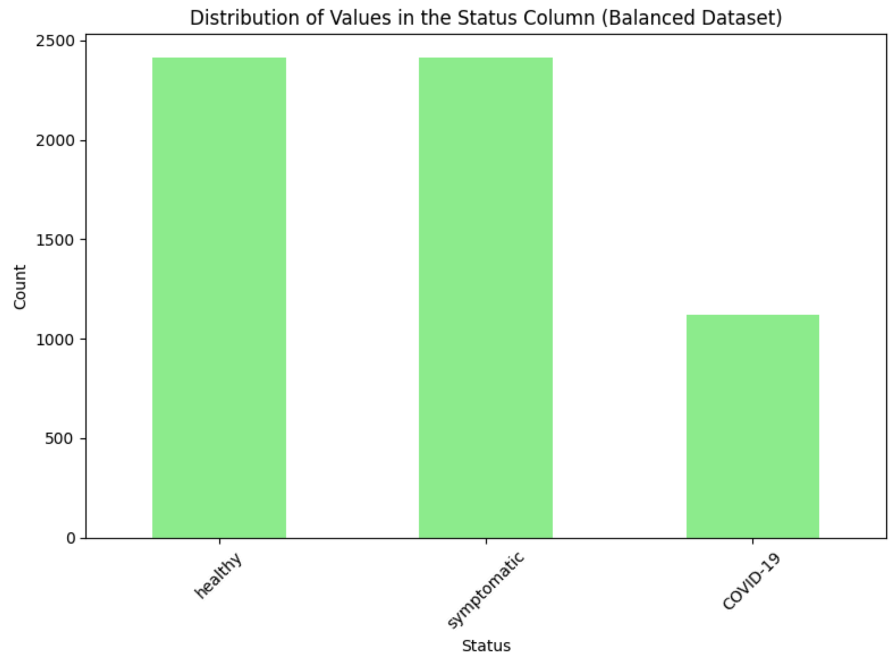
- Task 1:

Despite having a usable CNN for testing, initial results continue to be yielding low results due to inaccuracies within the dataset. Until the dataset is further cleaned and confirmed to match the given labeling testing with either our developed CNN or the benchmark model ResNet50 will be fruitless. Feature development has continued, and initially was showing no signs of improvement. With the discovery of the dataset being less than usable at this time, it is anticipated that results will improve once the dataset has been cleaned.

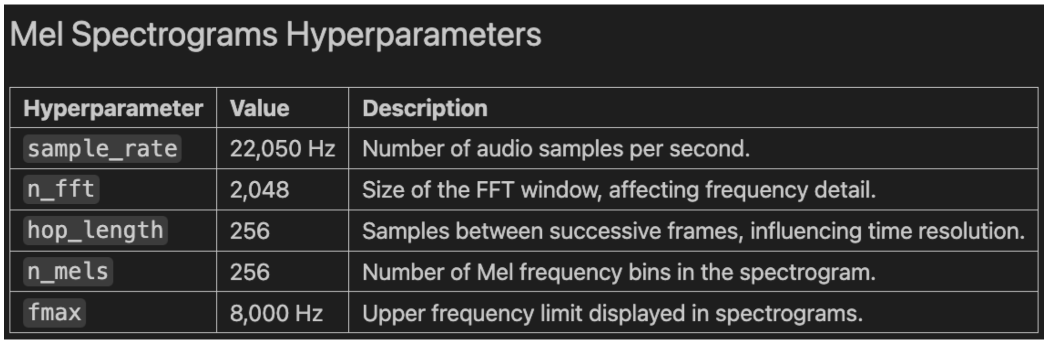
- Task 2 (Lamine):

In Milestone 3 of the COVID-19 cough detection project, significant advancements were made in data handling and preprocessing to enhance the model's learning capacity. Unlike previous milestones where undersampling was employed, this phase utilized oversampling for the COVID-19 class to address class imbalance. Specifically, three random data augmentation techniques were applied to the COVID-19 cough samples, effectively increasing the dataset size and diversity.

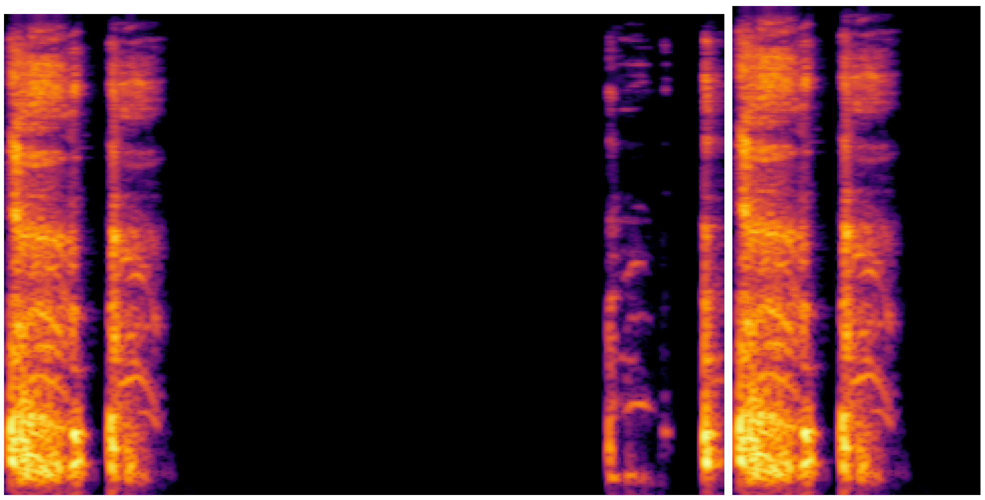




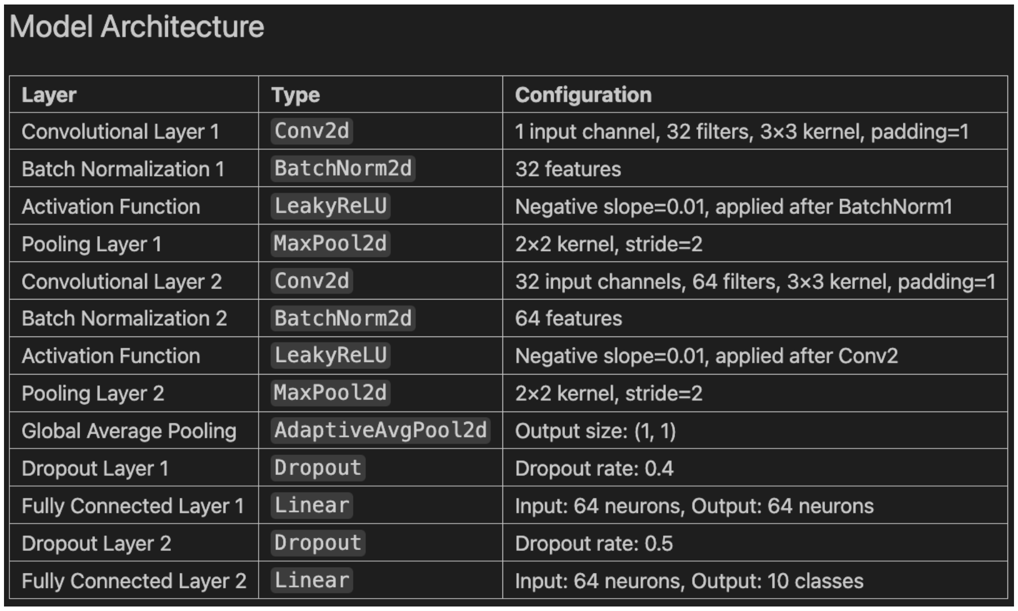
Mel spectrograms were generated using the librosa library, allowing for higher resolution spectrograms by fine-tuning hyperparameters to capture detailed audio features. Additionally, the spectrograms were converted to grayscale and normalized, reducing the input complexity by eliminating the three RGB channels and focusing on essential auditory information. To simulate single cough events and eliminate the need for padding silences, all images were resized to one-third of their original dimensions.


Mel Spectrograms Hyperparameters

Hyperparameter	Value	Description
`sample_rate`	22,050 Hz	Number of audio samples per second.
`n_fft`	2,048	Size of the FFT window, affecting frequency detail.
`hop_length`	256	Samples between successive frames, influencing time resolution.
`n_mels`	256	Number of Mel frequency bins in the spectrogram.
`fmax`	8,000 Hz	Upper frequency limit displayed in spectrograms.

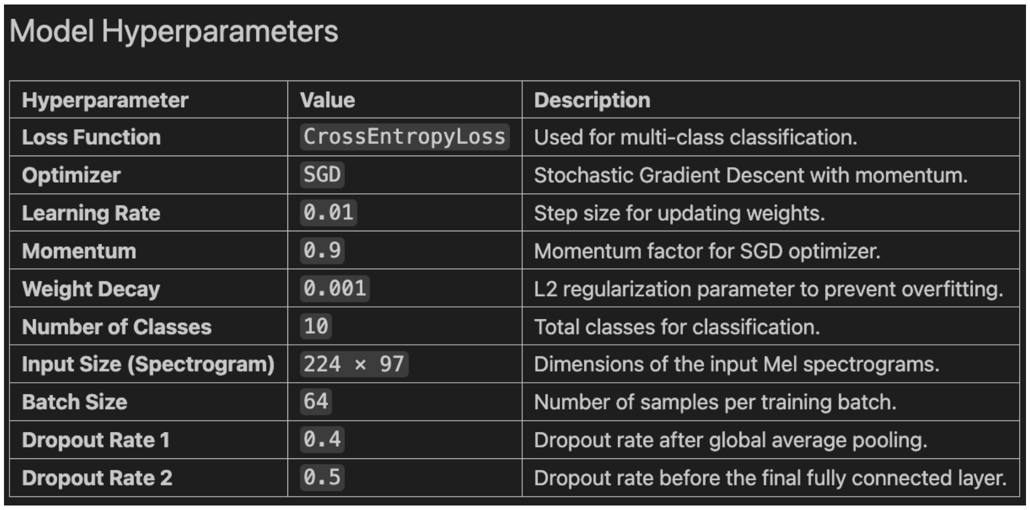


Model training procedures were meticulously refined to provide a more comprehensive evaluation of performance. The hyperparameters related to mel spectrogram generation were optimized to ensure maximum detail was retained in the spectrograms. During training, confusion matrices were incorporated alongside traditional metrics such as epoch counts, accuracies, and training times, offering deeper insights into the model's classification capabilities across different classes. Loss line graphs were also introduced to visualize the model's performance trajectory over each epoch, facilitating the identification of overfitting or underfitting trends. To enhance the model's ability to learn from the augmented dataset, the patience parameter in the training process was increased, allowing the model more epochs to converge effectively. Importantly, the model architecture remained consistent with previous experiments to maintain the validity of comparative analyses.



Model Architecture

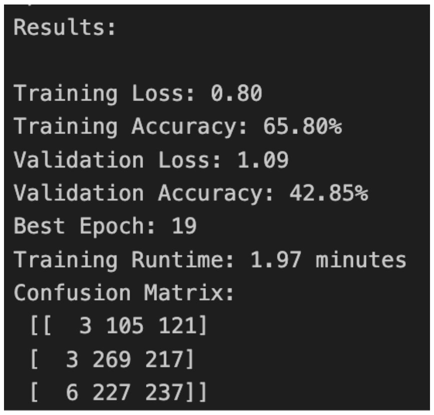
Layer	Type	Configuration
Convolutional Layer 1	`Conv2d`	1 input channel, 32 filters, 3×3 kernel, padding=1
Batch Normalization 1	`BatchNorm2d`	32 features
Activation Function	`LeakyReLU`	Negative slope=0.01, applied after BatchNorm1
Pooling Layer 1	`MaxPool2d`	2×2 kernel, stride=2
Convolutional Layer 2	`Conv2d`	32 input channels, 64 filters, 3×3 kernel, padding=1
Batch Normalization 2	`BatchNorm2d`	64 features
Activation Function	`LeakyReLU`	Negative slope=0.01, applied after Conv2
Pooling Layer 2	`MaxPool2d`	2×2 kernel, stride=2
Global Average Pooling	`AdaptiveAvgPool2d`	Output size: (1, 1)
Dropout Layer 1	`Dropout`	Dropout rate: 0.4
Fully Connected Layer 1	`Linear`	Input: 64 neurons, Output: 64 neurons
Dropout Layer 2	`Dropout`	Dropout rate: 0.5
Fully Connected Layer 2	`Linear`	Input: 64 neurons, Output: 10 classes



Model Hyperparameters

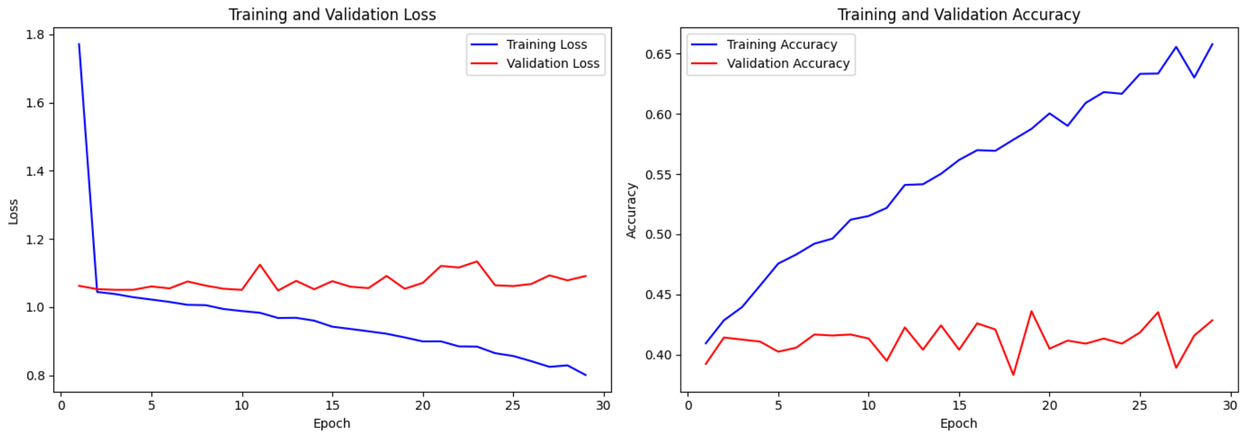
Hyperparameter	Value	Description
Loss Function	`CrossEntropyLoss`	Used for multi-class classification.
Optimizer	`SGD`	Stochastic Gradient Descent with momentum.
Learning Rate	`0.01`	Step size for updating weights.
Momentum	`0.9`	Momentum factor for SGD optimizer.
Weight Decay	`0.001`	L2 regularization parameter to prevent overfitting.
Number of Classes	`10`	Total classes for classification.
Input Size (Spectrogram)	`224 × 97`	Dimensions of the input Mel spectrograms.
Batch Size	`64`	Number of samples per training batch.
Dropout Rate 1	`0.4`	Dropout rate after global average pooling.
Dropout Rate 2	`0.5`	Dropout rate before the final fully connected layer.

The implementation of these enhancements led to noticeable improvements in the model's training dynamics. Experiments 4 and 5 demonstrated that the model could better learn from the augmented and higher-quality data, evidenced by increased training accuracy and reduced training loss. Specifically, Experiment 4 focused on high-resolution, grayscale spectrograms with augmented COVID-19 data, while Experiment 5 further incorporated new data samples processed with librosa and additional oversampling techniques. Despite these advancements, the validation accuracy did not see a corresponding increase, indicating that while the model was better at learning the training data, its generalization to unseen data remained unchanged. Nevertheless, the improvements in training performance underscore the effectiveness of the data augmentation and preprocessing strategies implemented in Milestone 3, laying a solid foundation for future enhancements aimed at boosting validation accuracy.

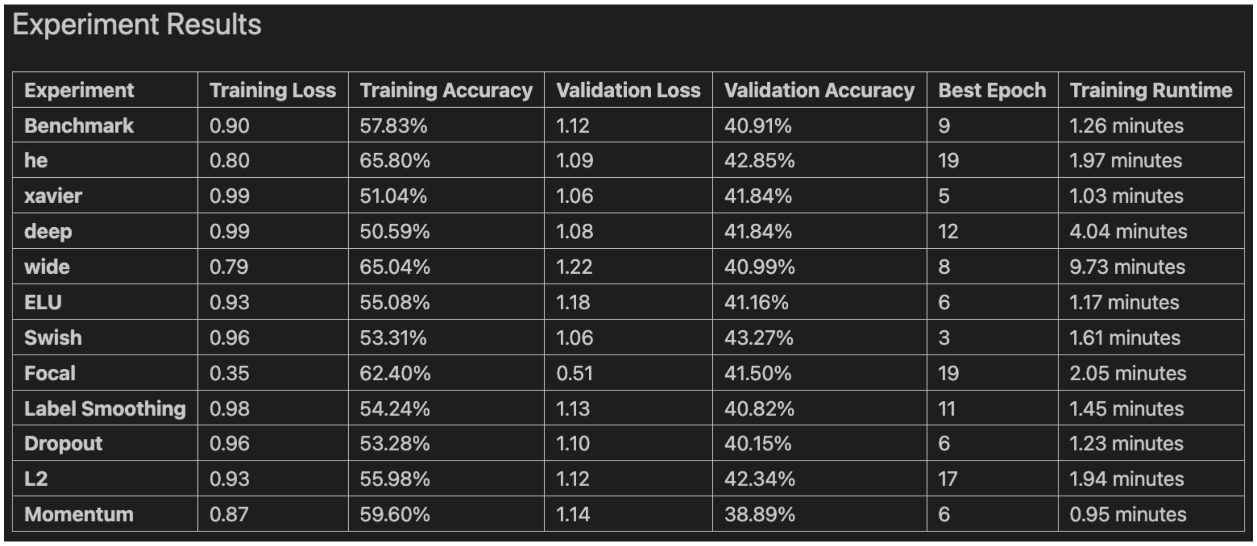


```
Results:

Training Loss: 0.80
Training Accuracy: 65.80%
Validation Loss: 1.09
Validation Accuracy: 42.85%
Best Epoch: 19
Training Runtime: 1.97 minutes
Confusion Matrix:
 [[  3 105 121]
 [  3 269 217]
 [  6 227 237]]
```



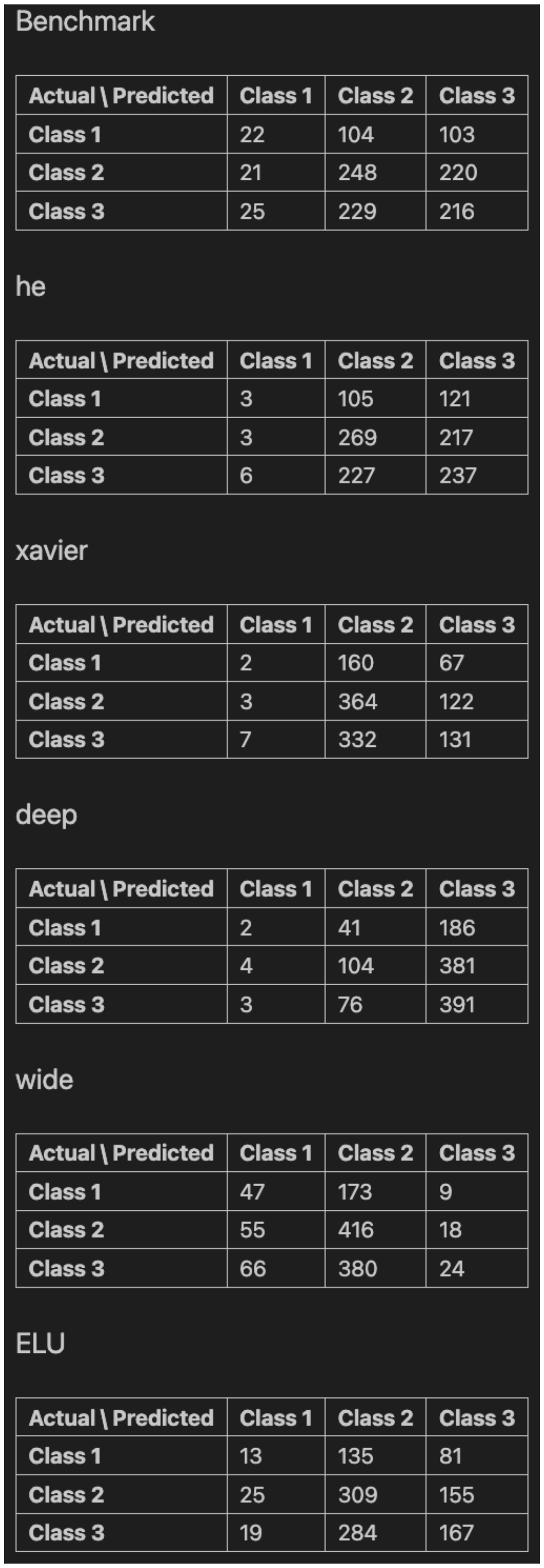
For best model overall performance



Experiment Results

Experiment	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Best Epoch	Training Runtime
Benchmark	0.90	57.83%	1.12	40.91%	9	1.26 minutes
he	0.80	65.80%	1.09	42.85%	19	1.97 minutes
xavier	0.99	51.04%	1.06	41.84%	5	1.03 minutes
deep	0.99	50.59%	1.08	41.84%	12	4.04 minutes
wide	0.79	65.04%	1.22	40.99%	8	9.73 minutes
ELU	0.93	55.08%	1.18	41.16%	6	1.17 minutes
Swish	0.96	53.31%	1.06	43.27%	3	1.61 minutes
Focal	0.35	62.40%	0.51	41.50%	19	2.05 minutes
Label Smoothing	0.98	54.24%	1.13	40.82%	11	1.45 minutes
Dropout	0.96	53.28%	1.10	40.15%	6	1.23 minutes
L2	0.93	55.98%	1.12	42.34%	17	1.94 minutes
Momentum	0.87	59.60%	1.14	38.89%	6	0.95 minutes

Confusion matrices:



Benchmark

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	22	104	103
Class 2	21	248	220
Class 3	25	229	216

he

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	3	105	121
Class 2	3	269	217
Class 3	6	227	237

xavier

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	2	160	67
Class 2	3	364	122
Class 3	7	332	131

deep

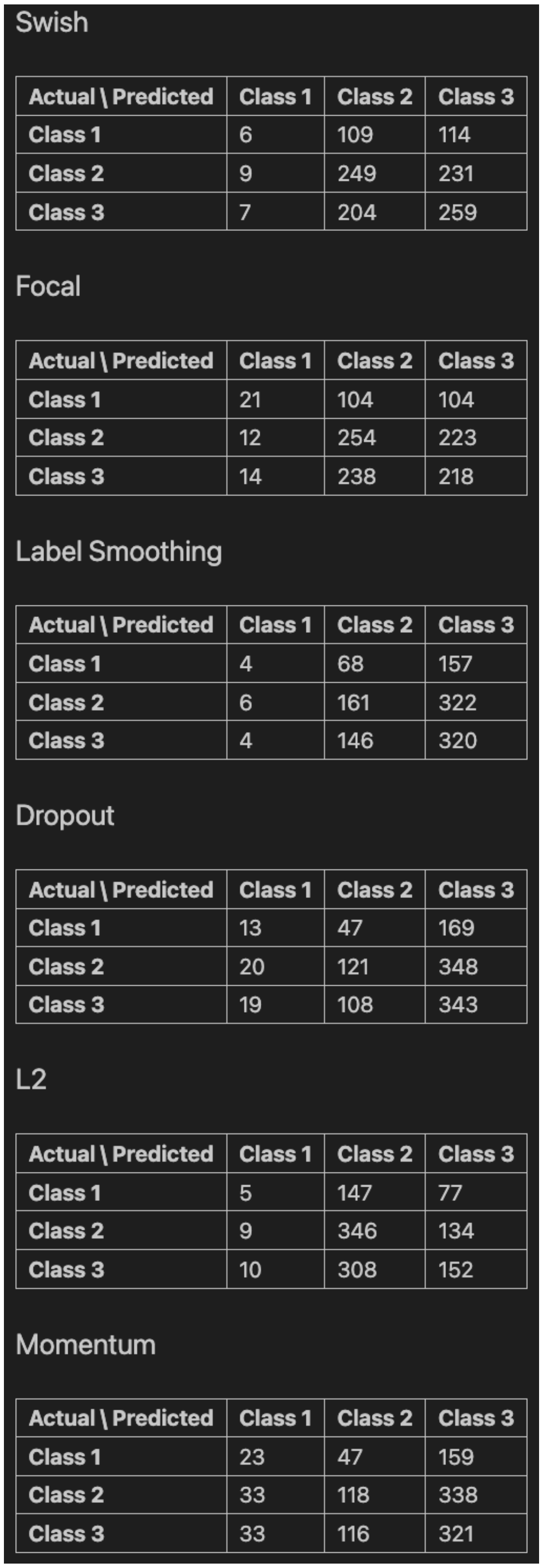
Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	2	41	186
Class 2	4	104	381
Class 3	3	76	391

wide

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	47	173	9
Class 2	55	416	18
Class 3	66	380	24

ELU

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	13	135	81
Class 2	25	309	155
Class 3	19	284	167



Swish

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	6	109	114
Class 2	9	249	231
Class 3	7	204	259

Focal

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	21	104	104
Class 2	12	254	223
Class 3	14	238	218

Label Smoothing

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	4	68	157
Class 2	6	161	322
Class 3	4	146	320

Dropout

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	13	47	169
Class 2	20	121	348
Class 3	19	108	343

L2

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	5	147	77
Class 2	9	346	134
Class 3	10	308	152

Momentum

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	23	47	159
Class 2	33	118	338
Class 3	33	116	321

- Task 3:

Using the Appliku services and Amazon Web Service (AWS), the website is now being hosted live. Appliku provides a simplified connection to AWS which allows for a more streamlined process to host a website. This also provides a secure connection to the website using the HTTPS protocol. Additional testing still remains with the current database as additional user information needs to be saved, but also ensure security within the database. Using the command ‘django-admin startapp (app_name)’ the ML container app is created which will delegate all ML related tasks on the website. A ‘urls.py’ file must also be created to provide proper routes within the website from page to page or link to link. The final thing that must be done is to add the app in the main projects ‘settings.py’ file. This allows the entire project to communicate with the new created app. This will allow the initial testing to begin relating to the ML portion of the website as the audio recording will be the first thing to be created.

- Task 4:

For this task, integration of the Django framework and the machine learning model began. To integrate these, apps to store the trained model and record new audio and convert it to mel spectrograms had to be created in the Django framework. To use an ML model, it must be saved as a ‘.pth’ file and stored in the app created to hold it. Once this was successfully done, the next step would be to run the application on a server. This step has not yet been completed, as there have been many issues with directory structure, packages, and simply learning the Django framework. For example, librosa is a necessary package for the audio app, but it is not compatible with the newest version of Python. This version was the one used in the environment attempting to run the application, and it was a time-consuming process to identify and resolve this incompatibility. Issues such as this have delayed the completion of this task.

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

Rodrigo Alarcon

The focus on this milestone was ensuring that the website was being hosted properly and could be accessed by anyone. With Appliku and AWS, the process was simplified and streamlined with the next step of working on expanding the user database. Hosting was accomplished by reworking the structure of the website in a form that was compatible with these services. Another aspect I worked on was creating the new ML container app using the command ‘django-admin startapp (app_name)’ along with the steps mentioned in the task 3 description. In regards to the general website, I met with Emma to explain some of the inner workings of Django, as well as discuss some of the content that will go on the blank pages. That content just needs to be uploaded. I have also been looking into the different methods of recording audio samples, but with no implementation yet.

Emma Conti

I met with Rodrigo and have begun to familiarize myself with the web app development aspect of the project. I designed both the Home Page and the Continuing Research Page and wrote the content to be implemented so users will be able to understand and better navigate the web app.

Additionally, I did further research into ResNet50 and running our dataset with a ResNet50 model. It has yet to yield any usable results as testing has just begun on my end, but I will be able to compare the results against our current model in order to better determine what features can be improved on our model. I will also be assisted with our dataset in cleaning the data to ensure it is usable. Until this is complete, testing with both our model and ResNet50 will not yield any usable results and is therefore fruitless.

A primary part of my work was towards justifying the choices our teams was making in terms of the model and the components we are using.

Mel Spectrograms

Mel Spectrograms are being used because the way to determine whether or not a cough is covid is based on the way they are breathing. What a mel spectrogram does is makes a visual representation of the types of noise happening in an audio file. This will allow for the specific types of noise, in our case restricted breathing prior to the cough, to become more identifiable. Differences between “wet” and “dry” coughs can also be determined using Mel Spectrograms.

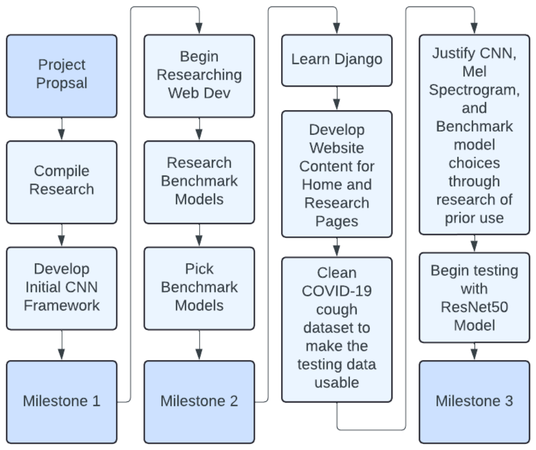
Mel Spectrogram the x-axis is time and the y-axis is frequency. They are specifically designed in order to better represent what humans hear in a visual format. The vibrancy of the color represents the amplitude at a given time. The main benefit of using Mel Spectrograms is the ability to compression the frequency scale to help reduce the data’s dimensionality, which assists in bringing down the computational load on a learning model. Capturing important pitch that can then be analyzed further by a CNN.

Mel Spectrograms are ideal for our project because there is a short and limited amount of data per sample that will need to be evaluated. Mel Spectrograms will be able to attain the specific audio frequencies that are needed to evaluate COVID 19 in a cough recording. Other ways of transferring audio into usable data are less efficient when compared to Mel Spectrograms for our specific needs. Mel Spectrograms also let us focus on specific frequencies and amplitudes and look for repetition amongst our dataset. Because the data is set up to be a visual representation of human sound, it will also be possible for us to visually identify what in a Mel Spectrogram is defined as “COVID 19”, which also allows for better training and testing.

CNNs

CNNs and RNNs are generally heralded as the best choices for a cough detection algorithm. Additionally, because mel Spectrograms are two dimensional, they are primarily paired with CNNs. A downside of CNNs is that they are a training model for a local space and can result in the loss of information based on how the information is prepared and filtered. This cannot be solved entirely, only mitigated.

CNNs are ultimately more useful over RNN’s because they are able to produce a feature map. This allows for further development on dataset augmentation to be based on which


Project Propsal
Compile Research
Develop Initial CNN Framework
Milestone 1
Begin Researching Web Dev
Research Benchmark Models
Pick Benchmark Models
Milestone 2
Learn Django
Develop Website Content for Home and Research Pages
Clean COVID-19 cough dataset to make the testing data usable
Justify CNN, Mel Spectrogram, and Benchmark model choices through research of prior use
Begin testing with ResNet50 Model
Milestone 3

features are most important to be identified within the CNN. Mel Spectrograms can therefore be enhanced to more clearly show the features needed specifically to identify COVID-19 during testing.

ResNet50

ResNet50 was used as a benchmark model for testing our own CNN. As a CNN, ResNet50 is already uniquely equipped for audio processing and testing specifically, and yields fantastic results. A variety of cough based mel-spectrogram models using ResNet50 as a basis have published their findings, making it a strong choice for our testing.

One of the benefits of ResNet50 over other CNNs is the accuracy without needing any kind of preprocessing for imbalance data handling on an audio benchmark set. This allows for data sets to be used without any kind of data augmentation, and still yields usable results. Results from testing with our dataset using ResNet50 will be included soon.

Progression of Work from Emma

Lamine Deen

During Milestone 3 of the COVID-19 cough detection project, I focused extensively on enhancing our data handling and preprocessing techniques to improve the model's learning capabilities. Recognizing the class imbalance issue from previous milestones, I implemented an oversampling strategy for the COVID-19 class instead of undersampling. To further augment the dataset, I applied three random data augmentation techniques to the COVID-19 cough samples, which significantly increased both the size and diversity of our data. Additionally, I generated high-resolution Mel spectrograms using the `librosa` library, meticulously fine-tuning hyperparameters to capture detailed audio features. By converting these spectrograms to grayscale and normalizing them, I reduced input complexity and emphasized essential auditory information. To accurately simulate single cough events and eliminate the need for padding silences, I resized all images to one-third of their original dimensions, ensuring that each spectrogram represented a concise and relevant cough sample.

In refining the model training procedures, I optimized the hyperparameters associated with Mel spectrogram generation to retain maximum detail. I introduced confusion matrices alongside traditional metrics such as epoch counts, accuracies, and training times to provide a more comprehensive evaluation of the model's performance across different classes. Additionally, I incorporated loss line graphs to visualize the model's performance trajectory over each epoch, which was instrumental in identifying trends related to overfitting or underfitting. To allow the model ample opportunity to learn from the augmented dataset, I increased the patience parameter, enabling the model to train for more epochs and achieve better convergence. Throughout this process, I maintained consistency in the model architecture with previous experiments to ensure the validity of comparative analyses, thereby providing a reliable basis for assessing the impact of the implemented changes.

The culmination of these efforts led to significant improvements in the model's training dynamics, as evidenced by the results from Experiments 4 and 5. In Experiment 4, I utilized high-resolution, grayscale spectrograms with augmented COVID-19 data, which resulted in increased training accuracy and reduced training loss. Experiment 5 further advanced this approach by incorporating new data samples processed with librosa and additional oversampling techniques, reinforcing the model's ability to learn from higher-quality data. Although these enhancements did not translate into higher validation accuracy, they demonstrated the model's improved capacity to learn from the training data. This progress underscores the effectiveness of the data augmentation and preprocessing strategies I implemented in Milestone 3. Moving forward, these foundational improvements provide a solid platform for future enhancements aimed at boosting validation accuracy and overall model generalization.

Audrey Eley

During this milestone I had two main tasks: Work on aesthetic/branding improvements, and work to integrate the ML model and the Django framework. The branding element was much simpler and less time consuming. For this, I simply identified some potential color palettes, made potential official names for the project, and created some logo mock-ups combining these color palettes and names. For the integration task, I made the two apps required to implement ML/Django integration. I attempted to verify successful integration, but was not able to do so in the allotted time due to issues highlighted in the task description.

7. Plan for the next Milestone 4 (Task Matrix)

Task	Rodrigo	Emma	Lamine	Audrey
1. ML Testing	Test using benchmark model (ResNet50) and initial testing from our model.			
2. Refined ML Workflow	Continue to improve the ML model. Determine which improvement strategies to implement based on testing results.			
3. Web Testing	Continue implementing a framework for users to access the CNN and upload their coughs.			

4. Integrating WebApp and CNN	Determine 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: For this task, once the data has been cleaned and is made usable for testing, testing with both our model and the benchmark model (ResNet50) will resume. Based on the results from the benchmark tests, it should become clearer which features need to be adjusted to achieve better results from our model. As our model is developed more layers may be added for more accurate results.
- Task 2: To refine the machine learning workflow for the COVID-19 cough detection project, the team will continue enhancing the ML model by systematically implementing improvement strategies informed by comprehensive testing results. Initially, the dataset will undergo meticulous cleaning, which involves manually reviewing and listening to each audio file to eliminate noisy or irrelevant samples, thereby ensuring high data quality and integrity. Following this, the team will integrate and stack the results from experiments that demonstrated increases in training or validation accuracy, leveraging these successful approaches to build a more robust foundation for the model. This strategic selection of enhancement techniques will be pivotal in addressing any residual class imbalances and optimizing data representation.

  Furthermore, the project will advance by developing three more complex neural network architectures designed to extract intricate patterns and features from cough audio samples. These architectures will incorporate deeper layers, advanced activation functions, and sophisticated regularization methods to maximize the model's learning capacity. In parallel, ResNet50 will be employed as a benchmark to compare its performance against the custom-designed models, providing valuable insights into the strengths and potential areas for improvement of each approach. Additionally, extensive fine-tuning will be conducted, including hyperparameter optimization and the application of advanced data augmentation techniques, to further enhance model accuracy and generalization. This comprehensive refinement process aims to create a highly accurate and reliable COVID-19 cough detection system, building upon the successes of previous milestones and setting the stage for ongoing innovation.

- Task 3: As research continues and the model develops, the website will continue to grow. As pages are added, the webapp will become more robust, and it will become more user-friendly. As integration continues, allowing for test cases may become possible to ensure that the webapp is working as needed.

- Task 4: For this task, as the model continues to develop, it is important to continue to integrate the most recent model with the website to determine whether it can be accommodated on the webapp.

9. See Faculty Advisor Feedback Below

10. Meeting Date: November 25th, 2024

11. Faculty Advisor feedback on each task for the current Milestone 2

- Task 1: The data needs to be cleaned.
- Task 2: The model should be trained on the cleaned data.
- Task 3: Then the hyperparameters and CNN architecture can be selected accordingly.
- Task 4: Based on the current data, the result was not successful.
- Task 5: Web testing is done gradually and needs to be implemented and completed for user file uploading.
- Task 6: Check the input shape.
- Task 7: Update the document accordingly

Faculty Advisor Signature: ________________________________ Date: ________

Evaluation by Faculty Advisor

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: __________