Automated COVID-19 Detection Using Deep Learning

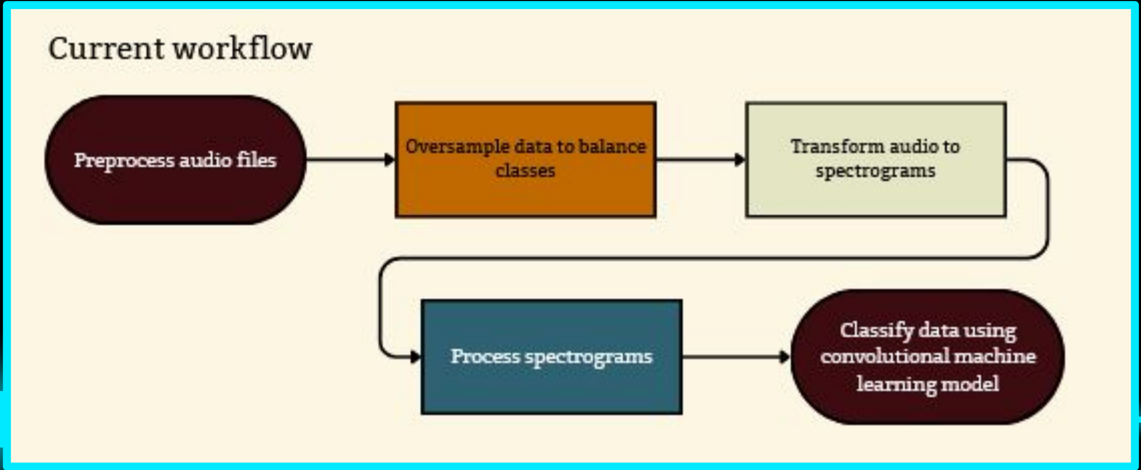
Rodrigo Alarcon, Emma Conti, Lamine Deen,
Audrey Eley
Advisor: Dr. Zahra Nematzadeh

Task Matrix: Milestone 2

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
5. Pick 3 benchmark models	100%	10%	70%	10%	10%	All benchmarks selected

Task 1 - Refine ML Workflow

- Researched and explored potential workflows
- Initial workflow was defined
- More workflow phases were added



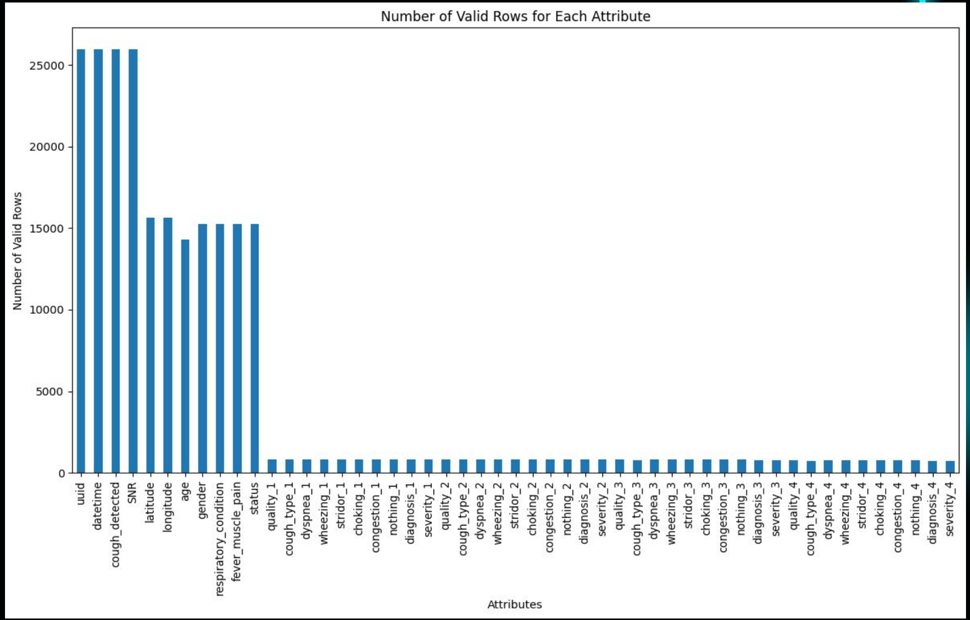
Task 2 - Data Exploration and Cleaning

Dataset: Audio files in .webm format with metadata on cough status.

Process:

- Loaded metadata and audio files.
- Matched metadata UUIDs to available audio files.
- Removed entries missing audio files.

Outcome: Consistent dataset aligning metadata with existing audio.



Handling Missing Data

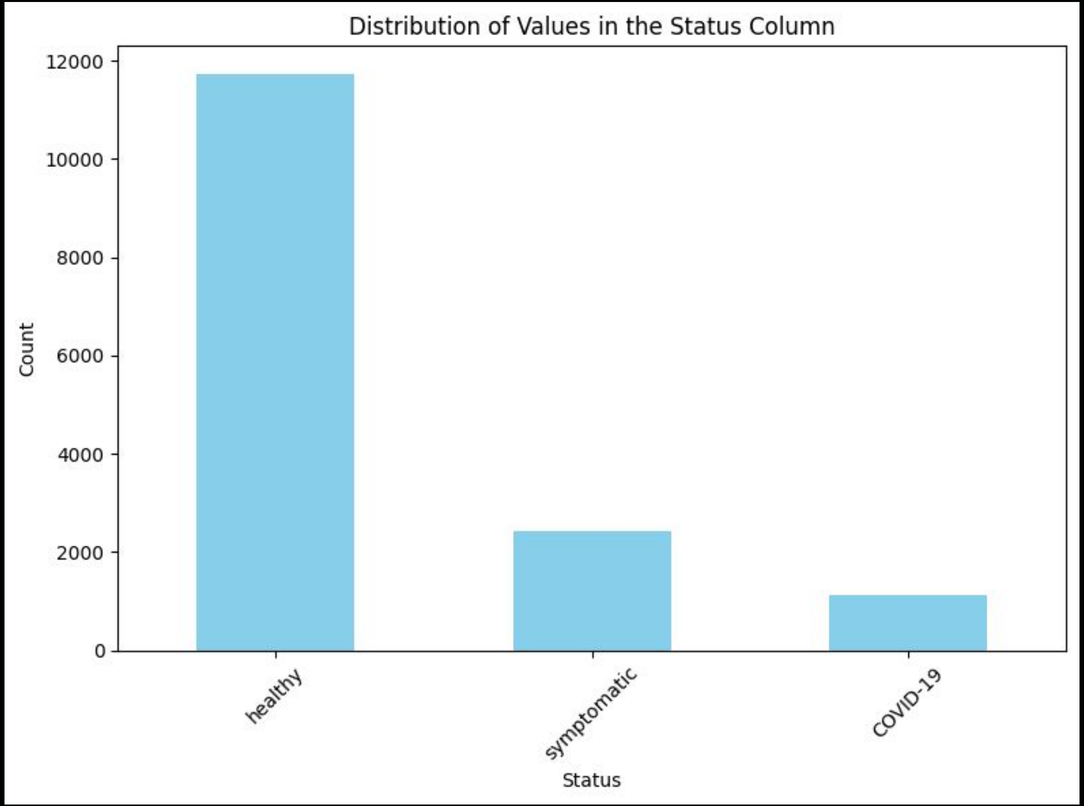
Steps:

- Checked attributes for missing values.
- Focused on essential fields like `status`.
- Visualized valid data counts per attribute to verify data quality.

Outcome: Final cleaned dataset with valid and relevant attributes retained for processing.

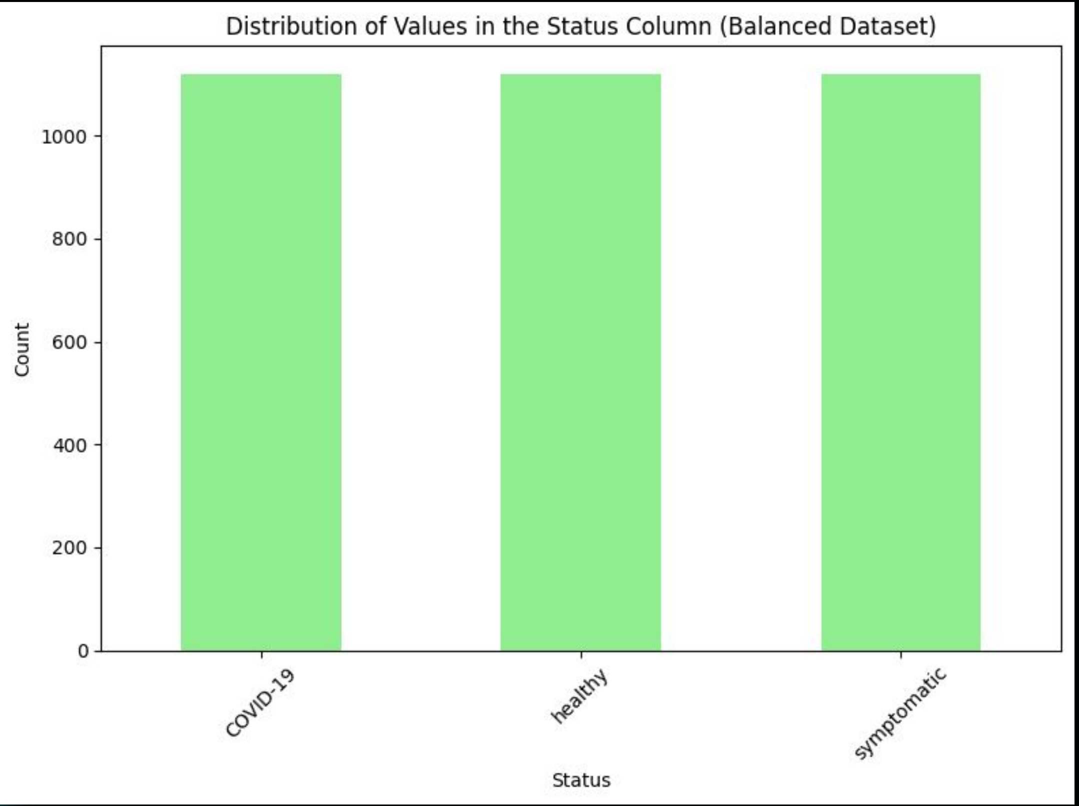
```
(27550, 51)
Number of extra files in the archive: 0
Number of missing files in the archive: 1565
New cleaned dataset shape: (25985, 51)
```

Distribution of all Classes for 'Status'



status	
healthy	11715
symptomatic	2411
COVID-19	1118

Distribution after Sampling



status

COVID-19 1118

healthy 1118

symptomatic 1118

Data Transformation - Mel Spectrograms

Purpose: Convert audio from .webm compresses audio files into .wav and then transform data into Mel spectrograms to capture frequency patterns.

Techniques:

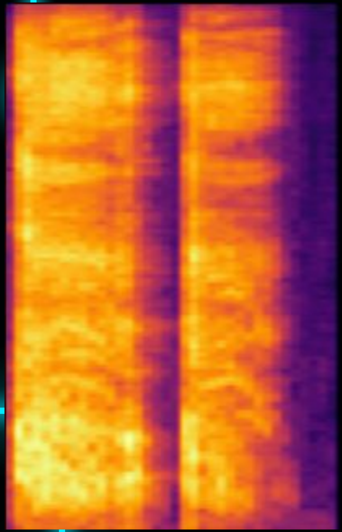
- Short-Time Fourier Transform (STFT) for frequency analysis.
- Mel scaling to focus on human-hearable ranges.

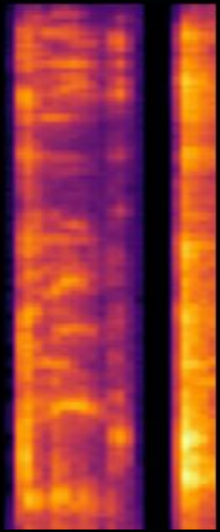
Challenges: Configuring optimal spectrogram settings to ensure relevant features are extracted for cough detection.

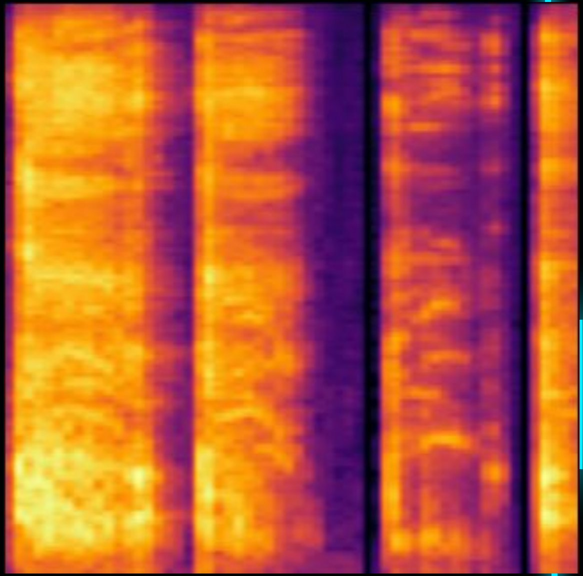
Mel Spectrograms Hyperparameters

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

Example of Mel Spectrogram







Data Augmentation Techniques

Objective: Increase dataset diversity and improve model generalization.

- **Noise Addition**: Random background noise to simulate real-world conditions.
- **Pitch Shifting**: Adjusting pitch to account for variability in cough sounds. (+/- 3 semitones)
- **Time-Stretching**: Altering playback speed to simulate different cough speeds. (+/- 10%)

Benefits: Helps reduce overfitting and improves robustness.

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.

Extra Preprocessing

- Audios were trimmed at both beginning and end to remove silence and reduce noise, which decreased audio files sizes by 67%

- Mean and Standard Deviation were computed on the entire dataset just like for Imagenette Dataset and RGB values were used for normalization during Tensor Transformation.

    - `- Mean: tensor([0.6716, 0.2803, 0.2594])`
    - `- Std: tensor([0.2539, 0.1936, 0.1297])`

Custom CNN Model Architecture

Layer Overview:

- **Convolutional Layers**: Capture spatial features from spectrograms.
- **Pooling Layers**: Reduce dimensionality, retain key information.
- **Fully Connected Layers**: Integrate learned features for final classification.

Activation: ReLU in hidden layers, softmax for final classification output.

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

Training Process and Hyperparameters

Data Split: 60% Training, 20% Validation set, and 20% Test set.

Hyperparameters:

- Learning rate, batch size, number of epochs.

Optimizer: Stochastic Gradient Descent, balancing speed and accuracy.
Loss Function: Cross-entropy loss, suited for multi-class classification.

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

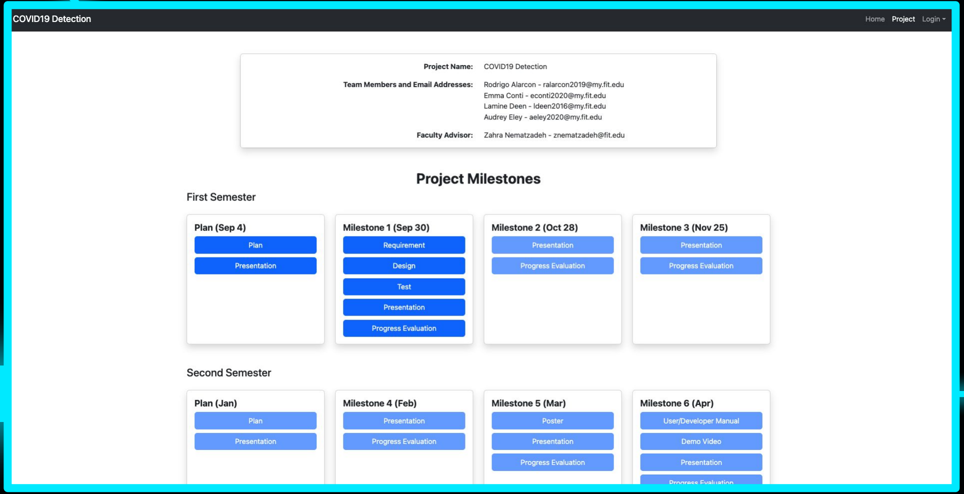
Model Performance

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 - Framework Frontend

- Created initial website in django
    - Currently displays project documentation
- Includes page to create a new user account
- Using bootstrap for styling and layout



Task 4 - Framework Backend

- Implemented user functionality with Django’s built-in authentication system
- Integrated views and url routing to handle authentication requests
- Working on extending current DB to allow for additional user info as well as a more secure authentication service

Task 5 - Selected Benchmark Models

- VGG 16 (90%)
  - Mel Spectrogram Classification
- Resnet 14 (40%)
  - Medical Image Classification
- Resnet 50 (87%)
  - Tuberculosis Cough Mel Spectrogram Classification
- Inception v4 (71%)
  - Complex Soundscape (Bird Call) Classification

Task Matrix: Milestone 3

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.			

Milestone 3

NOV 25 — Begin ML Testing

NOV 25 — Refine ML workflow

NOV 25 — Begin web testing

NOV 25 — Integrating base ML model with web using a Neural Network framework

Questions?