Automated COVID-19 Detection Using Deep Learning

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

Task Matrix: Milestone 3

Task	Rodrigo	Emma	Lamine	Audrey
1. Begin ML Testing	Test using chosen 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.			

Task 1 - Begin ML Testing

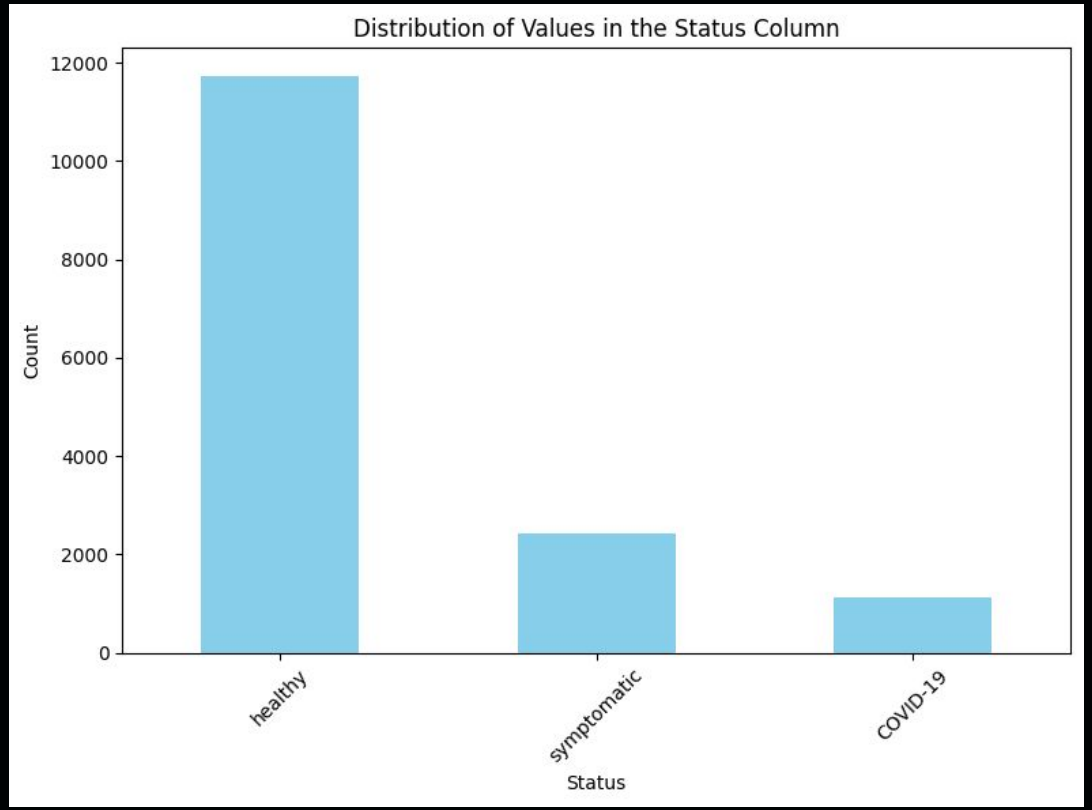
- Feature engineering continued
- ResNet50 model has been set up for testing on COVID-19 dataset

- Dataset cleaning must be completed before further testing can be completed

Task 2 - Refine ML Workflow

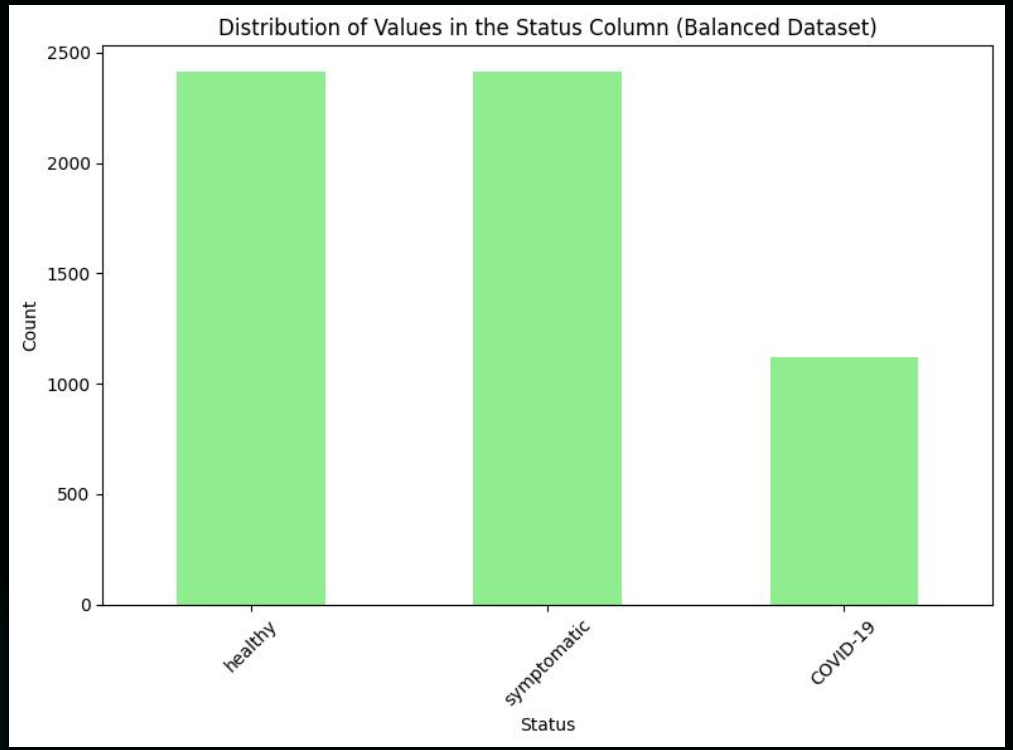
- **Oversampling**: Used for the COVID-19 class to address class imbalance.
- **Augmentation**: Applied three techniques to increase dataset size and diversity.
- Changed the data augmentation method to be applied randomly to covid 19 class replaced timestretch with loudness and quietness
- **Improved Learning**: Enhanced the model's capacity with better preprocessing.
- **Data Split**: 60% Training, 20% Validation set, and 20% Test set.

Distribution of all Classes for ‘Status’



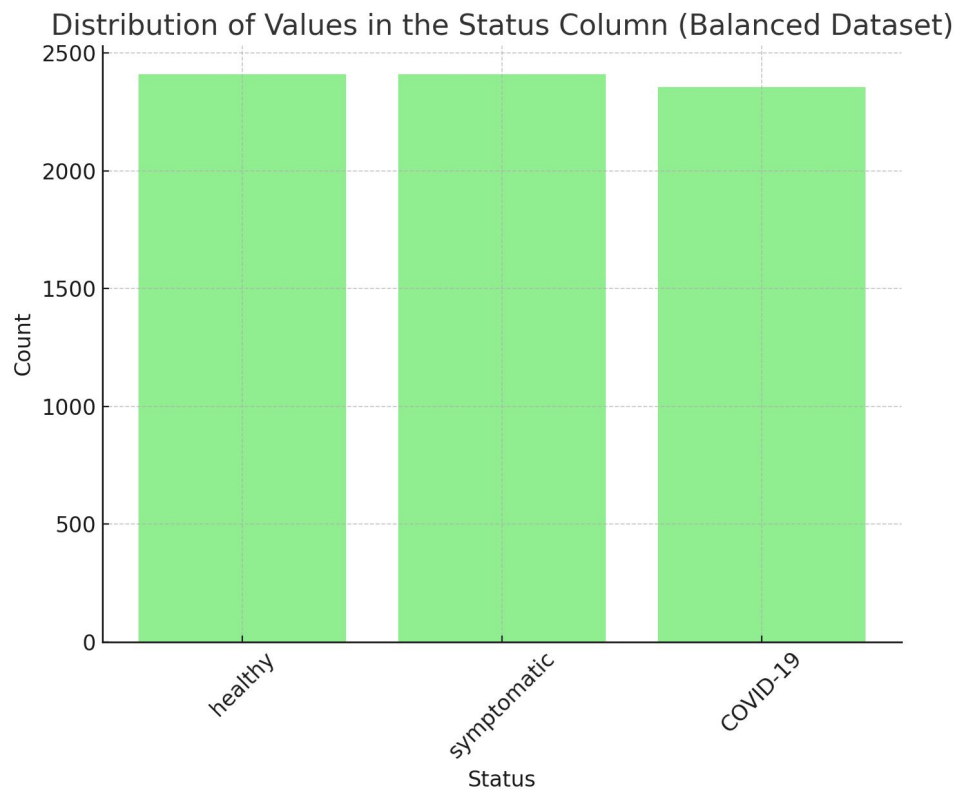
status	
healthy	11715
symptomatic	2411
COVID-19	1118

Distribution after Oversampling



status	
COVID-19	1118
healthy	2411
symptomatic	2411

Distribution after Data Augmentation



status	
COVID-19	2355
healthy	2411
symptomatic	2411

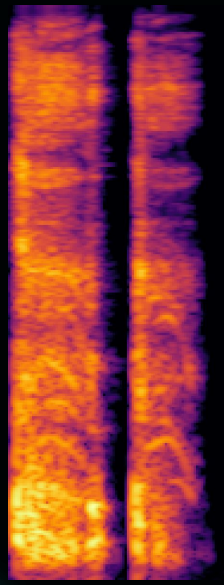
Data Transformation - Mel Spectrograms

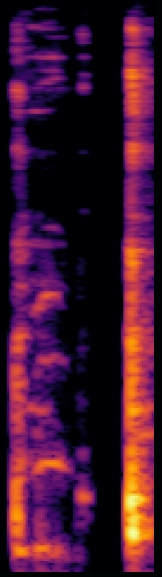
- **High-Resolution Mel Spectrograms:** Generated with Librosa instead of pytorch to capture audio features.
- **Grayscale Conversion**: Reduced input complexity by eliminating RGB channels and focusing on auditory information.
- **Normalization:** Standardized spectrogram values to ensure consistency across data.
- **Resizing**: Images were resized to one-third of their original dimensions to simulate single cough events and remove padding silences.

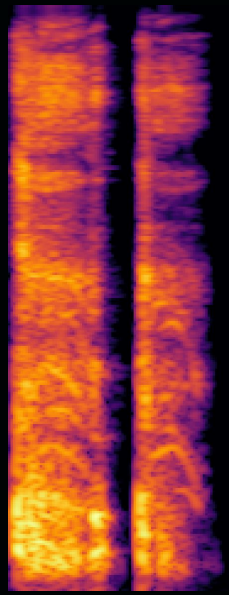
Mel Spectrograms Hyperparameters

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.

Sample







Custom CNN Model Architecture

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

Training Process and Hyperparameters

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.

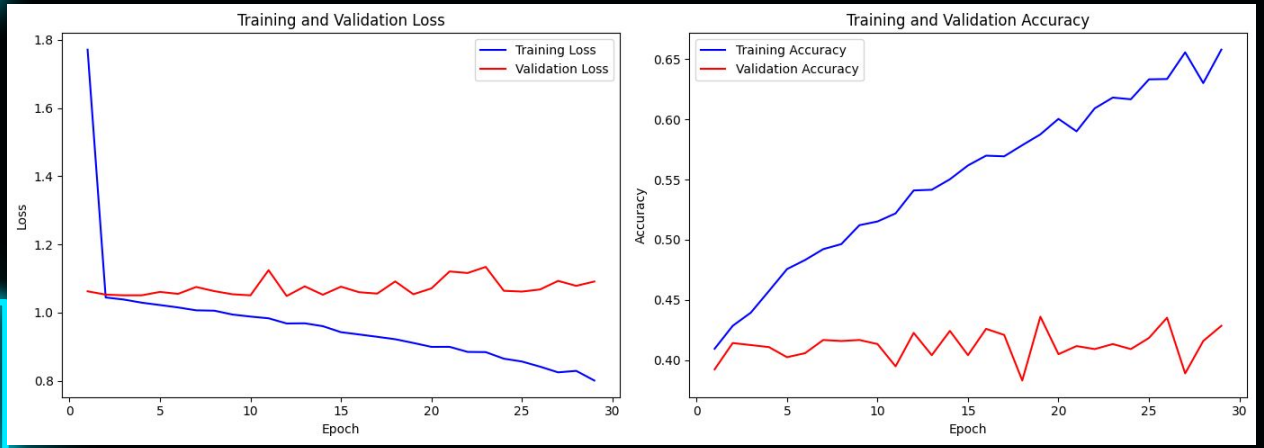
Improvement

Improved Training Dynamics: Enhancements led to higher training accuracy and reduced loss.

Experiment Focus: Experiment 4 used augmented, high-resolution grayscale spectrograms; Experiment 5 added new data processed with librosa and oversampling.

Validation Challenges: Despite training improvements, validation accuracy remained unchanged, highlighting generalization issues.

Results for best model




```
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]]
```

Results for all models

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

Results for all models

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

Focal

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

Label Smoothing

Actual \ Predicted	Class 1	Class 2	Class
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 - Begin Web Testing

- Database has been developed for users
  - Will be changed to include additional user info and improved security
- Home page and research page have been added
- Audio model has been added
- Initial audio recording ability has been started
- Integration and testing with user model is in progress

Task 4 - Integration Testing

- Refining user model to handle data required for covid predictions.
- Added initial audio recording capability required for ML model
- Planned tests for data flow (audio uploads) ensuring it is isolated to each user

Task Matrix: Milestone 4

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.			

Milestone 4

NOV 25 — ML Testing

NOV 25 — Refined ML workflow

NOV 25 — Web testing

NOV 25 — Continued Integration of CNN and WebApp

Questions?