

Respiratory Disease Classification API

This project uses Google's HeAR (Health Acoustics Representations) model to classify respiratory sounds into different disease categories using audio embeddings.

Overview

This service takes short audio clips (2 seconds, .wav format) of respiratory sounds and classifies them into one of the following categories:

- Asthma
- COPD
- Pneumonia
- Bronchial
- Healthy

The system uses Google's HeAR model to generate embeddings from the audio data, which are then fed into a Random Forest classifier to predict the respiratory condition.

Technical Architecture

1. **Frontend:** Streamlit web interface for easy file upload and classification
2. **Backend API:** Flask REST API for programmatic access
3. **Models:**
 - Google HeAR model for audio embedding extraction
 - Random Forest classifier trained on respiratory sound dataset

Getting Started

Prerequisites

- Python 3.8+
- PyTorch
- Google's HeAR model
- Required Python packages (see `requirements.txt`)

Installation

1. Clone the repository:

```
bash
```

```
git clone https://github.com/yourusername/respiratory-disease-classification.git  
cd respiratory-disease-classification
```

2. Install the required packages:

```
bash
```

```
pip install -r requirements.txt
```

3. Download and set up the HeAR model:

```
bash
```

```
# Follow instructions at https://developers.google.com/health-ai-developer-foundations/hear  
# to download and set up the HeAR model
```



4. Make sure the model paths in `app.py` and `streamlit_app.py` are correct for your environment.

Running the API

Start the Flask API:

```
bash
```

```
python app.py
```

The API will be available at `http://localhost:5000`.

Running the Streamlit Interface

Start the Streamlit app:

```
bash
```

```
streamlit run streamlit_app.py
```

The web interface will be available at `http://localhost:8501`.

API Usage

POST /predict

Endpoint for classifying respiratory audio.

Request:

- Method: POST
- Content-Type: multipart/form-data

- Body: Form data with a "file" field containing a .wav audio file (2 seconds)

Response:

- 200 OK: JSON object with predicted class

```
json
{
  "predicted_class": "asthma"
}
```

- 400 Bad Request: Error if no file or wrong format

```
json
{
  "error": "No file provided"
}
```

- 500 Internal Server Error: Other errors

```
json
{
  "error": "Error message details"
}
```

Test Examples

The repository includes test audio files in the `test_samples` directory:

- `asthma_sample.wav` - Sample of asthmatic breathing
- `healthy_sample.wav` - Sample of healthy breathing
- `copd_sample.wav` - Sample of COPD breathing
- `pneumonia_sample.wav` - Sample of pneumonia breathing
- `bronchial_sample.wav` - Sample of bronchial breathing

You can use these to test the API:

```
bash
curl -X POST -F "file=@test_samples/asthma_sample.wav" http://localhost:5000/predict
```

Model Performance

The current model shows the following performance metrics:

Classification Report:

	precision	recall	f1-score	support
Bronchial	0.50	0.04	0.07	28
asthma	0.75	0.91	0.83	57
copd	0.77	0.80	0.79	82
healthy	0.44	0.36	0.40	22
pneumonia	0.68	0.85	0.75	54
accuracy			0.71	243
macro avg	0.63	0.59	0.57	243
weighted avg	0.68	0.71	0.67	243

Known Limitations

- The model requires exactly 2-second audio clips. Shorter clips will be padded, and longer clips will be truncated.
- Performance on bronchial conditions is currently limited due to dataset imbalance.
- Audio must be in .wav format with a sample rate of 16kHz.

Future Improvements

- Enhance model performance with additional training data
- Add confidence scores to predictions
- Support for longer audio clips with sliding window analysis
- Add model versioning and A/B testing

Dataset

This project uses the "Asthma Detection Dataset Version 2" which includes labeled respiratory sounds for various conditions. The dataset was split 80/20 for training and testing.

License

[Insert your license information here]

Acknowledgments

- Google's HeAR team for the audio embedding model
- [Include any other acknowledgments]