Surround Sound: Real-Time Environmental and Sound Event Classification Using Log-Mel Spectrogram CNNs

Everett-Alan Hood
University of Washington Bothell
Bothell, WA, USA
elhood@uw.edu

Abstract

Real-time auditory scene understanding is a major challenge in machine learning, particularly outside the domains of speech recognition and music information retrieval. This project, *Surround Sound*, presents a multi-model audio analysis system capable of classifying both background environments and fine-grained sound events from raw microphone input. Using large-scale datasets including AudioSet and FSD50K, the system trains two convolutional neural network (CNN) classifiers on log–mel spectrogram representations: an eight-class environment classifier and a sixty–two–label multi-label sound event detector. A unified preprocessing pipeline standardizes all audio to 16 kHz mono, normalized ten-second segments, and consistent log–mel features.

Training incorporates class rebalancing, mixed-precision optimization, and threshold tuning for multi-label analysis. Results show moderate environment classification performance (macro-F1 ≈ 0.50) and early-stage event detection reliability (macro-F1 ≈ 0.39), but data imbalance and noise contribute to recurring errors. A supplemental real-time demo integrates both models with a lightweight large language model (LLM) scene summarizer, showcasing the system's practical potential for live acoustic awareness. This paper details the datasets, preprocessing, training pipeline, evaluation results, and future directions for unifying environment and event recognition into a cohesive auditory framework.

Keywords

Audio Classification, Acoustic Scene Recognition, Sound Event Detection, Machine Learning, Log-Mel Spectrograms, Real-Time Systems

1 Introduction

Machine learning for audio understanding has advanced rapidly in domains such as automatic speech recognition, music tagging, and medical acoustics. Yet general-purpose auditory scene awareness, the ability for a system to interpret what is happening around it based solely on sound, remains comparatively underexplored. Unlike speech or music, real-world acoustic environments contain overlapping sources, variable noise conditions, and highly imbalanced events, making the problem difficult to organize, process, and evaluate.

The Surround Sound project aims to address this by building a unified, real-time auditory scene understanding system. Rather than relying on a single classifier, the system uses two models: one specializes in background environment classification, and another performs multi-label sound event detection. Combined with lightweight natural-language summarization, these components enable a system capable of interpreting an audio scene in both broad contextual terms (e.g., *indoor*, *crowd*, *vehicle*) and fine-grained auditory details (e.g., *human laugh*, *glass clink*, *wind*, *music*). This approach is intended to reflect how people can describe a scenario based on audio: as an overall setting or specific details that make up that scenario.

1.1 DCASE and Related Work

The Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge series [6] has played a central role in advancing acoustic machine listening. DCASE tasks are typically divided into two categories: *Acoustic Scene Classification* (ASC), which assigns a single global label to an audio segment, and *Sound Event Detection* (SED), which identifies one or more overlapping events. Overall, DCASE provides carefully curated benchmarks that demonstrate what is achievable under controlled conditions, but they are not designed as end-to-end, real-time scene understanding systems.

Surround Sound diverges from these paradigms in three ways:

(1) **Multi-head design.** Instead of framing the problem as ASC *or* SED, this work trains two complementary classifiers that operate simultaneously, enabling richer interpretation of a single audio segment.
(2) **Unified preprocessed datasets.** AudioSet and FSD50K contain a diverse mix of real-world audio scenarios curated and labeled from online recordings [2, 3]. This project adds light preprocessing to harmonize them into compatible training inputs.
(3) **Real-time, user-facing inference.** Unlike DCASE benchmarks, which evaluate offline performance on curated datasets, this project targets real-time deployment, including microphone input, spectrogram computation, and an optional natural-language summarization stage powered by an LLM.

1.2 Limitations of Existing Approaches

Existing ASC and SED systems face several challenges that motivate an integrated approach:

- **Dataset noise and availability.** AudioSet clips are sourced from YouTube, meaning many recordings become unavailable over time, resulting in incomplete or noisy training sets [3].
- **Severe class imbalance.** Many sound events occur extremely rarely, while others dominate the dataset. This imbalance greatly affects supervised learning.
- **Lack of real-time considerations.** Most models are optimized for batch inference rather than streaming inputs.

These limitations motivate building a flexible, modular system rather than a single unified classifier.

1.3 Contributions

This project makes the following contributions:

- A complete dataset acquisition pipeline for both AudioSet environment clips and FSD50K event audio, including robust downloading, metadata generation, and label handling.
- A unified preprocessing framework that converts all audio into normalized, fixed-length, log–mel spectrogram tensors suitable for CNN-based learning.
- Two lightweight but effective convolutional neural network models: an eight-class environment classifier and a sixty-two-label multi-label event detector.
- A real-time inference and visualization demo that integrates both classifiers with a large language model to generate natural-language scene summaries.
- A detailed analysis of the system's performance, failure modes, and limitations, incorporating class imbalance, confusion patterns, and dataset noise.

1.4 Paper Outline

The remainder of this paper is organized as follows. Section 2 reviews relevant literature in acoustic scene recognition and sound event detection. Section 3 describes the datasets and label taxonomies used. Section 4 outlines the preprocessing and feature extraction pipeline. Section 5 details the model architectures and training procedures. Section 6 reports experimental results for both classifiers. Section 7 presents the supplemental real-time demo system. Finally, Section 8 concludes with future directions.

2 Related Work

Acoustic Scene Classification (ASC) assigns a single global label (e.g., *park*, *bus*, *office*) to an audio segment, achieving strong performance on curated benchmarks such as the DCASE challenges [6]. However, ASC models are typically evaluated offline and do not capture the fine-grained acoustic events embedded within a scene. Sound Event Detection (SED) complements ASC by identifying one or more overlapping events, often using multi-label CNN architectures with sigmoid outputs and threshold tuning, similar to the events classifier used in this project.

Large-scale datasets such as AudioSet [3] and FSD50K [2] have accelerated progress in ASC and SED but introduce challenges due to missing clips, corrupted downloads, and long-tailed label distributions. These limitations directly affect reproducibility and motivate robust preprocessing and metadata handling, as implemented in Surround Sound.

Spectrogram-based CNN architectures similar to those used here have been widely adopted for large-scale audio tagging and environmental sound classification [4, 8]. Overall, this project builds on established spectrogram-based CNN pipelines while addressing practical considerations that are less emphasized in prior ASC/SED research, including dataset volatility, class imbalance, and real-time inference requirements.

3 Datasets

This project uses two large-scale, publicly available datasets to train the environment and sound event classifiers: a filtered subset of AudioSet [3] for background environments and the FSD50K dataset [2] for multi-label sound event detection. Although both datasets provide substantial coverage of real-world audio, they differ in taxonomy, labeling methodology, and reliability, requiring additional harmonization and preprocessing.

3.1 AudioSet Environment Subset

AudioSet consists of more than two million 10-second audio clips sourced from YouTube and annotated with over 500 ontology-based labels [3]. For this project, a focused subset of eight high-level environment categories was selected: *animal*, *crowd*, *indoor*, *other_ambient*, *outdoor_nature*, *vehicle_ground*, *vehicle_water*, and *water*. These classes were chosen based on the broad semantic groupings present in the AudioSet ontology, their suitability as contextual "scene" descriptors, and their alignment with the project's goal of real-time environmental awareness.

Each AudioSet clip is associated with a unique YouTube ID and timestamp interval. However, because AudioSet relies on third-party hosting, many clips are unavailable, removed, or corrupted. The download pipeline (`environment_download.py`) attempts segmented extraction via `yt-dlp` and falls back to full-video downloads with post-trimming. Additional metadata is recorded for each clip, including status, timestamps, and source URLs.

A substantial portion of the manifest could not be retrieved or failed to decode. After filtering unusable files, a total of 11,207 environment samples were successfully incorporated into the dataset.

3.2 FSD50K Event Dataset

For fine-grained sound event detection, this work uses FSD50K, an open dataset of human-labeled sound events across 200+ categories [2]. Unlike AudioSet, FSD50K audio files are directly hosted on Zenodo and are consistently accessible.

To unify the label space and reduce complexity, a custom canonical taxonomy of 62 event classes was constructed from the raw FSD50K metadata (`01_events_setup.py`). This process merges semantically redundant labels and propagates hierarchical relationships. Because clips often contain multiple simultaneous events, the resulting formulation is a multi-label classification problem.

As with the environment data, each FSD50K clip is standardized into a normalized 10-second waveform and converted into a log–mel spectrogram. All 51,197 available samples were successfully processed for model training.

3.3 Dataset Statistics

After filtering and preprocessing, the final datasets used in training consist of:

- **Environment clips:** 9,526 training samples and 1,681 test samples.
- **Event clips:** 43,517 training samples and 7,680 test samples.

0: animal	1: crowd
2: indoor	3: other_ambient
4: outdoor_nature	5: vehicle_ground
6: vehicle_water	7: water

Table 1: Eight-class environment taxonomy used for classification.

0: Aircraft	1: Alarm Beep	2: Alert
3: Animal	4: Appliance	5: Bell Chime
6: Bicycle	7: Bird	8: Bird Call Song
9: Bird Chirp	10: Cabinet	11: Car Engine
12: Car Pass By	13: Cat	14: Cough/Sneeze
15: Dog	16: Domestic	17: Door Open/Close
18: Siren	19: Farm Animals	20: Fire Crackle
21: Flowing Water	22: Footsteps	23: Glass Clink
24: Guitar Instr.	25: Human	26: Breathing
27: Cry/Laugh	28: Human Group	29: Impact
30: Insect Buzz	31: Keyboard Instr.	32: Kitchen Appl.
33: Mechanical	34: Motorcycle	35: Music
36: Musical Instr.	37: Nature	38: Ocean Waves
39: Other Event	40: Percussion	41: Perc. Instr.
42: Rain	43: Rhythm/Beat	44: Scissors
45: Shouting	46: Singing	47: Speech
48: Strings	49: Tearing	50: Thunder
51: Flush	52: Tools	53: Train
54: Truck/Bus	55: Typing/Writing	56: Vehicle
57: Water Running	58: Weather	59: Wild Animals
60: Wind	61: Wind Instr.	

Table 2: Canonical 62-class event taxonomy derived from FSD50K.

Overall, the combined dataset preparation pipeline produces clean, uniformly formatted, and label-consistent log–mel spectrograms suitable for supervised learning across both environment and event classification tasks.

4 Preprocessing and Feature Extraction

Both the environment and event classifiers operate on log–mel spectrograms derived from 10-second mono audio segments. A preprocessing pipeline for both datasets standardizes raw audio into consistent feature tensors suitable for supervised learning.

4.1 Waveform Standardization

All clips are converted to 16 kHz mono and normalized to a fixed 10-second duration. Environment audio from AudioSet is preprocessed using `environment_preprocess.py`, which scans both the balanced and unbalanced environment directories, avoids duplicate files, and processes each unique WAV file. Event audio from FSD50K is handled by `events_preprocess.py` based on a manifest that records clip IDs and file paths.

For each input file, the pipeline performs:

(1) **Loading and resampling.** Raw audio is loaded with `librosa` [5] at 16 kHz in mono.
(2) **Peak normalization.** The waveform is scaled by its maximum absolute value to avoid clipping and ensure consistent dynamic range.
(3) **Duration fixing.** If the clip is shorter than 10 seconds, zero-padding is applied; if longer, it is center-cropped (or truncated) to exactly 10 seconds.

The resulting standardized waveforms are written back to disk as cleaned WAV files for reproducibility and potential reuse.

4.2 Log–Mel Spectrograms

From each normalized waveform, a log–mel spectrogram is computed using a common configuration for both tasks:

- Sample rate: 16 kHz
- Number of mel bands: 128
- Hop length: 512 samples
- Maximum frequency: 8 kHz

The mel power spectrogram is converted to decibels using `librosa` [5], with values clipped to a fixed decibel range (approximately $[-80, 0]$) for numerical stability. Environment features are stored under `data/environment/processed/features` and event features under `data/events/processed/features`, each as a NumPy file containing a 2D array of shape (mel, time).

These spectrograms serve as the direct inputs to the convolutional models. During training, a channel dimension is added so that each example has shape $(1, F, T)$, where F and T represent frequency and time, respectively.

4.3 Index Files and Metadata

To connect features with labels and metadata, both pipelines construct index files that summarize all usable samples. For the environment head, `01_environment_setup.py` merges metadata from the balanced and unbalanced AudioSet subsets, maps raw label strings to the eight environment categories, and produces a `data_index` file that stores, for each clip, the feature path and integer label ID. A similar script, `01_events_setup.py`, processes FSD50K metadata to create a canonical 62-class taxonomy and multi-label index with per-clip lists of label IDs.

These index files are saved as both CSV and Parquet for convenience and are the only sources consulted by the training scripts. Any clip that fails preprocessing (e.g., missing file, decode error, or mismatch between metadata and features) is excluded.

4.4 Consistency Across Tasks

Using a unified preprocessing backbone for both environment and event data provides several benefits:

- **Shared feature space:** Both models learn from spectrograms with identical frequency × time resolution and scaling.
- **Simplified deployment:** The real-time demo can compute a single log–mel representation and feed it to both heads without having to do each process uniquely.
- **Easier experimentation:** Changes to spectrogram parameters or normalization strategies can be applied across all models and datasets.

This standardized preprocessing pipeline forms the foundation for the CNN architectures and training procedures described in Section 5.

5 Models and Training

Both classification heads in the Surround Sound system use convolutional neural networks operating on log–mel spectrogram inputs. Although the environment and event models share a similar architectural backbone, they differ in output dimensionality and loss formulation due to the single-label vs. multi-label nature of the tasks [4, 8].

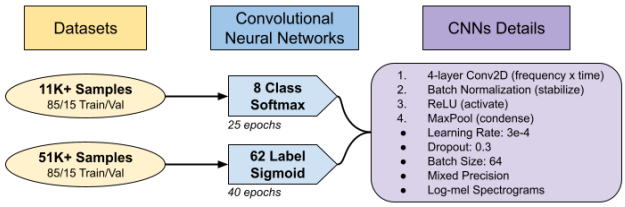

Figure 1: Overview of the environment and event CNN architectures used in the project.

5.1 Environment Classifier

The environment classifier predicts one of eight mutually exclusive scene labels. Its architecture is implemented in `02_environment_training.py` and consists of four convolutional blocks with progressively increasing channel depth. Each block contains:

- a 2D convolution (`Conv2d`) with kernel size 3×3,
- batch normalization,
- ReLU activation,
- 2×2 max pooling.

After the convolutional stack, global average pooling reduces spatial dimensions, followed by a fully connected layer that outputs an 8-dimensional logit vector. A final softmax operation produces class probabilities.

The network is trained using cross-entropy loss with optional class weighting to compensate for class imbalance. Optimization uses Adam with a base learning rate of 3×10^{-4} and cosine annealing. Training is performed for 25 epochs using mixed precision (`torch.cuda.amp`) to reduce memory usage.

5.2 Event Classifier

The event classifier predicts the presence or absence of 62 canonical sound events derived from FSD50K metadata. Because multiple events may co-occur in each clip, the classifier uses a multi-label formulation with sigmoid activations.

The architecture, defined in `02_events_training.py`, follows a deeper convolutional stack than the environment model to accommodate the greater label complexity. The model includes:

- three convolutional stages with channel depths from 64 to 256,
- batch normalization and ReLU activations,
- max pooling between stages,
- global average pooling,
- connected 62-dimensional output layer.

A sigmoid function is applied independently to each output dimension. Training uses binary cross-entropy with logits (BCEWithLogitsLoss). To mitigate extreme class imbalance in FSD50K, class-dependent weight values are computed from label frequencies and passed directly into the loss function.

The optimizer is Adam with learning rate 2×10^{-4} and cosine annealing. Mixed precision is enabled for reduced VRAM usage. During validation, event presence is determined via thresholding, and the threshold is tuned globally to optimize macro-F1.

5.3 Batching, Augmentation, and Sampling

Training uses batch sizes ranging from 32–64. Spectrograms are stacked into tensors of shape $(B, 1, F, T)$, where F=128 mel bands and T varies with hop length. Lightweight augmentation (e.g., noise scaling, time masking) was tested but not heavily used due to limited performance gains relative to training time.

For the event model, label imbalance needs additional handling. Rare classes receive increased gradient emphasis through weights, while the data loader samples uniformly across the training set without oversampling to preserve clip diversity.

5.4 Training Stability and Convergence

Both models benefit from mixed-precision training and cosine learning-rate schedules, which improve stability over long training runs on a 6 GB VRAM GPU. Early stopping is avoided in favor of full training cycles. The validation metrics tend to vary greatly in early epochs.

The final environment and event classifiers reach stable convergence and are summarized in Section 6.

6 Experimental Results

This section summarizes the performance of both classifiers on held-out validation sets. All results use log–mel spectrogram inputs generated by the unified preprocessing pipeline described in Section 4. We train all models using PyTorch [7] on a laptop with an NVIDIA RTX 3060 Ti (6 GB VRAM) and a Ryzen 9 5900HS CPU. Environment and event datasets are split into training and test sets as described in Section 3.

6.1 Environment Classification Results

Table 3 presents the macro- and weighted-average metrics for the 8-way environment classifier. Macro-averaged metrics treat each class independently, while weighted metrics reflect class frequency.

Metric	Macro	Weighted
Precision	0.5404	0.6818
Recall	0.6376	0.4666
F1-score	0.4985	0.4466

Table 3: Macro and weighted average metrics for the environment classifier.

Class	Prec.	Rec.	F1	Support
animal	0.7271	0.5631	0.6347	705
crowd	0.4033	0.9932	0.5737	292
indoor	0.4106	0.7745	0.5367	439
other_ambient	0.5132	0.8584	0.6424	226
outdoor_nature	0.2853	0.5753	0.3815	365
vehicle_ground	0.9191	0.1842	0.3069	1911
vehicle_water	0.2848	0.7917	0.4189	533
water	0.7795	0.3605	0.4930	1118

Table 4: Per-class precision, recall, F1-score, and support for the 8-class environment classifier.

Performance varies substantially by class. High-frequency classes such as *crowd* and *other_ambient* achieve higher recall, while acoustically subtle classes like *vehicle_water* and *indoor* show lower F1-scores. These discrepancies are primarily attributed to dataset imbalance and the inherent acoustic similarity between several categories.

To visualize label-level confusion, Figure 2 displays the normalized confusion matrix for the 8-class classifier.

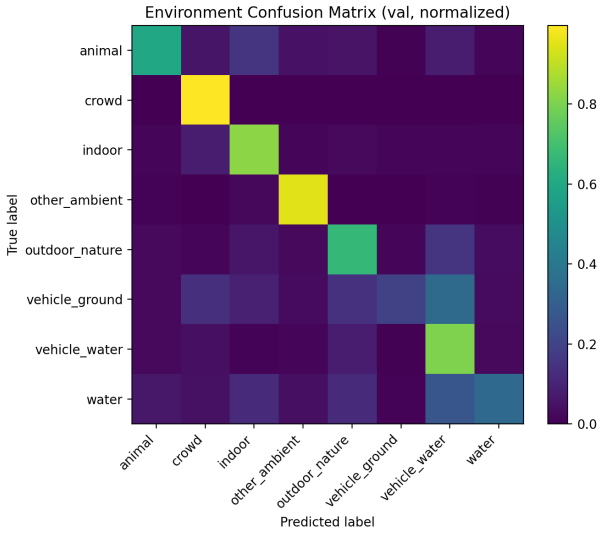

Figure 2: Normalized confusion matrix for the 8-class environment classifier.

6.2 Event Classification Results

Table 5 reports overall macro- and weighted-averaged metrics for the 62-label event classifier.

These results reflect the inherent difficulty of long-tailed multi-label classification. Many sound events in FSD50K occur infrequently, and several exhibit significant acoustic overlap (e.g., *bird*, *bird chirp*, *bird call song*). The high macro recall relative to precision suggests that the classifier frequently identifies candidate events but generates false positives for semantically similar classes.

Metric	Macro	Weighted
Precision	0.33	0.58
Recall	0.62	0.58
F1-score	0.39	0.54

Table 5: Macro and weighted metrics for the 62-class event classifier.

Class	F1	Prec.	Rec.	Support
Music	0.86	0.92	0.80	2512
Musical Instr.	0.79	0.84	0.75	1937
Percussion	0.69	0.62	0.77	588
Perc. Instr.	0.69	0.62	0.77	588
Rhythm Beat	0.69	0.62	0.77	588

Table 6: Top 5 event classes by F1-score.

Class	F1	Prec.	Rec.	Support
Scissors	0.15	0.09	0.52	21
Aircraft	0.15	0.09	0.77	35
Wind	0.14	0.08	0.53	59
Mechanical	0.18	0.10	0.64	28
Tools	0.17	0.10	0.61	28

Table 7: Bottom 5 event classes by F1-score.

6.3 Per-Class Performance

Per-class metrics reveal patterns that are not visible from macro averages alone. For the environment classifier, several trends emerge:

- **Clear, high-SNR classes perform best.** Labels such as *crowd* and *outdoor_nature* show higher F1-scores because they contain distinct sound signatures.
- **Acoustically similar scenes cause confusion.** Classes like *indoor*, *other_ambient*, and *vehicle_water* overlap heavily in background noise patterns, leading to lower precision.
- **Rare categories underperform.** Limited representation of some classes reduces the model's ability to learn stable features.

A similar pattern appears in the event classifier:

- **Common events generalize well.** Events such as *Speech*, *Bell Chime*, and *Car Engine* achieve relatively strong performance due to abundant training examples and clear spectral cues.
- **Fine-grained or similar labels are difficult.** Classes like *Bird*, *Bird Chirp*, and *Bird Call Song* often co-occur acoustically, making exact boundary distinctions challenging.
- **Long-tailed categories struggle.** Events such as *Wind Instruments* or *Insect Buzz* have fewer samples and higher variability, depressing recall.

Overall, the accuracy distribution across classes reflects the realities of real-world audio: overlapping sources, uneven label frequencies, and subtle differences in timbre that lightweight CNNs may not fully capture.

6.4 Discussion

The results highlight several practical challenges that shape the performance of both models:

- **Class imbalance remains the dominant issue.** Many event classes in FSD50K appear only a handful of times. Even with weight balancing, the model struggles to form strong representations of these rare labels.
- **Label noise affects the environment classifier.** Because AudioSet clips originate from YouTube, many samples contain incorrect labels, low-quality audio, or mismatched timestamps. These inconsistencies tend to blur boundaries between visually similar categories such as *indoor* and *other_ambient*.
- **Lightweight CNNs have limited expressive capacity.** The models are intentionally compact for real-time use, but this reduces their ability to differentiate between fine-grained events or model long-term temporal structure.
- **Some confusion comes from genuine acoustic overlap.** Certain categories are simply hard to separate. For example, *wind* vs. *ocean waves*, or the family of bird-related labels all sound similar even to the human ear. A lightweight CNN cannot currently handle the intricacies between very similar sounding noises, especially without surrounding context.

Despite these limitations, the classifiers are sufficiently stable for real-time use and produce coherent outputs when combined with the live microphone pipeline. The observed challenges also point toward clear next steps, including improved data balancing, better thresholding strategies, and exploring architectures that can model longer temporal context.

7 Supplemental: Real-Time Demo and Inference Pipeline

In addition to offline evaluation, Surround Sound includes a real-time demonstration that classifies live microphone audio using both the environment and event models. Although not part of the formal final paper grading criteria, the demo provides a practical illustration of how the system behaves in real scenarios.

7.1 Audio Streaming and Feature Extraction

The demo captures audio between 1 and 15 seconds and processes it through the same preprocessing pipeline described in Section 4. Each window is:

- recorded in real time at 16 kHz mono,
- normalized and padded to a fixed duration,
- converted into a log–mel spectrogram using the unified feature extractor.

Because the models assume 10-second inputs, the streaming version cuts longer recordings to match the expected input.

7.2 Dual-Head Inference

Once a spectrogram window is computed, it is passed simultaneously to the environment classifier and the event classifier:

- The **environment head** produces an 8-class softmax distribution, estimating the most likely background scene.
- The **event head** outputs the best 5 independent sigmoid probabilities for the presence of each of the 62 labels.

Simple probability thresholds are applied to the event predictions to generate a compact list of active events. Thresholds are tuned on the validation set to balance recall and precision.

7.3 Natural-Language Scene Summarization

To present results in a human-friendly format, the demo optionally uses a large language model to convert raw classifier outputs into a short textual summary. The summarizer receives:

- the top environment prediction,
- the subset of event labels above threshold,
- and the model's confidence values.

The LLM then generates a brief sentence such as: *"You may be in an outdoor nature environment with birds, water flow, and wind present."*

This summarization layer is an interpretability aid and does not influence the model in any way.

7.4 User Interface and Visualization

A simple Streamlit [1] interface displays the live spectrogram, classifier outputs, event probabilities, and generated summary. Users can start or stop live recording, inspect prediction trends, and observe how small acoustic changes influence the models. Although the interface is lightweight, it provides valuable insight into the behavior of both classifiers under real-time constraints.

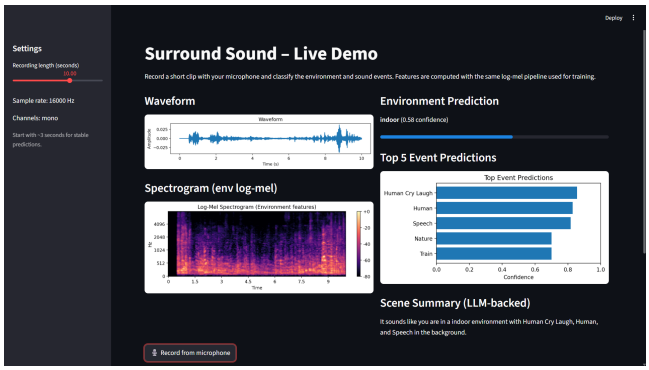

Figure 3: Example live demonstration of Surround Sound on a noisy indoor classroom recording.

7.5 Pipeline Overview

Figure 4 summarizes the architecture of the real-time system, from audio capture to dual-head classification and optional LLM-based interpretation.

Figure 4: Overview of the real-time inference pipeline used in the demo.

In summary, the demo highlights the strengths and limitations of the classifiers when deployed in a realistic streaming context.

Despite occasional noisiness in the outputs, the models successfully capture broad scene characteristics and identify salient events, demonstrating the feasibility of low-latency multi-task audio understanding.

8 Conclusion and Future Work

This project introduced Surround Sound, a two-head audio classification system capable of identifying background environments and fine-grained sound events from both offline audio and real-time microphone input. Using a unified preprocessing pipeline and lightweight CNN architectures, the system produces consistent log–mel spectrogram inputs and achieves moderate performance on two large-scale datasets: AudioSet for environment scenes and FSD50K for sound events. While the models are intentionally compact to support real-time use, they demonstrate meaningful predictive ability and provide a strong foundation for future development.

The experimental results highlight several practical challenges, including dataset imbalance, label noise, and acoustic overlap between similar classes. These issues limit accuracy but reveal many areas of relatively feasible improvement. The real-time demo further illustrates how the system behaves in dynamic acoustic environments as well.

Future work will focus on several directions:

- **Enhancing data balance and variety.** Increasing the diversity and quantity of training examples, or adjusting how samples are selected, may help the models better handle rare or subtle sounds.
- **Exploring stronger model designs.** Trying more flexible or modern neural network architectures may allow the system to capture richer patterns in audio while still running efficiently in real time.
- **Organizing labels into clearer groups.** Structuring related environment and event categories together could reduce confusion between similar sounds and lead to more intuitive predictions.
- **Training models specifically for streaming audio.** Teaching the models to work directly with shorter or shifting audio windows may improve stability during real-time use.
- **Combining tasks more tightly.** Integrating environment and event prediction into a single unified model could help each task benefit from the other and produce more coherent scene summaries.
- **Adding additional specialized models.** Extending the framework with new heads, such as a dedicated speech or language model, could further enrich scene understanding once the core environment and event components are mature.

Overall, Surround Sound provides a flexible and extensible framework for multi-level auditory scene understanding. With further refinement and exploration of the directions above, the system has the potential to support more robust, context-aware audio interpretation in both offline and real-time settings.

References

[1] 2021. Streamlit — The fastest way to build data apps. https://streamlit.io. Accessed: 2025-01-01.

[2] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. 2021. FSD50K: An Open Dataset of Human-Labeled Sound Events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 979–993.

[3] Jort F Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. AudioSet: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 776–780.

[4] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, and Dylan Platt. 2017. CNN Architectures for Large-Scale Audio Classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 131–135.

[5] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in Python. In *Proceedings of the 14th Python in Science Conference (SciPy 2015)*. 18–25.

[6] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. 2018. A Multi-Device Dataset for Urban Acoustic Scene Classification. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*. 9–13.

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*.

[8] Yuki Tokozume and Tatsuya Harada. 2017. Learning Environmental Sounds with End-to-End Convolutional Neural Networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2721–2725.