JOHANNES KEPLER
UNIVERSITY LINZ

UE MLPC 2025: CHALLENGE

Tara Jadidi, Florian Schmid, Paul Primus
2025-05-26
Institute of Computational Perception

JOHANNES KEPLER
UNIVERSITY LINZ

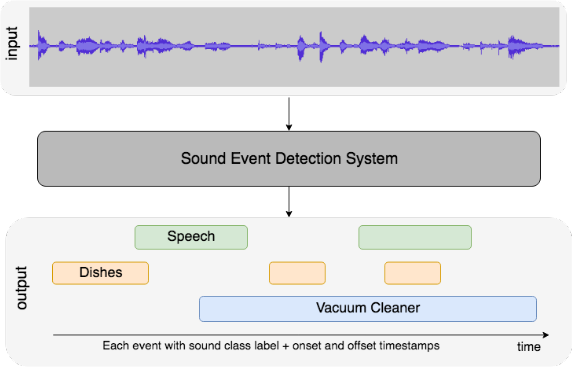
Menu for Today

- State of the Project & Schedule
- Task 3: Selected Presentations
- Task 4: Intro to Challenge

STATE OF THE PROJECT & SCHEDULE

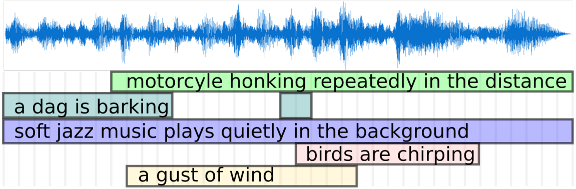
The Project Vision

- Goal: Train models that can detect a set of arbitrary sound events with their respective onsets and offsets.



The MLPC2025 Data Set

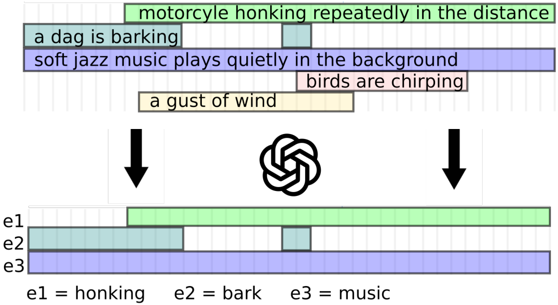
- Strategy: Annotate with free-text instead of a closed set of events.



Map Text to Class Labels

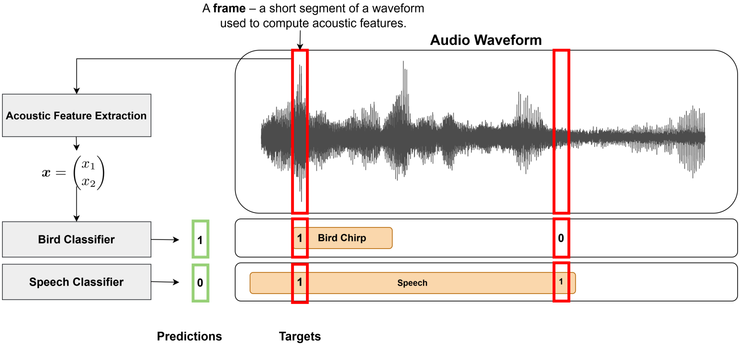
We used an LLM to map the text annotations to 58 classes.

- Each annotation was mapped to one or multiple classes.
- Annotations that cannot be mapped were ignored.



Training Classifiers in a Nutshell

Train a separate binary classifier for each of the 58 classes.



- Each frame is represented as a feature vector and can be used as a training example.

Project Schedule

		Date/Deadline
Meeting 1	Introduction, explain Tasks 0 and 1	March 10 ✔
Task 0	Form teams	March 24 ✔
Task 1	Data Annotation	March 24 ✔
Meeting 2	Release dataset, explain Task 2	April 7 ✔
Task 2	Data Exploration	April 24 ✔
Meeting 3	Discuss results, explain Task 3	April 28 ✔
Task 3	Classification Experiments	May 22 ✔
Meeting 4	Present results, release test data, explain Task 4	May 26 ◀
Task 4	The Challenge	June 18
Meeting 5	Final presentations	(12:00!) June 23

TASK 3: CLASSIFICATION EXPERIMENTS

Classification Experiments

Goals:

- Assess labeling function quality and identify key audio features.
- Explain data splits and leakage prevention.
- Summarize feature selection and preprocessing.
- Choose an appropriate evaluation metric and establish baseline performance.
- Train, tune, and evaluate classifiers.
- Visualize and assess classifier predictions.

Selected Presentations

1. **Team Observe**: Labeling Functions & Audio Features
2. **Team Waste**: Data Split & Evaluation
3. **Team Far**: Experiments
4. **Team Arrange**: Analysing Predictions

TASK 4: CHALLENGE

A Customer Approaches KIAL

The customer wants to monitor **urban noise pollution.** Their aim is to:

- Understand which acoustic sources contribute most to the noise pollution.
- Quantify their individual impact.
- Develop countermeasures for high-impact sources.

A Customer Approaches KIAL

Goal: Detect the *temporal occurrence* of 10 common urban noise events:

Speech, Dog Bark, Rooster Crow, Shout, Lawn Mower, Chainsaw, Jackhammer, Power Drill, Horn Honk, Siren

This is a typical **Sound Event Detection (SED)** task:

- **What** sound occurred?
- **When** did it happen?

The Customer's Secret Test Set

To select the best-performing Sound Event Detection (SED) system, the customer:

- Provides a **test set** of audio recordings *without annotations*.
- Evaluates submitted systems based on a **cost-based metric.**
- Awards the contract to the system with the **lowest total cost.**

Predictions Format

Task: Predict presence of sound events in **1.2s non-overlapping segments**.

Submission: One CSV file with binary predictions per class.

`filename`	`onset`	`Speech`	`Dog Bark`	`Siren`	⋯
1922.mp3	0.0	1	0	0	⋯
1922.mp3	1.2	1	0	0	⋯
1922.mp3	2.4	1	0	1	⋯
⋮	⋮	⋮	⋮	⋮	⋱

Each row corresponds to the predictions for one particular 1.2s segment in one specific file.

Cost-Based Evaluation

- Each sound class has **custom costs** for FP and FN.
- TP and TN incur **zero cost**.
- Final score: *average cost per minute*, averaged across all classes.

Class	TP	FP	TN	FN
Speech	0	1	0	5
Dog Bark	0	1	0	5
Rooster Crow	0	1	0	5
Shout	0	2	0	10
Lawn Mower	0	3	0	15
Chainsaw	0	3	0	15
Jackhammer	0	3	0	15
Power Drill	0	3	0	15
Horn Honk	0	3	0	15
Siren	0	3	0	15

Evaluation: Step 1 — Generate Ground Truth

Use `get_ground_truth_df` to create a reference CSV for your custom test set.

```
from compute_cost import get_ground_truth_df

df = get_ground_truth_df(
    list_of_files_in_custom_test_split,  # e.g., [123.wav, ...]
    path_to_the_development_set          # e.g., path/to/MLPC2025_dataset
)
df.to_csv("ground_truth.csv", index=False)
```

This creates `ground_truth.csv` with binary labels per 1.2s segment.

Evaluation: Step 2 — Generate Predictions

Use `get_segment_prediction_df` to convert your model's outputs (120 ms frames) into the required prediction format.

```
from compute_cost import get_segment_prediction_df

df = get_segment_prediction_df(
    model_outputs_dict,    # e.g., {'123.wav': {'Siren': [0, 1, 1, ...]}}
    class_names            # e.g., ['Speech', 'Dog Bark', 'Siren', ...]
)

df.to_csv("predictions.csv", index=False)
```

This creates `predictions.csv` with one row per 1.2s segment and predictions for each class in `class_names`.

Evaluation: Step 3 — Compute Cost

Run the script to evaluate predictions against the ground truth.
Returns *average cost per minute*.

Command:

```
python compute_cost.py \
  --dataset_path=path/to/MLPC2025_dataset \
  --ground_truth_csv=path/to/ground_truth.csv \
  --predictions_csv=path/to/predictions.csv
```

Arguments:

- `dataset_path` — must contain `audio_features/`
- `ground_truth_csv` — from `get_ground_truth_df`
- `predictions_csv` — from `get_segment_prediction_df`

Evaluation: Step 4 — Check CSV Format

To validate the structure of your predictions file on the secret test set:

```
python compute_cost.py \
  --dataset_path=path/to/MLPC2025_dataset \
  --predictions_csv=path/to/predictions.csv
```

This checks whether the CSV:

- Covers all required classes
- Matches the expected onsets of 1.2s segments
- Has valid binary values (0 or 1)

TASK 4: SUBMISSION GUIDELINES AND DELIVERABLES

Report: What to Investigate (1/4)

Evaluation Setup[1]

1. How did you split the development dataset?
2. Implement a **naive baseline system** and report its cost.

[1] Refer to the task description for full details.

Report: What to Investigate (2/4)

Build A Simple SED System ...[2]

... to predict the presence and absence of the 10 sound classes for 1.2s segments.

1. How did you threshold and combine predictions to derive a label for the 1.2-second segments?
2. What strategies did you apply to minimize the cost-based metric?
3. Does your initial SED system outperform the naive baseline?

[2] Refer to the task description for full details.

Report: What to Investigate (3/4)

Improve Your SED System[3]

Explore **three different improvement strategies**:

1. For each, state your hypothesis and verify or falsify it experimentally.

[3] Refer to the task description for full details.

Report: What to Investigate (4/4)

Real-World Deployment[4]

1. Could your final system be deployed in a real-world application? What aspects would need to be adapted to meet real-world requirements?

[4] Refer to the task description for full details.

Report: What to Investigate (5/4) — Bonus

Fine-Tune the Embedding Model[5]

- Audio embeddings were extracted using a *pre-trained, frozen* transformer from the PretrainedSED repository.
- Fine-tune one of the transformers on the MLPC dataset end-to-end.
- Compare its performance to your best existing system.

[5] Refer to the task description for full details.

Challenge Task: Report

Compile a technical report that addresses all the previous questions.

- One report per group.
- Template is available on Moodle.
- max. 6 pages (including tables, figures)
  - max. 4 pages of text (excluding tables, figures)
  - bonus question can occupy one extra page

Challenge Task: Slides

In addition to your report, prepare a short presentation.

Requirements:

- Include a **system architecture overview**.
- Present your **most interesting hypotheses** and their outcomes.
- Limit: **max. 6 slides** + 1 title slide.

Challenge Task: Submission

- Submit your report and slide deck as two separate PDF files via Moodle by June 18th (Wednesday), 23:59.
- Submit your predictions csv file via Moodle by June 18th (Wednesday), 23:59.
- Selected groups will be asked to present their results in the lecture timeslot (12:00-13:30) on June 23rd (Monday).
- Presenters will be informed on June 22nd (Sunday).
- At least one team member must be available to present in-person or via Zoom.

Challenge Task: Grading

- Completing all tasks **is mandatory** to pass the course!
- The report is worth 27 (+10 bonus) points and the slides 3 points
- Grading criteria for the report in the task description on Moodle.
- Submitting a day late will cost you $\frac{1}{3}$ of the total points:
  - Up to June 18th, 24:00: 100 %
  - June 19th 00:00–24:00: 66.66%
  - June 20th 00:00–24:00: 33.33%
  - Afterwards, we will not accept submissions.

Challenge Task: Group Restructuring

If there are inactive team members in your group, or if you are looking for a new team member, please contact florian.schmid@jku.at until this Friday (30th of June).

Best practices for teamwork:

- Coordinate early
- Distribute tasks
- Define deadlines
- Check in regularly

Challenge Task: Summary

- Completion of Task 4 is mandatory.
- Answer all aspects and the corresponding questions in your **written report**.
- Use the LaTeX template, stick to the page limit (6 pages, 4 pages text max.) and include a contributions statement.
- Create a **slide deck** with 6 slides + 1 title slide max.
- Create a **predictions.csv** file and check its format using the provided scripts.
- Upload until June 18th to get up to 30 (+10 bonus) points.

Tutorial Session — Monday, June 2

Topics Covered:

- Advanced classification concepts
- Introduction to **PyTorch**, **PyTorch Lightning**, and **Weights & Biases**
- Training a bidirectional recurrent neural network for Sound Event Detection on the MLPC dataset
- Generating ground truth and predictions csv, estimating costs, generating predictions on secret test set for submission