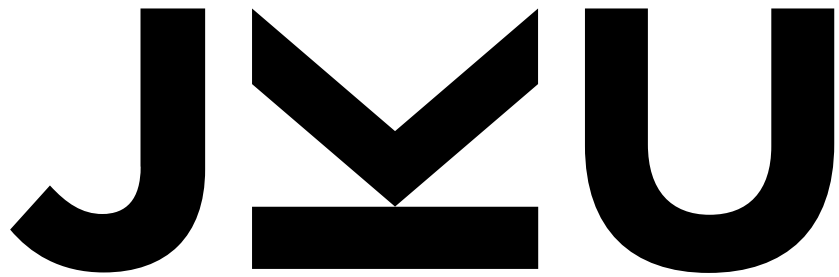


JOHANNES KEPLER
UNIVERSITY LINZ

UE MLPC 2024: CHALLENGE START

Katharina Hoedt, Verena Praher
Florian Schmid, Paul Primus

2024-05-27

Institute of Computational Perception

JOHANNES KEPLER
UNIVERSITY LINZ

WHERE ARE WE

Project Schedule

		Date/Deadline
Meeting 1	Introduction, explain Tasks 0 and 1	March 4 ✔
Task 0	Form teams	March 18 ✔
Task 1	Data Collection	March 13 ✔
Meeting 2	Release dev data, explain Task 2	March 18 ✔
Task 2	Data Exploration	April 10 ✔
Meeting 3	Discuss results, explain Task 3	April 15 ✔
Task 3	Classification Experiments	May 22 ✔
Meeting 4	Present results, release test data, explain Task 4	May 27 ◀
Task 4	The Challenge	June 19
Meeting 5	Final presentations	June 24

Today's Plan

1. Look back at the classification task
2. Have some teams present their results
3. Explain the challenge

Today's Plan

- We have multiple interesting presentations
- + want to set up the challenge appropriately
- This might take a bit longer - if you need to leave, feel free to do so (quietly)!

RECAP: CLASSIFICATION

Classification Task

What we gave to you:

- A dataset with $\approx$ 45,000 snippets of your recorded audio
- Pre-computed audio features + speech annotations

What you had to do:

- Train and evaluate at least 3 classifiers
- Analyze the effect of changes in hyperparameters
- Report your findings
- Experiment with 4 realistic scenes

What we got from you:

- 70 groups, handed in a total of ...
- ... 459 pages written reports and 275 slides

Title Slide Gallery

How did you do?

- How much time did you invest?
  - ☐ a day or less
  - ☐ 1–2 days
  - ☐ 3–5 days
  - ☐ more

How did you do?

- How much time did you invest?
- How was the page (text) limit for you?
  - ☐ too long
  - ☐ too short
  - ☐ about right

How did you do?

- How much time did you invest?
- How was the page (text) limit for you?
- How balanced was the group work?
  - ☐ pretty good
  - ☐ one of us did almost everything
  - ☐ one of us did nothing
  - ☐ half of us did nothing

About the group work:

1. Have you been ghosted by any of your team members and are underpowered now? Send us an email (katharina.hoedt@jku.at), we can merge such teams.
2. Give each other feedback (in time)!

TEAM PRESENTATIONS

Team Presentations

- Will ask some groups to present their main findings
- 5 minutes presentation, then a bit of time for questions
- Will hold up a sign at 4 minutes
- Team presentations and discussions will not be recorded!

Team Presentations

- Data Split: Team Color
- Classes / Features: Team Fumbling
- Evaluation: Team Noise
- Experiments: Team Pocket
- Analysis of Realistic Scenes: Team Sparkling

TASK 4: CHALLENGE

Background

- The (fictitious) Austrian company “SmartVoiceControl” is developing a smart home solution
- Turn on and off devices via **speech commands**
- App for smart devices (phones, tablets) that constantly listen for spoken commands
- App recognizes command $\rightarrow$ sends “turn on/off”-signal to respective household device
- The company lacks machine learning expertise
- They provide us with a cost function which is better aligned with the actual goals of the application

Data: Development and Test Set

Development:

- Speakers who gave consent for academic purposes.
- Word recordings.
- Scenes without speech commands split and used as “other” class.
- **New:** Scenes with speech commands + strong labels (annotations)

Test:

- Speakers who gave consent for use in the MLPC class.
- Scenes with speech commands.
- Will be released June 12th.

Challenge: What to look at (1)

1. Establish a naive baseline system
2. A starting point: Build a simple speech command detection system (e.g., with one of the keyword classifiers you trained in the previous phase of the project).
   - Describe how your system was used to detect keywords in the longer domestic recordings
   - How did you threshold and combine keyword predictions?
   - What strategies did you apply to minimize the task-specific cost function?
   - Describe your evaluation setup and provide evaluation results.
   - Comparison to the naive baseline system (step 1).

Challenge: What to look at (2)

3. Improvements:
   Investigate at least three diverse strategies to improve your starting point (e.g., via hyperparameter tuning, ensembling, data augmentation, training on scenes, ...)
4. Critical Reflection:
   Do you think your final system could be deployed in a real-world application? In your opinion, which aspects of the project or your system would need to be adapted to fulfill possible real-world requirements?

Evaluation: Cost Function

- True Positives: -1
- False Negatives: 0.5
- False Positives:
  - {“Fernseher”, “Licht”, “Radio”, or “Staubsauger”} + {“an” or “aus”}: 2
  - {“Heizung” or “Lüftung”} + {“an” or “aus”}: 3
  - {“Ofen” or “Alarm”} + {“an” or “aus”}: 4
- Cross-Triggers[1]:
  - incorrect device key word (“Licht” statt “Radio”): 1
  - correct device with incorrect action (“Licht an” instead of “Licht aus”): 0.1

Cost needs to be minimized!

[1] `https://pypi.org/project/psds-eval/`

Our baseline detector

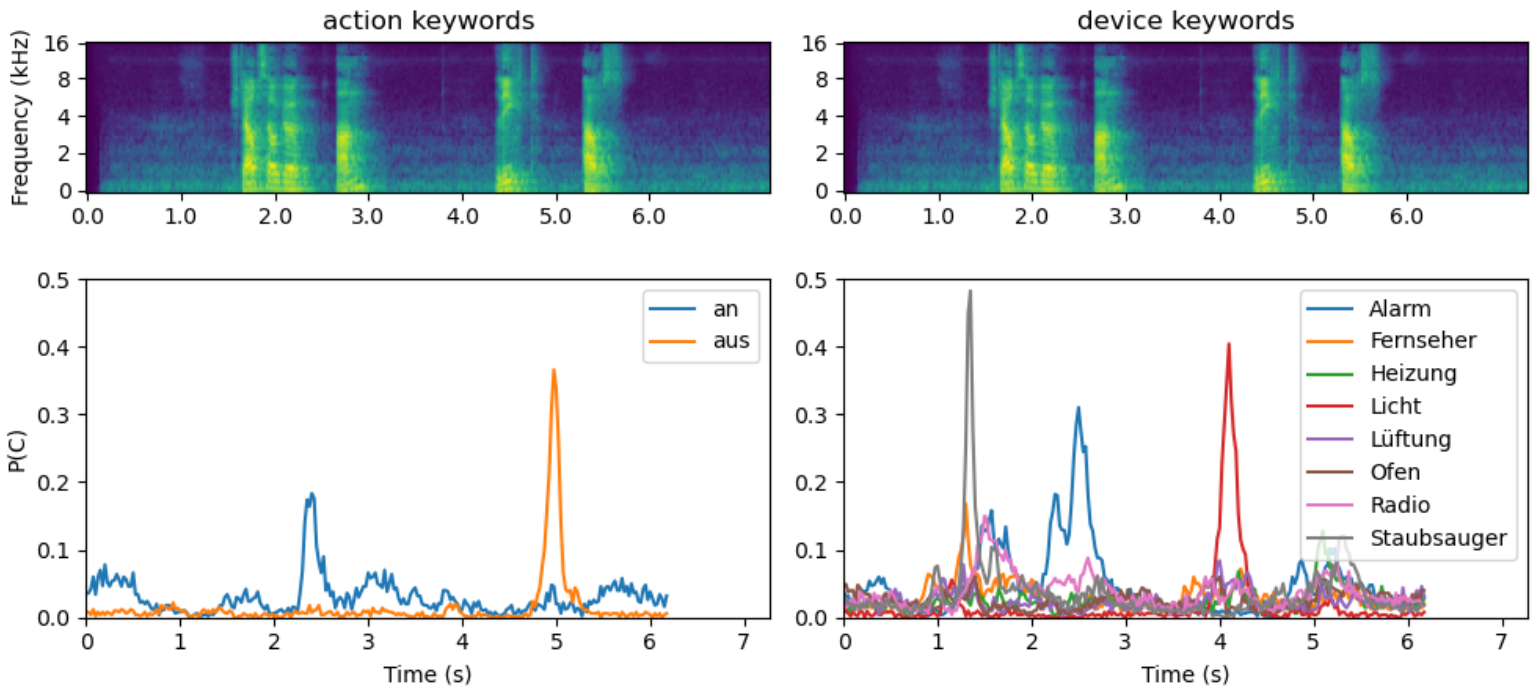
A simple Random Forest classifier with 500 trees

- trained on the 1.1s snippets in the keyword development set
- takes the 32 MFCC features as input
- outputs a posterior distribution over the 20 keywords and the 'other' class (21 classes in total)
- accuracy (on our custom validation set) is around 83%

Our baseline detector

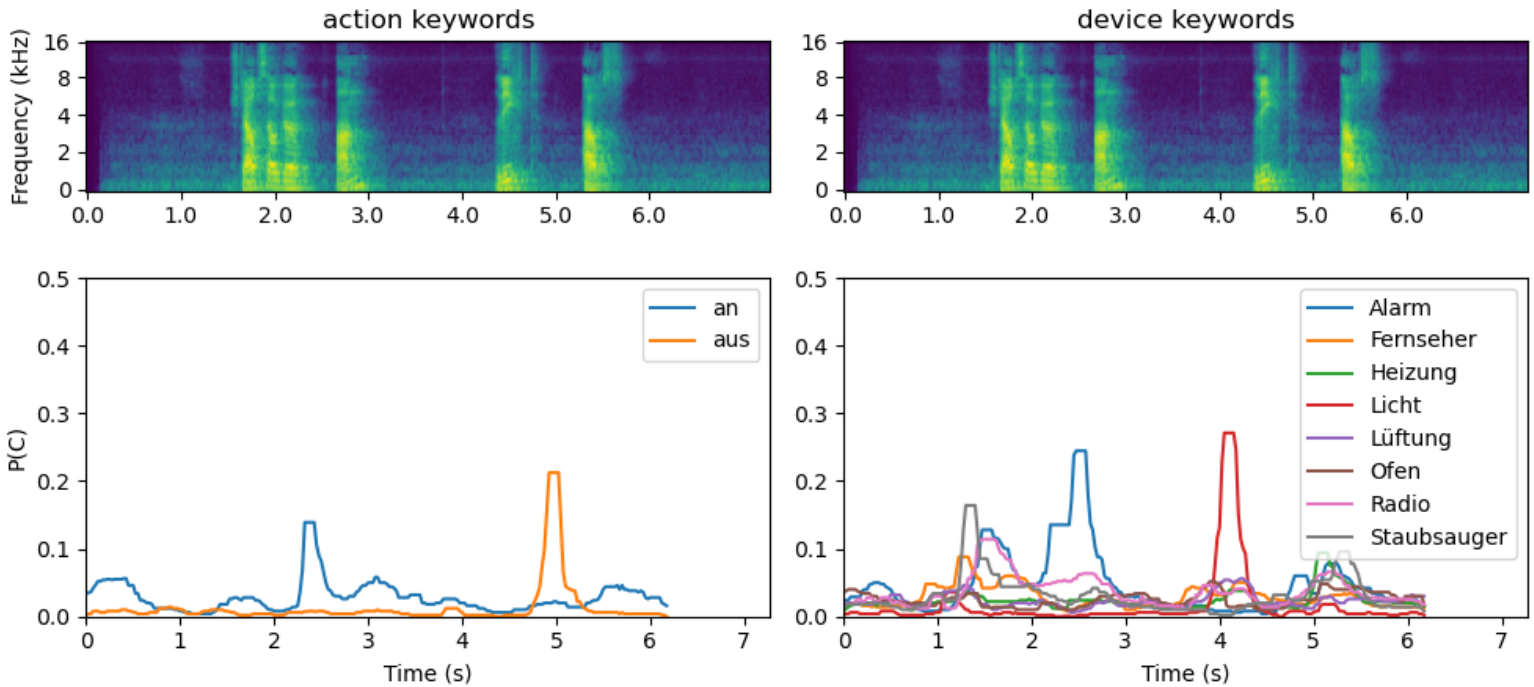
How to detect complete speech commands:

1. Extract overlapping 1.1s windows from the input and apply the classifier to each of them. Use a small hop size such as one time frame (25 ms).



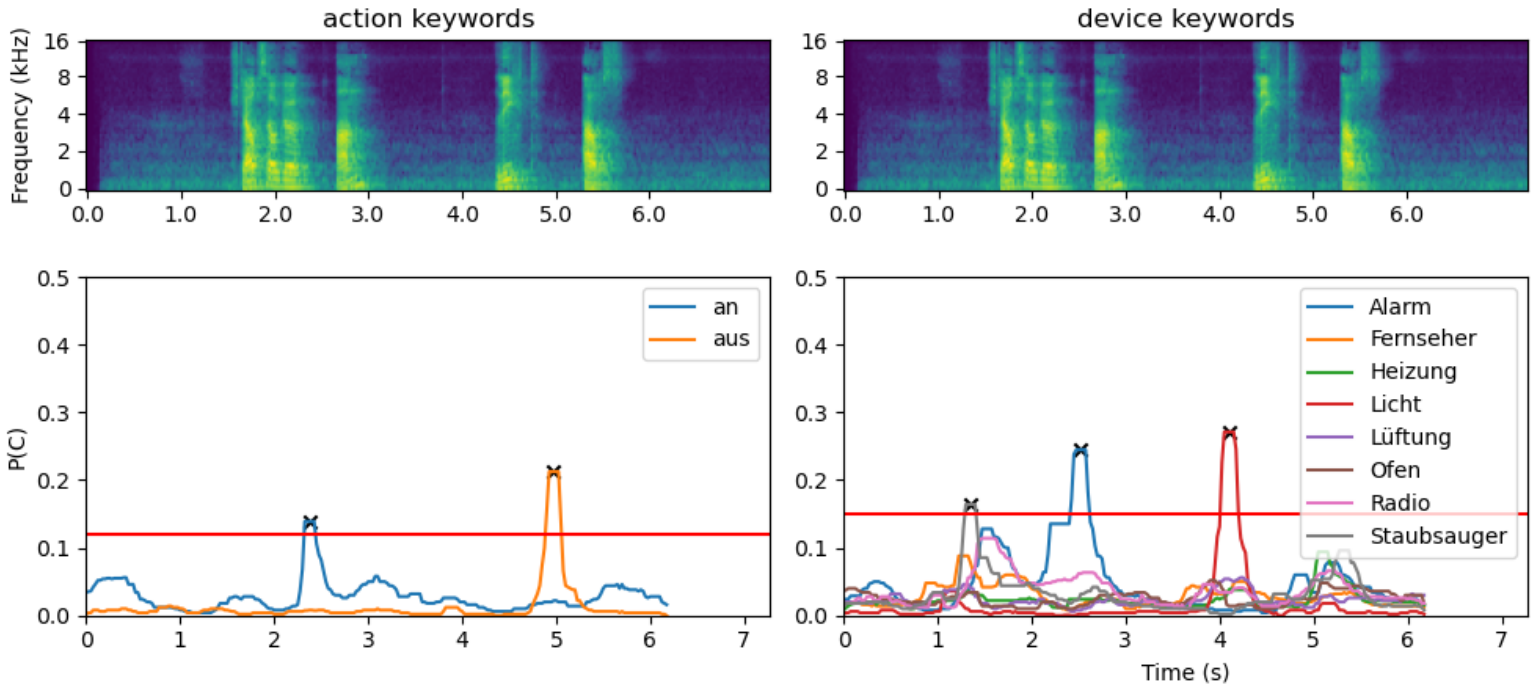
Our baseline detector

2. Apply a median filter to smooth the predictions.



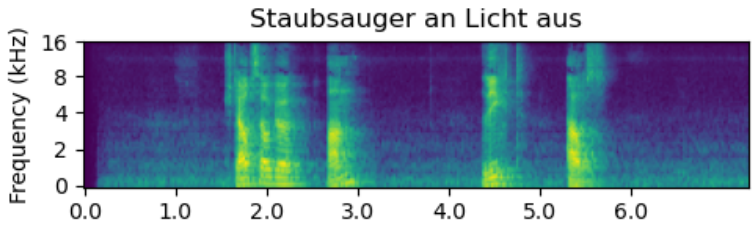
Our baseline detector

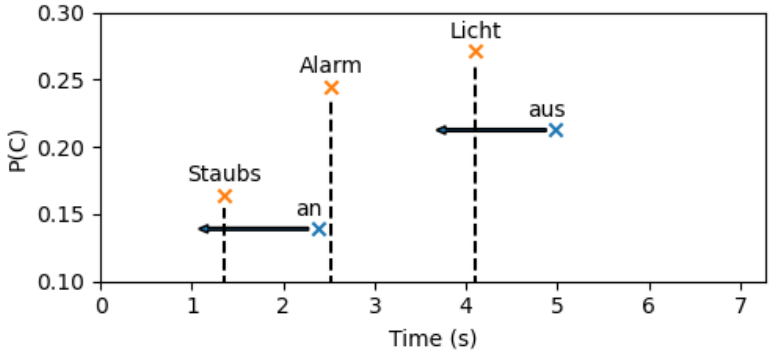
3. Find peaks in the predictions; use a minimum height and minimum distance between peaks.



Our baseline detector

4. Combine keyword peaks with a heuristic: We, for example, combined each action keyword with the most prominent device keyword within the preceding 1.2 seconds.





Our system's cost on a custom validation set is lower than not predicting anything.

Practical Hints

- Make sure to convert between frames and seconds correctly; the duration of one timestep of the pre-computed features corresponds to 25 ms (sample rate = 16kHz; STFT hop size = 400 samples)
- Do not implement all post-processing steps yourself! Use:
  - Median filtering: `scipy.signal.medfilt` (Matlab: `medfilt1`[2])
  - Peak picking: `scipy.signal.find_peaks` (Matlab: `findpeaks`[3])
- The recorded scenes have arbitrary lengths; extract overlapping windows before applying your classifier; avoid incorrect reshape operations!

[2] `https://de.mathworks.com/help/signal/ref/medfilt1.html`

[3] `https://de.mathworks.com/help/signal/ref/findpeaks.html`

Practical Hints

- Avoid data leakage when creating your own train/ validation splits for the new scenes data set!
- Using keywords and scenes from the same speaker might lead to overestimating the performance of the system.
- Use the speaker ID (provided in development_scenes.csv) to create your own scenes train/ validation split.

Practical Hints: Score Predictions

Provided Python script to score your predictions and to check the format of your submission file:

```
python score_predictions.py \
--predictions=predictions.csv \
--annotations=annotations.csv
```

- `predictions.csv` contains your system's output
- `annotations.csv` holds a list of events in your validation split
- use the `--check_format` flag to check the structure of your predictions file before uploading it to Moodle

Challenge: Predictions

Submit your predictions as a CSV File.

filename	command	timestamp
5bcf8028f5	Radio aus	4.102
…	…	…
92ba262ee	Licht aus	12.012

Figure: Schematic Representation of the CSV File.

Challenge: Report

Compile your results into a short report.

- Cover **all** of the four previous aspects, the corresponding subquestions, and other interesting things that you've discovered.
- You may use any kind of statistical computation or visualization that suits your purpose
- use the LaTeX template that is available on Moodle
- max. 6 pages (including tables, figures)
- max. 4 pages text
- include a statement about the contributions of each team member

Challenge: Slides

In addition to a detailed report, compile a short presentation.

- Describe the general architecture of your system
- Describe the most interesting hypotheses you investigated and their outcomes
- max. 5 slides + 1 title slide
- Create presentable slides!

Challenge: Submission

- Submit your predictions as a CSV file via Moodle by June 19th.
- Submit your report and slide deck as two separate PDF files via Moodle by June 19th.
- Selected groups will be asked to present their results in class on June 24th.
- At least one team member must be available to present in-person or via Zoom.

Challenge: Grading

- Completion of Task 4 (“Challenge”) is **mandatory**
- The report is worth 27 points and the slides 3 points
- Submitting a day late will cost you $\frac{1}{3}$ of the total points:
  - Up to June 19th, 24:00: 100 %
  - June 20th 00:00–24:00: 66.66%
  - June 21st 00:00–24:00: 33.33%
  - Afterwards, we will not accept submissions.

Challenge: Summary

- Provided to you:
  - Development data set (already available)
  - Test data set without annotations (Release date: June 12th)
  - Cost function provided by the company "SmartVoiceControl"
- Your task:
  - Provide speech command predictions for each scene in the test set
  - Focus on minimizing cost
  - Upload predictions, report (and slide deck) to get up to 30 points
- Deadline: June 19, 24:00