Project (we are updating it regularly)

EE698V – Machine Learning for Signal Processing

[Introduction 1](#_Toc57459722)

[TASK 1 2](#_Toc57459723)

[Evaluation 2](#_Toc57459724)

[TASK 2 2](#_Toc57459725)

[Evaluation 2](#_Toc57459726)

[Competition Rules 2](#_Toc57459727)

[Codes and Format 3](#_Toc57459728)

[Useful Resources 3](#_Toc57459729)

# Introduction

For the rest of the course, we will be having a hands-on project. You can do it individually or form a team of up to 2 students for the project. This project will serve as a part of end-sem. The other part will be either an MCQ or a coding exam.

There are two tasks given below – both will be evaluated. You can build your system and train it using the data given here. The evaluation will be done on a test set that will be released later around end sem. Students doing the best on test set will get more marks.

Final submission consists of at least the following:

* <yourRollNo>\_task1\_labels\_test.csv
* <yourRollNo>\_task2\_labels\_test.csv
* <yourRollNo>\_task1\_code.ipynb
* <yourRollNo>\_task2\_code.ipynb
* <yourRollNo>\_report.pdf

It is an audio classification problem, technically known as "Audio Event Detection". The content is available here:

<https://iitk-my.sharepoint.com/:f:/g/personal/vipular_iitk_ac_in/EqibR9AbEi9GiPeGGOZUeVsBn78UyhMHVF0hL_yAl8IxAg?e=c7upzH>

See README.docx for description.

Each audio file corresponds to an event class, e.g., children-playing, dog-barking, drilling, etc. The folder "audio\_train\_1ch/" contains the mono audio files (1761 wav files) and the file "labels\_train.csv" contains the class labels.

If you find g-drive to be convenient, the same mono audio files are also available at <https://drive.google.com/drive/folders/1idzxvhkKEFh5TbfZ4Rk9cMKZhtLcSbwJ?usp=sharing>

# TASK 1

Given an audio file corresponds to a single event, find out that event.



Task example: street\_music (right click to Play)

## Evaluation

Accuracy

# TASK 2

Given an audio file contains a sequence of events occurring one after the other, find out that sequence of events. A sequence can contain at least 1 and at most 5 events.



Task example: street\_music, dog\_bark, engine\_idling (right click to Play)

## Evaluation

Edit distance

* Return sequence of classes separated by hyphen "-".
  + E.g. street\_music-dog\_bark-engine\_idling
* Labels would not repeat consecutively, e.g., street\_music-dog\_bark-dog\_bark will be labeled as street\_music-dog\_bark

# Competition Rules

* No extra/external data allowed for training. You can use only the training data provided.
* Allowed libraries:
  + For data processing: Numpy, pandas
  + General libraries: glob, pdb, string, os, sys,
  + For ML: Scipy, Scikit-learn, Levenshtein
  + For audio processing/feature extraction: librosa
  + For deep learning: Keras, tensorflow, pytorch
  + For HMMs: <https://github.com/hmmlearn/hmmlearn> and <https://github.com/larsmans/seqlearn>
  + Let the instructor know if you want to use any other library
* You cannot use someone else's codes. You have to write your own codes and submit as a single ipynb file. But of course, you can use ideas from others' papers.
* The test set will not contain audio (wav) files but spectrograms. The feature extraction function is available in "shared\_train/utils.ipynb".
* The test set will be released for a limited time. You have to run your codes to predict the labels and submit in csv files. The format will be shared soon.
* Grading will be done based on evaluation results, the approach used and the quality of report (presentation – crisp and clear). Weightage: task 1 results (30%), task 2 results (30%), report and codes (40%).
* The report should explain your method/approach. It can be upto 2 pages (A4, 11 font size) long + 1 page for figures. You are encouraged to use figures/block-diagrams (a picture is worth 1000 words).

# Codes and Format

On the day of exam, we will release "feats/\*.npy" for both the tasks, as given in "sample\_test\_task\*/".

* Functions in utils.py are used to generate "feats/\*.npy"
* During test, you should have your codes ready. Read each npy file in a folder (say test\_task1/) (using e.g., glob) and write your prediction to a csv. Submit this csv. You have limited time (~1hr) to submit.
* For evaluation:

Your predicted labels (est.csv) should be in the same format as labels.csv

We will use eval\_model.py to get your score.

E.g.,

$ python eval\_model.py sample\_test\_task1/labels.csv est.csv 1

Your score is: 90/100

$ python eval\_model.py sample\_test\_task2/labels.csv est.csv 2

Your score is: 97/100

# Useful Resources

* (educational) AED talk by Justin Salamon <https://www.youtube.com/watch?v=zvccOFz2KxI&ab_channel=SpeechandAudiointheNortheast%28SANE%29>
* (interesting) Application of AED in Amazon Alexa: <https://www.youtube.com/watch?v=-nKelNVVblM&ab_channel=Amazonre%3AMARS>