
Layout:

PB:

- **CONFIG:** Contains configuration files(.yaml)
- **CHECKPOINTS:** To store model weights during training
- **SRC:** Contains all modules
  - **MODELS:** To define the DL model (Fingerprinter)
  - **INDEX:** To create a reference database of fingerprints and perform audio retrieval
  - **TRAIN:** To train the model
  - **UTILS:** Helping modules used by modules in INDEX, TRAINS. Also used by the *main.py* file.
- **main.py:** Integrates all the above modules. This is called for training the model.

Modules in subdirectories:

CONFIG:

- *main.yaml*: Used for parameters defined in *main.py.* This contains all the important parameters of the system.
- *create_refdbase.yaml*: Used for parameters defined in */src/index/create_refdbase.py*.
- *search.yaml:* Used for parameters defined in */src/index/search.py.*

MODELS:

- *custom_CNN.py*: DL model used as fingerprinter
- *feedforward.py:* projection layer (NN architecture)

TRAIN:

- *contrastive_learning.py:* Pytorch Lightning module for training the model.

UTILS:

- *audio.py:* Reads and preprocess the audio files.
- *callbacks.py:* Used during training to track progress
- *dataclass.py:* Custom datatype to store reference database. Helps in fast appending to numpy array.
- *dataset.py:* Custom dataset class compatible with our model training.

features.py: To transform raw audio into time-frequency representation.
losses.py: Loss metric defined used for training.
similarity.py: Similarity metric used to find similarity between embeddings during training.

main.py: Integrates all modules.
demo.ipynb: For audio retrieval demo purposes.

—---

Commands Execution:

For Training the model:

1. Update *main.yaml* file: mainly the paths corresponding to the train/validation data and noise/rir files need to be specified.
2. Execute the command from the src/ directory: *python main.py --subdir <repository name> --config <main.yaml path> -d <PB directory path>.* <repository name> will be created inside PB/checkpoints/ repository
3. To resume training from a checkpoint
    a. Execute command from the src/ directory: *python main.py --subdir <repository name> -c <checkpoint(*.ckpt) path> -d <PB directory path>*

For creating a reference database:

1. Update *create_refdbase.yaml* file: mainly the patch corresponding to reference audio files need to be specified
2. Execute the command from index/ directory: *python create_refdbase.py --config <create_refdbase.yaml path>*

For audio retrieval:

1. Update *search.yaml* file: Specify the fingerprints database and metadata paths and model weights path.
2. Execute the command from index/ directory: *python search.py --config <search.yaml path>.* For now, it will perform audio retrieval for 10 noisy query files of length 5s for demo purposes.

—---