Processing steps for encoding spectrograms into latent features and then using classifiers

Note that the 7 first steps are the same as the ones for classification and will NOT be repeated here. It is assumed that we already have created the CSV with file information and executed the frontend.

# Steps for mel spectrogram using fixed dimension

Assumes resizing was used.

8)

python machine-learning\fixed\_dim\_encoder\_clusterer.py --input\_file ..\mel\_D50T60\features\_melD50T60.hdf5 --triplet\_epochs 2 --epochs 100.

One may use parameter--continue\_training to load previously trained model and improve it.

9)

python machine-learning\write\_latent\_vectors.py --input\_file ..\mel\_D50T60\features\_melD50T60.hdf5 --input\_model ..\mel\_D50T60\features\_melD50T60\encoder\_models\encoder.h5

10)

python machine-learning\train\_dnn\_latent\_space.py --input\_file ..\mel\_D50T60\features\_melD50T60\encoder\_models\latent\_vectors.h5

11)

python machine-learning\evaluate\_classifier.py --input\_file ..\mel\_D50T60\features\_melD50T60\encoder\_models\latent\_vectors.h5 --input\_model ..\mel\_D50T60\features\_melD50T60\encoder\_models\best\_model.hdf5

12) To visualize results in case latent space dimension was chosen to be 3

Note: we call “tsne” below because it can be used for t-SNE results, but it is not mandatory to adopt t-SNE. This code can also be used in case the latent space dimension was chosen to be 3.

python machine-learning\plot\_3d\_tsne.py --input\_tsne\_file ..\outputs\3melD50T60\features\_melD50T60\encoder\_models\latent\_vectors.csv --input\_file ..\outputs\3melD50T60\features\_melD50T60.hdf5