Urban Sound Classification

FLAVIO FURIA

Introduction

- Predicting urban sounds is a research field of increasing interest in the recent years.

- It is a quite challenging task since sounds are recorded in a noisy environment and thus hard to classify.

- We work on the UrbanSound8K dataset, a collection of 8732 audio sounds coming from 10 possible different classes (street music, car horn, etc.).

Related Works

The most successful works on urban sound classification involve:

- 1D CNNs on raw audio signals

- 2D CNNs on spectrogram-based images

- Recurrent network, like LSTM, to exploit their capability of ‘‘remembering sequences’’.

Our Contribution

We compare two different approaches in terms of feature extraction and chosen models. We also show the issues of facing audio analysis with low available resources . In particular:

- Random Forest and Support Vector Machine are trained on data obtained by applying statistics on features coming from Time, Frequency, MFCCs and Chroma representations

- 2D Convolutional Neural Network are fed with melspectrogram images obtained from the same audio signals.

Data preprocessing and Feature Extraction for the first approach

- Using Librosa, each audio sample is imported with a sampling rate of 22050Hz and a single channel.

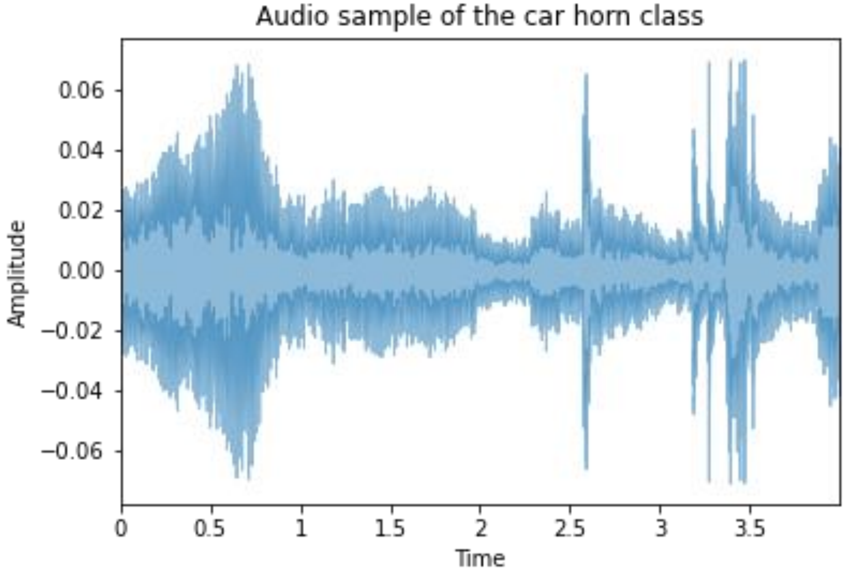
- The vast majority has a duration of exactly 4 seconds (the sorther ones are zero padded), resulting in 88200 values each.

- With a frame size of 1024 and a hop length of 256, features are extracted and statistics are computed from them, more precisely min, max, mean, median and standard deviation.

Time Domain Features

They rely on the raw audio signal, thus give information about how the amplitude changes over time. In particular, these features have been considered:

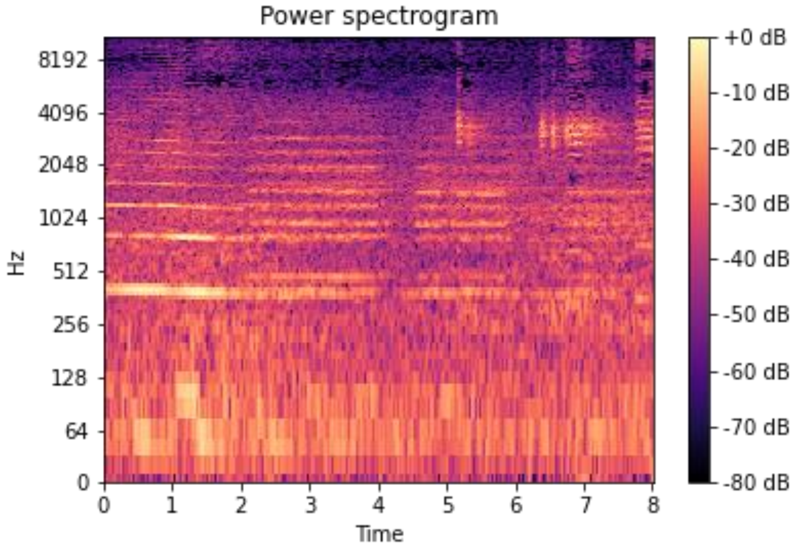
- amplitude envelope
- root-mean square energy
- zero crossing rate.



Frequency Domain Features

They rely on the spectrogram of the audio signal, obtained by applying the Fourier Transform in order to move to the frequency domain. In particular, the following features have been chosen:

- spectral centroid
- spectral bandwidth
- spectral rolloff.



Mel Cepstrum Representation

Mel is a logarithmic scale measuring pitches as a function of frequency. It well approximates how humans perceive sounds,thus it has found many application in speech recognition. In particular:

- The first 13 Mel-frequency Cepstral Coefficients are computed, together with the delta and the delta-delta
- In the end, each of the 39 row vectors is considered as a single features and used to compute statistics.

Chroma representation

This kind of representation is used to aggregate spectral information into a given number of bins, usually 12, representing harmonic characteristic of music. In particular:

- Chromagrams with 12 coefficients are computed and each vector is used to compute statistics

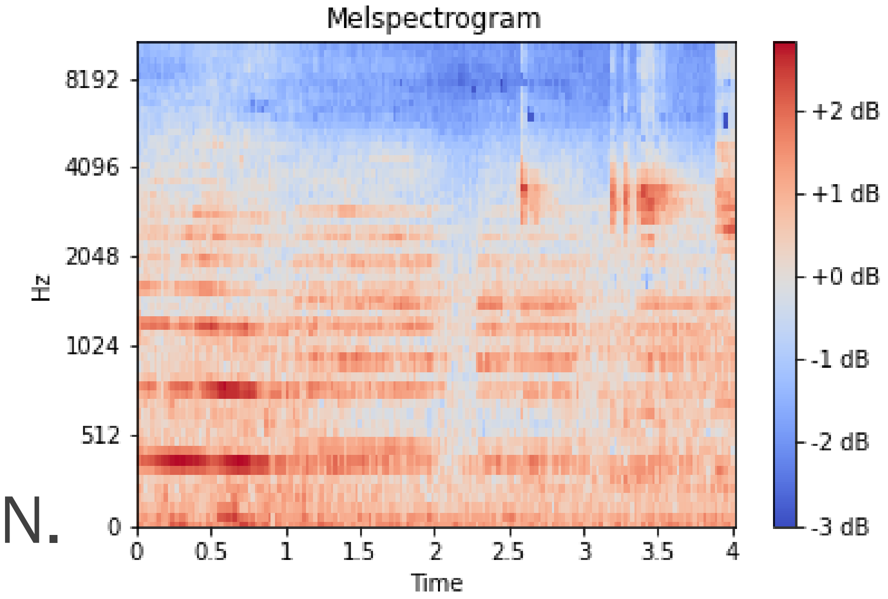
- The same holds for Tonnetz featues (with only 6 coefficients instead).

Second approach:Melspectrogram

- Melspectrogram images are generated from the same audio signals, using 64 mel filter banks, 1025 frame size and 512 hop length, resulting in 64x173 grayscale images

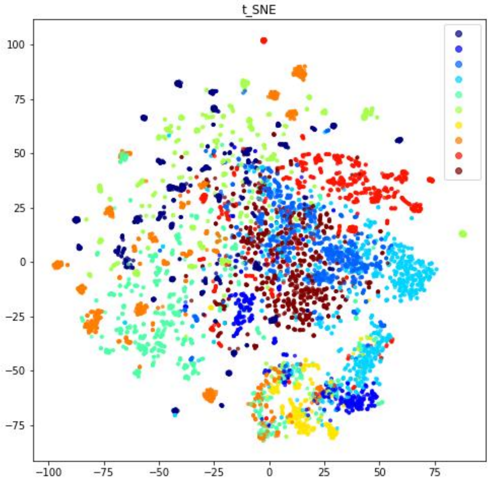
- values are converted to dB and scaled to be centered around 0 with a std dev of 1

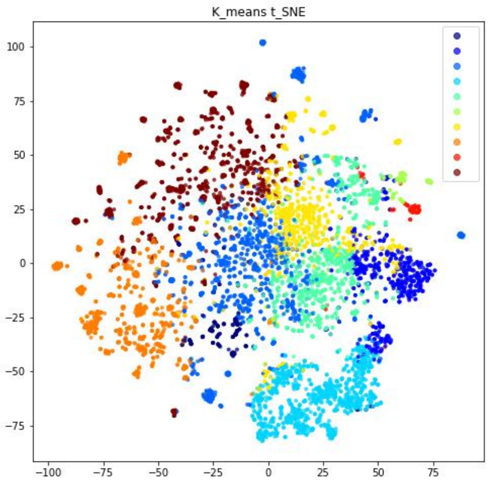
- No statistics is computed on this images, as they will be only used as input for the CNN.

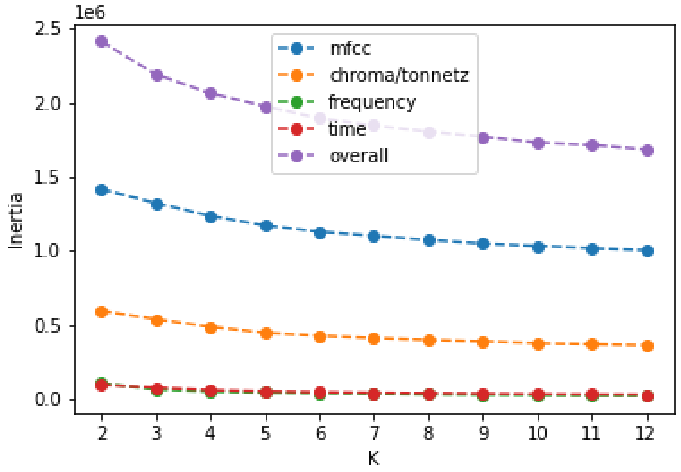


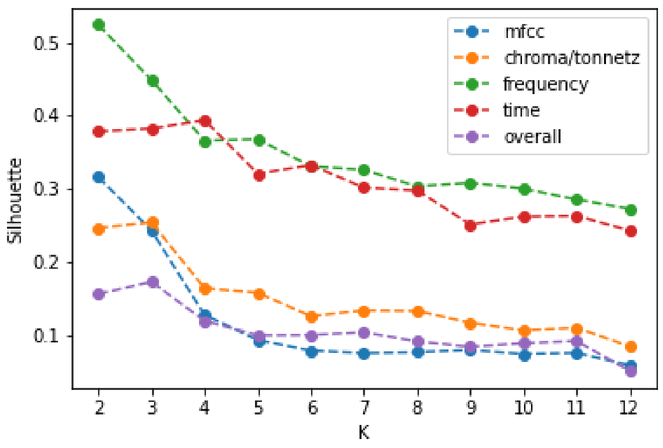
Visualization and Clustering

A tabular dataset of size 8732x320 is obtained. After scaling it, dimensionality reduction and clustering (KMeans) are applied in order to visualize the feature space.







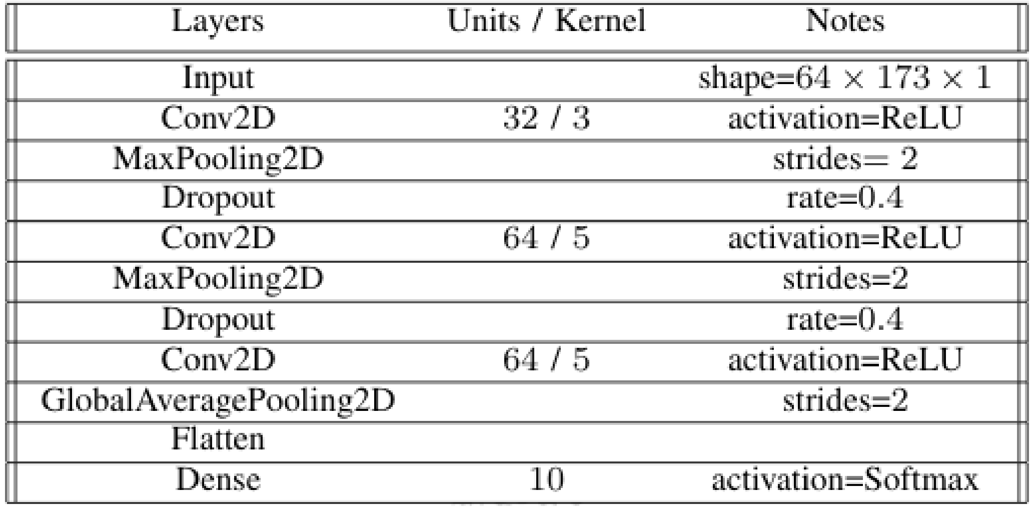


Chosen Models and Training

- Random Forests and Support Vector Machines are trained with tabular dataset obtained by statistical features

- CNNs are trained with the melspectrogram image data

- Scores are validated through 10-Fold CV with both randomly generated splits and folds provided by the authors of the dataset.

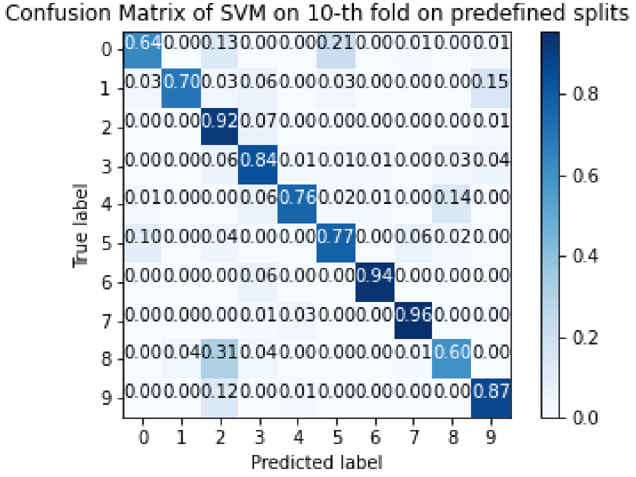


Layers	Units / Kernel	Notes
Input		shape=$64 \times 173 \times 1$
Conv2D	32 / 3	activation=ReLU
MaxPooling2D		strides= 2
Dropout		rate=0.4
Conv2D	64 / 5	activation=ReLU
MaxPooling2D		strides=2
Dropout		rate=0.4
Conv2D	64 / 5	activation=ReLU
GlobalAveragePooling2D		strides=2
Flatten		
Dense	10	activation=Softmax

CNN architecture

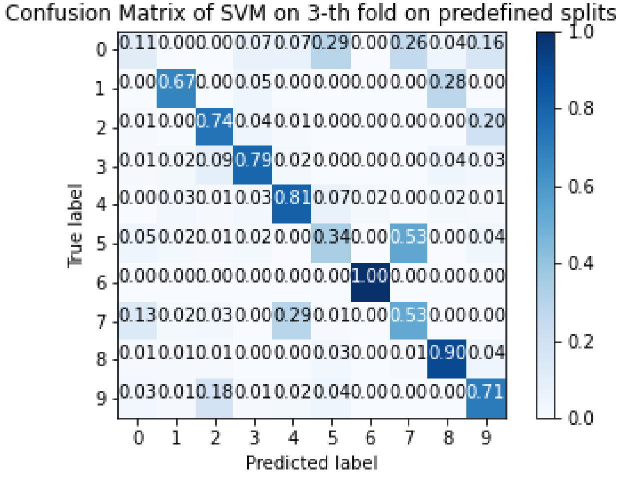
Results

- Random Splits: an avg test accuracy of 0.897 has been obtained by the RFs, while a remarkable score of 0.945 by the SVMs. Both of them slightly overfitted, reaching an avg training accuracy of 0.99 CNNs suffered less from overfitting, but avg test accuracy stopped at 0.903.
- Scores on predefined splits are definitely worse, with an avg test accuracy of 0.667 for RFs and 0.71 for SVMs. Also CNNs stopped around an average of 0.705 accuracy on the test set.



Confusion Matrix of SVM on 10-th fold on predefined splits

	0	1	2	3	4	5	6	7	8	9
0	0.64	0.00	0.13	0.00	0.00	0.21	0.00	0.01	0.00	0.01
1	0.03	0.70	0.03	0.06	0.00	0.03	0.00	0.00	0.00	0.15
2	0.00	0.00	0.92	0.07	0.00	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	0.06	0.84	0.01	0.01	0.01	0.00	0.03	0.04
4	0.01	0.00	0.00	0.06	0.76	0.02	0.01	0.00	0.14	0.00
5	0.10	0.00	0.04	0.00	0.00	0.77	0.00	0.06	0.02	0.00
6	0.00	0.00	0.00	0.06	0.00	0.00	0.94	0.00	0.00	0.00
7	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.96	0.00	0.00
8	0.00	0.04	0.31	0.04	0.00	0.00	0.00	0.01	0.60	0.00
9	0.00	0.00	0.12	0.00	0.01	0.00	0.00	0.00	0.00	0.87

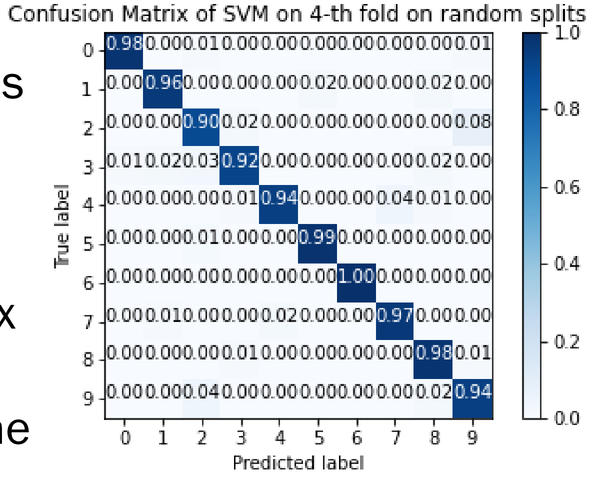


Confusion Matrix of SVM on 3-th fold on predefined splits

	0	1	2	3	4	5	6	7	8	9
0	0.11	0.00	0.00	0.07	0.07	0.29	0.00	0.26	0.04	0.16
1	0.00	0.67	0.00	0.05	0.00	0.00	0.00	0.00	0.28	0.00
2	0.01	0.00	0.74	0.04	0.01	0.00	0.00	0.00	0.00	0.20
3	0.01	0.02	0.09	0.79	0.02	0.00	0.00	0.00	0.04	0.03
4	0.00	0.03	0.01	0.03	0.81	0.07	0.02	0.00	0.02	0.01
5	0.05	0.02	0.01	0.02	0.00	0.34	0.00	0.53	0.00	0.04
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
7	0.13	0.02	0.03	0.00	0.29	0.01	0.00	0.53	0.00	0.00
8	0.01	0.01	0.01	0.00	0.00	0.03	0.00	0.01	0.90	0.04
9	0.03	0.01	0.18	0.01	0.02	0.04	0.00	0.00	0.00	0.71

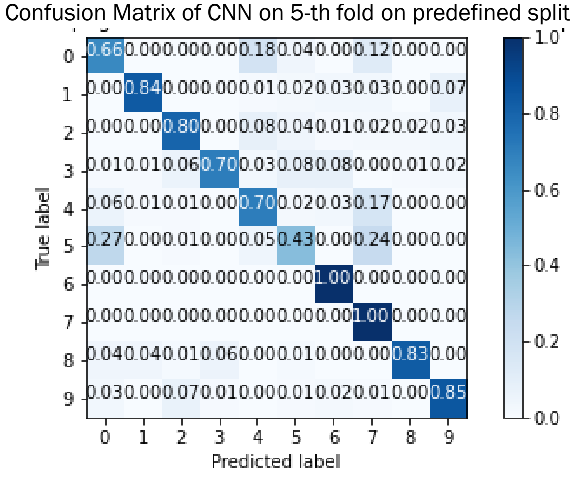
LEFT - Confusion Matrices of SVMs on worst and best predefined splits

RIGHT - Confusion Matrix of SVM on the 4-th random split, the best one



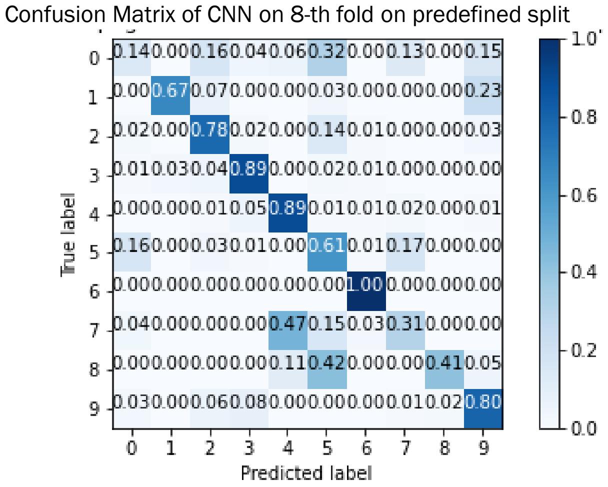
Confusion Matrix of SVM on 4-th fold on random splits

	0	1	2	3	4	5	6	7	8	9
0	0.98	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01
1	0.00	0.96	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.00
2	0.00	0.00	0.90	0.02	0.00	0.00	0.00	0.00	0.00	0.08
3	0.01	0.02	0.03	0.92	0.00	0.00	0.00	0.00	0.02	0.00
4	0.00	0.00	0.00	0.01	0.94	0.00	0.00	0.04	0.01	0.00
5	0.00	0.00	0.01	0.00	0.00	0.99	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
7	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.97	0.00	0.00
8	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.98	0.01
9	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.02	0.94



Confusion Matrix of CNN on 5-th fold on predefined split

	0	1	2	3	4	5	6	7	8	9
0	0.66	0.00	0.00	0.00	0.18	0.04	0.00	0.12	0.00	0.00
1	0.00	0.84	0.00	0.00	0.01	0.02	0.03	0.03	0.00	0.07
2	0.00	0.00	0.80	0.00	0.08	0.04	0.01	0.02	0.02	0.03
3	0.01	0.01	0.06	0.70	0.03	0.08	0.08	0.00	0.01	0.02
4	0.06	0.01	0.01	0.00	0.70	0.02	0.03	0.17	0.00	0.00
5	0.27	0.00	0.01	0.00	0.05	0.43	0.00	0.24	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
8	0.04	0.04	0.01	0.06	0.00	0.01	0.00	0.00	0.83	0.00
9	0.03	0.00	0.07	0.01	0.00	0.01	0.02	0.01	0.00	0.85

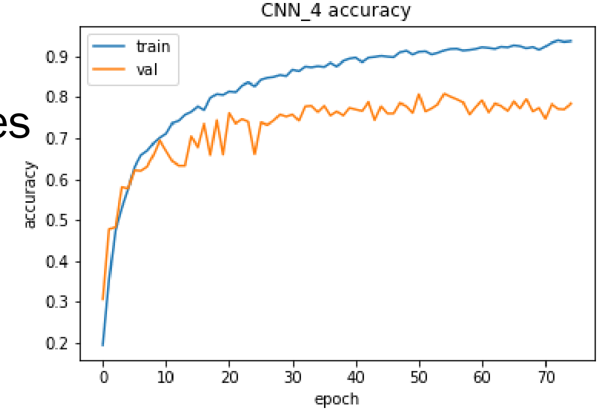


Confusion Matrix of CNN on 8-th fold on predefined split

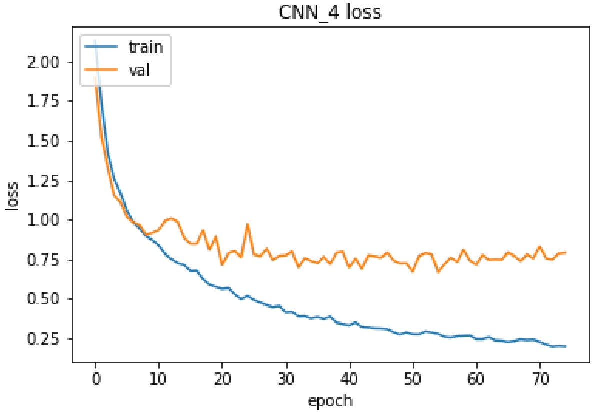
	0	1	2	3	4	5	6	7	8	9
0	0.14	0.00	0.16	0.04	0.06	0.32	0.00	0.13	0.00	0.15
1	0.00	0.67	0.07	0.00	0.00	0.03	0.00	0.00	0.00	0.23
2	0.02	0.00	0.78	0.02	0.00	0.14	0.01	0.00	0.00	0.03
3	0.01	0.03	0.04	0.89	0.00	0.02	0.01	0.00	0.00	0.00
4	0.00	0.00	0.01	0.05	0.89	0.01	0.01	0.02	0.00	0.01
5	0.16	0.00	0.03	0.01	0.00	0.61	0.01	0.17	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
7	0.04	0.00	0.00	0.00	0.47	0.15	0.03	0.31	0.00	0.00
8	0.00	0.00	0.00	0.00	0.11	0.42	0.00	0.00	0.41	0.05
9	0.03	0.00	0.06	0.08	0.00	0.00	0.00	0.01	0.02	0.80

LEFT - Confusion Matrices of SVMs on worst and best predefined splits

RIGHT - Confusion Matrices of the CNNs on 5-th and 8-th predefined folds, the best and the worst respectively.



CNN_4 accuracy



CNN_4 loss

Conclusion and future works

- We have shown that urban sound classification can be faced in different ways, also when using features that are deeply used in speech recognition and music genre classification.
- Deep learning is not always the best, since RFs and SVMs performed similarly (somehow even better) to CNNs.
- Results could be improved by refining the chosen features and statistics and, maybe, with some data augmentation.