Supervised Audio Classification

SGN-26006 Ismael Peruga
Supervisor: Shuyang Zhao, Shayan Gharib Jorge Morte
Lab Date: 20/12/18 Asier Alcaide

Problem Statement

This paper has the main goal to present a performance comparison when using different features. We will use the same model based on a convolutional neural network and we will train it using different features as input. The impact on the performance depending on the features used will be surveyed. The data-set is composed with 15 different types of audios, corresponding to the following 15 scenes:

1. Bus
2. Cafe
3. Beach
4. City center
5. Forest
6. Car
7. Grocery store
8. Home
9. Library
10. Metro station
11. Office
12. Park
13. Residential area
14. Train
15. Tramway

Feature Extraction

We will train different networks using the following features, and we will present a comparison.

1. **Mel-frequency cepstral coefficients (MFCCs)**. is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.MFCCs are commonly derived as follows[3]:
   - I Take the Fourier transform of (a windowed excerpt of) a signal.
   - II Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
   - III Take the logs of the powers at each of the mel frequencies.
   - IV Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
   - V The MFCCs are the amplitudes of the resulting spectrum.
2. **Constant-Q chromagram**. transforms a data series to the frequency domain. It is related to the Fourier transform[1] and very closely related to the complex Morlet wavelet transform.

All the proposed are spectral features based, to extract them we will use the python library Librosa [1]. Both feature extractions are also combined into a new model, as we consider the possibility of a better improve in the results if the model is trained with more than one feature extractor. It will be trained with more information than the other models.

Network Architecture

The Convolutional Neural Network used in this project is based in Han, Park and Lee paper [2]. The following image shows a basic representation of the Neural Network:

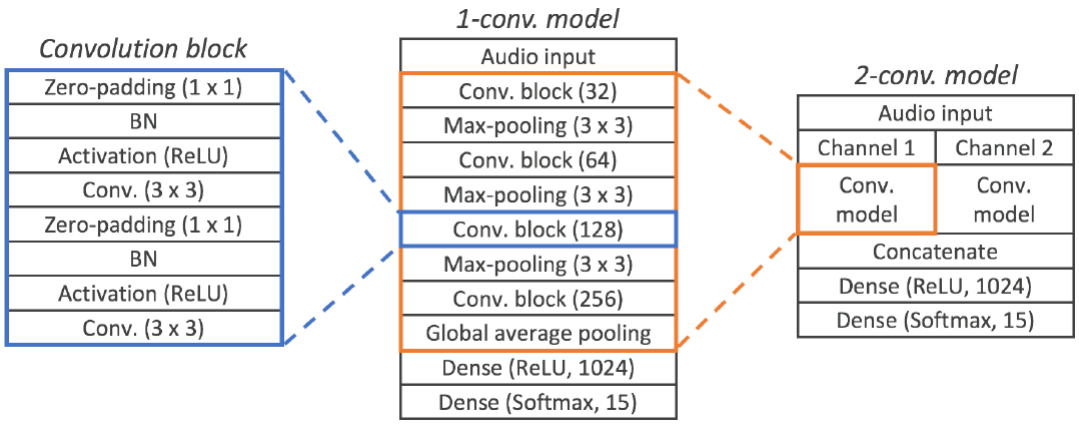


Figure 1: Caption

For each stereo audio signal, we will convert it to mono and extract the features from it. The features will be the input for a CNN with the structure defined in the Figure 1, and after that we will include two fully connected layers from where we will estimate the category. When combining the two features, we will concatenate them and change the input size of the CNN.

Implementation

The implementation was done by using a *TensorFlow* as background and *Keras* as the main library. The dataset used is the *TUT Urban Acoustic Scenes 2018* training dataset [4], which has been split into a 70% for training phase, 20% for evaluation and 10% for validation.

We developed a process to separate the audio files into the three subsets: training, validation and evaluation. We also created a process to extract the features of all audios and save them as a pickle file, file containing a Python variable which contains all the features from a subset. To train the model, we load the corresponding features file.

Results

After the training and evaluation phases, the results obtained show some differences between them. First and second models, the simple feature extraction ones, they have been finished almost at the same total time; however, the last model which combines both features extractions, finished with doubled training and evaluating time than the rest. That is explained with the duplication of the dimensions of the input data.

Table 1: Confusion matrix using both features

0.94	0	0.04	0	0	0	0	0	0	0	0	0	0	0	0.02
0	0.92	0.02	0	0	0	0.04	0	0	0	0	0	0	0	0.04
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0.05	0.94	0	0	0	0	0	0	0	0	0.01	0	0
0	0	0.03	0	0.97	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.9	0	0	0	0	0	0.02	0	0.01	0.07
0	0.02	0.05	0	0	0	0.94	0	0	0	0	0	0	0	0
0	0	0.1	0	0.03	0.02	0.01	0.84	0.02	0	0	0.02	0	0.06	0
0	0.02	0	0	0.02	0	0	0	0.95	0	0	0	0	0	0.01
0	0	0	0	0.02	0	0.13	0	0	0.86	0	0	0	0	0
0	0	0.02	0	0	0	0	0	0.02	0	0.95	0.01	0	0	0
0	0.03	0.03	0.13	0.02	0	0	0	0	0	0	0.77	0.02	0	0
0	0	0.02	0.1	0.06	0	0.02	0	0	0	0	0.06	0.75	0	0
0.02	0.03	0	0	0	0.04	0	0	0.04	0	0	0	0.04	0.76	0.12
0.02	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0.96

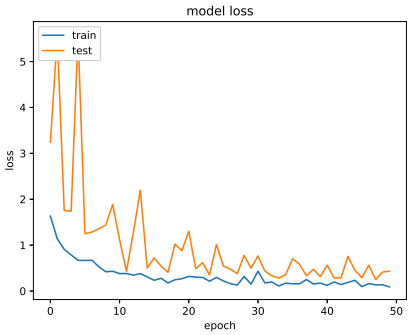
Table 2: Confusion matrix using CQC

0.58	0.02	0	0.02	0	0.03	0	0.02	0.1	0	0	0.18	0	0.04	0.02
0	0.81	0	0	0	0	0.03	0.03	0.08	0	0	0	0.03	0.02	0
0.02	0	0.86	0.06	0	0	0	0	0	0	0	0.03	0.02	0.02	0
0	0	0.06	0.42	0	0	0	0	0.02	0	0	0.16	0.34	0	0
0	0	0.1	0.29	0.35	0	0	0	0	0	0	0.02	0.024	0	0
0.11	0	0.02	0	0	0.71	0	0	0.05	0	0	0.06	0	0.02	0.03
0	0.02	0	0	0	0	0.84	0.05	0.02	0.05	0	0	0.03	0	0
0	0	0	0.08	0.02	0	0.03	0.69	0.03	0	0.06	0.02	0.06	0	0
0	0.02	0.1	0	0.02	0	0	0	0.73	0.06	0	0.05	0.03	0	0
0	0	0.02	0	0	0	0.02	0	0.05	0.81	0	0.02	0.1	0	0
0	0	0	0	0	0	0	0	0.03	0	0.95	0.02	0	0	0
0	0	0.03	0.11	0	0	0	0	0	0	0	0.68	0.18	0	0
0	0	0.11	0.21	0.02	0	0	0	0	0.02	0	0.15	0.5	0	0
0.05	0	0.03	0	0	0.02	0	0	0.06	0.08	0	0.13	0	0.61	0.02
0.03	0	0.06	0	0	0	0	0	0.02	0.18	0	0.06	0.02	0.03	0.60

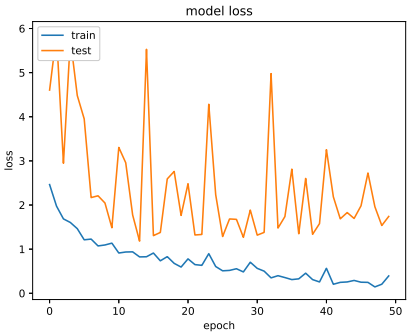
Table 3: Confusion matrix using MFCC

0.90	0	0	0	0	0	0	0	0	0	0	0	0	0.03	0.07
0	0.90	0	0	0	0	0.03	0	0	0.05	0	0.02	0	0	0
0	0	0.95	0	0.02	0	0.02	0	0	0	0	0.02	0	0	0
0	0	0	0.95	0	0	0	0	0	0.03	0	0	0.02	0	0
0	0	0	0.02	0.98	0	0	0	0	0	0	0	0	0	0
0.02	0	0	0	0	0.93	0	0	0	0	0	0.01	0	0.01	0.03
0	0.03	0	0	0	0	0.97	0	0	0	0	0	0	0	0
0	0.05	0	0	0	0	0	0.90	0.02	0	0.02	0	0	0.02	0
0	0	0	0	0	0	0.02	0	0.92	0.02	0.02	0	0	0.02	0
0	0	0	0	0	0	0.06	0	0	0.93	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0.02	0.02	0.02	0	0	0.03	0	0.78	0.15	0	0
0	0	0.02	0.08	0.03	0	0.03	0	0	0	0	0.02	0.82	0	0
0	0.02	0	0	0	0	0.02	0	0.02	0	0.02	0	0	0.89	0.05
0	0	0	0	0	0	0	0	0.02	0.02	0.01	0.01	0	0.03	0.89

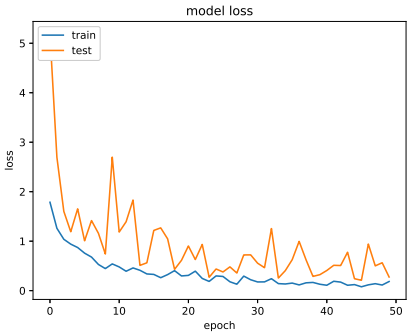
In the Figure 3, it is shown the accuracy curve of the model depending on which feature we have used. We can observe that MFCC produces a model with less variance, difference between training and validation/test accuracy, than the model produced by the CQC features. Both achieve a similar accuracy in the training phase but when using CQC the validation accuracy is not able to reach the training accuracy value, that is called model variance. This could be solved using a model less complex for CQC features, including more regularization techniques in the model or increasing the training dataset size. This also could mean that our model is somehow overfitting over our training data when using CQC features.



(a) Only MFCC features



(b) Only CQC features



(c) MFCC and CQC features

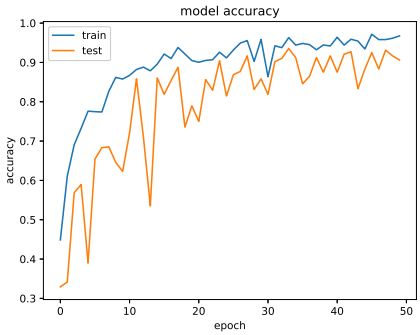
Figure 2: Model losses curves comparing different features

From the Figure 2, where the model losses, depending on which features are used, are shown, we obtain a similar result than before. The MFCC features achieve lower losses than CQC features.

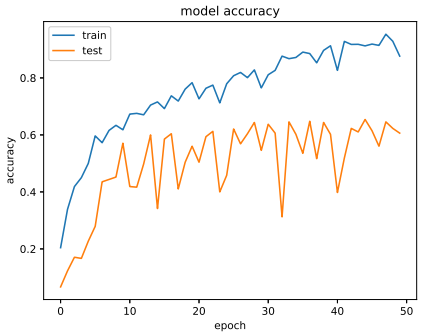
In the Table 1, 2 and 3, we show the confusion matrix of the model. As explained in the previous paragraphs, when using only CQC features, the model has more errors predicting the classes. For example, for input data of residential area, the network only achieves 50 % of accuracy, and selects beach and city center with 21 and 11 % respectively. Also, for tramway data, the network gives an output of metro station with 18 % probability. The worst case is with forest data, when the accuracy of success is only 35 %. When using only MFCC, the network works pretty well, with more than 90 % of success for almost all the classes, but, for example, for input data of park, it has only 78 % of success, saying the 15 % of times, that the audios are taken in a residential area. When using both feature extraction methods, the smaller success value is 77, 75 and 76 % for park, residential area and train audios. But the rest of the classes have more than 90 % of success, or close to that value.

We also tested the model using audios recorded on our own. Doing this we can check if our model would generalize correctly to audios from different sources recorded in different conditions, different sample rate, different places and a different physical device. Our results using self recorded audios achieve around 15 % of accuracy. From the results, we can observe that the model does not work properly when using an audio from a different device.

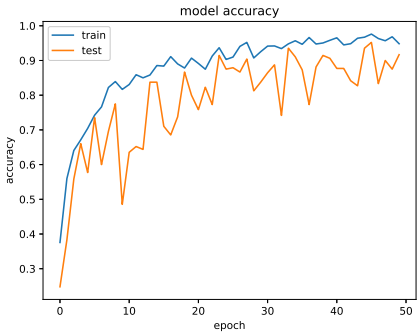
We could think that our model is overfitting over our device data, and to achieve a good generalization, we should include audio from different devices and sources in the training dataset and probably modify somehow the model.



(a) Only MFCC features



(b) Only CQC features



(c) MFCC and CQC features

Figure 3: Model accuracy curves comparing different features

Conclusions

With this laboratory project we understood the basics of audio classification problems. We studied the state-of-the-art solutions to classify audio signals and we have developed our own solution. We have used and gained familiarity with Keras library, and the importance on which feature extractor we should choose to have the best performance.

References

[1] Librosa, 2013. *Spectral Features* (https://librosa.github.io/librosa/feature.html#spectral-features).

[2] Yoonchang Han, Jeongsoo Park, Kyogu Lee 2017. *Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification* Retrieved from (http://www.cs.tut.fi/sgn/arg/dcase2017/documents/challenge_technical_reports/DCASE2017_Han_207.pdf)

[3] Sahidullah, Md.; Saha, Goutam (May 2012). *Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition.* Speech Communication. 54 (4): 543565. (https://www.sciencedirect.com/science/article/pii/S0167639311001622?via%3Dihub)

[4] Heittola, Toni 2018. *TUT Urban Acoustic Scenes 2018, Development dataset* (http://dcase.community/challenge2018/task-acoustic-scene-classification)

[5] Huy Phan, Philipp Koch, Fabrice Katzberg, Marco Maass, Radoslaw Mazur, and Alfred Mertins, 2017 *Audio Scene Classification with Deep Recurrent Neural Networks* Institute for Signal Processing, University of Lubec. (https://arxiv.org/pdf/1703.04770.pdf)

[6] Dawei Feng, Kele Xu, Haibo Mi Feifan, Liao and Yan Zhou 2018 *Sample Dropout for Audio Scene Classification Using Multi-Scale Dense Connected* Science and Technology on Parallel and Distributed Laboratory, School of Computer, National University of Defense Technology; Changsha, 410073, China. (https://arxiv.org/pdf/1806.04422.pdf)

[7] A. Eronen, *Comparison of features for musical instrument recognition* in IEEE Workshop on the Apps. of Signal Proc. to Audio and Acoustics, pp. 1922, Citeseer, 2001.

[8] T. Heittola, A. Klapuri, and T. Virtanen, *Musical instrument recognition in polyphonic audio using source-filter model for sound separation.* in ISMIR, pp. 327332, 2009.

[9] T. Heittola, Automatic classification of music signals, Master of Science Thesis, 2003.

[10] T. Kinnunen and H. Li, *An overview of text-independent speaker recognition: From features to supervectors* Speech communication, vol. 52, no. 1, pp. 1240, 2010.