

Lab 2: Music Genre Classification - CNNs

1 Files included

You can find the files below at the following link:

https://drive.google.com/drive/folders/1ijzsLkkSZHZW2JDaoH2fAm6f9IwMg8dk?usp=drive_link

- features_30_sec.csv – csv file with global song descriptors
- features_3_sec.csv - csv file with 3sec-window song descriptors
- Original songs in wav format
- Songs spectrograms

The first two files are the same files of Lab1. In addition, the link provides access to a folder (*genres_original*) with the wav files of the songs in the genre dataset, and a folder (*images_original*) with the mel spectrograms of the songs in the dataset.

2 Background

In recent years, Convolutional Neural Networks (CNNs) have become increasingly popular for audio classification tasks, driven by their remarkable success in computer vision. Originally designed to exploit the spatial hierarchies of visual data, CNN architectures have been effectively adapted to analyse time-frequency representations of audio signals, such as spectrograms or Mel-spectrograms. This adaptation leverages the CNNs' ability to learn local and hierarchical features, enabling them to capture important temporal and spectral patterns in sound. Many techniques and architectures that proved successful in vision—such as VGG, ResNet, and Inception—have been transferred to audio analysis, where they demonstrate strong performance in tasks like speech recognition, music genre classification, and environmental sound identification. This cross-domain success illustrates how insights from visual deep learning research have profoundly shaped and accelerated progress in computational audio understanding.

3 Tasks

Implement the following tasks using a Google Colab notebook.

Task 1. Using the spectrogram data provided, train a CNN model with the highest accuracy possible. Explore a simple CNN with a couple of convolutional layers. Optimise your CNN model by exploring different configurations (i.e. different numbers of convolutional layers and filters). Shuffle the data and perform an 80%-20% split for training data and test data. Optimise your model using the 80% training data, and once you have decided on the best configuration, evaluate your model on the 20% test set.

Task 2. Fine-tune one pretrained existing deep learning model, such as VGGish, and use the same training data and test data split to evaluate the model.

Submitting your answer

The Lab can be solved individually or in teams of two people (1 submission per team). Submission is through the Aula Global. Submissions should contain a PDF file with your name(s), the link to your Google Colab notebook (make sure you give me access to it) and a summary and discussion of the results. Include a comparison of the results obtained in Lab 1

using manually extracted features. Provide all the results for every model you trained (including Lab 1 results). The Lab submission deadline will be discussed in class.