# Coursework 1 – Music Genre Classification

## Brief

Due date: Tuesday 29th April, 16:00.
Handin: https://handin.ecs.soton.ac.uk/handin/2425/COMP6252/1
Required files: `report.pdf and code.zip`
Credit: 20% of overall module mark

## Overview

The goal of this assignment is to build different classifiers (which will be specified below) for the music genre dataset GTZAN using different network architectures and report their performance, together with a brief discussion regarding the results.

## Details

### Data

The GTZAN dataset can be downloaded from: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

The dataset includes audio samples for 10 different music genres (i.e., jazz, classical, ...). Also, each audio sample has a visual representation constructed using MEL spectrograms.

### Specified neural architectures

The following six architectures need to be used to build your six different classifiers:

1. A fully connected network with two hidden layers.
2. The convolutional network shown in **Figure 1** with your own choice of parameters.
3. A convolutional network obtained by modifying the one in **Figure 1** by adding a batch normalisation layer.
4. The same architecture in the above 3, with the RMSProp optimiser.
5. An RNN network with LSTMs.
6. The same architecture in the above 5, together with GANs generating audio samples augmenting the training audio samples (we suggest you generate the same number of audio samples as the original number of training audio samples).

### Steps for each neural architecture

The following steps need to be implemented for each of the above 1 - 4 neural architectures:

1. Resize the images to 180x180. This must be done when the images are loaded using torchvision.transforms.Resize.
2. Use PyTorch to randomly split the dataset into a training (70%), validation (20%), and test (10%) datasets.
3. Run the training for 50 epochs and 100 epochs.

The following points are suggested for each of the above 5 - 6 neural architectures:

1. Just use the **audio samples**.
2. Use PyTorch to randomly split the original dataset into a training (70%), validation (20%), and test (10%) datasets.
3. Run the training for a certain number of epochs (e.g., until convergence according to your stopping criterion).

### Code

We suggest you use a **single** Jupyter notebook to manage your code, and suggest:

- defining the specified neural architectures in sequence, e.g., Net1, Net2, Net3, Net4, Net5, and Net6.

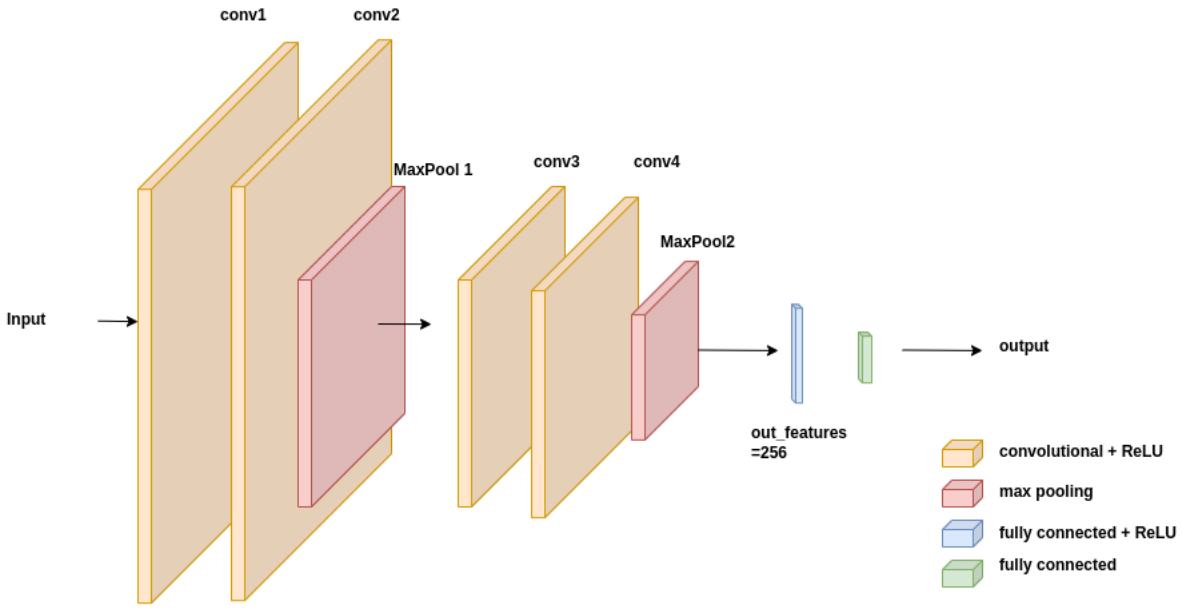


**Figure 1:** A convolutional network architecture.

## The report

The report must be no longer than 3 pages of A4 with the given Latex format, and must be submitted electronically as a PDF. The report must include:

- Your name and ECS user ID.
- A description of the implementation of the methods, including information on how they are trained and tuned, and the specific parameters used.
- The performance of all the methods used.
- A thorough and insightful discussion regarding the comparison of the results obtained by the methods used.

## What to hand in

You are required to submit the following items to ECS Handin:

- Your 3-page report (as a PDF document in the CVPR format; max 3 A4 pages, no appendix).
- Your code enclosed in a zip file.

## Marking and feedback

You will receive a grade out of 20 for this coursework. Marks will be awarded for:

- Successful completion of the task.
- Evidence of understanding.
- Well structured and commented code.
- Evidence of professionalism in the implementation and reporting.
- Quality and contents of the report.

Standard ECS late submission penalties apply.

Individual feedback will be given covering the above points

## Useful links

- Module COMP6252 website: https://secure.ecs.soton.ac.uk/module/2324/COMP6252/43095/
- Jupyter notebook: https://docs.jupyter.org/en/latest/

## Questions

If you have any problems/questions, use the Q&A channel on Teams, or email Hikmat, Zhiwu, or Xiaohao.