Speech and Speaker Recognition
Music Genre Classification: Different Methods Exploration

Maria Bjelikj bjelikj@kth.se

Fernando García fegs@kth.se

Carlos Jordán Rosado cjrt@kth.se

Andrés Alonso Toledo aatc@kth.se

Abstract

This project explores different techniques for genre classification of the music tracks provided by the FMA small dataset. Several different state-of-the-art approaches were analyzed, including support vector machines, k-nearest neighbors, convolutional neural networks, recurrent neural networks, as well as the combination of the two. The content of this paper shows that convolutional networks trained in parallel with recurrent networks achieve best results for the task, although obtaining a proper architecture can be an arduous task. The amount and quality of the data have proven to be two fundamental features when performing this kind of experiments. Augmentation methods were applied, which slightly boosted the data efficiency. The best overall test accuracy obtained was 52.5%.

1 Introduction

Music genre classification is a task that belongs to the field of music information retrieval (MIR), an interdisciplinary science aimed at studying the processes, methods, and knowledge representations required to retrieve information from music. MIR can be used broadly, for example for building music recommendation systems, or for instrument recognition and separation, and even for automatic music transcription and music generation. This report focuses on music classification, which with the ever-growing number of music collections has been posed the challenge of how to retrieve, browse, and categorize the data. Musical genres are the main top-level descriptors for music data organisation.

There are many approaches for classifying songs, a lot of which are manual and make use of "social tagging"; hand-crafted annotations that are added to characterize each song 2469. For example, the internet radio Pandora[1] hires musicians to manually analyze each song that is played, a process which takes at least 20 minutes per track, resulting in a classification of over 400 different "genes". Surely such an approach comes with a heavy workload, and due to the high human effort required for manual annotations, the automation of the process is a smart solution. Take Spotify[2], a company that has developed a machine listening tool which takes into account a number of factors when performing classification, such as characteristics they have named tempo, energy, danceability, strength of the beat and emotional tone, and so on. Different systems work with the principle of finding users whose listening history is similar, and using this knowledge in order to suggest unheard music.

1.1 Related Work

Nowadays, machine learning is often times the first choice when it comes to automatization of classification. Techniques such as support vector machines (SVM) seem to perform well, achieving small error rates, especially in comparison with other methods such as nearest neighbor (NN), Gaussian mixture models (GMM) and hidden Markov models, as concluded by Xu et al. [1]. However, the results aren't astonishing, reaching classification accuracy such as 76.6% by Mutiara et al. [2]

[1] https://www.pandora.com

[2] https://artists.spotify.com/blog/how-spotify-discovers-the-genres-of-tomorrow

on the GTZAN[3] dataset, or 82% by Mandel and Ellis [3], on a subset of the *uspop2002*[4] collection. Clearly, the difference in results often depends on which dataset was used as well as how the features were extracted and processed. Deep belief networks (DBN) using Restricted Boltzmann Machines (RBM) have also been used for automatic classification, which some papers further compare to SVMs, reaching accuracy of around 80% Xiaohong Yang [4] on the GTZAN dataset, and 72.18% Son N. Tran [5] on the MagnaTagATune[5] dataset.

Convolutional neural networks (CNN) seem to be just as effective, if not more, as an approach to solving this type of problem. Researches Li et al. [6] achieved accuracy of 85% on the GTZAN dataset. Their network architecture consists of five layers total. The first layer inputs raw MFCC features, which then connects sequentially to three different convolutional layers using different kernels and filters, linked to a fully connected output layer. The overview of the classification system boils down to several steps: MFCC extraction from audio signals, MFCC map transformation, which is then segmented to fit the input size of the CNN, and lastly supervised learning is employed.

Similarly, Zhang et al. [7] used a CNN model with three convolutional layers and three dense layers on the GTZAN dataset pre-processed to STFT features, reaching an accuracy of around 85%, too. Furthermore, they explored using averaging between max-pooling and average-pooling for feature extraction, as well as using shortcut connections to skip one or more layers, a method inspired by residual learning, which overall improved the performance of their networks.

The FMA dataset, described in section 2, seems to be a bit trickier to learn and classify. CrowdAI's 2018 competition[6] on this dataset resulted in a highest F1 test score of only 63%. Similarly, Bian et al. [8] report accuracy as high as 66.3%, obtained with a 4 convolutional layer ResNet, the results of which are then fed to an SVM classifier. Additionally they recommend data augmentation, which on average has improved their accuracy by about 3%. SongNet by Chi Zhang [9] et al. achieved accuracy of 65.23%, built as a three-layer CNN which is followed by a Recurrent Neural Network (RNN). Adiyansjah et al. [10] obtain an F1 score on the FMA small dataset of 74.9% with their C-RNN network, a 4-layer CNN complemented by two Gated Recurrent Units (GRU) layers as the RNN component, used to summarize 2D temporal patterns from the results of the CNN. Overall, the results with all these different approaches are not as good as those obtained with the GTZAN dataset.

Based on this research, it was concluded that this project will explore CNNs, C-RNNs and parallel CNN-RNNs for automatic music genre classification as a suitable approach, using the FMA dataset.

2 Dataset

The dataset used in this project is the Free Music Archive [11], a free and open library directed by WFMU[7], a free-form radio station in the United States. It provides high-quality audio, pre-computed features, together with track and user-level metadata, tags, and free-form text such as biographies. The small version of the dataset is sufficient for the purposes of this project, which consists of 8,000 tracks that are 30s long, sampled from 8 top-level genres, balanced with 1,000 clips per genre, 1 root genre per clip. It is a subset of the original dataset which has a selection of the top 1,000 clips from the 8 most popular genres of the dataset, which are:

'Folk', 'International', 'Experimental', 'Pop', 'Hip-Hop', 'Instrumental', 'Electronic', 'Rock'

This subset in comparison to the large version is similar to the very popular GTZAN dataset in terms of number and type for classes (GTZAN has 10 top-level genres, whereas FMA as has 8), with the benefit that FMA is more updated and suitable in terms of genre completeness and audio quality. Additionally, the fine genre information for each track was claimed by the artists themselves.

Out of the 8,000 tracks, 5 tracks were removed, 3 of them because they were too short, and other 2 because the files were corrupted, either a mistake by the creators of the dataset or an issue that occurred during downloading. The pre-processed dataset as described in the following subsection was split into 80% data for training, 10% data for validation and 10% data for testing.

[3] http://marsyas.info/downloads/datasets.html
[4] https://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html
[5] http://mirg.city.ac.uk/codeapps/the-magnatagatune-dataset
[6] https://www.crowdai.org/challenges/www-2018-challenge-learning-to-recognize-musical-genre
[7] https://wfmu.org/

2.1 Pre-processing

Spectrograms allow for the audio classification problem to be converted to an image classification task, or rather pattern recognition task, which can then be applied to a CNN. Spectrogram preparation is key to a successful model. Thus, the audio tracks were converted into log mel-spectrograms, a visual representation of frequencies over time. A regular spectogram represents the squared magnitude of the short term Fourier transform (STFT) of the audio signal.

The next step was converting this spectrogram by means of mel scale, providing an output that is more interpretable to a human eye. The package LibROSA[8] was used for these functions. All audio tracks were sampled with sampling rate 22,050 Hz, since most but not all original tracks are sampled at 44,100 Hz, with duration of only 3s for a faster computation. The parameter setup for the window length−the window of time for the STFT−was 2048, which amounts to 10ms, the shortest reasonable period a human ear can distinguish. The parameter hop length−the number of samples between successive frames− was set to 604. After all of these steps were applied, the mel-spectrograms were transformed by log function, which maps the spectrogram to the normal logarithmic scale used to determine loudness in decibels, again applied for human interpretability. The results of this procedure are log mel-spectrograms of size 128x128.

3 Methods

The main steps in music classification are pre-processing of raw audio data and design of a classifier.

3.1 Data

The experiments and research in this project highlighted the dependency of the performance of the classifier on the data pre-processing. Finding the best shape and method for the data for a selected type of classifier and approach is the main key. Generally, there are three main ways of using the data:

- Acoustic features extraction.
- Spectrograms transformations (mel-spectrograms, log mel-spectrograms, MFCC).
- Using raw audio.

At first, all 30s of the original tracks were used for computing first MFCC image data, then mel-spectrograms, and finally log mel-spectrograms, as the related work did not provide a consensus for which of these features are most suitable. A basic 3-layer CNN was used for the initial experiments, and the results pointed to the log-mel-spectrogram data being most suitable for the CNN approach.

The dimension of the log-mel-spectrograms affected the network performance, too. As using all 30s from the tracks amounted to very long computations, experiments were performed with 10s, 5s and finally only 3s from the tracks. Although the accuracy improved by about 4% using all 30s of the provided tracks when compared to the smaller spectrograms with the same model, the results did not vary significantly between the datasets computed using 10s, 5s and 3s. Thus, using 3s of the tracks is a sufficient compromise given computational and time constraints.

Data Augmentation

Data augmentation is a technique used in machine learning to avoid overfitting−the model learning the training data patterns too well and performing badly on unseen data−by increasing the volume of data. The raw audio training data was augmented before the spectrograms were computed, on varying 3s samples from the original tracks for slight diversity, as follows:

- sampling different 3s from the tracks.
- pitch shift, $steps = 2$ per octave.
- pitch shift, $steps = -2$ per octave.
- sampling different 3s from dataset.
- time stretch, $rate = 1.1$.
- time stretch, $rate = 0.9$.

[8] https://librosa.github.io/librosa/

Principal Component Analysis, Support Vector Machine and k-Nearest Neighbors

Principal Component Analysis is an analytical method that finds which dimension's variances best fit the data volume. This method rotates the data along the axis of the highest variance, where the relative contribution of each feature towards the variances between classes for simple k-NN and SVM models. To apply PCA to the data, the mean and variance of the mel-spectrograms were normalized. Using 15 PCA components provided the best results in terms of accuracy.

The k-Nearest Neighbors (k-NN) classifier is a non-parametric classification system that clusters data based on the 'k' nearest training points and classifies a given points based on the majority vote of the 'k' nearest neighbors. Through trial and error, setting $k = 8$ provided the accuracy results and weighting the label of each neighbor by distance.

The Support Vector Machine (SVM) is a supervised classification system that finds the maximum margin hyper-plane between classes of the data. For the SVM's model an RBF (radial basis function) kernel which corresponds to an infinite dimensional feature space is related to Euclidean distance.

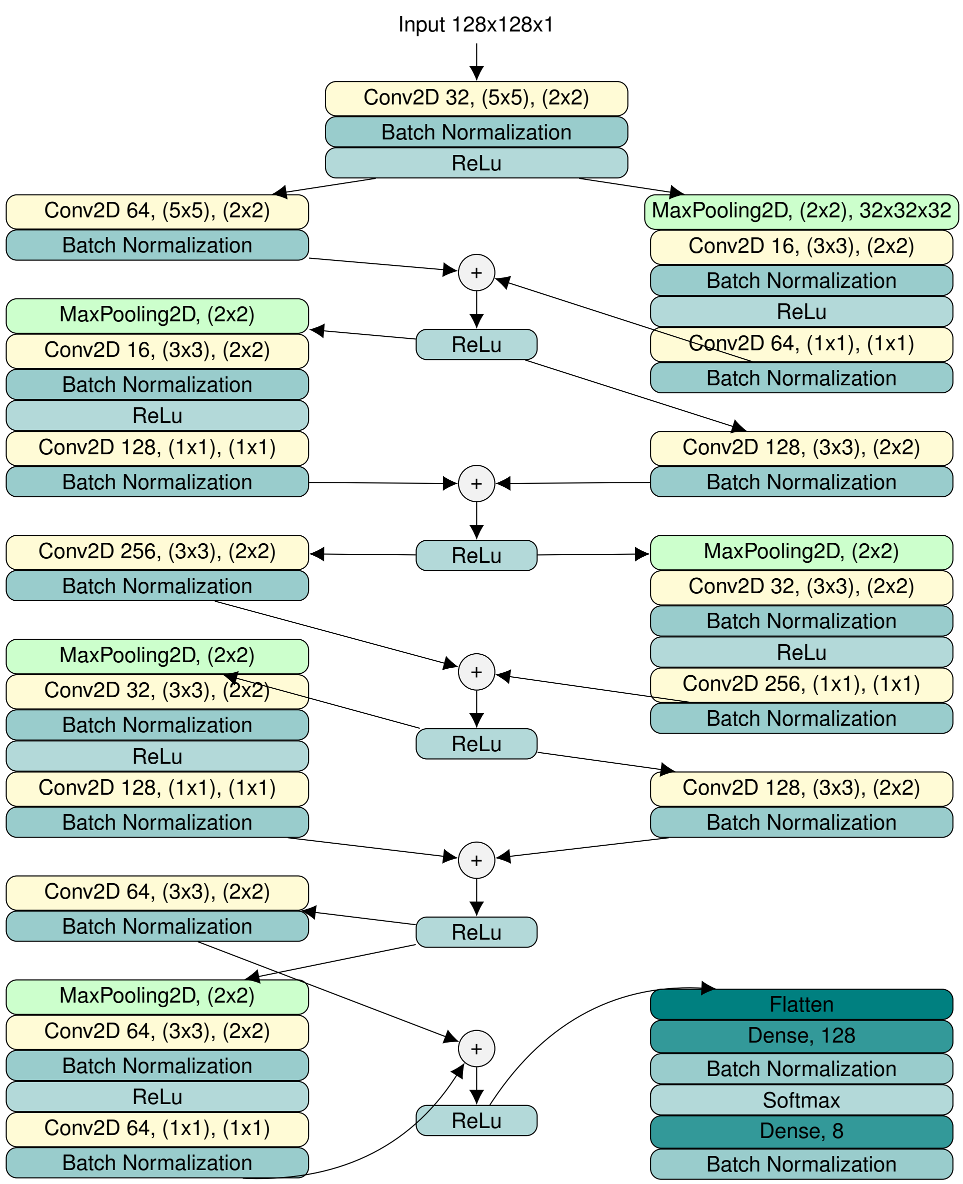

Figure 1: CNN model architecture with 5 blocks created with shortcut connections.

3.2 Networks

For music genre classification[9], k-NN and SVM were implemented as a baseline and for comparisons. Furthermore, CNN, C-RNN and CNN-RNN structures were built. The k-NN and the SVM were implemented with scikit-learn [10]. For the neural networks, Tensorflow.Keras[11] provides great tools.

CNN

CNNs are well suited for pattern recognition such as spectrogram features, both frequency and temporal patterns, as they are characterized with hierarchical learning of structures. The *convolutional layer* (Conv2D) uses 2D filters with various digital image processing techniques for feature extraction, which "slide" through the width and height of an input image, performing convolution−the dot product of the input's region with the filter. This in turn produces a 2D activation map that consists of responses of the filter at a given region−the extracted features of the data.

Next, the *pooling layer* (MaxPooling) reduces the size of the Conv2D layer output. As a result, the number of parameters is down-sampled, so computation becomes faster and overfitting is minimized. An *activation function* is used for introducing non-linearities in the computation. Without it, the model would only learn linear mappings. *Batch normalization* is a relatively new technique for improving the speed, performance, and stability of neural networks, by means of normalizing the input data, and experimentally this type of layer showed better performance.

The CNN architecture proposed in Figure 1 was inspired by the residual architecture principles presented in [12], and although from the studied literature it is in [8] that they present the idea of residual blocks for this type of tasks, the architecture designed for this project is still unique in comparison. The method of shortcut connections allows for increasing the depth of the network substantially, while still managing overfitting, though the bigger motivation for using the skip connections is to avoid the problem of vanishing gradients, by reusing activations from a previous layer until the adjacent layer learns its weights.

The final cluster of layers are used for representing the output of the network, where a fully-connected layer (Flatten) distributes the accumulated scores for all its units; in classification purposes the number of units is equal to the number of classes, and in Figure 1 a Softmax activation function is used to have as an output the normalized probability distribution of each class.

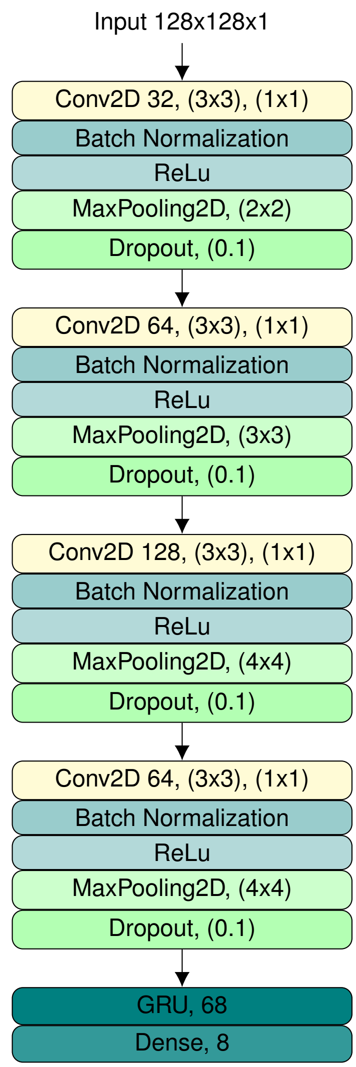

Figure 2: C-RNN model.

C-RNN

C-RNN are networks which use the outputs of the CNN an the input for an RNN, and as such are very useful for extracting spectrogram feature for prediction. This type of network architecture is not only looking at the frequency related features than can be extracted from the image data, but the RNN part of the network excels at learning time sequence patterns.

The C-RNN structure is constructed with 4 convolutional blocks, consisting of Conv2D, MaxPooling, Batch Normalization, ReLU activation, and finally, Dropout (randomly removing a percentage of activations to prevent overfitting). What follows is a layer RNN with Gated Recurrent Units (GRU) to capture 2D temporal patterns from the CNN results.

---

[9]Find the project code in https://github.com/fernando2393/DT2119-Final-Project

[10]https://scikit-learn.org/stable/

[11]https://www.tensorflow.org/guide/keras

CNN-RNN Parallel

The CNN-RNN parallel structure differs from the C-RNNs in that the CNN and RNN sections of the network are performing independent feature extraction on the input data, and then the results of each are concatenated. The key idea behind this type of network is that even though C-RNN has RNNs to be act as a temporal summarizer, it can only summarize temporal information from the output of CNNs−not the input data. The temporal relationships of original musical signals may not be necessarily preserved during CNN operations. RNNs are good in understanding sequential data by modelling the time dependence of the hidden state at time t and hidden state at time $t-1$.

The flattened output data from the CNN in Figure 1 is concatenated with an RNN block trained in parallel on the same input data, which is shown in Figure 3. The RNN consists of The flattened output data from the CNN in Figure 1 is concatenated with an RNN block trained in parallel on the same input data, which is shown in Figure 3. The RNN consists of a MaxPooling rectangular layer first, to cut down the data dimension. Next, an Embedding layer is used, which provides vector representation for music segments. This layer can be capable of capturing structural and stylistic information of the music in this low dimensional space. Additionally, a Bidirectional GRU layer is used to find forward and backward hidden states and then uses attention mechanism to form a weighted sum of these hidden states to output as the representation, the GRU indicating the output.

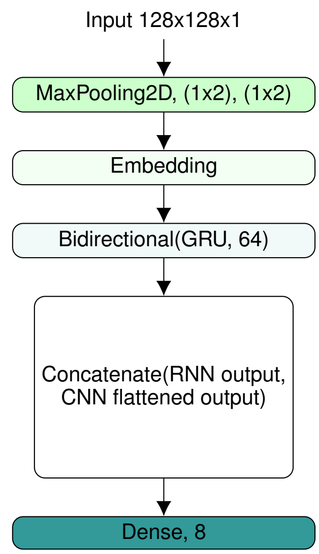

Figure 3: CNN-RNN Parallel model.

4 Experiments and Results

Initially, k-NN and SVM were implemented as guidance. The performance was judged by accuracy, which is the percentage of the correct test labels. The k-NN achieved 30.5% in test accuracy, and the SVM 33%, none of which were satisfactory results. Using the data-augmented spectrograms seems to be a most suitable approach for neural networks, and thus only such networks are further explored.

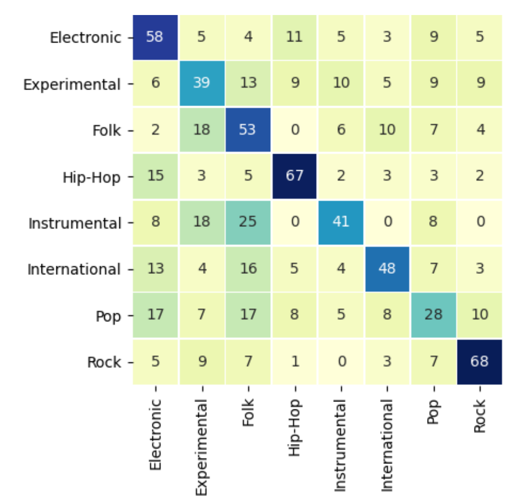

Figure 4: Confusion matrix for CNN.

4.1 CNN

The described CNN was trained for 30 epochs, batch size 16, with an initial learning rate $2e\text{-}4$ that is decayed by factor $learning\_rate/epochs$ by Adam optimizer. Early stopping monitoring the validation accuracy with patience 10 was employed, as well restoration of the best model weights during the training process. The final test accuracy was 50.25%, the precision, recall and F1 score shown in Table 2, and the confusion matrix in Figure 4.

For further result interpretation, let's observe the confusion matrix. The genres 'Hip-hop' and 'Rock' seem to be most distinctive ones, closely followed by 'Electronic'.

Surprisingly, the 'Instrumental' tracks were not classified as well as expected, which for the human ear would be a very distinctive genre, which could be due to the fact the selected random 3s from the tracks may have only contained instrumental parts for the other genres too. Particularly noting here that many samples are misclassified as 'Folk' or 'Experimental', both genres that are popular for containing distinctive instrumental sections. For this purpose alone, with more computational power and time available, larger spectrograms enframing more seconds of the provided data could help.

4.2 C-RNN

The C-RNN architecture, as shown in Figure 2, was heavily inspired by the work [10], which obtained F1 score on the test data of 74.9%. It was trained for 50 epochs, batch size 32, optimized by Adam optimizer with a learning rate 0.001. Early stopping monitoring the validation accuracy with patience 10, and restoration of the best model weights during training were employed. The final test accuracy was 41.12%; precision, recall and F1 score shown in Table 1, and the confusion matrix in Figure 5.

In the initial experiments, convolutional feature maps (68-137-137-137) were implemented to replicate the network in [10], however this never lead to results higher than 40%. One reason could be that the input data shape in [10] is 96x1366, unlike the 128x128 spectrograms used in this project. Additionally, simply adding one GRU layer on top of the 4 convolutional blocks increased training an epoch from 2min to 15min when GPU was used, so no second GRU layer was added, as proposed in [10].

'Instrumental' and 'Folk' are confused here again, and 'Hip-hop' remains the most distinctive class. However, similarly to the results in Chi Zhang [9], the most problematic classes are 'Experimental' and 'Pop'. To interpret this, let's consider the genre definitions. Experimental music expands upon existing genre definitions and boundaries, such electronic, jazz, rock etc, and as such is inherently characterized by similar features of the individual labels. Pop music, on the other hand, also interchangeably called "popular music", is not quite a particular genre in regards of characteristics, but refers more to songs which are popular, that could be from varying genres.

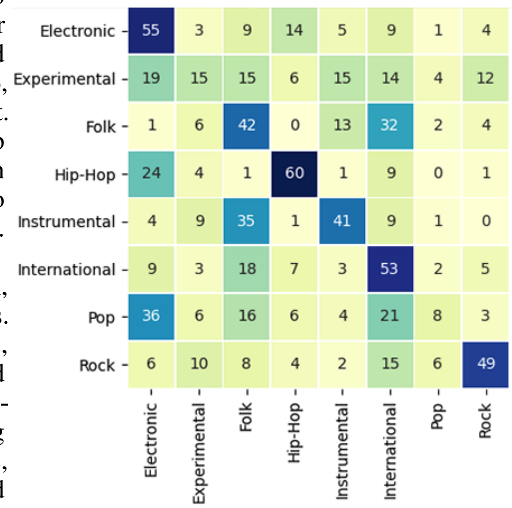

Figure 5: Confusion matrix for C-RNN.

Genre	Precision	Recall	F1-score
Electronic	43.63%	48%	45.71%
Experimental	28.28%	28%	28.14%
Folk	30.43%	35%	32.55%
Hip-hop	53.84%	63%	58.06%
Instrumental	44.64%	50%	47.16%
International	34.10%	59%	43.22%
Pop	29.41%	5%	8.54%
Rock	71.92%	41%	52.22%
Average:	42.04%	41.12%	39.46%

Table 1: Classification report for the prediction of the C-RNN.

Genre	Precision	Recall	F1-score
Electronic	46.77%	58%	51.79%
Experimental	37.86%	39%	38.42%
Folk	37.86%	53%	44.17%
Hip-hop	66.34%	67%	66.67%
Instrumental	56.16%	41%	47.40%
International	60.00%	48%	53.33%
Pop	35.90%	28%	31.46%
Rock	67.33%	68%	67.66%
Average:	51.02%	50.25%	50.20%

Table 2: Classification report for the prediction of the CNN.

4.3 CNN-RNN Parallel

The CNN-RNN parallel network, as shown in Figure 3, combines the output of the CNN in Figure 1 and with an RNN component output. It was trained for 50 epochs, batch size 32, optimized by Adam optimizer with a learning rate 0.001. Early stopping monitoring the validation accuracy with patience 20 was employed, as well restoration of the best model weights during the training process. The final test accuracy was 52.25%, the highest accuracy obtained in this work, the precision, recall and F1 score shown in Table 3, and the confusion matrix in Figure 6. The results here are similar to those of the basic CNN, only improved by 2%.

Genre	Precision	Recall	F1-score
Electronic	47.55%	68%	55.97%
Experimental	36.29%	45%	40.18%
Folk	44.52%	65%	52.85%
Hip-hop	78.89%	71%	74.74%
Instrumental	50.68%	37%	42.77%
International	51.22%	63%	56.50%
Pop	50.00%	17%	25.37%
Rock	80.60%	54%	64.67%
Average:	54.97%	52.50%	51.63%

Table 3: Classification report for the prediction of the CNN-RNN parallel architecture.

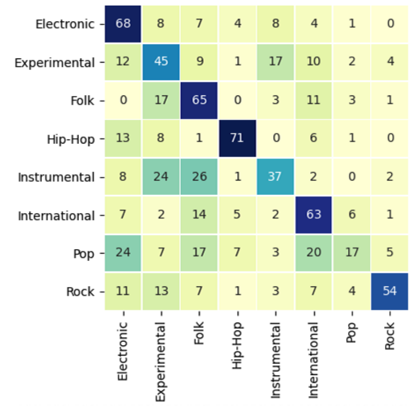

Figure 6: Confusion matrix for the prediction by the CNN-RNN parallel architecture.

5 Discussion And Future Work

Across all experiments, the genre 'Pop' was misclassified more than any other genre. Although the accuracy was higher than random (12.5%), the ranking is quite low. In machine learning, models are often a mystery, so there is no definite explanation for this. As was discussed previously, 'Pop' as a genre is not very tightly bound and contains subsets of other genres, depending on which songs were popular. For example, often times Ed Sheeran's music is classified as pop, although it's primarily guitar based, and the same could be said for Lady Gaga's, despite its heavy synthesizer effects. One could argue that the FMA dataset has taken a rather ambiguous definition of 'Pop'.

Similar trends can be observed for the different approaches. Across most experiments, 'Hip-hop' obtained highest accuracy, which can be explained by the rapped vocal passages as distinctive features. 'Electronic' was the second best, which unlike 'Pop' using synthesized music on occasion, has more unique characteristics as a genre. Interestingly, 'Electronic' and 'Hip-hop' were interchangeably misclassified as each other more than any other genre. This can be explained by the very nature of the two genres, especially as hip-hop tracks are often produced using electronic samples. Instrumentally speaking, 'Hip-hop' and 'Folk' bear few similarities, so there was very little confusion between them.

For future work it could be useful to look into specific features of the data, as well as to incorporate more of the metadata−additional information such as artists and album years could help with the more general genres, for example 'Experimental' and 'Pop'. As discussed in the C-RNN experiments section, larger spectrograms could be used for better classification of 'Instrumental' and 'Folk' genres. The next step to be taken with an improvement of the model, would be developing a recommendation engine which, by means of a proper analysis of the main genres a person listens to, could provide accurate recommendations about what to listen next. But for initial improvements, larger spectrograms using all 30s of the tracks should be used.

6 Conclusion

Classifying tracks by genre is a challenging yet very interesting task, especially with the FMA small dataset, which many have taken upon as a challenge to improve the benchmark accuracy. The amount of available data as well as its pre-processing plays a crucial role in here, since music has a lot of variations, genres that at the very same time encapsulate sub-genres, which can be quite different between each other. Moreover, inside the very same song, the style of some sections might be quite different to others, adding a new level of complexity. From what has been tested, CNN-RNN parallel structures yielded the best result of 52.25%.

The computational cost was the most challenging part of this project, triggering that new approaches had to be figured out. The mechanisms of augmenting the amount of available data by taking short random samples from the tracks seem to be a good choice, although ideally, it would have been better to use the entire length of the tracks and provided in the dataset. Data normalization has also proven to be a good method for the purpose of enhancing the models generalization properties.

References

[1] Changsheng Xu, Namunu Maddage, Xi Shao, Fang Cao, and Qi Tian. Musical genre classification using support vector machines. volume 5, pages V – 429, 05 2003. ISBN 0-7803-7663-3. doi: 10.1109/ICASSP.2003.1199998.

[2] Achmad Mutiara, Rina Refianti, and N.R.A. Mukarromah. Musical genre classification using support vector machines and audio features. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol. 14:1024–1034, 05 2016. doi: 10.12928/telkomnika.v14.i3.3281.

[3] Michael I. Mandel and Daniel P.W. Ellis. Song-level features and support vector machines for music classification. *LabROSA, Dept. of Elec. Eng., Columbia University, NY NY USA*, 2005.

[4] Shusen Zhou Xiaolong Wang Xiaohong Yang, Qingcai Chen. Deep belief networks for automatic music genre classification. *Department of Computer Science and Technology Key Laboratory of Network Oriented Intelligent Computation Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China*, 2011.

[5] Tillman Weyde Artur Garcez Son N. Tran, Daniel Wolff. Feature preprocessing with rbms for music similarity learning. *Department of Computer Science, City University London, Northampton Square, EC1V 0HB, UK*, 2014.

[6] Tom Li, Antoni Chan, and Andy Chun. Automatic musical pattern feature extraction using convolutional neural network. *Lecture Notes in Engineering and Computer Science*, 2180, 03 2010.

[7] Weibin Zhang, Wenkang Lei, Xiangmin Xu, and Xiaofeng Xing. Improved music genre classification with convolutional neural networks. In *Interspeech 2016*, pages 3304–3308, 2016. doi: 10.21437/Interspeech.2016-1236. URL http://dx.doi.org/10.21437/Interspeech.2016-1236.

[8] Wenhao Bian, Wang Jie, Bojin Zhuang, Jiankui Yang, Shaojun Wang, and Jing Xiao. *Audio-Based Music Classification with DenseNet and Data Augmentation*, pages 56–65. 08 2019. ISBN 978-3-030-29893-7. doi: 10.1007/978-3-030-29894-4_5.

[9] Chen Chen Chi Zhang, Yue Zhang. Songnet: Real-time music classification, 2018.

[10] Adiyansjah, Alexander Gunawan, and Derwin Suhartono. Music recommender system based on genre using convolutional recurrent neural networks. *Procedia Computer Science*, 157:99–109, 01 2019. doi: 10.1016/j.procs.2019.08.146.

[11] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference*, 2017. URL https://arxiv.org/abs/1612.01840.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016. doi: 10.1109/CVPR.2016.90.

[13] M. Sordo. *Semantic Annotation of Music Collections: A Computational Approach*. PhD thesis, Universitat Pompeu Fabra, Barcelona (Spain), 12/2011 2012.

[14] Benjamin Graham. Fractional max-pooling, 2014.

[15] Clevert, Djork-Arné, Unterthiner, Thomas, Hochreiter, and Sepp. Fast and accurate deep network learning by exponential linear units (elus), Feb 2016. URL https://arxiv.org/abs/1511.07289v5.

[16] Goodfellow, Warde-Farley Ian J., David, Mirza, Mehdi, Courville, Aaron, and Yoshua. Maxout networks, Sep 2013. URL https://arxiv.org/abs/1302.4389.