Multi-class classification of music genres*

Bartosz Karpiński[1] and Olga Wawryk[1]

Wrocław University of Technology, Wrocław, Poland pwr.edu.pl

Abstract. This research shows the summary of work that we have done in relation to the classification of music genres on the music data gathered by us.

Keywords: Classification · Music data

1 Introduction

In recent years due to the popularity of music streaming services the music business has changed dramatically. Users oftentimes want to be able to catalogue or simply categorize and organize their music collection. Music genres are one of the best ways to do so. Additionally, the ability to easily recommend users new music which they would find interesting has become one of the most important problems for the music providing services. Also here, music genres are one of the best ways to classify music, as they look at the music in a much broader perspective than simpler parameters, and provide much more insight about the piece of music in relation to other pieces. Thus the tools that would ease and help automate such work could prove very useful. A multi-class classifier trained on a broad-spectrum data can be very effective to manage such task. The more objective results of one will be able to detect more nuances and as a result be able to describe the data in the more precise and meaningful way. Most of the works in this field are based on just few datasets, that can be found in most of the studies. Our goal is to train and validate the classifiers on the new dataset, completed from scratch from the real-world data, that is sure a good representation of the current trends and in general the state of the music industry.

2 Related works

The music classification topic has been brought to the surface in many research articles, discussing the various approaches and methodologies that can be applied to the problem. The first work [1] did a case study of the performance of different classification approaches, varying from the very basic ones like kNN to more advanced ones like Convolutional Neural Networks. This research gave interesting insight about the solutions, as depending on the length of the sample (as the two discussed were 3 and 30 second samples) the results varied drastically. In the shorter samples kNN was definitely dominant, whereas in case of

* Supported by PWr.

30 second samples its result were rather poor. Another article [2] discussed another approach to analyzing the sound data, by means that are oftentimes more known to the researches, so by analyzing spectrograms. Those are the projections of sounds onto an image, thus providing the ability to use Neural Networks designed for pattern-recognition in images to be used for the music classification tasks. The next work [3] dives deeper into the usage and usefulness of simpler Machine Learning algorithms such as SVM or Logistic Regression to fulfill this task. Similar idea is presented in the other work [4], but here the authors extract the information through the use of Mel Frequency Cepstral Coefficients. The use of representation learning in pair with Convolutional Networks is brought up and tested in the next article [5]. Besides using the spectrograms to perform the classification, the authors also performed experiments on the fusion of both learned features and handcrafted features, to assess the complementarity of those two approaches. The use of Convolutional Recurrent Neural Networks [6] shows, that the usage of CNN and RNN hybrid brings a very strong performance. Usage of time and space decomposition with combination of ensemble classifiers was the main idea of the next work [7]. The authors used among others KNN, Naïve-Bayes, Decision Trees and SVM on the novel, Latin dataset. The last article [8] dives into the usage of more classical features, such as chord or melody, to perform the classification tasks.

3 Experimental Evaluation

3.1 Research questions

The two research questions we want to answer are:

- Do the models trained on longer snippets give better results than the ones trained on the shorter ones?
- Does the Mel-scaled spectrogram give better results when compared with the non-scaled spectrogram?

3.2 Dataset

The dataset base is created from the popular music from years 1960 to 2023. It consists of 6 classes, and the samples were chosen so they belong to one class only in a broader classification. The files used for the extraction of snippets were mostly MP3 files, with some being in M4A format or FLAC format. The samples were taken at both 5 second length, and 15 second length from the same songs, and were picked from the songs that were at least one minute long, at the 40 second mark and 30 second mark for 5 second and 15 second samples, respectively. The labels were extracted from the RateYourMusic [9] online database.

Then this base was processed into three datasets, each consisting of 2670 samples in total:

- Tabular data, consisting of 15 distinct features extracted through the librosa library

Table 1. Dataset classes and samples

Class	N. of samples
Classical	438
Electronic	443
Hip-Hop	477
Metal	364
Pop	504
Rock	444

- Spectrogram images, extracted through the ffmpeg library
- Mel Spectrogram images, also extracted through ffmpeg library and then converted into the Mel scale

3.3 Experimental Environment

The experiment was done on Arch-based Linux distrubtion, namely EndeavourOS. The code was written in Python 3.11. The base for all models was Tensorflow 2.15, and was using CUDA 12.3 drivers. The processor used was a 6-core 12-threads AMD 5600X, and the graphic card was 8GB VRAM Nvidia RTX 3070.

3.4 Model Architectures

Below is the description of each model used during the experiments:

- Initial Model - the first tested solution. Consists of three Dense layers, each consisting of 128 neurons and having activation of type 'relu'. Last layer has 6 neurons and 'softmax' activation
- Middle Model - also has 3 Dense layers, each 64 neurons and also 'relu' activation. Between each there is performed a BatchNormalization, and a 0.5 Dropout. Last layer is also a 'softmax' activated 6 neuron Dense layer
- Last Model - 3 Dense, 32 neuron and 'relu' activated layers, after each there is performed a BatchNormalization and a Dropout of 0.3, 0.5 and 0.7, respectively. Finally a 6 neuron 'softmax' activated Dense layer.
- Image Model - model used on the spectrogram data. Has 3 consecutive pairs of Conv2D layers and MaxPooling2D layers. Conv2D have (3,3) kernel sizes, 'relu' activation, and 32, 64 and 128 filters, respectively. MaxPooling2D layers have each (2,2) pool size. They are followed by a Flatten layer, one Dense layer of 'relu' activation and 128 neurons, then a 0.5 Dropout and finally a 6 neuron 'softmax' activated Dense layer.

Each model was a sequential one, used 'adam' for optimizer and had batch size of 128. First three models use Sparse Categorical Crossentropy for loss function, and the number of epochs for each was 4000. The Image Model had 200 epochs and used Categorical Crossentropy loss function.

3.5 Experiment scenarios

First experiment comparison of performance of Neural Networks model depending on the snippet length.

Scenario Models of different parameters and build (First Model, Middle Model, Last Model) were trained on the Tabular dataset, independently for 5s samples and 15s samples. Then for each model the pair of results was compared using different metrics.

Goals The goal is to compare the the performance of the models of the same parameters against dataset of different sample length, to check whether additional computational complexity during the preprocessing of the data connected with the longer snippets yields better results.

Second experiment comparison of the results of the models depending on the spectrogram type: basic and Mel scaled.

Scenario Using the same model (Image Model) on two datasets (Spectrogram images and Mel Spectrogram Images) and for two snippet lengths (5s and 15s) the results from different datasets were compared using various metrics.

Goals The goal is to compare the the performance of a model using same parameters against differently obtained datasets

3.6 Evaluation Protocol

Both experiments were performed using Repeated Classified K Fold, with 2 splits and 5 repeats, for each model and dataset configuration. After each iteration the base accuracy, F1 score, Balanced Accuracy, Precision and Recall were computed and stored. Using the means of each of those metrics, the graphs were obtained. For the statistical analysis, the t-tests were performed for a chosen p-value significance level of 0.05.

4 Results

4.1 First experiment

The table below shows the comparison of different metrics for each model and snippet length:

Model	Length	Accuracy	F1	Balanced	Precision	Recall
Initial	5s	0.462	0.461	0.464	0.461	0.462
	15s	0.497	0.495	0.5	0.494	0.497
Middle	5s	0.52	0.512	0.523	0.515	0.52
	15s	0.545	0.538	0.548	0.541	0.545
Last	5s	0.511	0.505	0.513	0.511	0.511
	15s	0.531	0.525	0.533	0.527	0.531

Table 2. Performance Metrics for Different Models

The models were built differently in order to combat the overfitting of the data, which was very apparent when processing the Initial model. Thus the Middle and Last models have different structures, with reduced number of neurons to prevent the data generalization. In addition, common methods for preventing overfitting like Batch Normalization and Dropout were used, with different parameters. From the results we can see that the Middle Model performed the best; Last model was probably too simplified, with reduced number of neurons in comparison to the Middle one.

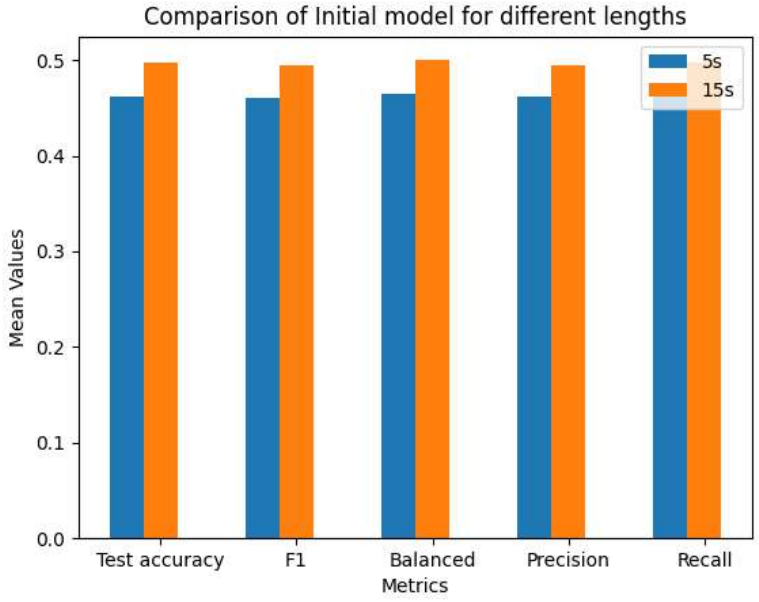

Fig. 1. Initial model metrics scores

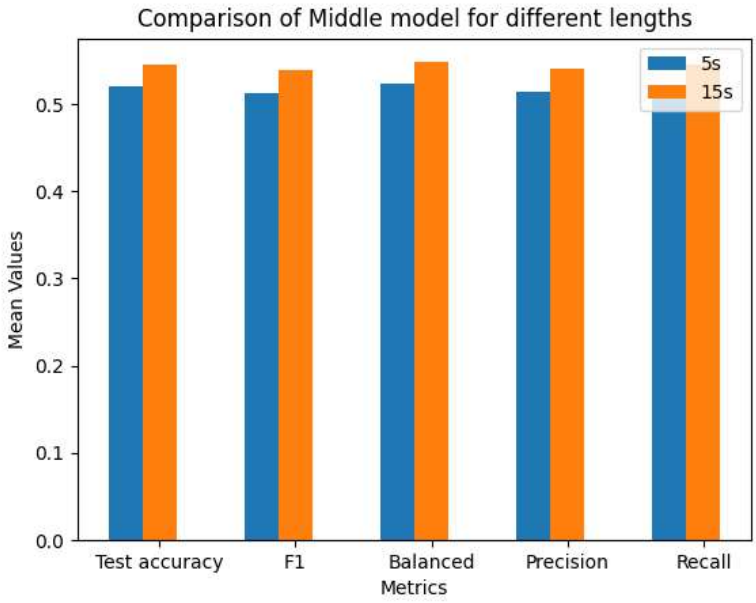

Fig. 2. Middle model metrics scores

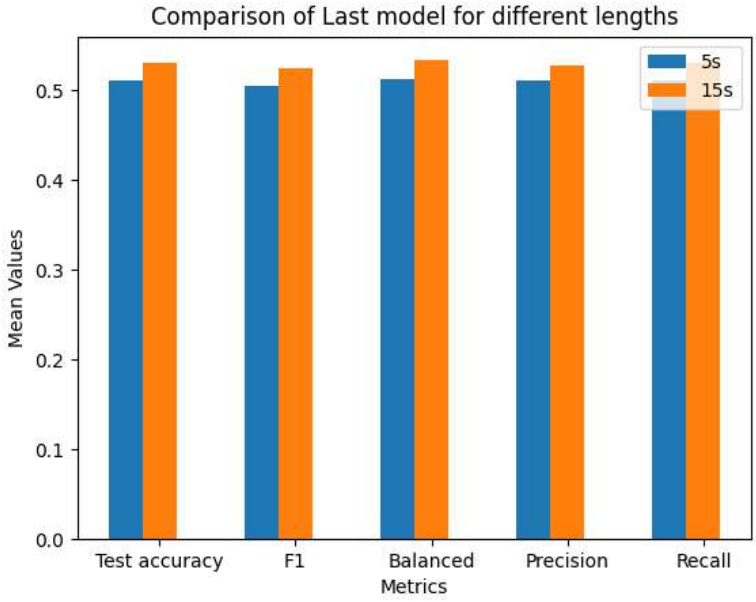

Fig. 3. Last model metrics scores

For the T-Test the Middle Model's results were used, due to it's best performance. The results are presented below:

	5s	15s
5s	NaN	-4.10010307
15s	4.10010307	NaN

Table 3. T-statistic for Middle Model

	5s	15s
5s	NaN	0.00267629
15s	0.00267629	NaN

Table 4. P-value for Middle Model

We can see that the T-Statistic favours 15s dataset, and the P-value is much smaller than the chosen significance level, being 0.05. Below are presented the comparison of performances of models for each dataset:

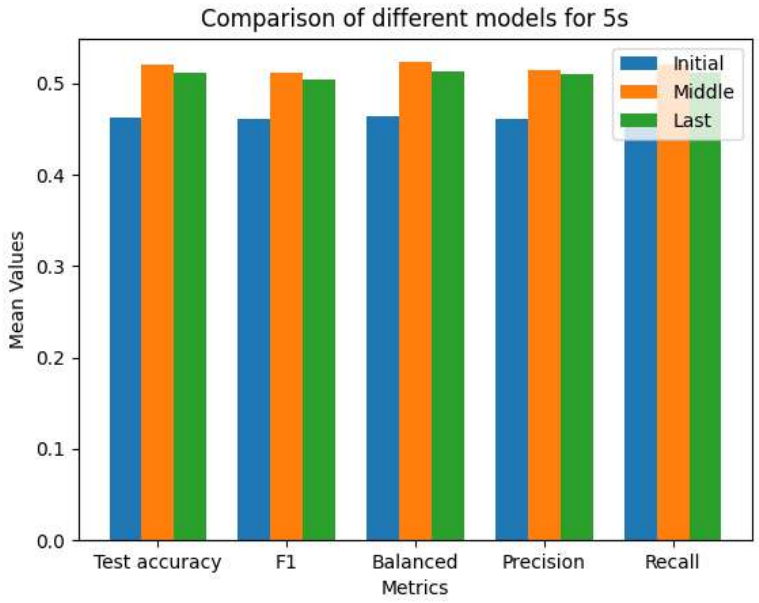

Fig. 4. Performance comparison for 5s snippets

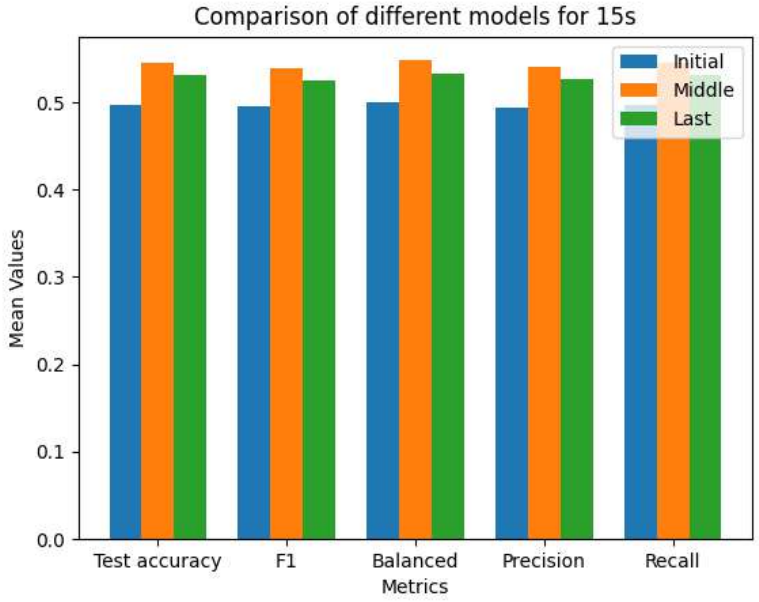

Fig. 5. Performance comparison for 15s snippets

4.2 Second experiment

The table below shows the comparison of the results for different metrics for different spectrogram types:

Type	Length	Accuracy	F1	Balanced	Precision	Recall
Basic	5s	0.484	0.481	0.485	0.484	0.484
	15s	0.512	0.508	0.512	0.513	0.512
Mel	5s	0.481	0.478	0.48	0.486	0.481
	15s	0.519	0.511	0.52	0.515	0.519

Table 5. Performance Metrics for Different Spectrograms

The T-Test was performed for the clearly better results of 15s dataset and are presented below:

	5s	15s
5s	NaN	-1.71442452
15s	1.71442452	NaN

Table 6. T-statistic for Spectrograms

	5s	15s
5s	NaN	0.12059452
15s	0.12059452	NaN

Table 7. P-value for Spectrograms

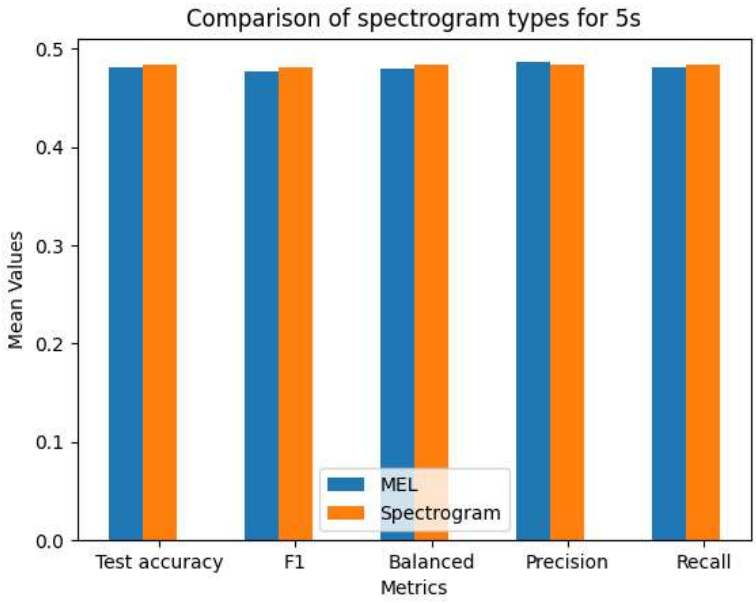

Fig. 6. Performance comparison for Spectrogram 5s snippets

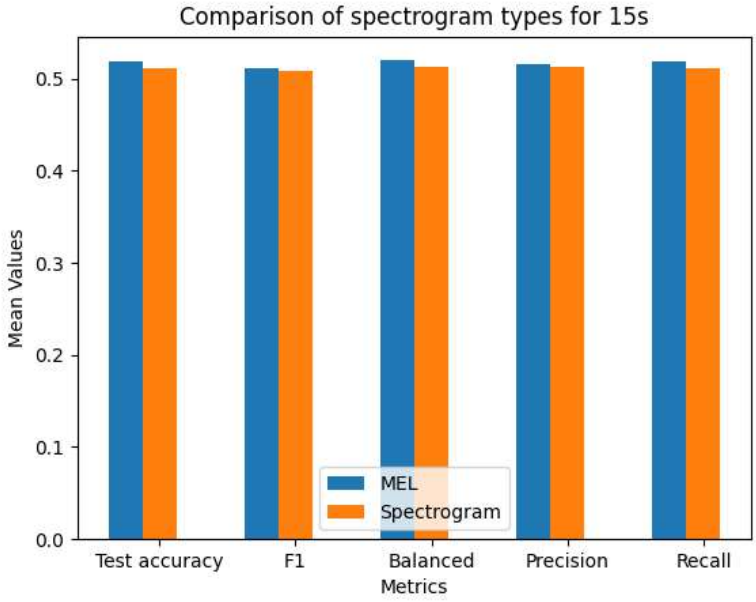

Fig. 7. Performance comparison for Spectrogram 15s snippets

Looking at both the T-Test tables, as at the graphs, we can see that the results obtained from by the Mel spectrogram processing were slightly better than the basic spectrogram ones. However, the value of P-Value of around 0.12 and T-Statistic magnitude of around 1.71 suggest, that the differences are rather minuscule.

5 Conclusions

The obtained results from the First Experiment clearly favour the longer snippets. For each model, independent of the performance against each other, the results for the 15s snippets were better than for the 5s ones. Additionally, the T-Test done for the best, Middle Model shows that the differences are rather significant. In the second experiment however, the Mel Spectrograms had slight, albeit insignificant advantage over the basic spectrogram. This experiment also showed, that the longer snippets make better datasets, as the results here in comparison between the length clearly favoured the longer ones.

References

1. Ndou, Ndiatenda, Ritesh Ajoodha, and Ashwini Jadhav. "Music genre classification: A review of deep-learning and traditional machine-learning approaches." 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). IEEE, 2021.
2. N. M R and S. Mohan B S, "Music Genre Classification using Spectrograms," 2020 International Conference on Power, Instrumentation, Control and Computing (PICC), Thrissur, India, 2020, pp. 1-5, https://doi.org/10.1109/PICC51425.2020.9362364
3. Bahuleyan, Hareesh. "Music genre classification using machine learning techniques." arXiv preprint arXiv:1804.01149 (2018).
4. Ali, Muhammad Asim, and Zain Ahmed Siddiqui. "Automatic music genres classification using machine learning." International Journal of Advanced Computer Science and Applications 8.8 (2017).
5. Costa, Y. M. G., Oliveira, L. S., Silla, C. N. (2017). An evaluation of Convolutional Neural Networks for music classification using spectrograms. Applied Soft Computing, 52, 28–38. https://doi.org/10.1016/j.asoc.2016.12.024
6. Choi, K., Fazekas, G., Sandler, M., Cho, K. (2017). Convolutional recurrent neural networks for music classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp.2017.7952585
7. Silla, Carlos N., Alessandro L. Koerich, and Celso AA Kaestner. "A machine learning approach to automatic music genre classification." Journal of the Brazilian Computer Society 14 (2008): 7-18.
8. Y. -L. Lo and Y. -C. Lin, "Content-based music classification," 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 2010, pp. 112-116, https://doi.org/10.1109/ICCSIT.2010.5563642
9. RateYourMusic Homepage, https://rateyourmusic.com/.