Members
Anca-Mihaela Matei: s4004507
Óscar Nebreda Bernal: s3434745
Rajiv Jethoe: s3490750

Classifying Music Genres: A Deep Dive into Sound Analysis

Audio Processing and Indexing - Final Project

1 Introduction

This research aims to offer a scientifically accurate depiction of music genre classification. Therefore, the focus of this systematic study is to reveal the intricacies of the high-dimensional space within the network and to present a clear understanding of how newly classified songs fit into the wide range of musical genres.

This project evaluates various well-established large image classifiers and selects the one demonstrating superior performance. The chosen classifier is then employed, utilizing the transfer learning method to train on spectrogram images derived from the GTZAN dataset[1].

Another noteworthy feature of our approach is the incorporation of embedding layers into the redesigned architecture. These stratified layers deliberately occupy spaces inside deep neural networks, seizing high-level characteristics to increase our comprehension of the classification process.

The study goes beyond traditional techniques to include dimensionality reduction technologies. By methodically examining the connections between different musical genres, these algorithms provide light on how they tend to cluster together in high-dimensional space. The main goal is to provide an objective and understandable representation of the procedure so that users may better understand how songs fit into the network.

2 Related Work

Transfer Learning for Music Classification and Regression Tasks
Genre Classification using Word Embeddings and Deep Learning
Music Genre Classification with ResNet and Bi-GRU Using Visual Spectrograms

3 Dataset

The GTZAN dataset, a widely recognized and extensively used collection in the domain of music genre classification was chosen for conducting this research. The 1,000 audio recordings in the dataset are evenly split between 10 genres, offering a varied and well-balanced collection of musical samples. With each track lasting 30 seconds, it is considered to be a manageable and available corpus for training and assessment. The 10 genres are

blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. This variety not only facilitates the capture of nuanced high-level features but also plays a crucial role in the subsequent formation of clusters. The exploration of these clusters illuminates the intricate relationships and distinctions between the genres, providing a comprehensive understanding of their positioning in the high-dimensional space.

In the analysis, the dataset was predominantly composed of spectrogram images. Figure 1 provides an illustrative example of these spectrograms, each corresponding to a distinct genre. Notably, these images were generated from 999 recordings out of the total 1000 in the GTZAN dataset. Alas, one audio was difficult to preprocess with well-known libraries such as librosa, resulting in its exclusion from the dataset. This rigorous curation process ensures the integrity and reliability of the dataset for our deep dive into sound analysis and music genre classification.

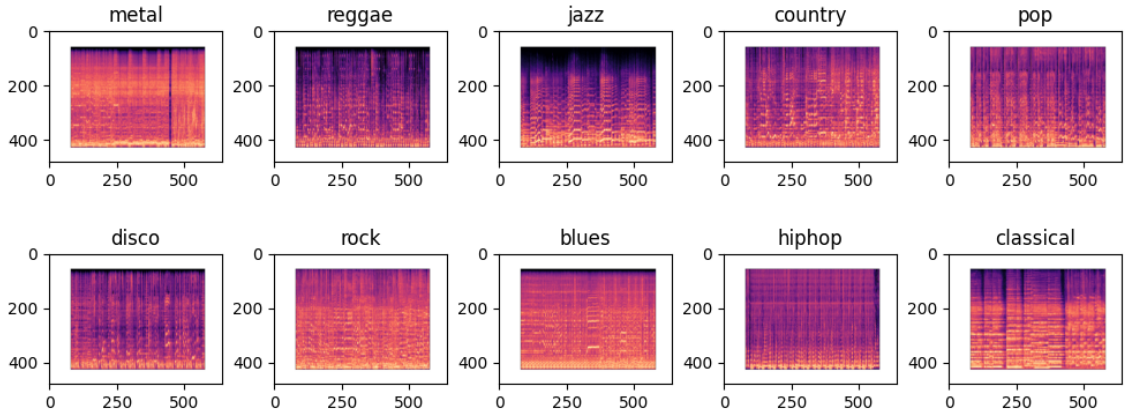

Figure 1: Spectrogram examples for distinct genres

4 Methodology

4.1 Transfer Learning

An investigation was conducted using transfer learning techniques with four pre-existing models: VGG16[6], Inception[7], Xception[3], and MobileNetV2[4], in order to find the most efficient and accurate spectrogram classification. The goal was to identify the most effective model for the spectrogram classification task. The results obtained after implementing these models can be observed in Table 1.

MobileNetV2 demonstrated superior performance, surpassing the accuracy of other models by a substantial margin of over 10%, elevating the classification accuracy from 60% to an impressive 73%. One of the main reasons for this outcome might be represented by the MobileNetV2's architecture. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers as it can be observed in Figure 2.

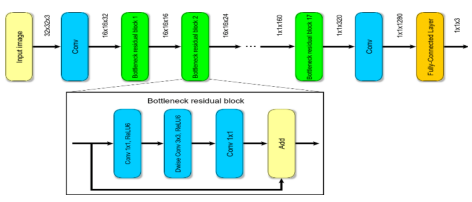

Figure 2: MobileNetV2 architecture [5]

The bottleneck layer consists of a depthwise separable convolution followed by a linear bottleneck and a shortcut connection. The depthwise separable convolution is an essential component, reducing computational cost by independently convolving each channel and before applying a 1x1 convolution [4]. This demonstrated to be effective for muti-scale pattern recognition and thus, might align well with the variety of patterns found in the spectrogram data.

No.	**Pretrained Model**	**Test Loss**	**Test Accuracy**[%]
1	VGG 16	9766.05	62.99
2	InceptionV3	1.4073	68
3	Xception	0.9703	68
4	MobileNetV2	0.8062	73

Table 1: Results obtained for various model implementations

It is important to notice that for this particular problem, the classification is not made in unequivocal classes. I.e., one can be certain about the difference between a *cat* and a *dog* classification, but it seems rather artificial to decide if a pop-rock song is only *pop* or *rock*. This means that too high values of accuracy probably represent an overfitting, even if they do well with the test set, to particular annotator preferences.

4.2 Data Augmentation

As discussed in 3 the GTZAN dataset was used, but what is evident is that the dataset is quite small. Since transfer learning is being applied whilst making use of pretty large well established CNNs it would seem like a logical conclusion that whilst in the process of retraining these image classifiers the model would lack a diverse set of training data in amount and scope and it would mean the model would start overfitting.

To this end, research was conducted on adding data augmentation methods to enlarge the dataset. The data augmentation focused on adding 4 data augmentation methods, namely:

- Gain variance
- Semitone shifting
- Noise addition
- Spectrogram segmentation

The thought process was to add all of these data augmentation methods to the affect of enlarging our database around at least 5 times. The librosa library was used for all of these data augmentation methods. This well known library has all these functionalities built in. This way the focus could lie on research of efficacy of the data augmentation methods instead of implementing these methods from scratch.

Spectrogram segmentation: the process of dividing up the music sample into same sized sub samples was thought of as a reasonable manner to increase the dataset size. When doing this several splits were tried, namely: 10 second segments and 5 second segments. Applying this data augmentation method should increase the dataset by 3

and 6 times respectively. What was noticed after training the networks is that the test accuracy stayed the same or varied by 3% up or down the baseline using the 10 second fragments whilst it went down around 5-10 percent with the 5 second fragments. This can be due to various reasons, but the main reason is probably the fact that when using segments that are to short we loose musical context. Aspects of certain musical genres are more evident when observing a larger part of the musical piece. This makes sense when the split is observed that was used.

The choice was made to continue with the 10 second fragments per song for further testing to save on VRAM and to save computational time. Now the other data augmentation methods were used on these three segments. An algorithm was devised where the 3 segments of a piece were equally likely to have one of the three data augmentation methods applied to it, it could also be that nothing would be applied. This way more diversity is created among the dataset by adding slight variances in the spectrograms.

Data leakage was a problem during creating the splits of the dataset. This is a very important problem that was cropping up. At some point an unlikely high test accuracy was achieved of close to 100 percent. This made it evident that data leakage was occurring and thusly more care was taken in to creating proper train/test/validation split which solved the problem.

Model	30 Second - clean	10 Seconds - DA	5 Seocnds - DA
MobileNetV2	75.9% Acc	74.6% Acc	67.3% Acc
Xception	66.0% Acc	64.3% Acc	59.6% Acc

Table 2: Data augmentation results, DA means data augmentation and this implies all augmentation methods were used during this run.

in 2 the results of training the different models that were applied can be observed. What is immediately evident is that using the full dataset without any augmentation works the best. This is likely because the model is starting to overfit on the other datasets. You can see that the test accuracy gets worse the more segments there are. This is likely because the spectrograms with data augmentation do not add enough variance in the data for the model to see them as completely new songs. This way the model sees the "same" training example and starts to heavily overfit on this data.
After the data augmentation phase the decision was made to finalize the project with MobileNetV2 on the unaltered dataset of full 30 second audio files.

4.3 Embedding introduction for cluster creation

In section 4.1, we introduced the problematic behind a classification on not closed classes. For this reason, it may be beneficial to explore the network beyond the one-hot classification output.

For doing so, we suggest two approaches. First, using the vector representation to extract distance measures as similarity between items. Second, a dimensionality reduction algorithm to 2D to provide a clear understanding on where the different songs lie with respect to each other. Working in high dimensions, however, can lead to counter-intuitive behavior about where items lie relative to each other [2].

The input of the network is extremely high dimensional, represents low level features, and is sparse, as most pixels carry little to no relevant information. As the input is passed forward to the consecutive layers of the neural network, the vector representation is mapped into lower dimensionalilty that captures high level features, and that has a dense representation, since the elements now represent the intensity of relevant features in an abstract latent space.

This means that somewhere inside our network we expect to be able to find a representation that balances the dimensionality problem without being the one-hot output of the classification. Figure 3 represents the choices and definitions for the embedding in our network architecture.

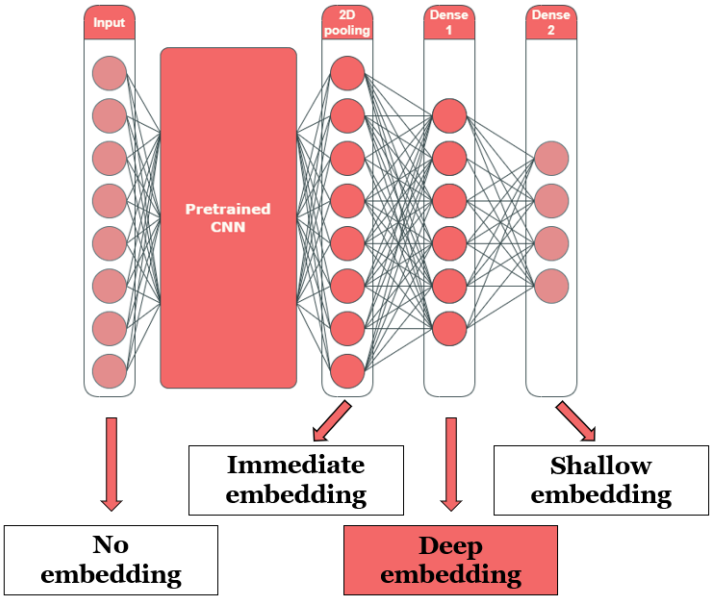

Figure 3: Different layers and embedding choices from our network architecture.

The first two embedding techniques are of particular interest to asses the relevance of our work, because they remain unaltered by our transfer learning process. The *immediate embedding* is extracted right after the pretrained CNN chosen. This means that if the dimensional feature extraction from the CNN is sufficient to capture the relation between the elements, our transfer learning efforts are useless for this task. The *no embedding* is even more drastic, as it represents the raw spectrogram image rearranged as a list of pixel values.

Figure 4 contains the results of the dimensionality reduction applied to the different

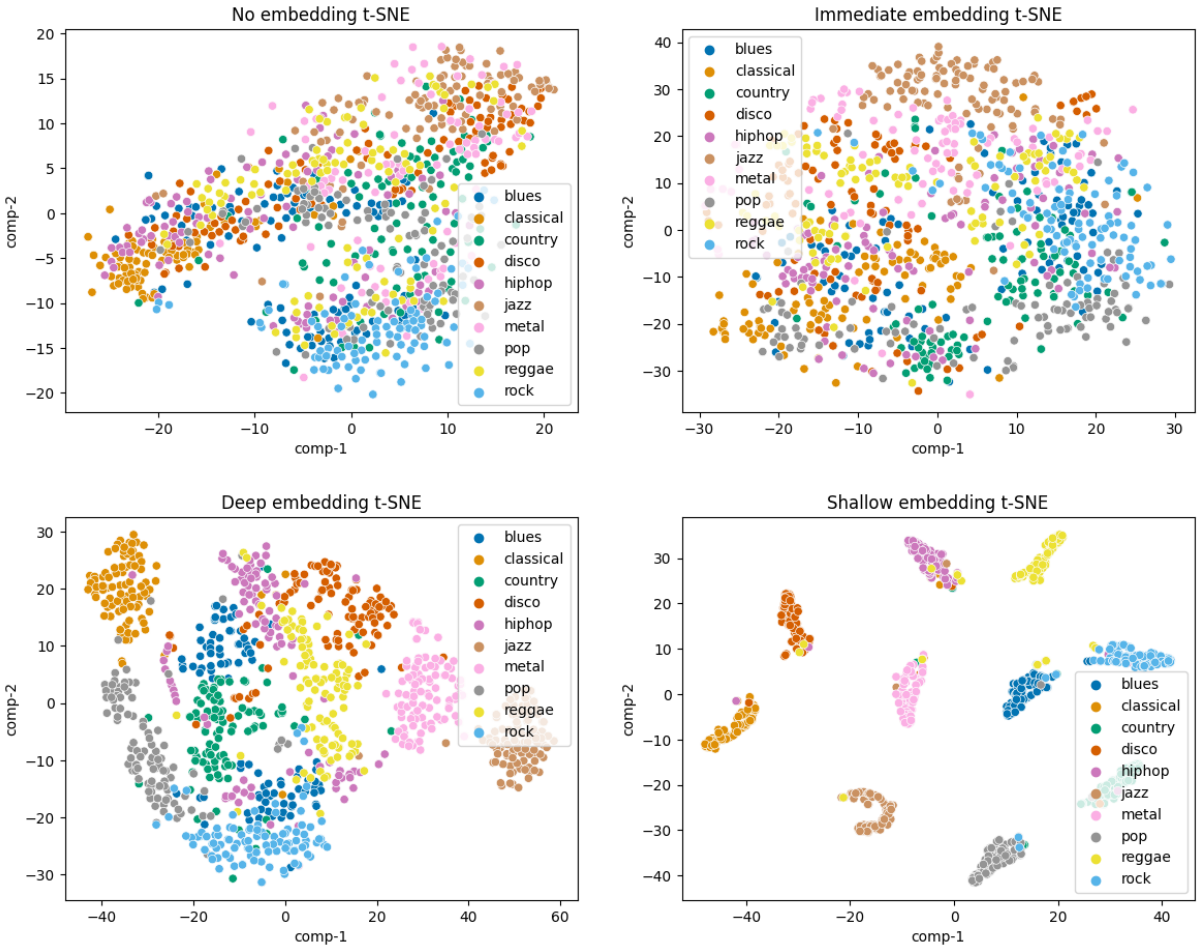

Figure 4: t-SNE 2D mapping applied to different embedding choices.

embeddings.

The results for t-SNE for *no embedding* are, as expected, not very informative. For the *immediate embedding*, there is more separation of classes, but it is clear that it is not enough (e.g., the classical music cluster spans across many unrelated genres). *Deep embedding* seems to balance things in the perfect way, where clear distinction between classes are captured, but information from their relative positions can be derived and insightful intersections are present. *Shallow embedding* is too far into the classification process and, although it has a clear clustering of classes, it provides no information about the relations between them. For this reason, we conclude that the *deep embedding* is the most adequate one for our task.

In the Appendix, the average of the cosine similarity between all pairs of songs grouped by genres is presented as a further exploration. The results confirm the same patterns in terms of clustering and differentiation within the different classes. Besides, the results in absolute scale further confirm the dimensionality problem [2] we intend to overcome, with all similarities in high dimensional spaces being extremely high (all elements lie too close to each other).

5 Conclusion

In this exploration of music genre classification through deep sound analysis, our journey has unfolded with a focus on leveraging advanced techniques to decode the intricacies of musical compositions. As we summarize the key findings and reflections, several crucial insights emerge, contributing to the broader landscape of machine learning applications in music analysis.

The use of pretrained models illustrated how important transfer learning is for classifying musical genres. The most successful model was MobileNetV2, which emphasized how crucial it is to select the appropriate architecture. Despite our efforts to improve robustness through data augmentation, the results were not as anticipated. This suggests that the choice and application of augmentation techniques require careful consideration and adaptation to the unique characteristics of music data.

Moreover, the adoption of embedding for cluster creation provided a more complex perspective on song similarity. Leveraging vector representations moved beyond traditional classification, offering a more detailed understanding of relationships between items. Another step consisted of the dimensionality reduction technique which facilitated a clearer visualization of song relationships. Thus, it offered valuable insights into feature space distribution and aided in the interpretation of the model's internal representations.

The project's exploration of similarity measures and clustering using embedding vectors suggests potential applications in music recommendation systems. Extracted features and clustering information could enhance personalized music recommendations based on user preferences.

Despite notable success, opportunities for future exploration exist. This may involve using other music genre datasets and reimplementing data augmentation techniques as well as considering refining the already used strategies.

In conclusion, our investigation serves as a demonstrative illustration of the symbiotic relationship between deep learning methodologies and the analysis of musical constructs. The efficacious assimilation of transfer learning protocols, inventive clustering methodologies, and the search for precision, particularly through the utilization of MobileNetV2, establishes this endeavor at the confluence of machine learning principles and the aesthetic intricacies inherent in auditory arts. The acquired insights hold resonance with wider ramifications concerning the deployment of deep learning paradigms within intricate and multifarious domains.

References

[1] Gtzan genre collection, 2002. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.

[2] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In Jan Van den Bussche and Victor Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.

[4] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[5] Ulzhalgas Seidaliyeva, Daryn Akhmetov, Lyazzat Ilipbayeva, and Eric Matson. Real-time and accurate drone detection in a video with a static background. *Sensors*, 20:3856, 07 2020.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

Appendix

Cosine similarity

Averaged cosine similarity between all pairs of songs in different classes, where the left column (Absolute) provides a color map with the values from 0 to 1 and the right column (Relative) from the min to the max similarity measured for each embedding.