Music Genre Classification: enhancing the baseline architectures with DenseNet

Eugenio Fella 2053854, Matteo Pedrazzi 2076719

***Abstract*—Due to the recent growth of music libraries and streaming platforms, the task of classifying music genres is something that needs to be automated on a large scale in order to deliver finely tuned and precise music recommendations for every kind of music listener. Using extracts of the music track whose genre is to be classified, fundamental results can be obtained from the implementation of deep learning techniques combined with prior and appropriate pre-processing of the analysed signal. In our case, we attempt to correctly classify the genre of audio tracks by first implementing basic and then advanced architectures inspired by state-of-the-art models in computer vision, which include significant typical elements such as convolutional neural networks and residual networks. We have found out that the Densely Connected Neural Network (DenseNet) architecture, which connects each layer to every other layer in a feed-forward manner, excels in capturing discriminative features that are pivotal for genre recognition. Using DenseNet architecture and building mel-spectrograms as input, we are able to overcome basic CNNs-based models and achieve slightly better results than by implementing late fusion with raw one-dimensional signals, which could still remain as a possible avenue on the sidelines of our work.**

***Index Terms*—Supervised Learning, Music Genre Recognition, Convolutional Neural Networks, Densely Connected Convolutional Neural Networks.**

I. Introduction

The rise of online platforms has revolutionised the way we share artistic content, especially in the area of music streaming. With a constantly growing number of collections and playlists, manual song classification and customised recommendations have become impractical. The automatic subdivision of music tracks into genres, which are based on characteristics such as rhythm, harmony and instrumentation, finds its solution in artificial intelligence, and particularly in deep learning. Despite the often blurred genre boundaries, artificial intelligence proves remarkably effective in this complex task. This field, known as Musical Genre Classification (MGC) is a particular branch of the vast field of Music Information Retrieval (MIR). Furthermore, the development of different music datasets greatly aids research in this field.

Critical to the success of this problem is the selection of suitable representations or features associated to the piece of music under consideration. These features can be extracted from the raw audio signal, spanning from time domain representations to advanced 2D frequency domain representations such as spectrograms [1]. Spectrograms have demonstrated efficacy in previous research when coupled with 2D convolutional neural networks (CNNs) for genre classification. However, working with 2D inputs requires adjusting parameters, which can vary depending on the types of input signal [2]. Some researchers have explored the application of one dimensional CNNs directly to music signal waveforms (referred in the following as 1D signal) to capture acoustic characteristics, although further improvements are needed to match the results obtained with 2D inputs. A significant gap in the current literature is the study of how a combination of these two input signals can improve performance in this task.

In this work, our goal is to try to apply a state-of-the-art computer vision architecture in the field of MGC, thus focusing on 2D spectrogram inputs. To obtain comparable results, having access to relatively limited computational resources, we will exploit appropriate data augmentation techniques. Another point of this work is to compare the results obtained with these architectures with those previously obtained using some basic standard classification models inspired by [3], both for one-dimensional audio waves and two-dimensional spectrograms. An attempt is made to implement mechanisms that combine both types of input data, with the intention of understanding whether this type of approach can improve the classification of musical genres.

The results obtained using a DenseNet architecture are encouraging, surpassing the proposed basic models and achieving performances close to the ones obtained by other state-of-the-art models. The great advantage of the proposed framework is that it requires minimal prior knowledge to build a consistent and efficient predictor over a large and heterogeneous set of music genres. It also has good scalability, which refers to the ability to easily adapt the model to any dataset containing music in mp3 format. The idea of considering both one and two dimensional data types as inputs, on the other hand, achieves results comparable to those of DenseNet, suggesting that further research on this strategy could lead to improvements over advanced architectures in the future, but investigations on this aspect currently come to a dead end.

Here we present a brief summary of the techniques and framework developed. Starting from the Free Music Archive (FMA) dataset [4], the labelled tracks are fed into two different baseline architectures. We then investigate an advanced architecture for the 2D input to obtain better overall performance, and finally the output of this model is combined with that of a one dimensional CNN, with the idea of obtaining a slight boost over the advanced model.

The rest of this paper is organised as follows: in Section II we discuss recent music classifiers proposed from the literature consulted, both on raw audio and spectrograms, in Section III we present the pipeline of the pre-processing procedure, as well as the dataset and the augmentation techniques

implemented on it are detailed in IV and in Section V all the different architectures proposed. In the final part, Section VI analyses the experimental results, followed by a conclusion and reflections on future work in Section VII.

II. Related Work

The classification of music genres has been studied for years and many different approaches have tried to tackle this problem in different ways. Given the recent success of CNNs in many image and sound recognition tasks, we have consulted some of the relevant literature, with a focus on publications in the MIR research area.

Among the first to have this intuition were Dieleman and Schrauwen in 2014 [5], who based their architecture on alternating convolutional and pooling layers and attempted to correctly classify music samples using only a 3-second input vector. They realised that the spectrogram approach would outperform the raw audio input technique, but the latter is still able to discover useful features, such as the frequency decomposition of the signal. A very effective architecture for classification of raw 1D waveforms is the one proposed in [6], where the implementation of seven consecutive convolutional blocks interchanged with max pooling layers achieved performances even better that the ones of spectrogram-based in the scenario of large-scale data.

Other approaches were attempted by [7], [8] with the aim of improving performance on raw one-dimensional audio samples. Both implemented and trained a deeper convolutional neural network (DCNN), and in the work of Kim *et. al.* some advanced blocks, taken from ResNets [9] and SENEts [10], were used to obtain relevant results.

In [11] an innovative approach was used for environmental sounds classification, initializing the first convolutional layer with a Gammatone filter bank, namely a linear filter described by an impulse response that is the product of a gamma distribution and a sinusoidal tone. The proposed 1D CNN uses large kernels since it is assumed that the first layer should have a more global view of the audio signal, obtaining optimal results in sounds classification tasks.
To conclude with the 1D signal approach, Allamy and Koerich [12] also based their novel deep CNN on the use of residual blocks, but achieving state-of-the-art results thanks to the presence of many small kernels with size three and stride one.

The possibility of 2D signal extraction methods from audio tracks, such as the STFT spectrogram or the mel spectrogram, has paved the way for various convolutional neural network (CNN) architectures that have achieved remarkable success in diverse tasks, notably in image classification. In [3], two CNNs are introduced: a first one combining max and average pooling, and another incorporating a residual block, both showcasing promising results.

Advanced networks have also emerged, delving into recurrent neural networks to preserve and harness temporal features. For instance, in [13], a combination of a CNN and a log short-term memory (LSTM) layer is explored. However, as suggested by [14] and [15], more advanced CNNs, primarily designed for image-based classification, tend to outperform these approaches. Notably, [15] conducts a comprehensive analysis of Densenet on Mel-spectrogram images, yielding robust results. In a similar vein, [1] follows a comparable approach but with the ResNet architecture.

Moreover, [16] investigates the behavior of a slightly modified VGG16 architecture, with a specific emphasis on the distinct outcomes achieved through transfer learning and fine-tuning. Additionally, innovative strategies such as bilinear convolutional neural networks, featuring both DenseNet and ResNet, are explored in [17].

Recent literature [18]–[20] is available also for the techniques trying to combine different types of data inputs, approaches commonly known as data fusion. In these works are covered the main differences and advantages of each of the candidate implementations, even if none of them is applied to the MGC task.

III. Processing Pipeline

The paper processing pipeline starts by reading the mp3 file and extracting the raw waveform, followed by randomly sampling a 10-second audio frame. If the architecture requires a two-dimensional input, the short-time Fourier transform (STFT) is applied, which converts the audio signal to the frequency domain using Fourier transforms on shorter segments. A mel-spectrogram is generated, which maps the frequencies onto the mel-scale. This leads to lower frequencies gaining more importance, which is how humans interpret sound. The signal is then converted to dB-scaled spectrogram relative to its maximum.

We experimented with different architectures. Firstly, a 1D CNN-based model was used to classify audio tracks directly from the raw waveform. This model comprised a 1D CNN module for feature extraction and a fully connected module for classification. A concatenation of average and maximum pooling layers was performed prior to linking these two blocks. An analogous architecture (referred as Baseline A)

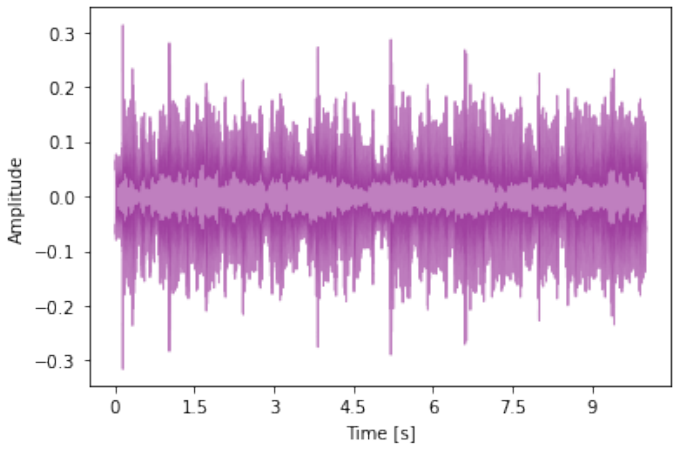

Fig. 1: Raw one-dimensional signal for a 10-second clip extracted from a Pop labelled track.

for 2D spectrogram input was tested in order to compare the performances of 2D and 1D input.

In addition, we explored another baseline architecture (referred as Baseline B) for 2D inputs, specifically the mel-spectrogram. This architecture also incorporated a residual block, which includes a shortcut connection from the first convolutional layer to the last, which is known to simplify optimisation and improve accuracy.

For the innovation, we integrated an advanced image classification architecture, DenseNet, which fosters dense connectivity between layers, promoting efficient feature propagation and mitigating the gradient fading problem, making it suitable for tasks such as image classification.

Lastly, a late fusion technique was developed to leverage both 1D and 2D information. Features extracted from DenseNet and the initial baseline were merged and inputted into a fully connected classifier for comprehensive analysis.

IV. Signals and Features

We used the Free Music Archive (FMA) dataset [4], tailored for music analysis, with a wide range of content. This dataset spans 161 genres and includes 106,574 uncut tracks from 16,341 artists and 14,854 albums. All FMA data are encoded in mp3 and are available in various sizes.

In particular, we focused on the subset *fma small*, a balanced dataset containing 8,000 songs divided into 8 genres: electronic, experimental, folk, hip hop, instrumental, international, pop and rock. This subset provides 30-second audio clips extracted from the middle part of songs (or the entire song if shorter than 30 seconds) in high-quality mp3 format. An example of raw audio signal extracted from a song is plotted in Figure 1, before applying transformations to it. The *fma small* dataset also includes pre-calculated features, along with complete track-level and user-level metadata, tags and free text such as biographies. The total size of the dataset is 7.2 GB. Before dividing it into training, validation and test set ($80\%, 10\%$ and 10% respectively), we shuffled the dataset. In addition, we identified and removed six corrupted files to ensure data integrity.

We began by extracting one-dimensional waveforms from mp3 files, using a sampling rate of 16,000 Hz for dimensionality reduction, a choice validated also by improvements in classification accuracy reported in [15]. To handle the computational requirements, we randomly selected 10-second clips (i.e. of length 160,000).

For the two-dimensional signal, we applied STFT on frames with 50% overlap as done in [3], using Hann's window function. This was followed by conversion into a mel-spectrogram expressed in dB. An example of mel-spectrogram is shown in Figure 2. In our A and B baseline approaches, we used a window length of 3472 (about 160 ms), in accordance with the architectural requirements (513x128). For DenseNet and the fusion network, we opted for a window of 1976 (about 90 ms) to obtain a final mel-spectrogram of 224x224, optimised for network performance. These operations were performed using the `librosa` library [21].

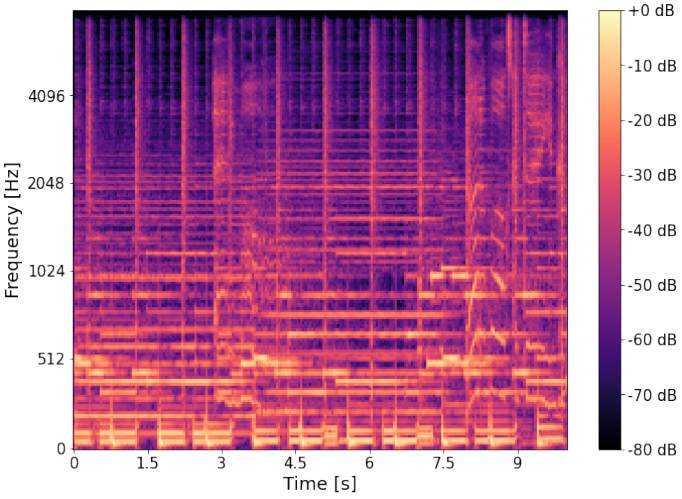

Fig. 2: mel-spectrogram for the same track considered in Figure 1.

Both 1D and 2D inputs were subjected to "z-score" normalisation. Specifically, the 1D waveform was normalised using the mean and standard deviation values calculated over the entire training set, while for the 2D spectrogram these statistics were calculated for each individual input.

During the training phase, we used data augmentation techniques. We randomly selected 10-second clips in each iteration, effectively expanding the dataset without distorting the audio content or increasing the computational requirements.

To introduce variability, we added random Gaussian noise with an amplitude uniformly chosen to remain below 3% of the maximum amplitude of the raw waveform, with a probability of 50% at each iteration. In addition, we considered time stretching with the same probability by altering the sampling rate by a uniformly selected factor between 0.9 and 1.1. This adjustment slightly affected the speed of the audio and its frequency distribution without compromising genre identification.

V. Learning Frameworks

As anticipated in Section III, we explored four distinct learning architectures, described below. After random data augmentation, each architecture was subjected to a training phase considering unweighted multi-class cross-entropy loss. We chose the Adam optimizer [22], adjusting the learning rate to 10^{-4} after experimentation. To reduce overfitting and eliminate insignificant weights, we incorporated a weight decay of 10^{-4}.

In all the networks illustrated Rectified linear units (ReLU) activation function, $ReLU(x) = max(0, x)$, are applied in all convolutional and dense layers except for the last one, where the softmax function $softmax(x)_i = e^{x_i}/\sum_j e^{x_j}$ is applied instead to convert the outcomes into the discrete probability distribution result of the music genre classification.

For the 2D architectures, we used a batch size of 32, while the 1D input was limited to a batch size of 16 due to memory constraints. For the same reason the batch size of

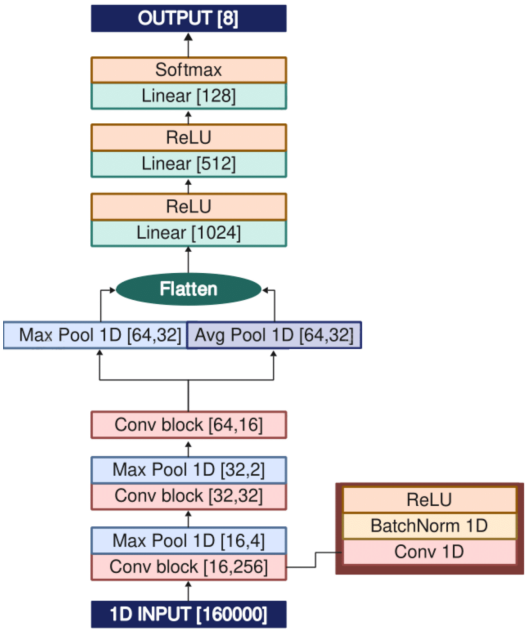

Fig. 3: 1D baseline A architecture, the 2D version uses the same blocks but with different dimensions.

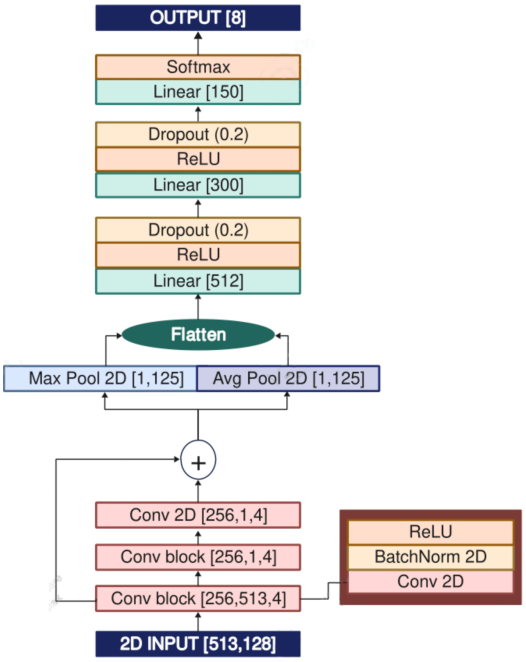

Fig. 4: Baseline B architecture.

the last architecture was fixed to 8. All architectures were subjected to 20 training epochs and only results with the lowest validation loss were retained, checking not to get into problems of overfitting.

Baseline A - 1D and 2D input

Even in light of the considerable amount of work concerning the analysis of raw audio wavelengths using CNNs, we decided to repurpose a neural network from [3], where it was originally applied to STFT spectrogram inputs. This architecture is rather simple and consists of three modules: a first convolutional module, with the purpose of learning features at different levels, consisting of 3 convolutional blocks, of which the first two are followed by a max pooling layer, which is used to allow the CNNs to look at non-overlapping regions of the signal. The last convolutional layer, receiving 64 feature maps from the initial convolutional layers, leads to the distinctive module of this network, a concatenation of max and mean pooling layers. This allows, according to the authors, to provide more statistical information to the subsequent layers. Finally, the linear module, which acts as a classifier, has an initial fully connected layer with 1024 neurons, followed by two more layers of 512 and 128 neurons each. The final layer has 8 neurons, to represent the probability of different genres. The architecture of this 1D baseline model is showed in Figure 3. An analogous network have been designed in order to compare the performances obtained from one dimensional and two dimensional input data (513x128) on similar architectures. In particular, the 2D kernels used in the convolutional blocks are the same size as those in the Baseline B model, and the classifier is also analogous to that architecture.

Baseline B - 2D input

The second architecture examined is the one that works with STFT spectrograms, i.e. a two-dimensional representation of the incoming music track, suitably adapted from the FMA dataset to fit into a 513x128 matrix. As for the previous baseline, this architecture is inspired by the work of Zhang [3], and in particular is similar to the one used for the baseline A, with the additional presence of a jump connection, typical of residual network architectures [9]. The initial convolutional block is composed of 3 convolutional layers, each associated with a respective batch normalisation layer. During the first convolution, the kernel inspects a fixed region 513x4 in the input STFT spectrogram, multiplying the input value with the associated weights in the kernel, adding the kernel bias and passing the result to the activation function. The shortcut connection sums the output of the first convolutional layer to the output of the third, increasing the possibility of obtaining better results on deeper networks. Before this operation, we used zero padding to make sure that the feature maps are of the same dimension. This jump connection is also useful for avoiding overfitting on the training data set. From this point to the end, the structure is the same as the previous baseline, the only difference being the kernel size of the pooling and max layers and the number of neurons in each classifier layer. Figure 4 presents the described architecture with the size of each individual layer specified.

Now that the baselines are defined, our interest will shift to two kinds of more advanced structures already anticipated, the DenseNet model and the user defined

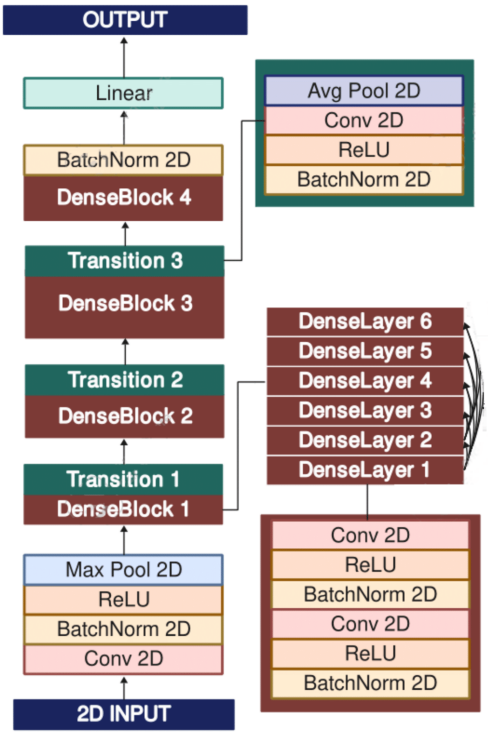

Fig. 5: DenseNet architecture. Only few links are showed among the ones existing between dense layers for the sake of clarity in the picture.

network implementing fusion, which is trying to improve the classification performances of the former by taking in input both one and two dimensional data at the same time.

DenseNet

The idea of improving classification on two-dimensional inputs was inspired by several state-of-the-art works in different research areas, where densely connected convolutional networks have been used with excellent results in image classification tasks. This architecture, known as DenseNet, was first introduced by [23] with the key idea of establishing dense connections between layers, allowing each layer to receive direct input from all previous layers. The architecture of DenseNet consists of several dense blocks, each containing a number of convolutional layers. Within each dense block, each layer is connected to every other layer in a feed-forward manner. In addition, transition layers are used to reduce the spatial dimensions of feature maps between dense blocks, while increasing the number of channels. We selected the `densenet121` PyTorch architecture, whose structure initially consists of a convolutional layer with filters of size 7x7, followed by a batch normalisation layer and a max pooling layer. Each sub-structure contains two batch normalisation layers and two two-dimensional convolutional layers of varying shapes. Six of these dense-layer structures are stored in the first dense macroblock, while 12, 24 and 16 are stacked within subsequent dense blocks, each of which is interspersed with a transition block consisting of batch

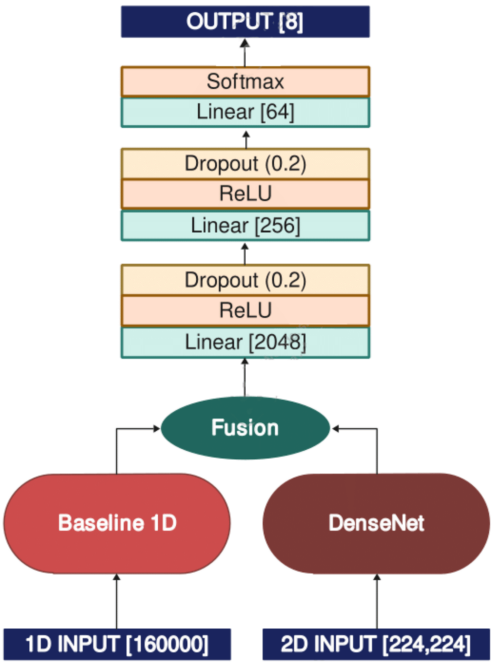

Fig. 6: Architecture implementing late fusion for 1D and 2D audio data types.

normalisation, convolutional and average pooling layers. The original `densenet121` architecture have been properly modified in order to input the correct number of channels and to obtain 8 output classes from the final linear classifier. A schematic picture for DenseNet is represented in Figure 5. For more details, see DenseNet-121 at [23].

FusionNet

Our idea at this stage was to consider the one- and two-dimensional data as merged inputs of a newly defined network. As explained in [19], [20], there are two possibilities: join the data before feeding them into the CNNs (early fusion) or, alternatively, concatenate the output at some point in the forward propagation steps of the network (late fusion). Although preferable according to [19], the former one is clearly unfeasible in our scenario, given the different shape of the data. The only possibility is to apply late fusion, concatenating along the time axis the outputs of the 1D baseline model and the DenseNet, which we have taken care of in order to obtain the same time length at this point. The classifier block has been removed for the two separate architectures, instead another classifier block is applied to find the outputs, see Figure 6 for a scheme of the global architecture.

VI. Results

In order to thoroughly evaluate the performance of the model, we used some standard classification metrics. Among these, accuracy and the F1-score stand out as important indicators. The F1-score, in particular, is a comprehensive

measure that takes into account both accuracy and recall, representing the harmonic mean of these two essential aspects of model evaluation:

$$\text{F1-score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

where precision and recall are defined as

$$\text{precision} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FP_i}$$

$$\text{recall} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FN_i}$$

Precision and recall, in their respective roles, exhibit a tendency to penalize certain scenarios. Precision penalizes classes that are frequently identified but incorrect (false positives), while recall penalizes those frequently misclassified when they are, in fact, correct (false negatives).

The accuracy is defined as:

$$\text{accuracy} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

Due to its inclusion of true negatives in the numerator, lacks informativeness when dealing with numerous classes and tends to be an overestimating indicator of model performance.

To derive the true positive (TP), true negative (TN), false negative (FN), and false positive (FP) values for each class, we aggregate across all audio tracks in the test set, where the total number of classes is represented by k. When we compute the metrics using these formulas, we are essentially performing an average across the classes, a method known as macro averaging [24].

The results, as depicted in Table 1, reveal that the DenseNet architecture outperforms all others in both considered metrics. Specifically, the F1 score surpasses the Baseline A with 2D input by nearly 2%, indicating a more balanced classification performance.

	F1-score	Accuracy	Precision	Recall
1D Baseline A	0.448	0.864	0.438	0.457
2D Baseline A	0.558	0.889	0.556	0.559
Baseline B	0.534	0.883	0.537	0.531
DenseNet	**0.574**	**0.892**	**0.578**	**0.571**
FusionNet	0.553	0.888	0.548	0.557

TABLE 1: Results of different metrics on test set for the architectures implemented.

Generally, the 2D input signal proves to be more informative, as demonstrated by the lower performance of the Baseline A model with 1D input compared to the same model with 2D input, which also unexpectedly outperforms even the Baseline B model. This trend persists when examining the results of the late-fusion-net, which yields lower accuracy and F1 scores compared to the standalone DenseNet. This suggests that either the late fusion approach might not be suitable or that further exploration of advanced architectures is warranted to fully leverage the combined input information.

The superior performance of DenseNet is further underscored by the results depicted in the confusion matrix, Figure 8. As indicated by the F1-score, this network, while not achieving exceptionally high values in any specific class, exhibits a noteworthy ability to consistently classify all genres more evenly compared to other architectures. On the other hand, the greater complexity of the Densenet architecture makes it more prone to overfitting due to the limited dataset in our experiments.

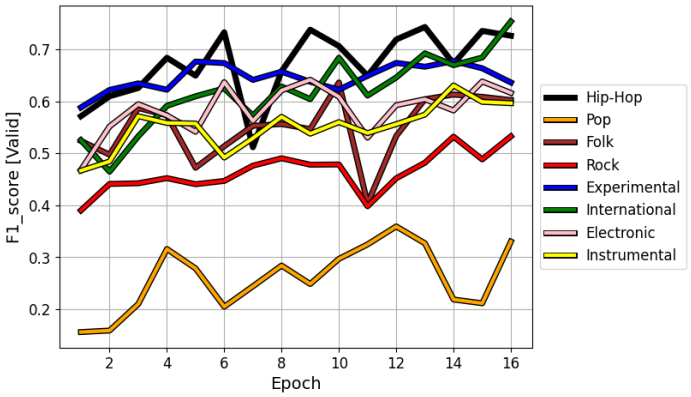

Fig. 7: F1-score achieved by the DenseNet architecture on each of the classified labels on the validation set as a function of the number of epochs.

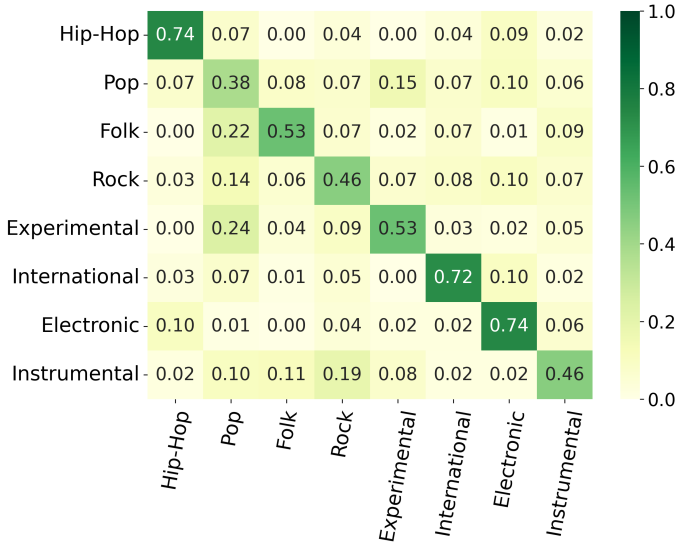

Fig. 8: Confusion matrix test set results with DenseNet.

In Figure 7, when examining the F1 scores per class for the validation set, we observe a subtle overall performance enhancement over the course of epochs. However, it's noteworthy that the "pop" class lags behind in performance, likely due to its less distinctive characterization. As can be seen

from the confusion matrix "pop" songs are often misclassified as "folk" or "experimental", leading to reduced recall and consequently lower F1 scores in this class.

VII. Concluding Remarks

Recognition of musical genres is not an easy challenge, partly due to the elusive boundaries that define musical genres and the inherent complexity that humans also face in tackling this task. In this study, we systematically use a number of strategies presented in the existing literature to analyse their performance distinctions on the Free Music Archive dataset.

Our approach starts with a pre-processing pipeline, extracting both 1D waveforms and 2D mel-spectrograms from the audio files. We then incorporate data augmentation techniques, adjusted to suit our classification task. Performance evaluation includes the use of state-of-the-art architectures, namely DenseNet, along with simpler convolutional neural networks that operate on both 1D and 2D inputs.

Our results indicate that DenseNet consistently exceeds other architectures, particularly in terms of F1 score, highlighting its ability to classify musical genres. Furthermore, mel-spectrogram representations prove more informative than their 1D counterparts, underlining the importance of spectral information in this domain.

Notably, we also explored the fusion of both types of input data, albeit with less encouraging results. This suggests the need for further exploration, potentially involving advanced network architectures for the 1D signal and refined fusion strategies that make better use of temporal information.

Although the results of DenseNet do not significantly surpass those of simpler architectures, the use of even deeper convolutional neural networks, that have been successful in various computer vision tasks, on mel-spectrograms emerges as promising for further exploration. This is especially true when larger data sets can be exploited, to effectively counteract overfitting problems.

In the end, we have understood how to scientifically process a dataset containing music files and how to handle its contents through the use of libraries, which allow us to extract various useful information for analysis purposes. We have increased our confidence in working with dataset classes and more elaborate models. In addition, a previously implemented network was also used with minor modifications to make it work on our data.

One difficulty may be related to the amount of time required to train the deep networks and the limited computational resources available, which did not allow us to test many different configurations of hyperparameters in order to understand the possible weaknesses of the different neural networks. In conclusion, we noted that in general the results obtained on the test set are slightly inferior when compared to those of the training and validation set.

References

[1] H. Bahuleyan, "Music genre classification using machine learning techniques," 2018.

[2] T. Kim, J. Lee, and J. Nam, "Sample-level cnn architectures for music auto-tagging using raw waveforms," 2018.

[3] W. Zhang, W. Lei, X. Xu, and X. Xing, "Improved Music Genre Classification with Convolutional Neural Networks," in *Proc. Interspeech 2016*, pp. 3304–3308, Sept. 2016.

[4] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[5] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6964–6968, 2014.

[6] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," *ArXiv*, vol. abs/1711.02520, 2017.

[7] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, 2018.

[8] T. Kim, J. Lee, and J. Nam, "Sample-level cnn architectures for music auto-tagging using raw waveforms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 366–370, IEEE Press, 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 12 2015.

[10] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

[11] S. Abdoli, P. Cardinal, and A. Lameiras Koerich, "End-to-end environmental sound classification using a 1d convolutional neural network," *Expert Systems with Applications*, vol. 136, pp. 252–263, 2019.

[12] S. Allamy and A. L. Koerich, "1d cnn architectures for music genre classification," *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 01–07, 2021.

[13] D. Ghosal and M. H. Kolekar, "Music Genre Recognition Using Deep Neural Networks and Transfer Learning," in *Proc. Interspeech 2018*, pp. 2087–2091, 2018.

[14] S. Chillara, S. A. Neginhal, and S. S. Haldia, "Music genre classification using machine learning algorithms: A comparison," 2019.

[15] L. T. Dao Thi, V. L. Trinh, T. Chu Ba, and H. C. Nguyen, "Music genre classification using densenet and data augmentation," *Computer Systems Science and Engineering*, vol. 47, no. 1, pp. 657–674, 2023.

[16] A. K. Hassen, H. JanÃŸen, D. Assenmacher, M. Preuss, and I. Vatolkin, "Classifying music genres using image classification neural networks," *Archives of Data Science, Series A (Online First)*, vol. 5, no. 1, pp. A20, 18 S. online, 2018.

[17] X. Zhangyong, G. Yutong, D. Shirong, and X. Qibei, "A study of music genre classification with bilinear convolutional neural network," *Academic Journal of Computing & Information Science*, vol. 5, no. 8, pp. 12–17, 2022.

[18] H. V. Joze, A. Shaban, M. L. Iuzzolino, and K. Koishida, "Mmtm: Multimodal transfer module for cnn fusion," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 13286–13296, IEEE Computer Society, jun 2020.

[19] K. Gadzicki, R. Khamsehashari, and C. Zetzsche, "Early vs late fusion in multimodal convolutional neural networks," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–6, 2020.

[20] S. Boulahia, A. Amamra, M. Madi, and S. Daikh, "Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition," *Machine Vision and Applications*, vol. 32, 11 2021.

[21] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[23] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[24] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.