International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I12P231
Volume 70 Issue 12, 328-337, December 2022

Original Article

Automatic Musical Transcription Applying Fine-Tuning by Composer in Neural Networks for the MusicNet Database

Leonardo Veronez Simões[1], Antônio Roberto Monteiro Simões[2], Karin Satie Komati[3], Jefferson Oliveira Andrade[4]

[1, 3, 4]*Graduate program in Applied Computing (PPComp), Instituto Federal do Espírito Santo (IFES) Campus Serra, Espírito Santo, Brazil*
[2]*Kansas University, Kansas, United States*

[1]*Corresponding Author : leo.vsimoes@gmail.com*

Received: 25 October 2022 Revised: 15 December 2022 Accepted: 21 December 2022 Published: 24 December 2022

Abstract *- The task of automatic music transcription is to build algorithms to convert acoustic music signals into some form of musical notation. Previous works train a single neural network architecture for a complete database of different composers. This work presupposes that each composer has their own characteristic style. The objective of this work is to carry out two stages of training. The first stage results in a generic trained model, and the second is a specialized and fine-tuned retraining step, generating a model for each compositor. To achieve this objective, experiments were carried out with two different Neural Network architectures —MLP and Convolutional— using the MusicNet database. Overall, the results with fine-tuning improved the average accuracy, except for composers with fewer musical works.*

Keywords *- Convolutional Neural Networks, Multilabel Classification, Multilayer Perceptron, Short Time Fourier Transform, Spectrogram, Transfer Learning.*

1. Introduction

Musical transcription is an old problem that has existed since its early developments, with the objective of documenting musical information in a legible writing form [1]. Recognition of music information is a problem addressed in different tasks, such as Optical Music Recognition (OMR) which performs optical scans on handwritten scores and can use Deep Learning to translate them into digital information, as did Nawade et al. [2]. Automated Music Transcription (AMT) is another task of Music Information Retrieval (MIR), with the aim of converting acoustic music signals into some form of musical notation [3]. The main idea can be represented as a sequence-to-sequence task, with the input of a sequence of audio frames and the output as a sequence of objects, which can be expressed as numbers or tokens, representing the notes [4].

The compilation of the study by Benetos et al. [3] presents several approaches to solving the AMT issue, including the methods of signal processing [5][6], Bayesian networks [7], probabilistic modelling [8], non-negative matrix factorization [9][10][11][12], and hidden Markov models [13][14]. Benetos et al. [3] also show that neural networks have become increasingly popular in solving this problem, as larger data sets are emerging and becoming accessible, as processing hardware is becoming more and more powerful. Even using neural networks, the authors indicate that there are still challenges associated with state-of-the-art methods based on neural networks, considering that experiments show that there is still detection of spurious notes in their results.

The article by Thickstun et al. [15] addresses the problem of labelling dataset size limitation and introduces a new large-scale music dataset, MusicNet. This same article deals with the multi-label classification method to predict musical notes in audio recordings, together with an evaluation protocol. It compares different machine learning architectures for this method. The best results were with MultiLayer Perceptrons (MLP) and Convolutional Neural Networks (CNN).

The work by Thickstun et al. [15] performs cross-validation using the holdout method, which separates part of the data for training and the rest of the basis for tests. This work assumes that each composer has a different style from one another [16]. Thus, the purpose of this work is to have two steps followed by training. The first is given by the model trained by Thickstun et al. [15] and includes a second

step of fine-tuning, generating a model for each composer. This approach of training steps followed by training has already been discussed in the work of Tamaazousti et al. [17], in which, in the first step, there is training with generic categories. Then there is a second training (retraining), making a fine-tuning of one or more models for more specific problems. Fine-tuning is defined as "the process of pre-training neural networks with a generative objective followed by an additional training phase with a discriminative objective on the same dataset" [34]. In the "Transfer learning" process, the second training step is performed with another general dataset of a smaller size than the one used in the first step.

This work investigates whether the fine-tuning of a separate model retrained by the composer (second stage) improves the result in relation to the general trained network (first stage) in the AMT task. As the work is based on the article by Thickstun et al. [15], the experiments use the same dataset (MusicNet), the same evaluation metrics, precision, recall and average precision, and the same five neural network models that obtained the best results (according to Table 4 and Appendix E of the base article). For each model, there are three types of experiments: the first one initializing the weights of the network and training for each composer; the second keeping the weights of the original work and running them with the test base of this work; and the third applying a second fine-tuning step by the composer.

As few musical datasets exist compared to image datasets, the contribution of this work is to improve the accuracy of musical transcription through fine-tuning due to the low amount of data.

The next sections are organized as follows: Section 2 describes related works; Section 3 presents a detail of the MusicNet database, following the proposal of the base article and the fine-tuning retraining; in Section 4, the results of the experiments are presented and discussed; and Section 5 concludes this study with a summary of the main points discussed here and suggestions for possible future work.

2. Related Works

Benetos et al. [3] summarize the AMT problem in six subtasks, which are the following: pitch and multi-pitch estimation that involve the extraction of fundamental frequency from an audio file; onset and offset detection, which is used in musical notes recognition; instrument recognition; rhythm and beat tracking for instrument separation; recognition of time and dynamics for each instrument; and score composition. All those methods can be applied to frame level, note level, and stream level approaches for source separation and problem resolution. For example, pitch estimation can be considered a frame-level approach, while onset and offset detection can be interpreted as a note level. Some of the main solutions to tackle these problems are listed in the introduction.

Also, as stated in the introduction, the work of Thickstun et al. [15] was used as the basis for this work. The article introduces the foundation of MusicNet and explains how automatic annotation was performed using the Dynamic Time Warping method, which performs event-based music segmentation on the score to find the best alignment between digital scores (MIDI) and audio recordings. The result of the best model was the architecture based on CNN, which achieved a value of 67.8% in the average accuracy metric, 60.5% in accuracy and 71.9% in the recall. A detailed description of the multi-label article classification basis and methods can be found in Section 3 of this text. The same authors published another article [19] exploring four CNN architectures the following year. In their work, the network that models the translation-invariant features obtained the best performance using the average precision metric of the first work as a comparison, reaching a value of 77.3%.

Cheuk et al. [20] also use the MusicNet database to evaluate different audio representations in automatic music transcription. The work analyzes the effect of four different frequency representations as input to the model: linear frequency spectrogram, logarithmic frequency spectrogram, constant-Q transform (CQT) and honey spectrogram. The model's performance is evaluated by varying the number of windows and frames in the audio representation. The best result found was the use of the logarithmic spectrogram, based on precision, accuracy and error metrics, with a performance of 66.6%, 48.1% and 56.0%, respectively.

Within the scope of audio processing with transfer learning D. Ghosal and M. H. Kolekar [21] proposes a music genre and style recognition approach for songs in the GTZAN dataset [22] and Ballroom dataset [23] containing a large amount of music of different styles. The experiment consists of a set of CNNs, Convolutional Long Short Term Memory (LSTM), and MLP with a transfer learning model running in a 10-fold cross-validation setup. The models for CNN and LSTM were tested with a wide set of spectral and rhythmic features such as Mel Spectrogram, Energy Chromagram, Constant Q Chromagram and others extracted from the raw music signals. The MLP model combined the features mentioned before with the transfer learning system trained for music labelling. The best result for the GTZAN dataset is obtained by MLP with transfer learning, achieving an 85.5% average accuracy score across all 10 folds. However, for the Ballroom dataset, the best result was obtained by the CNN LSTM with max pooling, with an average accuracy of 90%.

Another transfer learning proposal in the task of MIR is presented by L. Ou, X. Gu, and Y. Wang [35]; however, the focus is on ALT (Automatic Lyric Transcription). The

solution was developed to take advantage of the similarities of spoken and sung voices, achieved by performing transfer learning using a singing data system after pre-training and fine-tuning speech data and exploring the influence of different starting points. A variety of lyric transcription datasets were used for tests, including DALI [25], Hansen [26], Mauch [27], Jamendo [28] and DAMP Sing [29]. The results were measured with WER (Word Error Rate) and show that the work outperforms other state-of-the-art methods in all datasets, where the least error was measured in the DSING dataset at 12.34%.

Nawade et al. [2] investigate the performance of deep-learning methods for old handwritten music symbol recognition. The Alicia Fornes dataset was used, and the musical symbols were classified into seven different classes. The paper presents two approaches related to CNN, one purely based on the mobileNetV2 architecture and the other replacing the softmax layer with classifiers such as SVM, RandomForest, and KNN. The performance evaluation of the methods used the precision, recall, F1 score, and accuracy metrics, and Transfer Learning was also used in an attempt to improve the measures. Their method outperforms previous existing methods achieving an accuracy of 99.58% with the CNN combined with RandomForest without the Transfer Learning.

Tamaazousti et al. [17] bring three different approaches to address the issue of universal representations and knowledge transfer. The idea behind the article is that the data used for training can be structured in a certain way that allows different tasks to learn more or better from the same data and learning algorithms. The first proposal is a segregation of the initial problem to learn new features and then generically combine all the data. For example, categorize the network with images of Rottweilers, Pitbulls, and others, and then generalize to the category of dogs.

The second proposal is called Focused Retraining, which, unlike the first, retrains the model based on the tuning principle; that is, it starts as a generalist and then retrains in a specialized way. It is interesting to emphasize the need to form an independent network by speciality of the problem and that the data be the same in training and recycling. This second proposal is the one used in the experiments of this text. The third proposal is the performance evaluation for the problem of universality, combining its previous proposals, problem segregation, generic combination, and focused retraining. In the article, several experiments are carried out with ten different data sets comparing them with other methods of universal representations considered state of the art. In most of the experiments, the third proposal of the work stands out over the other methods based on the average accuracy and precision metrics, reaching the best value of 77.5% for the CA101 database.

3. Materials and Methods

This section is divided into three parts: a description of the MusicNet database, a description of the system proposed by Thickstun et al. [15], and the adjustment step, referring to retraining by the composer.

3.1. Musicnet

The MusicNet project was inspired by the ImageNet database and pursued the same idea of providing a rich set of labeled data to explore machine learning techniques in music, being a common reference to compare results. MusicNet [30] is a database containing 330 freely licensed classical music recordings and is featured in the article by Thickstun et al. [15]. The compressed base has a size of 11Gb, and the expansion reaches 30Gb. All compositions have a set of metadata containing the composer, musical movement (andantino, maestoso, andante, among others), set of instruments (solo piano, string quartet, among others), responsible for transcription, and music tempo. The recordings last an average of six minutes, the shortest being 55 seconds and the longest almost 18 minutes.

The dataset is labeled by musical composition. There are 513 starting classes using the naive definition of distinct instrument/note combinations. On the “MusicNet Inspector” page [31], it is possible to choose the musical composition, listen to it, and visualize the explanations about the musical notes in a video. The video contains a piano-roll representation of color-coded note annotations according to the instruments and the audio waveform to show the amplitude of the audio at each instant. The labels were obtained through automatic alignment with MIDI files rather than being manually tagged, which introduced a degree of labelling error that the authors estimate to be around 4%.

Because it is a continuous music event, many labels can overlap in a time series, creating multiple polyphonic labels. The data structure of the tags that support this information is an interval tree that builds a tree based on the time interval and expands to the leaves that exhibit the characteristics of that interval. The information contained in the leaves is as follows: start and end interval of the recording time (of the sheet), instrument, musical note, compass, rhythm (or beat) and note value.

In short, the dataset has more than 2,048 minutes of recording, almost 1.3 million stamps, exploring ten different classical music composers and eleven musical instruments. Table 1 shows the information by the composer: number of compositions, time in minutes and number of labels (leaf nodes). The database is unbalanced for composers. Practically half of the music dataset is related to Beethoven, but other composers such as Bach, Brahms, Schubert and Mozart have representative participation within the dataset. Haydn has the least number of compositions and time.

Table 1. Summary of information on musical composition by the composer in the Music-Net database, sorted by the number of compositions in descending order.

Composer	Number of musical compositions	Duration in minutes	Labels
Beethoven	157	1.057	736.072
Bach	67	184	62.782
Schubert	30	253	146.648
Brahms	24	192	133.109
Mozart	24	156	99.641
Cambini	9	42	24.820
Dvorak	8	55	46.261
Faure	4	32	22.349
Ravel	4	27	21.243
Haydn	3	14	6.404

Another imbalance is related to musical instruments. There are many solo piano musical compositions, while flute and oboe are poorly represented. Only 83 of the 88 keys are used on the piano and appear as labels.

3.2. The proposal from Thickstun et al.

The simplified architecture of the proposal by Thickstun et al. [15] is shown in Figure 1. The input audio goes through the window technique, part (a) of the figure; then, in step (b), an audio transformation is applied to generate a spectrogram that transforms the audio signal into a frequency graph. Next, there is the classifier, step (c), which is an MLP or CNN, which generates the result ŷ. This result converts into a vector of 0s (zeros) and 1s (ones) in step (d).

Thickstun et al. [15] analyze the compensation that the window size must be large enough to return relevant information but not so much as to lose the temporal resolution of the signal. The authors provided the values of the window size parameter of 2,048 and 16,384, a value that varies according to the model being illustrated by the cut of a red rectangle, part (a) of the figure, which is converted into a spectrogram, part (b) of the figure. Spectrograms represent the magnitude of the STFT (Short Time Fourier Transform).

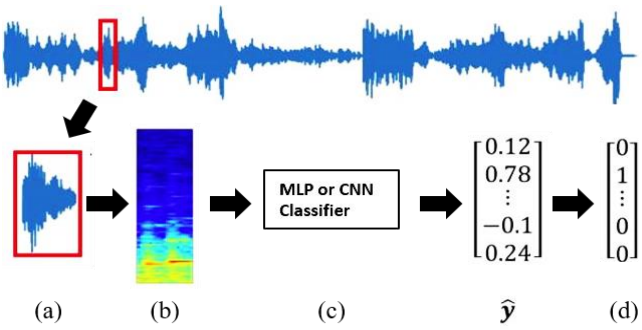

Fig. 1 The architecture of the work of Thickstun et al. (2017)

The spectrogram is an input for the classification methods, part (c) of the figure. In Thickstun et al. [15] work, four models based on MLP architecture and one on CNN architecture were developed. The list below shows the name and a summary of the models:

- Model 1: MLP with 2,500 nodes and a window size of 2,048;
- Model 2: MLP with 500 nodes and a window size of 2048;
- Model 3: MLP with 500 nodes and a window size of 16,384;
- Model 4: MLP that makes use of an average pooling with two strides in the input to try to modify the relevant characteristics. In addition, it uses a window of size 2,048;
- Model 5: Simple CNN with a convolution layer, a pooling layer (making use of the average pooling operation), and at the end, a fully connected layer for the result. The parameters are a convolutional window size of 16,384 and a sliding window of size 8 x 8. The first layer extracts features from the input using filters of 2048 samples from the network weights. The pooling layer reduces the number of features to be evaluated for the classification done on the fully connected layer at the end.

A multi-label classification, in the context of AMT, is the task of associating multiple musical notes to a piece of music [32]. Consider identifying notes in an audio segment $x \in \chi$ as a multi-label classification problem, modelled as follows. Assign each audio segment a binary label vector $y \in x = \{0, 1\}128$, part (d) of Figure 1. The 128 dimensions correspond to note frequency codes, and $yn = 1$ if note n is present at the midpoint of x. Multivariate linear regression is trained to predict ŷ given f(x), which is optimized for the squared error. The vector ŷ can be interpreted as a multiple-label estimate of the scores at x by choosing a threshold c and predicting the label n if and only if $\hat{y}n > c$. Then, the search is performed for the value of c that maximizes the f1-score on a sampled subset of MusicNet.

Models are evaluated on three metrics: accuracy, recall, and average accuracy. Accuracy counts the correct model predictions (through all data points) divided by all model predictions. The recall is the count of correct model predictions divided by the total number of labels (ground or baseline truth) in the test set. The precision recovery curve is constructed when parametrizing the accuracy and recall with the threshold c, and varying it. Average accuracy is the area under the accuracy recovery curve, i.e., the Precision-Recall Area Under Curve Score.

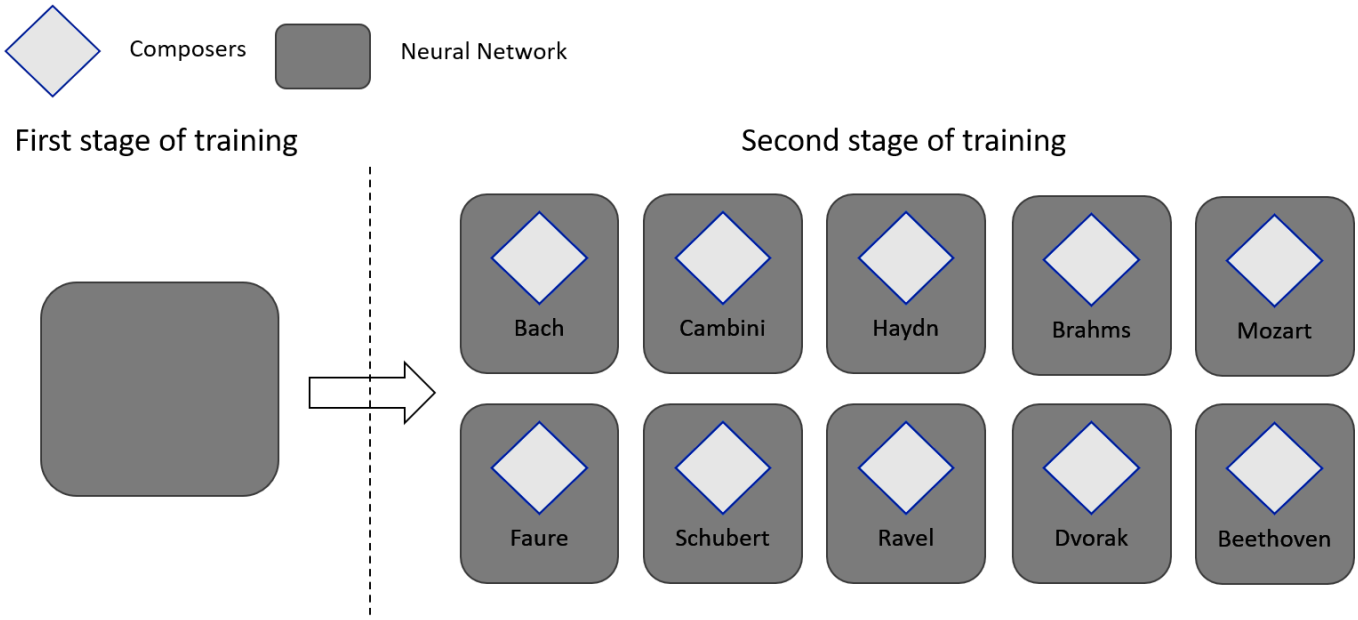

Fig. 2 Separation of composer and composition sets for the two stages of job training

3.3. Training

Figure 2 presents a conceptual drawing of the composers' separation. In the first stage of training, a network trained by all composers is generated. In the second stage, new models are generated, resulting in a separate model for each composer.

The test set and the parameters of the models were kept as the original ones, except for two of the parameters. One is the restore weights parameter, which indicates whether or not the original weights will be loaded and has changed depending on the experiment.

The other parameter was the number of iterations, or epochs, for training the networks. Originally, the models ran with 500,000 iterations to learn the network weights, but for Model 4 and Model 5, this number of epochs exceeded the computational power available for the experiments.

Thus, for Models 1, Model 2 and Model 3, 50,000 iterations were kept, but for Model 4, 10,000 and for Model 5, 1,000. The Loss Function parameter used for all models remained the original, the L2 Squared Error Loss, and the optimizer was the Stochastic Gradient Descent (SGD).

The same proportion in the test dataset of all compositions used in Thickstun et al. [15] was kept for the test set for the composer's Bach, Beethoven, Brahms, Mozart, and Schubert, who have more than 100 minutes of recordings. For Cambini and Dvorak, who have more than 40 minutes and less than 100 minutes of recording, a set of two records was chosen for testing, and for Faure, Haydn, and Ravel, only one record was chosen for the test set.

4. Experiments and Results

The experiments were carried out in the cloud, using the Google Cloud Services platform of type E2 Standard, with 16 vCPUs, 64Gb of RAM, without GPU, and with a standard installation of Tensorflow Enterprise 1.15. The five models developed by Thickstun et al. [15] and available on Github [33] were used to carry out the experiments of this work. The results of the experiments were divided into three parts:

- Resetting weights: The five models are loaded without previous training of the weights and are trained separately for each of the ten composers, thus generating 50 different models with a single training phase;

- Original weights: The five models are tested, with the weights resulting from the general training phase carried out by the work of Thickstun et al. [15]. There is no training phase in this experiment. This experiment is to ensure that the results presented were tested with the same holdout split test database performed in this investigation;

- Proposal of this investigation: To use the weights resulting from the training carried out by Thickstun et al. [15] on the five models. Each model is trained in a second step for the data of each of the ten composers, thus generating 50 models.

The results are shown in Table 2, Table 3 and Table 4. The columns represent the five models, from 1 to 5, and the lines show the grouping by composer, by type of experiment and by metrics. Bold marks are the highest column values (per model) of a given metric per composer.

Table 2. Results for five composers: Beethoven, Bach, Shubert, Brahms and Mozart

Composer	Experiment	Metric	Model 1	Model 2	Model 3	Model 4	Model 5
Beethoven	Resetting weights	Average precision	0.66	0.64	0.74	0.51	0.61
		Precision	0.68	0.67	**0.71**	0.22	0.57
		Recall	0.62	0.59	**0.70**	0.86	0.62
	Original weights	Average precision	**0.73**	**0.68**	**0.75**	**0.73**	**0.82**
		Precision	0.71	**0.68**	**0.71**	**0.20**	0.76
		Recall	**0.69**	**0.63**	**0.70**	**0.96**	**0.77**
	Proposal by this work	Average precision	**0.73**	**0.68**	0.74	**0.73**	**0.82**
		Precision	**0.73**	**0.68**	**0.71**	**0.20**	**0.78**
		Recall	0.67	0.62	**0.70**	**0.96**	0.76
Bach	Resetting weights	Average precision	0.65	0.62	**0.72**	0.06	0.35
		Precision	0.63	**0.62**	**0.65**	0.00	0.34
		Recall	0.68	0.63	**0.72**	0.00	0.46
	Original weights	Average precision	0.70	0.64	0.70	0.70	**0.81**
		Precision	0.62	0.61	0.61	0.11	**0.70**
		Recall	0.73	0.65	0.72	**0.93**	**0.80**
	Proposal by this work	Average precision	**0.71**	**0.66**	**0.72**	**0.71**	**0.81**
		Precision	**0.67**	**0.62**	**0.65**	**0.13**	**0.70**
		Recall	**0.73**	**0.69**	**0.72**	0.92	**0.80**
Schubert	Resetting weights	Average precision	0.59	0.59	**0.62**	0.10	0.18
		Precision	0.52	0.52	**0.54**	0.00	0.00
		Recall	0.65	0.64	**0.73**	0.00	0.00
	Original weights	Average precision	0.63	0.61	0.60	**0.64**	0.67
		Precision	0.55	0.53	0.53	0.18	0.57
		Recall	**0.72**	**0.69**	**0.73**	**0.95**	**0.77**
	Proposal by this work	Average precision	**0.64**	**0.62**	**0.62**	**0.64**	**0.68**
		Precision	**0.58**	**0.54**	**0.54**	**0.25**	**0.58**
		Recall	0.69	**0.69**	**0.73**	0.92	0.76
Brahms	Resetting weights	Average precision	0.52	0.51	**0.61**	0.14	0.16
		Precision	0.48	0.50	**0.57**	0.00	0.00
		Recall	0.64	0.59	0.67	0.00	0.00
	Original weights	Average precision	0.60	**0.57**	**0.61**	**0.61**	0.68
		Precision	**0.58**	0.52	0.53	**0.22**	**0.64**
		Recall	0.66	**0.68**	**0.72**	0.93	0.72
	Proposal by this work	Average precision	**0.61**	**0.57**	**0.61**	**0.61**	**0.69**
		Precision	0.55	**0.56**	**0.57**	0.20	0.63
		Recall	**0.69**	0.61	0.67	**0.94**	**0.73**
Mozart	Resetting weights	Average precision	0.52	0.51	**0.61**	0.12	0.14
		Precision	0.51	0.49	0.55	0.00	0.05
		Recall	0.61	0.62	**0.72**	0.00	0.00
	Original weights	Average precision	0.60	0.55	0.59	0.60	**0.70**
		Precision	0.55	**0.56**	**0.57**	0.21	**0.66**
		Recall	**0.71**	0.62	0.66	**0.91**	0.74
	Proposal by this work	Average precision	**0.61**	**0.57**	**0.61**	**0.61**	**0.70**
		Precision	**0.61**	0.54	0.55	**0.32**	0.63
		Recall	0.66	**0.69**	**0.72**	0.87	**0.78**

Numbers in blue are the highest values of a composer-determined metric, regardless of model. The gray shaded lines are the medium precision lines that showed the highest results for more than three models. The order of composers shown in Table 1 was maintained in decreasing order of the number of compositions. The results can be observed from the three dimensions of the dataset (Composer, Experiment Type and Models) for three different metrics (Accuracy, Average Accuracy and Recall). They will be discussed in the next section.

Table 3. Results for five composers: Cambini, Dvorak, Faure, Ravel, and Haydn

Composer	Experiment	Metric	Model 1	Model 2	Model 3	Model 4	Model 5
Cambini	Resetting weights	Average precision	0.61	0.61	**0.74**	0.20	0.20
		Precision	0.59	0.58	**0.65**	0.00	**0.98**
		Recall	0.59	0.60	0.75	0.00	0.01
	Original weights	Average precision	0.70	0.66	0.70	0.71	**0.81**
		Precision	0.62	0.59	0.60	**0.47**	0.74
		Recall	0.74	**0.72**	**0.76**	0.88	0.73
	Proposal by this work	Average precision	**0.74**	**0.71**	**0.74**	**0.74**	**0.81**
		Precision	**0.67**	**0.68**	**0.65**	0.43	0.67
		Recall	**0.75**	0.67	0.75	**0.91**	**0.85**
Dvorak	Resetting weights	Average precision	0.30	0.30	**0.45**	0.10	0.10
		Precision	0.33	0.32	**0.45**	0.00	0.00
		Recall	0.46	0.47	0.53	0.00	0.00
	Original weights	Average precision	0.41	0.35	0.39	0.41	**0.57**
		Precision	0.40	0.36	0.39	0.28	**0.51**
		Recall	**0.57**	0.52	**0.55**	**0.76**	0.61
	Proposal by this work	Average precision	**0.44**	**0.40**	**0.45**	**0.44**	**0.57**
		Precision	**0.45**	**0.40**	**0.45**	**0.38**	0.50
		Recall	0.55	**0.54**	0.53	0.66	**0.64**
Faure	Resetting weights	Average precision	0.31	0,31	0,55	0,13	0,12
		Precision	0.37	0.37	0.47	0.00	0.00
		Recall	0.34	0.34	0.70	0.00	0.00
	Original weights	Average precision	**0.57**	**0.50**	**0.57**	**0.57**	**0.69**
		Precision	**0.58**	**0.50**	**0.49**	0.40	**0.59**
		Recall	0.55	0.55	**0.71**	**0.76**	0.73
	Proposal by this work	Average precision	0.55	**0.50**	0.55	0,56	**0.69**
		Precision	0.50	0.48	0.47	**0.43**	0.57
		Recall	**0.63**	**0.58**	0.70	0.72	**0.75**
Ravel	Resetting weights	Average precision	0.10	0.09	**0.17**	0.04	0.04
		Precision	0.13	0.11	**0.19**	0.00	0.00
		Recall	0.26	0.30	0.41	0.00	0.00
	Original weights	Average precision	**0.23**	**0.19**	**0.17**	**0.23**	**0.31**
		Precision	**0.24**	**0.23**	**0.19**	0.19	**0.29**
		Recall	**0.44**	0.36	**0.46**	**0.59**	0.48
	Proposal by this work	Average precision	0.22	**0.19**	**0.17**	**0.23**	**0.31**
		Precision	**0.24**	0.19	**0.19**	**0.24**	0.28
		Recall	0.42	**0.47**	0.41	0.44	**0.52**
Haydn	Resetting weights	Average precision	0.29	0.28	**0.62**	0.13	0.13
		Precision	**0.84**	0.34	**0.58**	0.00	0.00
		Recall	0.01	0.31	0.62	0.00	0.00
	Original weights	Average precision	0.57	0.52	0.60	0.57	**0.73**
		Precision	0.53	0.51	0.54	**0.57**	**0.68**
		Recall	**0.66**	**0.60**	**0.64**	0.60	0.66
	Proposal by this work	Average precision	**0.59**	**0.54**	**0.62**	**0.59**	**0.73**
		Precision	0.60	**0.60**	**0.58**	0.52	0.62
		Recall	0.61	0.52	0.62	**0.68**	**0.73**

5. Discussion

Looking at the shaded rows in Table 2, all average accuracy rows of the "Proposal by this work" experiment type are marked. These lines represent the best average precision values by composer and type of experiment. There was a tie only for the composer Beethoven with the kind of experiment with the “Original Weights.” This result indicates that the second training stage improved average accuracy performance. Regarding Table 3, the marking of lines with the best average precision was better for Haydn in the

"Proposal of this investigation," but in the case of Faure and Ravel, the best marking was with the experiments in “Original weights.” One possible explanation for these results is that for composers with few compositions, the result of the fine-tuning may not improve the outcomes of the generic training step. It is also worth noting that no training lines resetting the original weights were marked in all Tables, with gray shading, indicating that only the separate training step has worse results than the other two forms of training.

Considering the marking off numbers in blue, the best results are concentrated in Model 4 and Model 5 (in both tables). Average accuracy metrics are highest on Model 5 (CNN) for the purposes of this investigation, often tied with the original weights. The precision metric of Model 5 for the purposes of this investigation is better for Beethoven, Bach, and Schubert, but for the rest of the composers, Model 5 with the original weights is better. Remembering the Model 4 with the original weights is better for almost all composers except Ravel.

In the experiment with the original weights, the same results from Thickstun et al. [15] were reproduced, the mean of the mean precision: Model 1 (0.562); model 2 (0.521); model 3 (0.600); Model 4 (0.564) and Model 5 (0.678). In the experiment proposed in this investigation, the average of the average precision: Model 1 (0.584), model 2 (0.544), model 3 (0.584), Model 4 (0.564) and Model 5 (0.678). There was an improvement in Model 1, Model 2, and Model 4. Probably the window size of 16,384 in Models 3 and 5 may have influenced the fine-tuning characteristics.

In Model 3, the results of the metrics for training separately by the composer are similar to the results of this proposal. It is as if the training of the second stage is superimposing all the weights of the training of the first stage. Model 4 (MLP with average pooling and window of 2048) with the type of experiment "Resetting the weights" only presents results greater than zero for Beethoven, who is the composer with the most significant number of musical compositions.

The style of the music may have influenced the results, as the rhythm of the music changes its patterns and consequently influences the results of the models. Cambini, Haydn, Beethoven and Mozart belong to the classical period, which may justify the good performance of Cambini and Haydn in the models, despite having fewer registers than the others. Analyzing the result by the composer, Ravel presents the worst result among the composers, regardless of the model or training method. Haydn has the fewest songs in the database but has better results than Ravel. One possible explanation is that the musical movements of Ravel's compositions vary more than Haydn's. Ravel has only four musical compositions and each one has different movements: Allegro moderato - très doux; Assez vif - très rythmé; Très lent; and Vif et agité. Haydn has only three songs, all in faster movements, Allegro moderato, Adagio – Cantabile, and Menuetto Allegretto. Faure also has four musical compositions in the database, but three of the four compositions have movements from Allegro, and this may justify his better performance than Ravel's results.

6. Conclusion

This work's objective was to investigate the fine adjustment given by a second training step, retraining models separated by composer, and comparing the results of the trained network in general in the AMT task. Based on experiments in the MusicNet database, a minimum amount of at least 42 minutes of music is required for this second stage to be effective. And the fine-tuning results made a greater difference in the results of the MLP-type neural networks than in the CNN-type ones.

Adding to the challenge of multi-label classification, the dataset is unbalanced to composers, musical instruments, and even the notes of each instrument. Also, the difference in period and musical style, as well as the type of musical movement of the songs in the database.

Several future works are anticipated, such as the use of the models already proposed in the article by Thickstun et al. [19], which models translation-invariant characteristics, in addition to other forms of transformations in the input audio signal, such as Cheuk et al. [20], but with different representations. The universality proposal of Tamaazousti et al. [17], the third approach, being a combination of generic training and retraining, can be applied to MusicNet to confirm the results that this representation is better than the one implemented in this investigation. It would also be interesting if the investigation were transdisciplinary, adding the opinion of a specialist in the field of classical music who could evaluate the database.

Acknowledgments

This work was supported by the CAPES/FAPES (process: 2021-2S6CD, nº FAPES 132/2021) in PDPG (Programa de Desenvolvimento da Pós-Graduação - Parcerias Estratégicas nos Estados). Authors thanks the support from Instituto Federal do Espírito Santo (IFES). Prof. The CNPq supports Komati under grant \#308432/2020-7 (Bolsa de Produtividade DT-2) and by the FAPES under grant #293/2021. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research.

References

[1] Dorothea Blostein, and Henry S. Baird, "A Critical Survey of Music Image Analysis," *Structured Document Image Analysis, Springer*, pp. 405–434, 1992. *Crossref,* https://doi.org/10.1007/978-3-642-77281-8_19

[2] Savitri Apparo Nawade, Mallikarjun Hangarge, and Shivanand S Rumma, "Deep Learning-Based Approach for Old Handwritten Music Symbol Recognition," *International Journal of Engineering Trends and Technology*, vol. 69, no. 7, pp. 208-214, 2021. *Crossref,* https://doi.org/10.14445/22315381/IJETT-V69I7P228

[3] Emmanouil Benetos et al., "Automatic Music Transcription: An Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019. *Crossref,* https://doi.org/10.1109/MSP.2018.2869928

[4] Josh Gardner et al., "Mt3: Multi-Task Multitrack Music Transcription," *arXiv preprint arXiv:2111.03017*, 2021. *Crossref,* https://doi.org/10.48550/arXiv.2111.03017

[5] Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010. *Crossref,* https://doi.org/10.1109/TASL.2009.2038819

[6] Li Su, and Yi-Hsuan Yang, "Combining Spectral and Temporal Representations for Multipitch Estimation of Polyphonic Music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 10, pp. 1600–1612, 2015. *Crossref,* https://doi.org/10.1109/TASLP.2015.2442411

[7] A. T. Cemgil, H. J. Kappen, and D. Barber, "A Generative Model for Music Transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 679–694, 2006. *Crossref,* https://doi.org/10.1109/TSA.2005.852985

[8] Paul H. Peeling, A. Taylan Cemgil, and Simon J. Godsill, "Generative Spectrogram Factorization Models for Polyphonic Piano Transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 519–527, 2010. *Crossref,* https://doi.org/10.1109/TASL.2009.2029769

[9] Andrea Cogliati, and Zhiyao Duan, "Piano Music Transcription Modeling Note Temporal Evolution," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE,* pp. 429–433, 2015. *Crossref,* https://doi.org/10.1109/ICASSP.2015.7178005

[10] P. Smaragdis, and J. C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription," *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No. 03TH8684), IEEE,* pp. 177–180, 2003. *Crossref,* https://doi.org/10.1109/ASPAA.2003.1285860

[11] Emmanuel Vincent, Nancy Bertin, and Roland Badeau, "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation*," IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010. *Crossref,* https://doi.org/10.1109/TASL.2009.2034186

[12] Emmanouil Benetos, and Simon Dixon, "Multiple-Instrument Polyphonic Music Transcription Using a Temporally Constrained Shift-Invariant Model," *The Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1727–1741, 2013. *Crossref,* http://dx.doi.org/10.1121/1.4790351

[13] Maksim Khadkevich, and Maurizio Omologo, "Use of Hidden Markov Models and Factored Language Models for Automatic Chord Recognition," *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009*, pp. 561–566, Citeseer, 2009.

[14] Sebastian Ewert, and Mark Sandler, "Piano Transcription in the Studio Using an Extensible Alternating Directions Framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1983–1997, 2016. *Crossref,* https://doi.org/10.1109/TASLP.2016.2593801

[15] John Thickstun, Zaid Harchaoui, and Sham M. Kakade, "Learning Features of Music from Scratch," *International Conference on Learning Representations (ICLR)*, 2017.

[16] S. Sadie, *The New Grove Composer Biography Series (Bach Family, Handel, Haydn, Mozart, Beethoven, Schubert, Masters of Italian Opera, Second Viennese School)*, Notes, vol. 32, no. 2, pp. 259–268, 1975.

[17] Youssef Tamaazousti et al., "Learning More Universal Representations for Transfer-Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2212–2224, 2020. *Crossref,* https://doi.org/10.1109/TPAMI.2019.2913857

[18] K. Pranathi, G. Ravi Kumar, and J. S. S. Aditya, "Fault Analysis on Multi-Terminal System Using Wavelet Transform and Wavelet Morphing Technique," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 8, no. 6, pp. 28-37, 2021. *Crossref,* https://doi.org/10.14445/23488379/IJEEE-V8I6P105

[19] John Thickstun et al., "Invariances and Data Augmentation for Supervised Music Transcription," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE,* pp. 2241–2245, 2018. *Crossref,* https://doi.org/10.1109/ICASSP.2018.8461686

[20] Kin Wai Cheuk, Kat Agres, and Dorien Herremans, "The Impact of Audio Input Representations on Neural Network Based Music Transcription," in *2020 International Joint Conference on Neural Networks (IJCNN), IEEE,* pp. 1–6, 2020. *Crossref,* https://doi.org/10.1109/IJCNN48605.2020.9207605

[21] Deepanway Ghosal, and Maheshkumar H. Kolekar, “Musical Genre and Style Recognition Using Deep Neural Networks and Transfer Learning,” in *Proceedings, APSIPA Annual Summit and Conference*, vol. 2018, pp. 2087-2091, 2018. *Crossref,* https://doi.org/10.21437/Interspeech.2018-2045

[22] G. Tzanetakis, and P. Cook, “Musical Genre Classification of Audio Signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002. *Crossref,* https://doi.org/10.1109/TSA.2002.800560

[23] Fabien Gouyon et al., “Evaluating Rhythmic Descriptors for Musical Genre Classification,” *Proceedings of the AES 25th International Conference*, vol. 196, p. 204, 2004.

[24] Murali Matcha et al., "Design and Performance Analysis of Multilayer Neural Network-based Battery Energy Storage System for Enhancing Demand Side Management," *SSRG International Journal of Electrical and Electronics Engineering,* vol. 9, no. 10, pp. 7-13, 2022. *Crossref,* https://doi.org/10.14445/23488379/IJEEE-V9I10P102

[25] Gabriel Meseguer-Brocal, Alice Cohen-Hadria, and Geoffroy Peeters, “Dali: A Large Dataset of Synchronized Audio, Lyrics and Notes, Automatically Created Using Teacher-Student Machine Learning Paradigm,” *arXiv Preprint arXiv:1906.10606*, pp. 431-437, 2019. *Crossref,* https://doi.org/10.5281/zenodo.1492443

[26] Jens Kofod Hansen, and I. Fraunhofer, “Recognition of Phonemes in A-Cappella Recordings Using Temporal Patterns and Mel Frequency Cepstral Coefficients,” in *9th Sound and Music Computing Conference (SMC)*, 2012. *Crossref,* https://doi.org/10.5281/zenodo.850135

[27] Matthias Mauch, Hiromasa Fujihara, and Masataka Goto, “Integrating Additional Chord Information into Hmm-Based Lyrics-to-Audio Alignment,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 200–210, 2011. *Crossref,* https://doi.org/10.1109/TASL.2011.2159595

[28] Daniel Stoller, Simon Durand, and Sebastian Ewert, “End-to-End Lyrics Alignment for Polyphonic Music Using an Audio-to-Character Recognition Model,” *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE,* pp. 181–185, 2019. *Crossref,* https://doi.org/10.1109/ICASSP.2019.8683470

[29] Gerardo Roa Dabike, and Jon Barker, “Automatic Lyric Transcription from Karaoke Vocal Tracks: Resources and a Baseline System,” *Interspeech*, pp. 579–583, 2019. *Crossref,* https://doi.org/10.21437/Interspeech.2019-2378

[30] The Musicnet Dataset Website, 2016. [Online]. Available: https://zenodo.org/record/5120004#.YXDPwKBlBpQ

[31] The Musicnet Inspector Website. [Online]. Available: https://musicnet-inspector.github.io/

[32] Hendrik Purwins et al., “Deep Learning for Audio Signal Processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019. *Crossref,* https://doi.org/10.1109/JSTSP.2019.2908700

[33] The Jthickstun Github, 2017. [Online]. Available: https://github.com/jthickstun/thickstun2017learning

[34] Christoph Käding et al., “Fine-Tuning Deep Neural Networks in Continuous Learning Scenarios,” in *Asian Conference on Computer Vision*, Springer, pp. 588– 605, 2016. *Crossref,* https://doi.org/10.1007/978-3-319-54526-4_43

[35] Longshen Ou, Xiangming Gu, and Ye Wang, “Transfer Learning of Wav2vec 2.0 for Automatic Lyric Transcription,” *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022.