MUSIC ROLE CLASSIFICATION USING AUDIOSET

Maria Korosteleva
KAIST Graduate School of Culture Technology
Motion Computing Laboratory
mariako@kaist.ac.kr

Jonas Gerstner
KAIST
Exchange Student
jonas.gerstner@kaist.ac.kr

ABSTRACT

In our term project we performed supervised training of various machine learning models on the music role subset of AudioSet, a large-scale dataset of manually annotated YouTube videos. First, we extracted various features from the audio files to identify the most relevant ones and combine those in a feature vector for conventional classification. Second, we trained a shallow convolutional neural network (CNN) from scratch and ImageNet [7] pre-trained CNNs VGG16 [19] and ResNet [10] on images of mel-spectrograms. Analyzing the dataset and our classification results, we found three vague classes to degrade our test accuracies. Discarding those classes and training on the remaining classes, we were able to achieve an accuracy of 64% using an ensemble of a feature engineering based approach and a transfer learning based approach, compared to our baseline accuracy of 41% obtained using transfer learning on the full dataset.

1. INTRODUCTION

Music role is the description of music according to its functional role [9]. The AudioSet ontology for music roles consists of ten classes such as music for wedding or Christmas celebrations. See Section 3 for details. The creation of a powerful music role classification model would be useful for various tasks, e.g. playlist creation, recommender systems for media creators and other professionals or automated tagging. Different tools are available to extract machine-readable information from audio files. Audio analysis libraries like LibROSA [1] offer the plotting of spectrograms as well as the extraction of features that aim at the loudness, harmonics, timbre and rhythm of an audio piece. Building a good music information retrieval system requires scientists to choose those features, that are the most beneficial to their task. This can be done manually using domain knowledge or automatically using feature selection algorithms. We study the importance of a variety of features in Section 5.4.1.

While CNNs are particularly successful in image classification tasks, their application to music information retrieval tasks has been proven to be meaningful too (e.g. [14], [17] and [23]). Operating on spectrogram images we trained different networks as described in Section 4.1.

The rest of this report is structured as follows: an overview over relevant literature is given in Section 2 followed by an introduction to the AudioSet dataset in Section 3. Section 4 refers to the detailed description of methods used in the study. Results of our experiments are described in Section 5, followed by a discussion in Section 6. The code of our project is available on GitHub [2].

2. RELATED WORK

Music role classification task is a rather new problem. To the best of our knowledge there is no published research on the subject at the time we worked on the project. This emptiness might be caused by the fact that there were no datasets that included the ontology of music role classes before the AudioSet [9] was published. Thus we use music role subset of AudioSet for our investigation of this classification task as the only possible choice. For the detailed description of the dataset refer to Section 3.

However, there were several achievements on large-scale classification of the dataset in question that could be helpful for our task. Another point of reference is the research on the music genre classification task. This problem has a similar setting as music role classification: it is focused on music classification and the classes in question are partially subjective. The task of the genre classification has been investigated for several decades with the most significant work published in 2002 by Tzanetakis and Cook [21]. Thus we can use these experience and apply it to the new task of music role classification.

Large-scale classification of AudioSet. Hershey et al. [11] compared different architectures of Deep Neural Networks (DNN) in application to large-scale audio classification task. The authors examined fully connected Deep Neural Networks (DNNs), AlexNet [13], VGG [19], Inception [20], and ResNet [10] and their behavior for different training set sizes and different numbers of labels to use.

Kong et al. [12] tried to challenge the result of the previous work by applying the probabilistic approach of Multiple Instance Learning (MIL) [8, 16] to the task of AudioSet dataset classification. The authors implemented this approach using fully-connected DNN and reported outperforming the results of Hershey et al. [11].

Music genre classification using subset of AudioSet. Bahuleyan [1] addressed the task of genre classification for

[1] http://librosa.github.io/

[2] https://github.com/jonasgerne/audioset-role

Publication	Approach	Reported accuracy
Sigtia and Dixon [18]	Fully-Connected DNN with ReLU units, Dropout and Hessian-Free optimization	83%
Zhang et al. [23]	CNN with shortcut connections and max- and average-pooling	87.4%
Schindler et al. [17]	Deep CNN with LeakyReLU and data augmentation	82.2%
Lee and Nam [14]	Multi-Level and multi-scale feature extraction using CNN pretrained with The Million Song Dataset (MSD) [2]	72%

Table 1. State-of-the-art results of music genre classification task performed on GTZAN dataset.

the subset of AudioSet. The author compared the feature engineering with traditional classifiers and classification using deep Convolutional Neural Networks (CNNs). We decided to use the code [3] of this research as our baseline since his task is similar to ours, and the techniques Bahuleyan uses correspond to the contents of the course [4] we were doing the project for. Bahuleyan achieved an AUC value of 0.894 by using the ensemble of the mentioned approaches.

Music genre classification using GTZAN dataset. GTZAN dataset was introduced by Tzanetakis and Cook [21] and became a popular dataset for those who want to perform music genre classification task. There are dozens of research papers published that address the genre classification task on GTZAN dataset. To survey the state-of-the-art results we reviewed several recent papers, they are briefly summarized in Table 1.

This work aims to obtain a reasonably good result on the music role classification task and provide a feedback on the dataset quality. To achieve these goals we applied the techniques we learned from the GCT634 course: feature extraction and classification using traditional approaches and deep learning techniques such as training Shallow CNN network or using a pretrained deep neural networks.

3. DATASET

Audioset has been published in 2017 by members of Google Research "to bridge the gap in data availability between image and audio research" [9], referring to the lack of audio datasets at the scale of ImageNet [7] containing 14 million sample images. The authors created an ontology of 632 audio event classes, divided into seven major classes: human sounds, natural sounds, animal sounds, music, source-ambiguous sounds, background/noise and sounds of things. See the publication [9] for detailed explanations on the creation of the ontology. Human labelers categorized 10s segments from YouTube clips. Currently, AudioSet contains 2.1 million annotated videos covering 527 classes. The dataset gives the alphanumeric YouTube video ID, start time, end time, and one or more labels for each segment. For music role category the dataset file with all 2.1 million labels contained 28,725 segments. While there are 10 classes of music role in the ontology, there is currently no segments labeled 'birthday music'. Since the subset is unbalanced with number of segments ranging from 6,510 to 682, we decided to sample 600 segments per class [5] . Using the video ID, start time and end time, audio clips can be retrieved by downloading from YouTube. However, about 5% of these IDs per class are no longer existent, probably because the corresponding video was deleted. Detailed numbers on the music role subset are given in Table 2.

Class label	Elements available	Sampled	Retrieved
Background music	4080	600	565
Christmas music	2000	600	573
Dance music	4574	600	568
Jingle	1257	600	566
Lullaby	3116	600	568
Soundtrack music	6510	600	572
Theme music	3780	600	570
Video Game music	2726	600	563
Wedding music	682	600	580
Birthday music	0	0	0
Sum	28725	5400	5125

Table 2. Music role subset of AudioSet

The dataset is split into 90% training data, 5% validation and 5% test data. See Section 5.2 for further explanations and a discussion on this.

4. METHODOLOGY

This section describes the approaches we used to perform the music role classification task

4.1 Deep Neural Networks

In image classification tasks images are fed into CNNs as three dimensional matrices where the first two dimensions are height and width of the image and the third dimension is the color channel, i.e. red, green and blue. Each entry represents the intensity of a each color channel at this position. To obtain 3-channel images of audio files, we plot the mel-spectrogram of the signal using LibROSA

[3] The code has been opensourced by the author and is available at https://github.com/HareeshBahuleyan/music-genre-classification

[4] Korea Advanced Institute of Science and Technology. GCT634 Spring 2018 course on Musical Applications of Machine Learning. (http://mac.kaist.ac.kr/~juhan/gct634/)

[5] Disregarding 13% more potential data while maintaining the balance happened when we wanted to obtain a first result before the project presentation and wasn't corrected afterwards.

to RGB images of size 216x216. Mel-spectrograms are spectrograms computed using Short-time Fourier transform (STFT) scaled by Mel-scale to incorporate human audio perception. We use sampling rate of 22,050 Hz, Frame/Window size of 2048, hop size of 512 and 96 Mel bins.

4.1.1 CNN

Our 'shallow' CNN consists of four 2D convolutional layers and two fully-connected layers. We used following parameters for our network: sigmoid activation function on the output layer, ReLU activation for every hidden layer, 50% of dropout after second and fourth convolutional layer and before the output layer. Additionally, second, third and fourth convolutional layer were L2 regularized with weight 0.01. For optimization we used Adam algorithm.

4.1.2 Transfer Learning

Deep neural networks with as many parameters as the ones performing best on the ImageNet challenge require millions of training samples and therefore extensive training time. Feature visualization of trained neural networks show that earlier layers are sensitive to more generic low-level features, while the features get more specific as we go deeper in the network [22]. In transfer learning, we train two to three fully-connected layers to classify our new but somehow related data using the feature map created by the pre-trained CNN from this data. For our project we downloaded both the weights for VGG16 [19] and ResNet50 [10] to use them as non-trainable convolutional layers of our classification network. We used sigmoid activation for the output layer and categorical cross-entropy as loss function.

Following [1], all experiments were conducted using Keras in Tensorflow (here version 1.8.0) and accelerated by a GPU.

4.2 Feature extraction

As was mentioned in the Section 2, we used the code created by Bahuleyan [1] as baseline for our own investigation. For the feature retrieval we use Python library LibROSA.

Here is the list of features that were extracted in the baseline code. For the detailed description refer to work of Bahuleyan [1].

- Time-Domain Features
  - Central Moments
  - Zero-Crossing Rate (ZCR)
  - Root Mean Square Energy (RMSE)
  - Tempo
- Frequency-Domain Features
  - Mel-Frequency Cepstral Coefficients (MFCCs)
  - Chroma
  - Spectral Centroid
  - Spectral Contrast
  - Spectral Bandwidth
  - Spectral Centroid
  - Spectral Rolloff

For the features that are extracted on per-frame basis, e.g. MFCC, mean and standard deviation were used as a final features of the song sample.

Overall a feature vector of 97 features is extracted for each song sample in the work of Bahuleyan [1]. Since the feature vector is long, it is possible that some of the extracted features introduce confusion among the samples and prevent them from being classified correctly. In Section 5.4.1 we describe experiments on identifying this issue. We managed to reduce the feature vector alongside with improvement of the classification performance.

4.3 Classifiers

In our study we experimented with four machine learning classifiers that are commonly used for Feature-engineering based classification. The description of experiments can be found in Section 5.4.2.

The four classifiers we used are:

- **Logistic Regression (LR) [6]:** a Linear binary classifier. Since music role classification task involves multiple classes, a separate classifier for each class was trained using one-vs-the-rest method.
- **Random Forest (RF) [3]:** method uses ensemble of the pre-defined number of decision trees. Each decision tree is required to be trained with only a subset of the training samples and to make predictions using a random subset of features. Final decision is based on the majority voting among the trees.
- **Extreme Gradient Boosting (XGB) [4]:** an implementation of boosting model that supports fast training of the model. Boosting model is an ensemble classifier that combines the prediction of the weak models. The difference with the RF is in sequential training of the boosting models using forward stagewise additive modeling.
- **Support Vector Machines (SVM) [5]:** a linear classifier, that addresses classification problem of non-linearly distributed classes by projecting the input data into higher dimensional spaces. That projection is performed using the kernel of the classifier. In this work we use Radius Basis Function (RBF) following the decision of Bahuleyan [1].

For more detailed description of the algorithms refer to the original papers.

5. EXPERIMENTS AND RESULTS

This section describes the experiments we conducted based on the methods introduced in Section 4 and results of these experiments.

5.1 Evaluation method

To evaluate the results of classification experiments we use these three metrics:

- Accuracy – percentage of correctly predicted labels among all the predictions for all the samples of the set.
- F-score [6] – harmonic mean of the precision and recall. The latter two are calculated based on the confusion matrix. Precision and recall are averaged among all the classes they are calculated for.
- AUC [7] – the area under the ROC curve, where ROC stand for Receiver Operating Characteristic. It indicates the relation between True-Positive Rate (TPR) and False-Positive Rate (FFP) for different threshold levels. Random guess corresponds to AUC value of 0.5, so the system is expected to have AUC value greater than 0.5 to be meaningful.

5.2 Dataset Reduction

After we performed several experiments on classification of the given dataset (see Section 3), it became evident that examples for three of the nine classes contained in the dataset are often misinterpreted in predictions. Figure 1 is an example of confusion matrix from one of our experiments. This confusion matrix was plotted for the prediction of SVM classifier using features extracted by the baseline code from Bahuleyan [1]. As can be seen from this example, instances of Background, Theme and Soundtrack music are frequently confused with other classes, especially with Video game class and between each other. Furthermore, opposed to the other classes, these three don't have a clear overweight of correct predictions against confused prediction in the table.

In addition we visualized extracted features using t-SNE [15] method for dimensionality reduction, that is often used for visualization purposes. Figure 2 confirms that examples of the classes Background, Theme and Soundtrack can be found all over the place. Video game class examples seem to have the same problem on this figure, but as mentioned earlier, this class has a clear overweight of correct predictions against confused prediction in confusion matrix, see Figure 1. Its distribution in the t-SNE diagram can be explained by the fact that the above-mentioned problematic classes are particularly often confused with the Video game class.

To confirm the hypothesis we conducted the experiments of removing potentially-problematic classes from the dataset and performing the classification task on the reduced dataset. Here, removing a class means removing all the samples from the dataset that are labeled with this class. SVM method is used for classification with features extracted by the baseline code from Bahuleyan [1]. Results can be found in Table 3.

[6] https://en.wikipedia.org/wiki/F1_score

[7] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

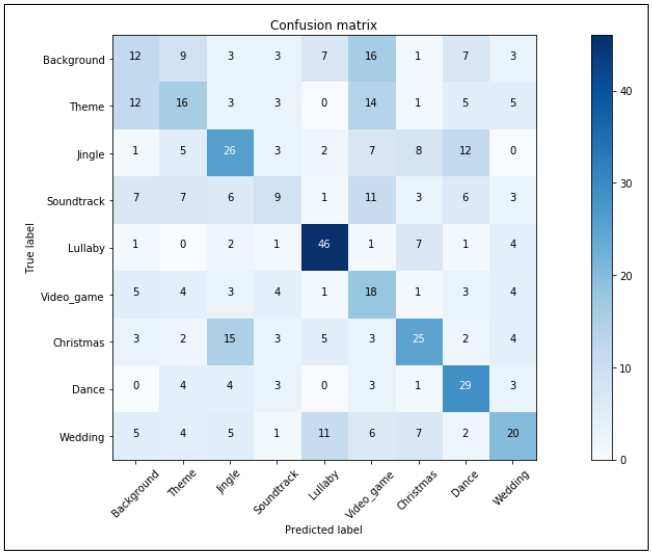

Figure 1. Confusion matrix of SVM classifier.

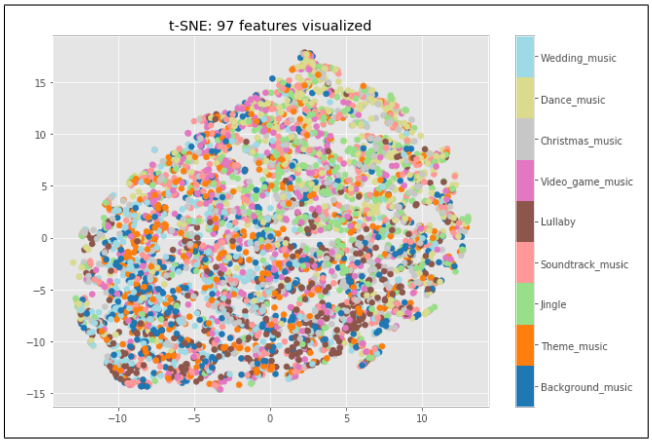

Figure 2. Visualization of the classes distribution using t-SNE [15].

Experiment	Accuracy	F-Score	AUC
Full dataset	0.40	0.39	0.807
Without Background	0.42	0.42	0.801
Without Theme	0.49	0.47	0.814
Without Soundtrack	0.40	0.39	0.804
Without Background and Theme	0.54	0.52	0.831
Without Theme and Soundtrack	0.45	0.45	0.818
Without Background and Soundtrack	0.54	0.53	0.856
Without Background, Theme and Soundtrack	**0.51**	**0.50**	**0.821**

Table 3. Results of experiments on dataset reduction. Reported for SVM classifier.

	Model	Test Accuracy	F-Score	AUC
Full dataset	CNN	0.37	0.34	0.794
	VGG16	0.42	0.40	0.831
	ResNet	0.46	0.45	0.823
Reduced dataset	CNN	0.55	0.53	0.837
	VGG16	0.61	0.60	0.874
	ResNet	see 5.3 for explanation		

Table 4. Best results for deep learning models

From the experiment results we can confirm that removing the classes in question increases the overall accuracy. Even though the best results were achieved on the dataset with only two classes removed (Background and Soundtrack), we preferred to remove all the classes from the dataset for future experiments. First, the classification results are not linear in terms of classes used, e.g. removing only Soundtrack class didn't contribute much, but removing it alongside with Background improved the results greatly, secondly, we realized that we used very small portion of the dataset for evaluation [8], thus the evaluation results might be biased.

We believe that our final decision might be inaccurate and more advanced techniques should be used to determine the problematic classes in the future work.

5.3 Deep Neural Networks

We used hyperparameter optimization over different learning rates, dropout and regularization weights to optimize the result of each model. Best result for each model is given in Table 4. Each model was trained on both the full and the reduced dataset, for details on this reduction see Section 5.2. "CNN" is the shallow network trained from scratch, "VGG16" and "ResNet" use the pre-trained convolutional layers from VGG16 [19] and ResNet50 [10] respectively. While ResNet performed best on the full dataset, we could not make it work on the reduced dataset. Although the training loss decreased and the training accuracy increased, the validation performance remained on the same low level. While this is normally a clear indication of overfitting, looking at the predicted classes, it turned out that the model assigns one class to all features. Despite our best effort we could not solve this issue.

Although the spectrograms lack distinctive structural features that can be found in ImageNet images like geometrical shapes, we were able to achieve superior results compared to training a CNN from scratch. We assume that the dataset is to small to train a network that performs and generalizes well given challenging input data.

5.4 Feature Engineering based model

As described in Section 4.2, in the baseline code adopted from Bahuleyan [1] a feature vector of 97 values consisting of various audio features was extracted for each data piece. This feature vector is than fed into a classifier. In this section we describe experiments conducted to reduce the feature space and to compare the results of different classifiers. All of the experiments use reduced dataset with Background, Theme and Soundtrack classes removed, see Section 5.2.

[8] Unfortunately, we revealed our mistake on using too little data for evaluation too late in a process and didn't have a chance to re-do the experiments due to time limit.

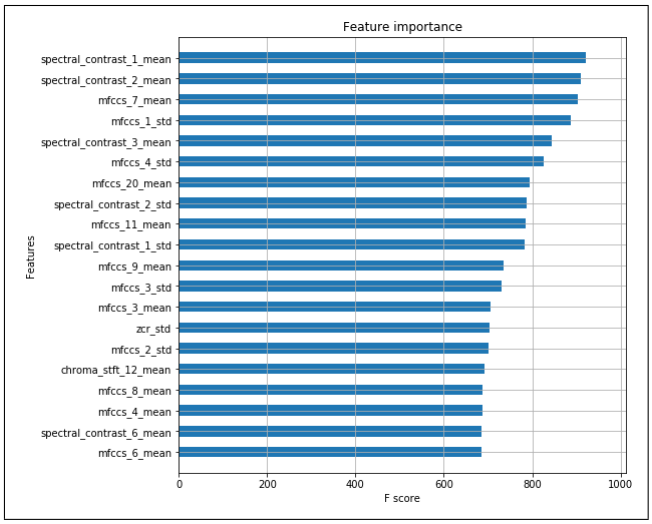

Figure 3. Top important features for the 97 features extracted. Classified using XGB method.

5.4.1 Feature Space Reduction

To further improve the performance of the classification task we conducted experiments on reducing the feature vector to eliminate elements that introduce confusion among the classes.

The feature importance graph in Figure 3 shows features contributed the most to determine classification result. According to these statistics we reduced the feature vector to contain only these features:

- MFCC mean and standard deviation
- ZCR mean and standard deviation
- Spectral Contrast mean and standard deviation
- Spectral Rolloff mean and standard deviation
- Spectral Centroid mean and standard deviation
- Spectral Bandwidth mean and standard deviation for three different values of power to raise deviation from spectral centroid: 2, 3 and 4.

This feature vector will be further referred to as reduced vector. Original feature choice will be referred to as full vector.

Another notion from Figure 3 is the fact that MFCC plays a great role in determining the prediction of the classifier. Thus experimenting with MFCC parameters could provide a raise in the classification performance.

We also conducted experiments with dimensionality reduction on the feature space using Principal Component

Experiment	Accuracy	F-Score	AUC
Full vector with 20 MFCC components	0.51	0.50	0.821
Full vector with 13 MFCC components	0.54	0.54	0.828
Full vector with 7 MFCC components	0.54	0.54	0.830
Reduced vector with 13 MFCC components	0.52	0.52	0.837
Reduced vector with 7 MFCC components	**0.58**	**0.58**	**0.838**

Table 5. Results of experiments on feature space reduction. Reported for SVM classifier.

Classifier	Accuracy	F-Score	AUC
Logistic Regression	0.56	0.54	0.827
Random forest	0.54	0.53	0.838
Extreme Gradient Boosting	0.58	0.57	0.829
Support Vector Machines	**0.58**	**0.58**	**0.838**

Table 6. Performance comparison of different classifiers.

Analysis (PCA) with different number of principal components, but those didn't provide any notable difference. Thus we don't report these experiments in detail.

Results of the experiments of the feature vector reduction are shown in Table 5. All of the results are reported for SVM classifier. The hypothesis proofed to be valid, reduction of the MFCC components and using reduced vector helped to increase performance. AUC evaluation parameter increased monotonically as the features were reduced even though the increase in accuracy was not monotonic. Best performance was achieved with MFCC of size 7 and reduced feature vector. Altogether they form a feature vector of 40 values, that outperform original classification with 97 features and provide less computational burden due to using the shorter feature vector.

5.4.2 Choosing Classifier

The last part of the feature engineering based model is the classifier itself. We compare performance for four different classifiers including Logistic Regression, Random Forest, Extreme Gradient Boosting and Support Vector Machine. These classifiers are described in Section 4.3.

Table Table 6 shows the results of the experiments conducted. Random Forest gives the least accuracy but outperforms Logistic Regression and XGB in terms of AUC and has the same AUC value as SVM. Best accuracy of 58% is achieved by SVM and XGB, but SVM outperforms other methods by the combination of all three values.

5.5 Ensemble classification

For the final part of the investigation we combined two models based on different paradigms: deep learning and feature engineering based, to form an ensemble of classifiers. This technique is widely used to incorporate the benefits of different prediction models to get a better performance.

There are two major strategies to combine several prediction models in one: majority voting and averaging the predicted probabilities. Since we have only two models to combine, number of voters is not enough for the first technique, thus we use the averaging predictions approach. For combination we use the best models from each side: tuned VGG16 and SVM classifier with reduced feature vector. Both are evaluated on the reduced dataset, see Section 5.2.

Expectedly, Table 7 and Figure 4 show ensemble to be beneficial and outperform both individual models. Each model uses different input data, spectrograms for VGG and extracted features for SVM, and use different paradigms to approach classification task.

Model	Accuracy	F-Score	AUC
VGG16	0.61	0.60	0.878
SVM	0.58	0.58	0.838
Ensemble of VGG16 and SVM	**0.64**	**0.64**	**0.885**

Table 7. Performance comparison of Deep Learning and Feature engineering based models and their ensemble.

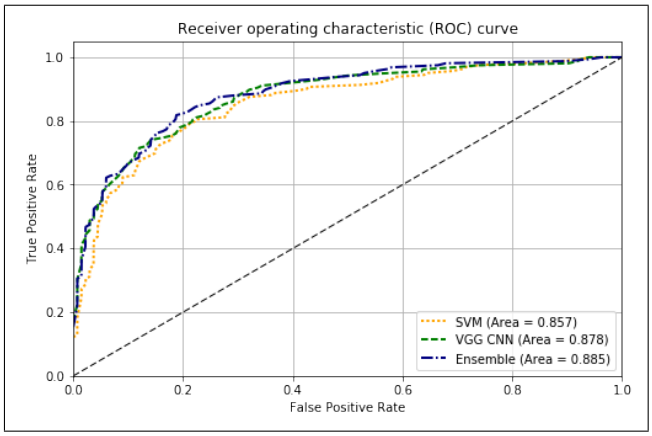

Figure 4. ROC Curves for best performing Deep Learning and Feature engineering based models and their ensemble.

6. DISCUSSION

In this report we outlined our work on the music role classification task. We were able to improve the classification performance using different machine learning techniques we learned during the semester. However, given the ambiguity of music role we believe that a multi-labeling task would be more appropriate and successful.

Another finding of our work for us is the necessity to question, discuss and adapt the design choices of baseline code to the given boundary conditions of our own problem at an early stage. This avoids problems like our test set choice and the uncertain choice of dataset reduction as described in Section 5.2. The problem we faced with some

of the dataset classes should also be adressed more thoroughly and properly, including more various experiments to correctly determine and eliminate problematic classes. In future work, we would switch to multi-labeling and conduct supplemental experiments using both more refined deep learning and feature engineering models.

7. AUTHOR CONTRIBUTIONS

Maria did the literature review, worked on the feature engineering approach and prepared the presentations. Jonas adopted the audio retrieval for our project and worked on the deep learning approach. To poster and report we contributed equally.

8. REFERENCES

[1] Hareesh Bahuleyan. Music genre classification using machine learning techniques. *CoRR*, abs/1804.01149, 2018.

[2] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference, October 24-28, 2011, Miami, Florida*, pages 591–596. University of Miami, 2011.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[6] D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242, 1958.

[7] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[8] James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):125, 2010.

[9] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 776–780. IEEE, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, March 2017.

[12] Qiuqiang Kong, Yong Xu, Wenwu Wang, and Mark D. Plumbley. Audio set classification with attention model: A probabilistic perspective. *CoRR*, abs/1711.00927, 2017.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[14] J. Lee and J. Nam. Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging. *IEEE Signal Processing Letters*, 24(8):1208–1212, Aug 2017.

[15] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[16] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 570–576. MIT Press, 1998.

[17] Alexander Schindler, Thomas Lidy, and Andreas Rauber. Comparing shallow versus deep neural network architectures for automatic music genre classification. 2016.

[18] S. Sigtia and S. Dixon. Improved music feature learning with deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6959–6963, May 2014.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[21] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, Jul 2002.

[22] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[23] Weibin Zhang, Wenkang Lei, Xiangmin Xu, and Xiaofeng Xing. Improved music genre classification with convolutional neural networks. *Interspeech 2016*, pages 3304–3308, 2016.