International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

MUSIC INFORMATION RETRIEVAL AND GENRE CLASSIFICATION USING MACHINE LEARNING TECHNIQUES AND DEEP LEARNING

Nirmal Vekariya[1], Hemang Vyas[1], Nirav Dedhiya[1]

[1]*Government Engineering College, Rajkot, Affiliated to Gujarat Technological University*

Abstract - *In this hectic world, music plays a vital role. There are many genres of music available that people love to listen to, and there is a dire need to classify them. Classifying the music according to their genre is indeed a challenging task. As music consists of various features, fetching the essential and appropriate features is a crucial task in the field of Music Information Retrieval (MIR) and Genre Classification. Previous research on music genre classification systems centered primarily on the use of timbral characteristics, which restricts the output. In this study, we have used various machine learning algorithms and Deep Neural Network to classify the music based on their genre. In machine learning, we have used the SVM classifier, Decision Tree classifier, K-Nearest Neighbour (KNN) classifier, and Random Forest classifier for the task of genre classification. These algorithms are prevalent in the task of classification. Our work compares the accuracy of different machine learning classification algorithms and Deep Neural Networks, where Deep Neural Network has the highest accuracy of 80%.*

Key Words: **music feature extraction, music information retrieval, deep neural network, machine learning, Librosa, TensorFlow.**

1. INTRODUCTION

As the number of songs keeps on growing, people find it relatively hard to manage the songs of their taste. Since listening to music online has become very convenient for people, thanks to the rise of online music streaming services such as Spotify, iTunes, and others, users expect the music to be recommended by the service. To make that possible, we need to study people's listening choices and identify the genre that they listen to, which is the best way to do so. Owing to the rapid growth of the digital entertainment industry, automatic classification of music genres has acquired significant prominence in recent years. One way to effectively classify the song is genre-based classification.

This paper focuses on the application of machine learning to automatically classify the audio file based on its genre. The feature extraction from musical data as a first step of the genre classification will significantly influence how the model behaves with the unseen data. All the algorithms are trained based on all the features of the GTZAN dataset. In the first part of the work, we train our models using the extracted features from the .wav music file of the dataset. In the second part, we extract the required features from the music file. These features are provided as input to various models like SVM, Decision Tree, KNN, Random Forest, and Deep Neural Network. Based on those features, we classified the music genre. The overview of the classification system is described in figure 1. In this paper, we compare the accuracy score of various models, highlighting other features like the confusion matrix.

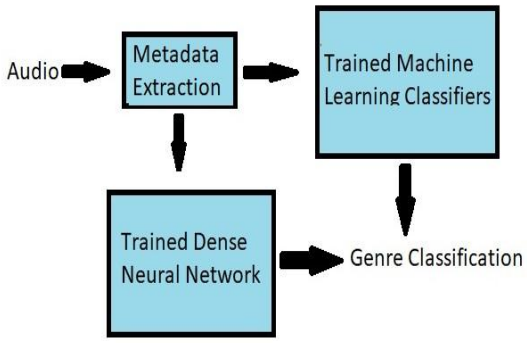


Figure-1: Overview of Music Genre Classification System

The remainder of the paper is structured as follows. Section 2 of this paper puts some light on the previous work related to this field, while in section 3, the structuring of the dataset is explained. Section 4 covers various classification algorithms and their details. The results and evaluations are mentioned in section 5 of this study, followed by the sections for conclusion and references.

2. LITERATURE REVIEW

Classifying the music without human interaction has been a fascinating problem for lots of people working from different branches like signal processing, machine learning, and music theory. There is a vast amount of research work related to audio and music classification.

The task of music classification is based on two different aspects, namely symbolic and audio. Symbolic classification mostly relies upon symbolic formats like MusicXML and MIDI. Several models have suggested conducting a symbolic classification of music genres. The input is used as a collection of instruments, musical sound, rhythm, dynamics, pitch figures, melody, etc. for a wide selection of multi-class generic classifiers. Symbolic music classification on audio files is highly impractical as making an effective audio transcription system ought to be more difficult than audio genre classification itself.

A work by Tzanetakis and Cook in (2002) [3], where researchers performed music genre classification using the timbral-related features, texture features, and pitch-related features based on the multi-pitch detection algorithm. Some of the features used in this work include MFCCs, roll-off, and spectral contrast. Their system achieved an overall accuracy of 61%. The work proposed by Lidy and Rauber (2005) [4] discusses the contribution of psycho-acoustic features to detect music genres.

A variety of experiments, with the recent popularity of deep neural networks, extend these methods to speech and other types of audio data (Abdel- Hamid et al., 2014; Gemmeke et al., 2017 [5]). The audio in the time domain is not entirely clear for feedback in neural networks due to the tremendous sampling rate. Nevertheless, it was discussed for audio generation tasks in Van Den Oord et al. (2016) [6]. The spectrogram of a signal that captures both frequency and time information is a common alternative representation.

In our proposed solution, we have compared the performance of several machine learning and deep learning algorithms that we have used for the task of music genre classification.

3. DATASET

In this proposed solution, we have used the GTZAN dataset, which is popular in the field of Music Information Retrieval. The dataset comprises the audio files which were gathered in the year 2000-2001 from a variety of sources like CDs, microphone recordings, radio.

This dataset contains 100 music files of each genre. There are a total of 10 genres so in total there are 1000 music files. 10 genres include Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, and Rock. It contains a 30 seconds audio clip of sampling rate 22050 Hz at 16 bit.

Source:- http://marsyas.info/downloads/datasets.html

4. METHODOLOGY

This section elaborates upon the task of data preprocessing followed by feature description and the two proposed approaches used for classification of music genre, Machine learning techniques and Deep Neural Network.

4.1 Preprocessing

To improve the model results, we processed the data by normalizing it and then converting the labels into categorical values. Since the dataset is very diverse in each feature, normalization of the data was necessary. We tried out different normalization methods like Standard Scaling, Z-score, Decimal Scaling, and Min-Max normalization, where Min-Max normalization gave the best results. In this technique of data normalization, a linear transformation is performed on the original data. The data is fetched along with the minimum, and maximum value and each value is replaced according to the following formula.

$$v` = \frac{v-min(A)}{max(A)-min(A)}(newmax(A) - newmin(A)) + newmin(A)$$

Where A is the given data, max(A) and min(A) are the minimum and maximum values of A, respectively. newmax(A), newmin(A) is the max and min value of the range(i.e., boundary value of range required), respectively. v' is the new normalized value and v is the old value of each entry in data.

To preprocess our dataset we have used pandas and NumPy library. Machine Learning related tasks for classification are done using the scikit-learn library, and the Deep Neural Network is written using Tensorflow Keras.

4.2 Manually Extracted Features

In this section, we have described various musical features used to train the machine learning algorithms and Deep Neural Network for the classification task. We have used Librosa, a python library for extracting the features.

4.2.1 Chroma

A chroma vector is typically a 12-element feature vector indicating how much energy of each pitch class (C, C#, D, D#, E, F, F#, G, G#, A, A#, B), is present in the signal.

4.2.2 Root Mean Square Energy (RMSE)

The RMSE of a signal corresponds to the total magnitude of the signal. For audio signals, that roughly corresponds to how loud the signal is. The energy in a signal can be calculated as follows:

$$\sum_{n=1}^{N} |x(n)|^2$$

After that, the root mean square value can be computed as:

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N}}$$

The calculation of RMSE is done frame by frame and then we take the average and standard deviation across all frames.

4.2.3 Spectral centroid

Every frame has a pre-specific frequency band number. And the spectral contrast is measured as the difference between maximum and minimum magnitudes within each frequency band.

$$f_c = \frac{\Sigma_k S(k) f(k)}{\Sigma_k fk}$$

4.2.4 Spectral bandwidth

Spectral Bandwidth is the difference between the upper and lower frequencies in a continuous band of frequencies of an audio signal. It is typically measured in hertz. The p-th order spectral bandwidth corresponds to the p-th order moment about the spectral centroid and is calculated as

$$\left[\sum_k (S(k)f(k) - f_c)^p\right]^{\frac{1}{p}}$$

4.2.5 Spectral Roll-off

For each frame, the roll-off frequency is specified as the center frequency for a spectral bin such that at least roll_percent (0.85 by default) of the energy of the spectrum in this frame is contained in this bin and the bins below. It can be used to, e.g., by setting roll_percent to a value close to 1 (or 0), we can approximate the maximum or minimum frequency.

4.2.6 Mel-Frequency Cepstral Coefficients (MFCC)

The mel frequency cepstral coefficients (MFCCs) of the signal are a small number of features that describe concisely the overall form of a spectral envelope (generally about 10-20). In MIR, it is often used to describe timbre.

4.2.7 Zero Crossing Rate (ZCR)

A zero-crossing point refers to one where the signal changes sign from positive to negative. The entire 10-second signal is divided into smaller frames, and the number of zero-crossings present in each frame is determined. The features are chosen by calculating the average and standard deviation of the ZCR score for all the frames.

4.3 CLASSIFIERS

This section provides insights into the classification techniques used to perform music genre classification. In this study, we have proposed two approaches for classification. The first approach, which is detailed in this section is based on Machine Learning techniques in

which we have used four classifiers K nearest neighbors (KNN), Support Vector Machine (SVM), Decision Tree and Random Forest.

4.3.1 Implementation Details

This section gives details about the implementation of machine learning algorithms that we have used. We have implemented all the machine learning classifiers using scikit-learn library.

1. **SVM:** Support Vector Machine is a supervised learning method for classification and regression. In this technique, we try to find a plane that has the maximum margin. So, there is a maximum distance between the data points of both classes. We have used Linear, Poly, and Radial Basis Function (RBF) kernels. It is implemented as a one-vs-rest classification task, and we got the best accuracy with Linear Kernel.

2. **KNN:** K Nearest Neighbors is simple and easy to implement a supervised learning algorithm that is widely used for the task of classification. The basic idea behind KNN is that similar things are near to each other, or in other words, the same traits exist nearby. The KNN classifier captures the notion of similarities among objects based on mathematics, like the calculating distance between the objects. In KNN, the test sample is assigned a class value to the class of the majority of its nearest neighbors. The KNN algorithm is based on the K value, which determines the number of training neighbors to which a test sample is compared. The most suitable value of K that we found is 13.

4.4 Deep Neural Network

In this section, we describe the second approach of classification, Deep Neural Network. A deep neural network is an architecture inspired by biological systems. DNN is Feed-Forward Networks where raw input flows from the input layer to the output layer without going backward. To extract the high-level features progressively from the raw input, it uses multiple layers.

4.4.1 Dense Neural Network

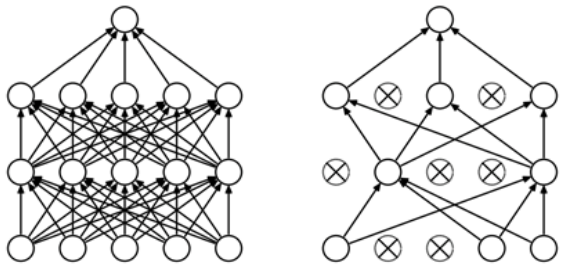
The name dense suggests that in the network, all the layers are fully connected by the neurons. Every neuron in a layer is input from all neurons in the last layer, so they are connected densely. This means that the dense layer is a completely connected layer, which means that all neurons in a layer are connected to those in the next layer.

- **ReLU:** ReLU stands for the *Rectified Linear Unit*. It is the most popular activation function that is chiefly implemented in hidden layers of Neural networks. It is non-linear in nature, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function**.** The ReLU layer applies the function f(x) = max(0, x) to all of the values in the input . In other words, this layer only changes all the negative activations to 0 and maintains the positive values.

- **Dropout:** The Dropout layer is used to prevent the problem of overfitting in neural networks. It randomly sets a fraction 'rate' of input units to 0 at each update during training time. This simplifies the neural network and decreases training. In each iteration, we use a different combination of neurons to predict the final output. Figure 2 provides insight into the structural change in the neural network after adding a dropout layer. In our work, a dropout rate of 0.3 is used, which means out of ten neurons, three will be shut off randomly.

- **Softmax:** Softmax is the form of logistic regression where it converts the input value into vectors of probability distribution that sums up to 1. The class having the highest probability is considered as the predicted class.

$$Softmax(x_i) = \frac{exp(x_i)}{\Sigma j \; exp(x_i)}$$



a) Neural Net without Dropout b) Neural Net with Dropout

Figure-2: The neural network structure with and without a dropout layer

4.4.2 Implementation Details

We created our Neural Network using Tensorflow Keras. In the first layer, we used 256 neurons with the 'ReLU' activation function. The input size of the neural network is a NumPy array of 26 elements, where each element represents the value of each feature extracted from the music. This layer is followed by three dense layers having 128 and 64 neurons respectively. We have also added dropout layers in-between these dense layers with a dropout rate of 0.3.

Since we have ten classes in total in the last layer, we used ten neurons and 'SOFTMAX' as an activation function where each neuron represents the probability of each class and then the class having maximum probability is considered. We have used sparse_categorical_crossentropy as a loss function and Adam as an optimizer.

Adam: We have optimized our model using Adam optimizer. Adam optimization algorithm can be seen as a combination of RMSprop and stochastic gradient descent algorithm with momentum. It is an adaptive learning rate method that computes individual learning rates for different parameters. Adam works by calculating the estimations of the first and second moment of gradient to adapt the learning rate for each weight of the neural network. We can explicitly provide the learning rate to the Adam optimizer to specify how well the model learns. We have used the default learning rate of 0.001.

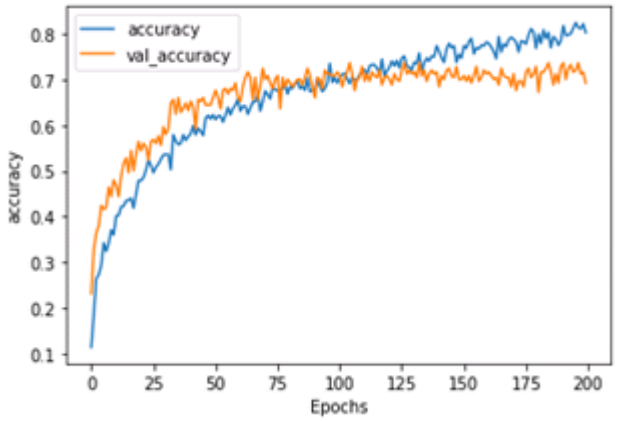
Sparse Categorical Cross-Entropy: The only difference between categorical cross-entropy and sparse categorical cross-entropy is that, if the class labels are one hot encoded then we can use categorical cross-entropy and if the class labels are in the form of integers then we can use sparse categorical cross-entropy.

The summary of the neural network is described in Table 1.

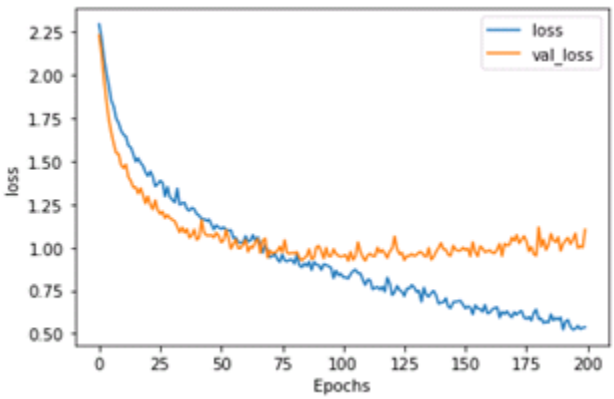
```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_40 (Dense)             (None, 256)               6912
_________________________________________________________________
dropout_30 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_41 (Dense)             (None, 128)               32896
_________________________________________________________________
dropout_31 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_42 (Dense)             (None, 64)                8256
_________________________________________________________________
dropout_32 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_43 (Dense)             (None, 10)                650
=================================================================
Total params: 48,714
Trainable params: 48,714
Non-trainable params: 0
```

Table-1 : Summary of the neural network with dropout layer

After adding the dropout layer, the difference between training and validation accuracy is less (as shown in fig. 3), hence overcoming overfitting. We got 80% training accuracy and a 71% validation score.



a) Accuracy



b) Loss

Figure-3: Learning curves: figure 3 (a) describes accuracy and figure 3 (b) describes the loss of neural network.

5. EVALUATION

In this section of the paper we have discussed the evaluation measures like accuracy, feature importance, and confusion matrix in order to evaluate the trained models.

5.1 Accuracy

It is defined as the percentage of correctly classified test labels. Table 2 provides the accuracy of the classifiers detailed in section 4.

Classifiers	Training accuracy	Validation accuracy
KNN	68%	62%
SVM	61%	62%
Decision Tree	60%	47.6%
Random Forest	77.6%	58.8%
Deep Neural Network with dropout	**80%**	**71%**

Table-2: Comparing the training and validation accuracies of various classifiers used

5.2 Results and Discussion

In this section, the different classifiers used in the study are evaluated based on the table 1 described in section 5.1.

In our study, the Deep Neural Network performs best as it has the highest training (80%) and validation (71%) accuracy. While the decision tree classifier performs the worst with the lowest accuracy due to its instability with large data. It is evident that SVM with RBF kernel outperforms decision tree. KNN is a widely used supervised learning classifier and it's easy to implement. KNN performs better than SVM and decision tree in our study. While a Random Forest classifier yields a far better training accuracy but it fails to classify the test samples correctly.

5.2.1 Feature Importance

In this section we can analyze which features play a vital role during prediction of genre, in the classification task. To do this analysis, we have ranked the top 25 features that are used to predict the genre of music. As shown in figure 4, the 'root mean square energy (rmse)', 'chroma_shift' and 'mel frequency cepstral coefficients 4 (mfcc4)' play a significant role in the music genre classification task. A previous study has shown that 'rmse' plays an important role in the music genre classification.

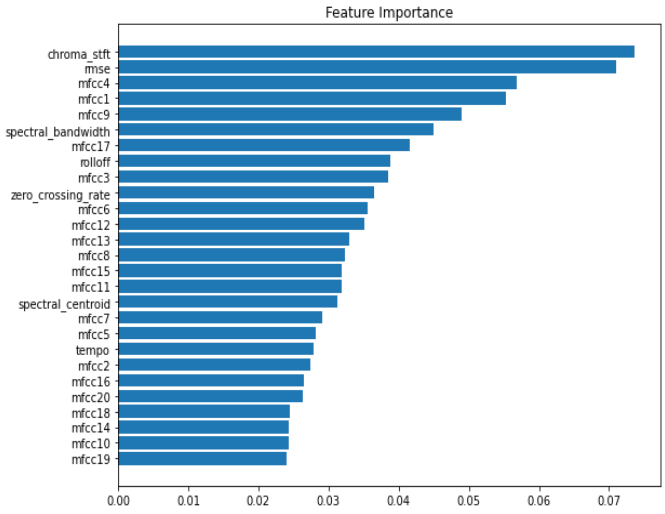


Figure-4: Feature Importance plot

5.2.2 Confusion Matrix

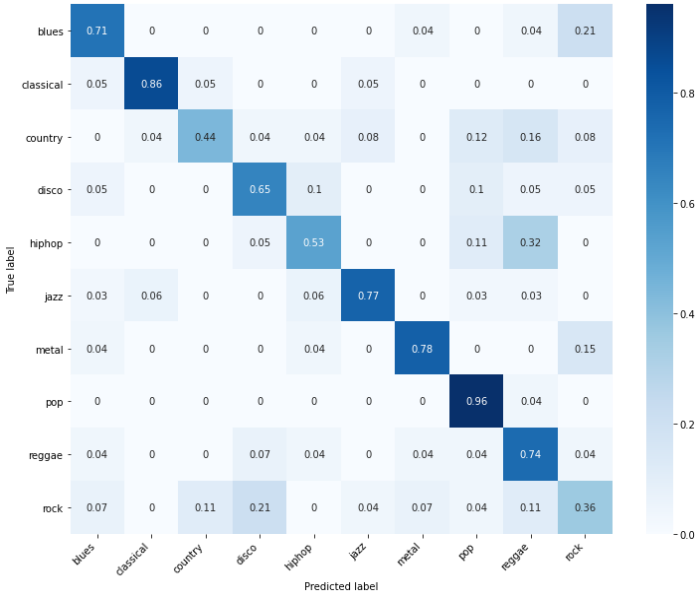


Figure-5: The confusion Matrix of the best model

Confusion matrix is a tabular representation that allows us to understand our model's strengths and weaknesses. Element A_{pq} in the matrix refers to the number of test instances of class p that the model predicted as class q. In the matrix, diagonal elements correspond to the correct predictions. It is clear from the confusion matrix, as shown in figure 5, our model predicts the best results for the 'classical' and 'pop' genre.

6. CONCLUSION

In this paper, we have provided the methodology for automatically extracting musical features from audio files and classifying the audio files based on their genre. We pre-process the data first, followed by feature extraction and selection, lastly followed by classification. Here, we focused our spectrum of features onto just Chroma-based features as these act as a useful metric for the human perception of music. For the task of classification, we have used various machine learning techniques and the Deep Neural Network. Our research concludes that the maximum accuracy of 80% is obtained using Deep Neural Network for ten genre classes. We have also highlighted the facts on feature importance where features like rmse and chroma_stft stand out to be the most vital features. It is evident from the confusion matrix that genres like disco and blues are quite tricky to classify, while genres like classical and pop are easy to classify accurately. One future direction of interest is to discover hidden relationships between music genres across time, which is not only a topic of interest, but it also has potential commercial applications. This exploration could lead to use of machine learning to determine artist influences that are directly applicable to playlist creation and song recommendation.

REFERENCES

1) McFee, Brian & Raffel, Colin & Liang, Dawen & Ellis, Daniel & Mcvicar, Matt & Battenberg, Eric & Nieto, Oriol. (2015). librosa: Audio and Music Signal Analysis in Python. 18-24. 10.25080/Majora-7b98e3ed-003.
2) Hareesh Bahuleyan, Music Genre Classification using Machine Learning Techniques, University of Waterloo, 2018.
3) Tzanetakis, G. and Cook, P. Musical genre classification of audio signals, IEEE Transactions on speech and audio processing Volume 10, Number 5, p293–302, 2002
4) Lidy, T. and Rauber, A. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR05) p34–41.
5) Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing 22(10):1533–1545.
6) Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.
7) Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1):1929–1958.
8) Leo Breiman. 1996. Bagging predictors. Machine learning 24(2):123–140.
9) Yali Amit and Donald Geman. 1997. Shape quantization and recognition with randomized trees. Neural computation 9(7):1545–1588.
10) Andrew Y Ng. 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In

Proceedings of the twenty-first international conference on Machine learning. ACM, page 78.

11) Corinna Cortes and Vladimir Vapnik. 1995. Support Vector networks. Machine Learning 20(3):273–297.
12) Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
13) Franc¸ois Chollet, "Keras: Deep learning library for theano and tensorflow," https://github.com/fchollet/keras, 2015.
14) Yann LeCun and M Ranzato, "Deep learning tutorial," in Tutorials in International Conference on Machine Learning (ICML13), Citeseer. Citeseer, 2013.
15) Basili, R. and Serafini, A. and Stellato, A. Classification of musical genre: a machine learning approach Proceedings of ISMIR 2004.
16) Wu H., Gu X. (2015) Max-Pooling Dropout for Regularization of Convolutional Neural Networks. In: Arik S., Huang T., Lai W., Liu Q. (eds) Neural Information Processing. ICONIP 2015. Lecture Notes in Computer Science, vol 9489. Springer, Cham.