MUSICAL GENRE AND STYLE RECOGNITION USING DEEP NEURAL NETWORKS AND TRANSFER LEARNING

Deepanway Ghosal, Maheshkumar H. Kolekar
Indian Institute of Technology Patna, India
deepanwayedu@gmail.com, mkolekar@gmail.com

***Abstract*—Music genre and style recognition are very interesting areas of research in the broad scope of music information retrieval and audio signal processing. In this work we propose a novel approach for music genre and style recognition using an ensemble of convolutional neural network (CNN), convolutional long short term memory network (CNN LSTM) and a transfer learning model. The neural network models are trained on a diverse set of spectral and rhythmic features whereas the transfer learning model was originally trained on the task of music tagging. We compare our system with a number of recently published works and show that our model outperforms them and achieves new state of the art results.**

I. Introduction & Related Work

Music information retrieval (MIR) is an interdisciplinary field dealing with the analysis of musical content by combining aspects from signal processing, machine learning and music theory. MIR enables computer algorithms to understand and process musical data in an intelligent way. Musical genre and style recognition is one of the most important subfields of MIR. Automatic musical genre and style analysis is a very interesting problem in the context of MIR because it enables systems to perform content based music recommendation, organizing musical databases and discovering media collections.

Music genre is defined as an expressive music style incorporating instrumental or vocal tones in a structured manner belonging to a set of conventions. The first significant work on automatic musical genre and style recognition were performed in [1] by Tzanetakis and Cook. Timbral texture, rhythmic content & pitch content based features were proposed and classification was done using Gaussian mixture model (GMM) and K-nearest neighbor (K-NN) algorithms. Musical genre recognition using support vector machines were proposed in [2] by Xu et al. In [3] Costa et al. proposed the approach of musical genre recognition using spectrogram features. Instead of directly extracting features from audio data, features are extracted from the visual representation of spectrograms. In [4], [5] specific musical features were used with feature selection techniques. Survey works performed in [6], [7] gives a comprehensive account of genre classification of musical content and evaluation techniques. Authors in [8] introduced the Million Song Dataset - a collection of audio features and metadata for a million contemporary popular music tracks. A wide range of musical information retrieval systems can be build using this dataset including genre recognition, automatic music tagging, music recommendation, etc.

II. Datasets

In this work we have focused on genre and style recognition of the songs in the GTZAN dataset [1] and the Ballroom dataset [9]. This two datasets has been widely studied in the area of music genre and style recognition. The GTZAN dataset contains songs of ten different genres - blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae & rock. Each genre is represented by 100 tracks. The Ballroom dataset consists of eight different styles - cha cha, jive, quickstep, rumba, samba, tango, viennese waltz & slow waltz. There are total 698 tracks. In both the datasets all the audio tracks are 30 seconds long and 22050Hz Mono 16-bit audio files in .wav format. Distributions of these two datasets across genres & styles are presented below in Table. I & Table. II.

TABLE I: Genre distribution in GTZAN dataset.

Genre	# instances
Blues	100
Classical	100
Country	100
Disco	100
Hip-Hop	100
Jazz	100
Metal	100
Pop	100
Reggae	100
Rock	100

TABLE II: Style distribution in Ballroom dataset.

Style	# instances
Cha Cha	111
Jive	60
Quickstep	82
Ramba	98
Samba	86
Tango	86
Viennese Waltz	65
Slow Waltz	110

III. Proposed Methodology

To recognize the genre/style of a song, we first train our deep neural network models on a set of extracted spectral and rhythmic features. We also utilize a transfer learning system to extract meaningful features from the songs. A multilayer perceptron network is then trained on this transferred features and the average of the spectral & rhythmic features to predict the genre/style. Finally the predictions of different models are combined using a majority voting ensemble.

A. Feature Extraction

1) Two Dimensional Spectral and Rhythmic Features:: A diverse set of spectral and rhythmic domain features are first extracted from the raw musical wav signals. In the features listed below, 'Tonnetz' and 'Tempogram' are rhythmic features, the rest are spectral features. The musical data in both the datasets are sampled at 22050 Hz and are 30 seconds long resulting in a total of 22020*30 = 661500 samples. We compute the features for each sliding window of 1024 samples. This results in $661500/1024 = 646$ windows and thus each song is represented as a $(646, k)$ dimensional feature matrix. The exact choice of k depends on the feature being computed.

• **Mel Spectrogram:** Mel-frequency cepstrum (MFC) representations introduced in [10] are widely used in automatic speaker and speech recognition. The Mel spectrogram produces a time-frequency representation of a sound imitating the biological auditory systems of human beings. We compute the magnitude spectrum from the time series musical data and then map it into the mel scale. We used k=128.

• **Mel, Delta and Double Delta Coefficients:** Mel coefficients (MFCCs) are the coefficients that collectively make up a Mel-frequency cepstrum. We use 20 (= k) of this coefficients as features. We also used derivative and double derivative of the Mel coefficients known as Delta and Double Delta coefficients.

• **Energy Normalized Chromagram:** Chroma audio features are extensively used in musical signal processing. Chroma features are effective in audio matching and retrieval applications [11], [12] as they capture melodic and harmonic characteristics of music and are robust to changes in instrumentation and timbre. In [13] authors introduced Chroma Energy Normalized Statistics (CENS) features by considering short time statistics over energy distributions within the chroma bands. We took k=12 as it represents 12 distinct semitones of the musical octave.

• **Constant Q Chromagram:** Constant Q transform [14] constitutes of a bank of filters with logarithmically spaced center frequencies $f_n = f_0 * 2^{\frac{n}{b}}$ where $n = 0, 1, ..$; central frequency of the lowest filter is denoted by f_0 and the number of filters in each octave is denoted by b. An appropriate choice of f_0 and b directly corresponds to musical notes. This transform also has increasing time resolution towards higher frequencies resembling the human auditory system. k=12 was taken.

• **Short Time Fourier Transform (STFT) Chromagram:** Chromagram of short-time chroma frames are used with k=12.

• **Tonnetz:** Tonal centroid features (tonnetz) are computed following works in [15]. Authors show that this features are successful in detecting changes in the harmonic content of musical audio signals, such as chord boundaries in polyphonic audio recordings. We used k = 6 tonnetz features.

• **Tempogram:** The aspects of tempo and rhythm are very important dimensions of music. In [16], the authors introduced a robust mid-level representation that encodes local tempo information by computing local autocorrelation of the onset strength envelope in music signals. This tempogram feature can act as a very important source of information for MGR, specifically where music reveals significant tempo variations. We used k = 128 tempogram features.

2) One Dimensional Averaged and Transfer Learning Features: We also compute the following one dimensional vectors as a summary statistic of the whole song.

• **Averaged Signal Vector**: This vector is calculated simply by taking the average of all the extracted two dimensional features listed above. After extracting $(646, k_1)$ dim matrix from Mel Spectrogram, $(646, k_2)$ dim matrix from Mel Coefficients, ..., $(646, k_n)$ features from tempogram, the averaging was performed over these 646 windows. Finally vectors of k_1, k_2, ..., k_n dimensions were obtained which were then concatenated to obtain the averaged signal vector. Our particular choices of k_1, ..., k_n led to this vector having dimension of 342.

• **Music Transfer Learning Vector**: Transfer learning is frequently used in computer vision problems. In this kind of systems, generally a deep convolutional net trained on the large scale ImageNet data [17] is used. Although the original network is trained on ImageNet data, it is able to capture a wide variety of visual features which are then used for other recognition tasks. In [18] authors introduce a musical transfer learning system. A deep convolutional neural network is first trained on a large dataset [8] for music tagging. This trained network is then used as a feature extractor for other related tasks. We use the model to extract a 160 dimensional vector for each song.

B. Models

Convolutional neural networks (CNN) are specially designed neural networks for processing data that has a grid-like topology [19]. Introduced by LeCun et al. [20] convolutional neural networks have produced excellent results in a wide variety of problems including computer vision [21], [22], speech recognition [23], natural language processing [24] and reinforcement learning [25]. In [24] Kim et al. introduced the idea of 1D convolutional neural network being applied to a variety of natural language processing (time series) problems and producing state of the art results. Long short term memory (LSTM) networks [26] are also widely used in sequential time series data to capture long term dependencies. LSTM networks effectively tackles the exploding or vanishing gradient problem by using an input, an output and a forget gate.

In this work, we apply variants of 1D (one dimensional) CNN & 1D CNN-LSTM models for musical genre/style prediction. The problem that we focus on is a time series problem as the extracted features from our musical data can be thought as a 1D grid data at regular time intervals. Considering that our spectral and rhythmic features have dimensions of $(646, k)$ (section III-A) one can expect that a 2D convolution would be appropriate, but the fact that 646 is the number of windowed time-steps and k is the 1D grid of features at each time-step, makes 1D CNN & 1D CNN-LSTM networks more suitable for our problem. Further 2D convolution is generally

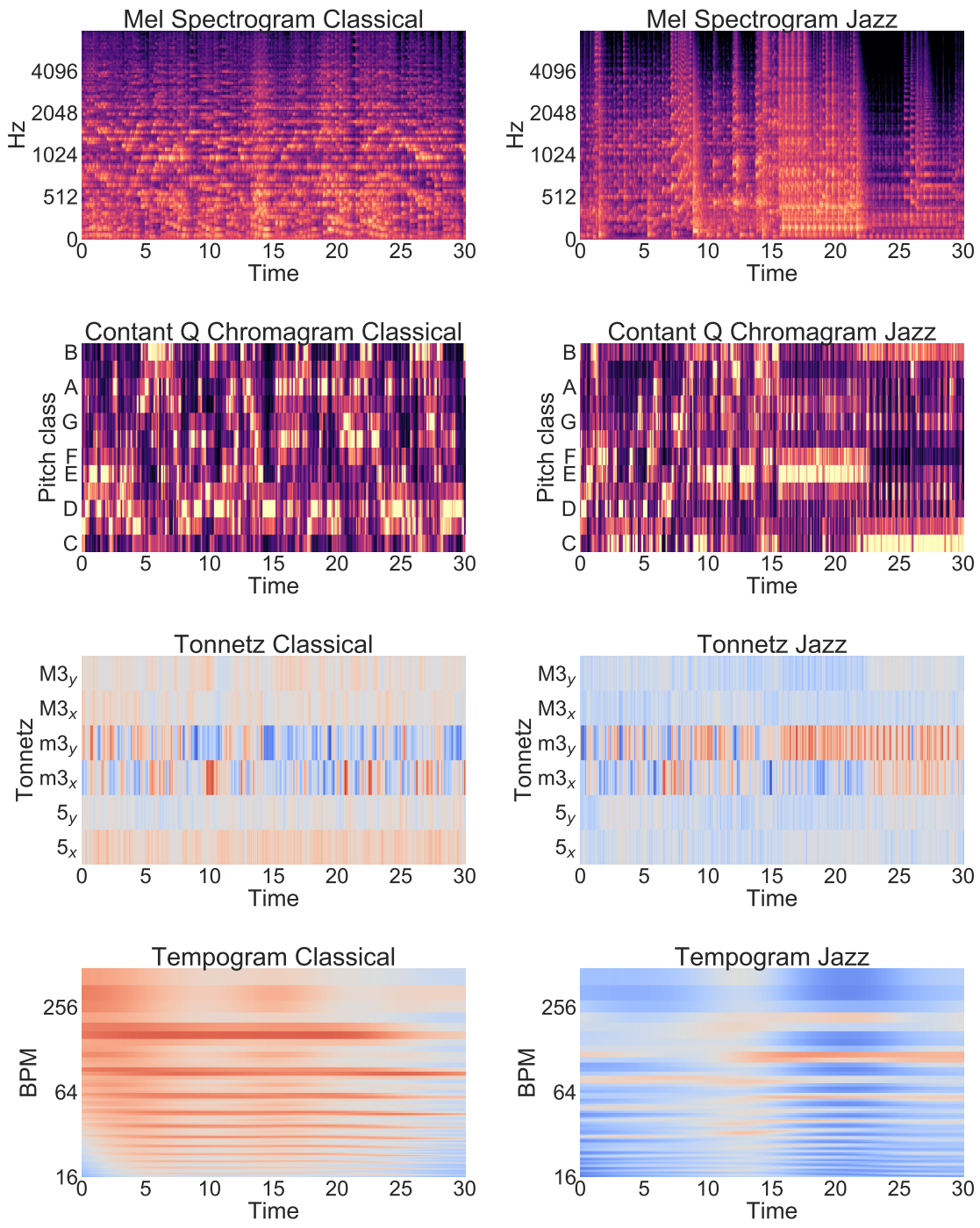


Fig. 1: Distinctive spectral and rhythmic features of two songs belonging to the 'Classical' and 'Jazz' genre in GTZAN dataset. Mel Spectrogram and Constant Q Chromagram are spectral domain features, whereas Tonnetz and Tempogram are rhythm domain features. Similar phenomenon is observed for the rest of the features across all the genres in GTZAN dataset and the styles in Ballroom dataset.

applied for spatial domain features (height-width-depth in images), whereas our features are time-series features and doesn't represent any kind of height-width-depth spatiality.

► Next we briefly discuss how these 1D CNN & 1D CNN-LSTM models operate. These models consists of a number of stages:

a) ***Convolution stage A***: This is the first convolution stage. A number of filters/kernels (generally tens to thousands) of very small dimension are slided and convoluted over the input data to create a feature map. The resultant 1D convolution (C_{A1}) between the data (D) and a filter (F_{A1}) is,

$$D * F_{A1} = C_{A1}$$

specifically,

$$\begin{bmatrix} a & b \\ c & d \\ e & f \\ g & h \\ i & j \end{bmatrix} * \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} aw+bx+cy+dz \\ cw+dx+ey+fz \\ ew+fx+gy+hz \\ gw+hx+iy+jz \end{bmatrix}$$

The important thing to note here is that the width and of filter (F_{A1}) and the data (D) should be same (here both have width of 2; the filter also has length of 2, whereas the data has length of 5). The filter would gradually slide over the length of the data and the resultant would be computed using dot product. Also note that, C_{A1} is the convoluted output of a single filter. Multiple such convoluted outputs (C_{A2}, C_{A3}, etc.) would be there for the multiple filters.

An element wise non-linear activation function is then applied over C_{A1}. In general *Rectified Linear Unit (ReLU)* [27] is widely used. It is defined as, $ReLu(x) = maximum(0, x)$. To illustrate,

$$C'_{A1} = ReLU(C_{A1})$$

$$\text{specifically if,} \quad C_{A1} = \begin{bmatrix} -5 \\ 4 \\ 3 \\ -2 \end{bmatrix} \quad \text{then,} \quad C'_{A1} = \begin{bmatrix} 0 \\ 4 \\ 3 \\ 0 \end{bmatrix}$$

b) ***Pooling stage***: Pooling is a sample based discretization process where we down-sample an representation to provide an abstracted form of the input. The pooling function is computed on the column-wise concatenated feature maps (outputs) from the *Convolution stage A*. Assuming there were 4 distinct filters, we will have 4 different outputs from the *Convolution stage A*. Let these are C'_{A1}, C'_{A2}, C'_{A3} & C'_{A4}. These are concatenated column-wise to obtain C'_A.

$$C'_A = concatenate[C'_{A1}, C'_{A2}, C'_{A3}, C'_{A4}]$$

The pooling operation is then computed along the length of C'_A. Generally max pooling or average pooling are used. If,

$$C'_A = \begin{bmatrix} 0 & 2 & 1 & 0 \\ 4 & 4 & 0 & 0 \\ 3 & 5 & 9 & 1 \\ 0 & 2.5 & 2 & 3 \end{bmatrix}$$

then max pooling and average pooling with factor 2 would result in,

$$\textit{max pooling } (C'_A) = \begin{bmatrix} 4 & 4 & 1 & 0 \\ 3 & 5 & 9 & 3 \end{bmatrix}$$

$$\textit{average pooling } (C'_A) = \begin{bmatrix} 2 & 3 & 0.5 & 0 \\ 1.5 & 3.75 & 5.5 & 2 \end{bmatrix}$$

c) ***Convolution stage B***: The next convolution and activation operation is performed on the max or average pooled output of C'_A. Let after the convolution and activation, the outputs of this stage are C'_{B1}, C'_{B2}, etc.

d) ***Global Pooling & LSTM stage***: At first we column-wise concatenate the outputs from the previous stage. Assuming there were 4 filters in the *Convolution stage B* we will have,

$$C'_B = concatenate[C'_{B1}, C'_{B2}, C'_{B3}, C'_{B4}]$$

We then apply a **global pooling** (max/average) operation on C'_B for the **1D CNN** models. If,

$$C'_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 4 & 3 & 2 \\ 3 & 2 & 2 & 0 \\ 6 & 0 & 3 & 1 \end{bmatrix}$$

then global max pooling and global average pooling would result in,

$$\textit{global max pooling } (C'_B) = \begin{bmatrix} 6 & 4 & 3 & 2 \end{bmatrix}$$

$$\textit{global average pooling } (C'_B) = \begin{bmatrix} 3 & 1.5 & 2 & 0.75 \end{bmatrix}$$

A single dimensional vector (having size equal to the no. of filters in the *Convolution stage B*) is thus obtained. This operation can also be thought as if, first taking the maximum along the length of C'_{B1}, C'_{B2}, C'_{B3} & C'_{B4}, and then concatenating the maximum values.

In contrast, we apply a **LSTM** operation for the **1D CNN LSTM** models. Here each row of the C'_B matrix works as the single time-step features for the LSTM network. We start from the top row and gradually move to the bottom row. We keep the hidden dimension of the LSTM network equal to the no. of filters in the *Convolution stage B*. The LSTM

network outputs a single dimensional vector after taking the C'_B matrix as input. Hence in this case also, we obtain a single dimensional vector (having size equal to the no. of filters in the *Convolution stage B*) after applying the LSTM operation.

e) ***Output stage***: The single dimensional vector obtained from the previous stage is then passed through a fully connected network to the output layer having 10 neurons (GTZAN) & 8 neurons (Ballroom) for the final genre/style classification.

► In total, we apply four different 1D CNN and 1D CNN-LSTM models on the extracted two dimensional features to predict the genre/style of the song. Structure of these models are outlined in Fig. 2. We briefly describe configurations of these models below. The first & third models are 1D CNN models, whereas, the second & fourth models are 1D CNN LSTM models. For the one dimensional averaged & transfer learning features, we use a simple multilayer perceptron (MLP) model for genre/style prediction. This is the fifth model described below.

• **CNN Max Pooling Model:** This sequence of stages are used: *convolution - max pooling - convolution - global max pooling - output*. The two convolution stages have 128 & 64 filters respectively, all with length of 3. The max pooling operation is performed with factor 2.

• **CNN Max Pooling LSTM Model:** This sequence of stages are used: *convolution - max pooling - convolution - lstm - output*. The two convolution stages have 128 & 64 filters respectively, all with length of 3. The LSTM network has hidden dimension of 64. The max pooling operation is performed with factor 2.

• **CNN Average Pooling Model:** The max pooling and global max pooling stages in the *CNN Max Pooling Model* are replaced with average pooling and global average pooling.

• **CNN Average Pooling LSTM Model:** The max pooling stage in the *CNN Max Pooling LSTM Model* is replaced with average pooling.

• **Multilayer Perceptron (MLP) Model:** The input to this network is an one dimensional feature vector. A single hidden layer with 256 nodes is used with *'ReLU'* activation. We also used 25% *Dropout* [28] in this hidden layer for regularization. The output layer has 10 (8) neurons corresponding to 10 (8) different genres (styles).

► We used *'Softmax'* activation for the output layer. All the models are trained with *'Adam'* optimizer [29] (learning rate of 0.001) with *cross entropy* loss function. We kept the *batch size* equal to 32 during the training process. We trained each model for 50 epochs with *Early Stopping* [30] having patience of 10.

IV. Experiments, Results and Discussion

The GTZAN dataset consists of 10 different classes of genres whereas the Ballroom dataset consists of 8 different classes of styles. We evaluate models for both the datasets in this multi-class classification framework. We run our experiments in a 10 fold cross validation setup. We maintain the uniform distribution of musical genres/styles in each fold i.e. we perform stratified 10 fold sampling.

For GTZAN dataset the average 10 fold accuracy score of our models are reported in Table. III. A number of interesting observations can be made from the results. First of all, we observe that the best result is obtained by the multilayer perceptron model when used with music transfer learning features. This result can be expected as the original system was trained on the very large Million Song Dataset [8] containing rich label sets for various aspects of music including 'mood', 'era', 'instrumentations' and most importantly 'genre'. Also further fine-tuning was performed on our experimental setup leading it to produce the best results. We also observe that the Mel Spectrogram features produces best results in CNN Max Pooling and CNN Average Pooling models, whereas Mel Coefficients produces best results for CNN Max Pooling LSTM and CNN Average Pooling LSTM models.

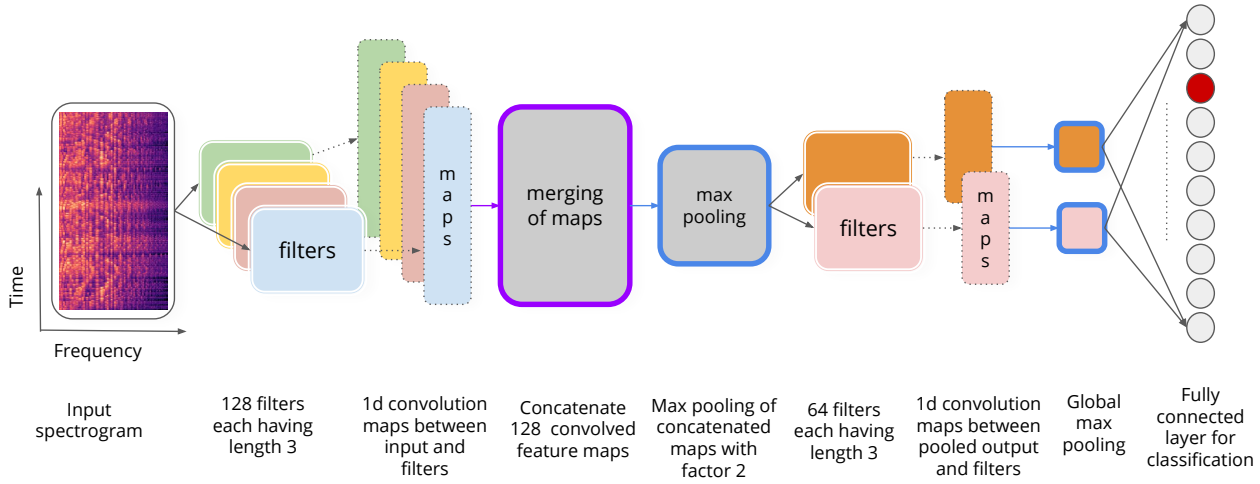
The introduction of LSTM resulted in improved performance for Delta Coefficients, Double Delta Coefficients, Tonnetz features and Tempogram features. The performance of Max Pooling and Average Pooling is somewhat consistent across all the feature sets. In some cases Max Pooling performs better, whereas in other cases Average Pooling performs better.

For Ballroom dataset the average 10 fold accuracy score of our models are reported in Table. IV. One important aspect to note here is that the Tempogram features are the best performing features in this dataset and outperforms all the other features by a big margin. This result is expected because the dataset was created in such a way, so that there are significant tempo variations between styles [9]. For the rest of the features, we observe similar kind of patterns in results as discussed above in GTZAN dataset.

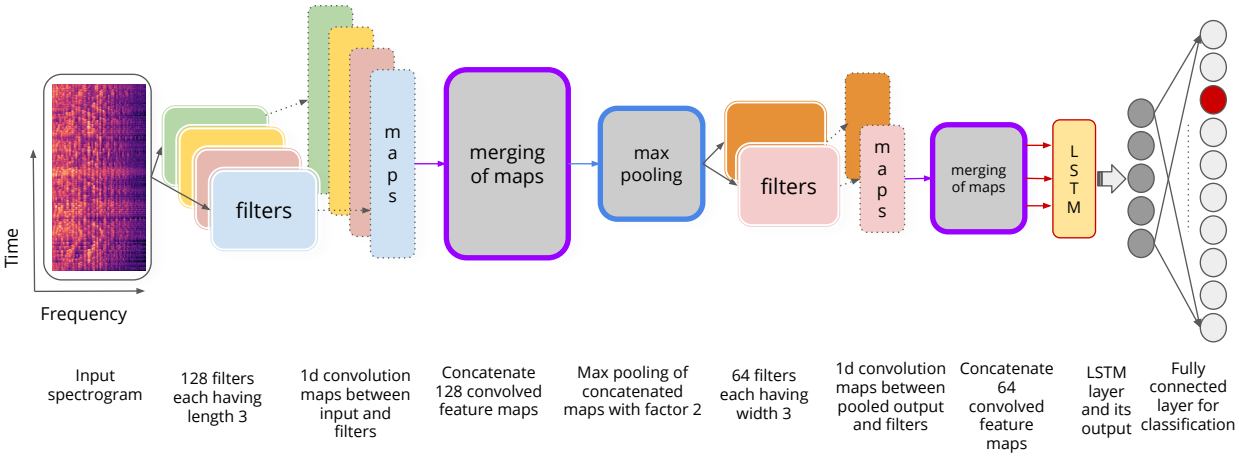
This heterogeneous and complimentary characteristics of the models led us to build an ensemble model which effectively improves the performance by combining the outputs of all the base systems. Our ensemble model is a simple majority voting ensemble of the deep learning and multilayer perceptron models; e.g. for a particular song, a number of genre/style predictions will be available from the base models. The genre/style which is predicted with most frequency will be the final assigned genre/style. If multiple genres are predicted with highest frequencies then the final decision is made based on the predicted softmax probabilities. By incorporating this simple rule, we were able to get a large improvement in performance as reported in Table V.

V. Comparative Analysis

In Table VI we compare our proposed model with other state-of-the-art systems. In [18] authors used a transfer learning system trained for music tagging to extract features for genre prediction. They reported scores of 89.8% by taking features from multiple layers of the transfer CNN model. Arabi and Lu [31] reported an accuracy of 90.79 % using



(a) ***CNN Max Pooling*** model.



(b) ***CNN Max Pooling LSTM*** model.

Fig. 2: a) ***CNN Max Pooling*** and b) ***CNN Max Pooling LSTM*** models for Mel Spectrogram input. For ***CNN Average Pooling*** and ***CNN Average Pooling LSTM*** models, the max pooling and global max pooling functions are replaced with average pooling and global average pooling functions respectively. Note that, this network structure is for GTZAN dataset; for Ballroom dataset the output layer will have 8 neurons.

a SVM classifier over selected combination of high (chord progression, chord distribution, beat) and low level (flux, flatness, roll-off, spectral centroid) musical features. In [4] Panagakis et al. used rich, psycho-physiologically inspired properties of temporal modulations of music with a sparse representation based classifier to achieve accuracy score of 91.0 %. Mostly pitch, temporal and timbre features were used with non negative matrix factorization as a feature reduction technique. Works by the same authors in [5] further increases the score to 93.7% by the utilization of topology preserving non-negative tensor factorization. For Ballroom dataset Klapuri et al. [32] reported an accuracy of 90.97 %. They account for subtle energy changes that might occur in narrow frequency subbands (e.g., harmonic or melodic changes) as well as wide-band energy changes (e.g.,drum occurrences). They also jointly determine three metrical levels (the tatum, the beat and the measure) through probabilistic modeling of their relationships and temporal evolutions. In [33] Marchand and Peeters proposed Modulation Scale Spectrum which achieved 93.1 % accuracy.

TABLE III: Average 10 fold cross validation accuracy in GTZAN dataset.

Features & Models (GTZAN)	**CNN Max Pooling**	**CNN Max Pooling LSTM**	**CNN Average Pooling**	**CNN Average Pooling LSTM**	**Multilayer Perceptron**
Mel Spectrogram	**83.0**	73.6	**82.5**	75.7	-
Mel Coefficients	80.2	**79.0**	81.6	**80.5**	-
Delta Mel Coefficients	70.4	77.2	74.5	77.0	-
Double Delta Mel Coefficients	72.1	72.9	72.1	76.5	-
Energy Normalized Chromagram	45.7	34.5	43.0	36.2	-
Constant Q Chromagram	60.0	49.4	57.5	45.6	-
STFT Chromagram	62.8	52.5	63.4	53.7	-
Tonnetz Features	50.2	53.5	51.0	55.8	-
Tempogram Features	41.5	42.0	41.6	43.3	-
Averaged Signal Features	-	-	-	-	77.1
Transfer Learning Features	-	-	-	-	**85.5**

TABLE IV: Average 10 fold cross validation accuracy in Ballroom dataset.

Features & Models (Ballroom)	**CNN Max Pooling**	**CNN Max Pooling LSTM**	**CNN Average Pooling**	**CNN Average Pooling LSTM**	**Multilayer Perceptron**
Mel Spectrogram	0.80	0.81	0.80	0.83	-
Mel Coefficients	0.28	0.26	0.32	0.34	-
Delta Mel Coefficients	0.68	0.65	0.64	0.64	-
Double Delta Mel Coefficients	0.74	0.70	0.72	0.71	-
Energy Normalized Chromagram	0.44	0.48	0.41	0.39	-
Constant Q Chromagram	0.57	0.60	0.57	0.58	-
STFT Chromagram	0.57	0.56	0.52	0.55	-
Tonnetz Features	0.50	0.52	0.54	0.53	-
Tempogram Features	**0.88**	**0.90**	**0.85**	**0.87**	-
Averaged Signal Features	-	-	-	-	0.22
Transfer Learning Features	-	-	-	-	**0.77**

TABLE V: Average 10 fold cross validation accuracy for ensemble models.

Models	Accuracy GTZAN	Accuracy Ballroom
Ensemble Models		
CNN Max Pooling & MLP	93.6	92.4
CNN Max Pooling LSTM & MLP	91.5	92.2
CNN Average Pooling & MLP	**94.2**	**93.8**
CNN Average Pooling LSTM & MLP	91.4	92.0

Our ensemble system of CNN Average Pooling and MLP models achieves an accuracy score of 94.2 % in GTZAN, which is at-least 0.5% more than the rest of the comparative systems. One important aspect to note here is the work by Sturm B. L. in [34].With rigorous examples and case studies, it is demonstrated that the perfect system in the GTZAN dataset would not be able to surpass the accuracy score of 94.5% due to the inherent noise in the some of the repetitions, mislabelings and distortions of the songs. Interestingly, our proposed system achieves accuracy of 94.2%, an almost perfect score. We also achieve state-of-the-art accuracy of 93.8% in Ballroom dataset which is 0.7% better than the previous best.

VI. Conclusion

In this work we proposed a novel approach for music genre and style recognition. Firstly variants of CNN and CNN-LSTM based models are trained on a variety of spectral and rhythmic features. Secondly, a MLP network is trained on the one dimensional averaged features & the extracted representational features from a transfer learning system trained for music tagging. Finally, these models are combined in a majority voting ensemble setup. With our experiments we showed that the ensemble model is effective in greatly improving the

TABLE VI: Comparative results with other state-of-the-art systems.

Models	Accuracy GTZAN	Accuracy Ballroom
Comparison with state-of-the-art systems		
Choi et al. [18]	89.8	-
Arabi and Lu [31]	90.8	-
Panagakis et al. [4]	91.0	-
Panagakis et al. [5]	93.7	-
Klapuri et al. [32]	-	91.0
Marchand and Peeters [33]	-	93.1
Proposed System	**94.2**	**93.8**

performance. Our proposed model outperforms the current state-of-the-art systems and achieves a near perfect score for musical genre recognition in the GTZAN dataset and musical style recognition in Ballroom dataset.

REFERENCES

[1] 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.

[2] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 5. IEEE, 2003, pp. V–429.

[3] Y. M. Costa, L. S. Oliveira, A. L. Koericb, and F. Gouyon, "Music genre recognition using spectrograms," in *Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on*. IEEE, 2011, pp. 1–4.

[4] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Music genre classification via sparse representations of auditory temporal modulations," in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 1–5.

[5] Y. Panagakis and C. Kotropoulos, "Music genre classification via topology preserving non-negative tensor factorization and sparse representations," in *Acoustics speech and signal processing (ICASSP), 2010 IEEE international conference on*. IEEE, 2010, pp. 249–252.

[6] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic genre classification of music content: a survey," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 133–141, 2006.

[7] B. L. Sturm, "A survey of evaluation in music genre recognition," in *International Workshop on Adaptive Multimedia Retrieval*. Springer, 2012, pp. 29–66.

[8] T. Bertin-Mahieux and D. P. Ellis, "The million song dataset."

[9] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "Evaluating rhythmic descriptors for musical genre classification."

[10] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *Readings in speech recognition*. Elsevier, 1990, pp. 65–74.

[11] M. Müller, F. Kurth, and M. Clausen, "Audio matching via chroma-based statistical features."

[12] F. Kurth and M. Muller, "Efficient index-based audio matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 382–395, 2008.

[13] M. Müller and S. Ewert, "Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), 2011. hal-00727791, version 2-22 Oct 2012*. Citeseer.

[14] J. C. Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[15] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*. ACM, 2006, pp. 21–26.

[16] P. Grosche, M. Müller, and F. Kurth, "Cyclic tempograma mid-level tempo representation for musicsignals," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5522–5525.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[18] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," *arXiv preprint arXiv:1703.09179*, 2017.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[23] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[24] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Advances in neural information processing systems*, 2001, pp. 402–408.

[31] A. F. Arabi and G. Lu, "Enhanced polyphonic music genre classification using high level features," in *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*. IEEE, 2009, pp. 101–106.

[32] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, 2006.

[33] U. Marchand and G. Peeters, "The Modulation Scale Spectrum And Its Application To Rhythm-Content Analysis,"

in *DAFX (Digital Audio Effects)*, Erlangen, Germany, Sep. 2014, cote interne IRCAM: Marchand14a. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01161080

[34] B. L. Sturm, "The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use," *arXiv preprint arXiv:1306.1461*, 2013.