Latent Space Shenanigans

Daniel Richards Ravi Arputharaj
KTH Royal Institute of Technology
drra@kth.se

Adhithyan Kalaivanan
KTH Royal Institute of Technology
adhkal@kth.se

1 Introduction

With deep learning based models continuing to be the state of the art for automatic music tagging, there is wide spread adoption of these methods in the entertainment industry. A prominent use case is in recommender systems where music considered to be *similar* to what users like, gets recommended or promoted. This naturally raises the question if these models learn sufficiently expressive representations to solve this task. And more importantly, the errors made by these models on downstream tasks is worth investigating for systemic bias. With recommender systems repeatedly been shown to possess feedback loops and bias amplification [1] [2], this can inflict disproportionate harm on some artists and music genre.

With this as the motivation, we investigate if popular music tagging models learn latent representation of music expressive enough to perform unsupervised clustering by genre. We ask if some model architectures are better suited than others and analyse what kind of errors these models make. We then conduct experiments which are aimed to provide a better understanding of their latent representation space, and test their generalization capability to music well beyond the training domain.

2 Methods

To examine the representation quality and understand the latent space of deep learning models, we use pretrained music taggers [3]. To cover a broad range of model architectures, we chose to evaluate Fully Convolutional Network [4], Musicnn [5], Convolutional Recurrent Neural Network [6], Self-attention based Network [7], Harmonic CNN [8], Sample-level CNN [9] and Sample-level CNN with Squeeze and Excitation layers [10] which are all trained on the MTG-Jamendo dataset [11]. The last two models operate on the waveform directly while the rest extract a spectrogram of the music and then perform feature extraction. All our experiments are performed using music files from the GTZAN dataset [12].

3 Experiments and Results

3.1 Clustering and Error Analysis

We resample all the GTZAN music files at 16kHz and perform inference with our set of models. Our aim is to use the 256-dimensional last hidden layer as the latent representation of a piece of music. In cases where the model input size is less than the length of a song, i.e., 30s, we use mean embedding of song sections with window size as model input size and no overlap between windows. Before performing any clustering and evaluation, we visually inspect the two dimensional t-SNE projection [13] of the embeddings. From figure 1, we can be hopeful that embeddings do cluster around their genre.

The next task would be to investigate how many natural clusters do these embeddings form, i.e., before we inject our prior knowledge that there are 10 genre in the dataset. We choose Gaussian Mixture Model to assign clusters of various sizes. Then we compute the silhouette coefficient [14],

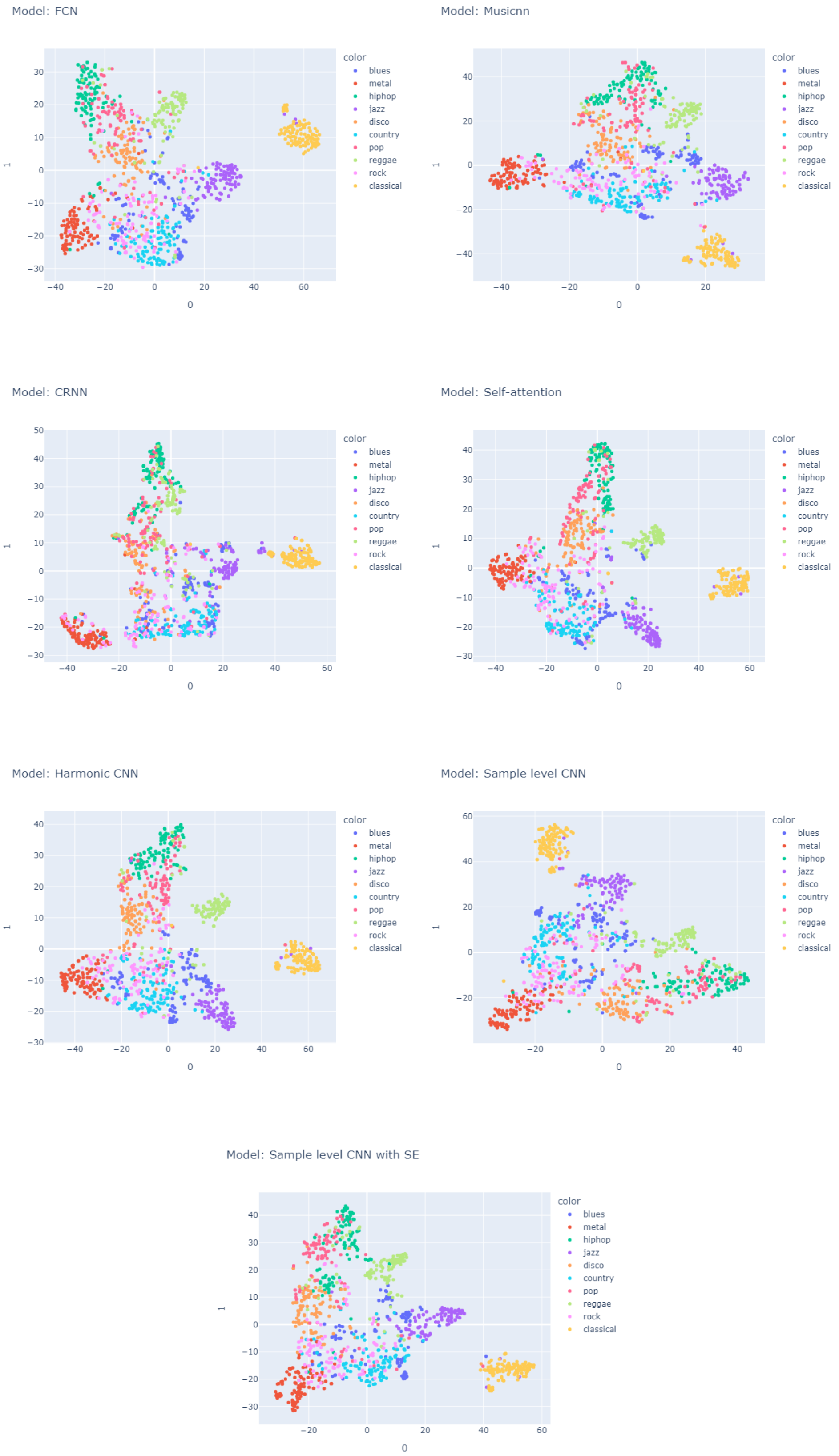


Figure 1: t-SNE projection of music embedding

Calinski-Harabasz index [15] and Davies-Bouldin index [16] for all our results. These scores are used to evaluate the cluster quality by measuring how well defined and separated the different clusters are.

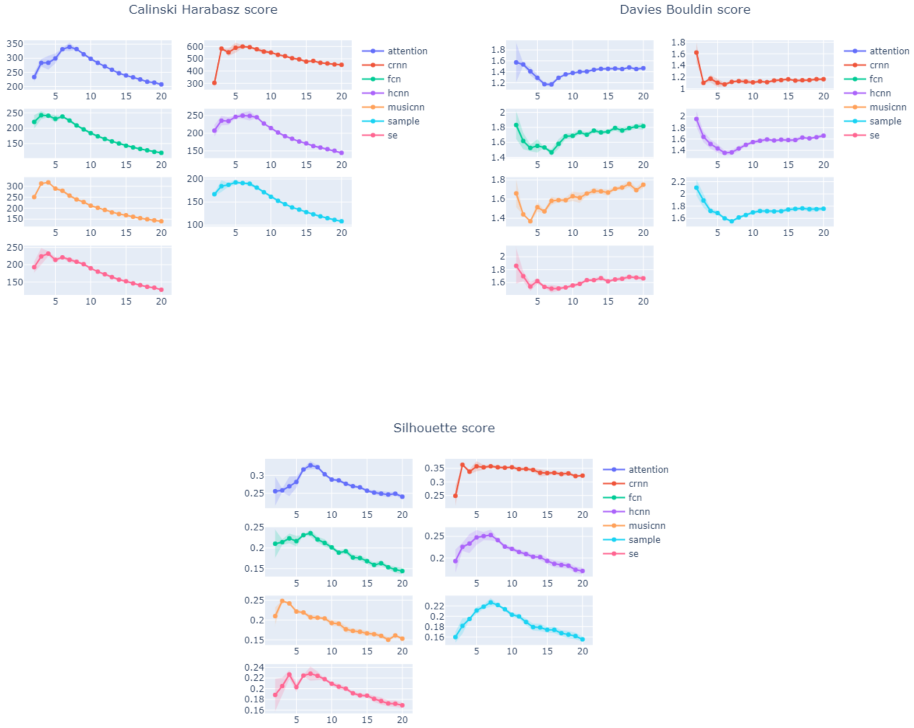


Figure 2: Clustering scores plot

We observe that for most models the metrics indicate the best number of clusters to be seven. We cluster the embedding from different models using both K-Means and GMM into seven or ten clusters. Then we treat the assignment of cluster labels as a linear sum assignment problem and solve it using the Hungarian algorithm [17]. We present the resulting classification accuracy in table below.

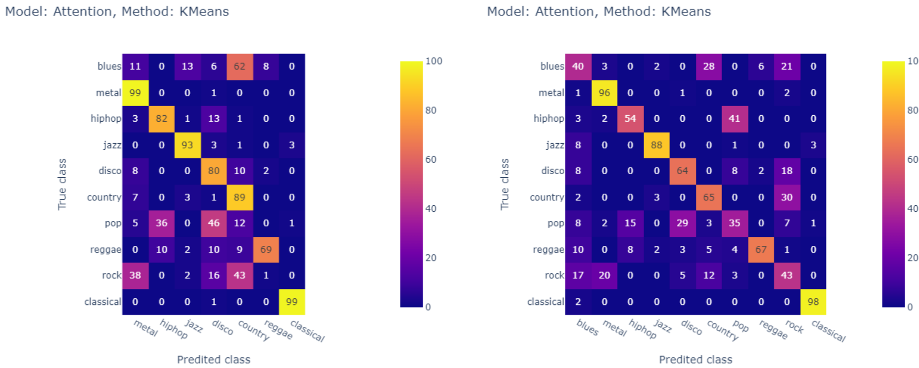
Table 1: Classification accuracy after label assignment on clusters

# Clusters	Model	K-Means	GMM
7	FCN	0.611 (0.0003)	0.593 (0.0293)
	Musicnn	0.547 (0.0054)	0.535 (0.0132)
	CRNN	0.500 (0.0029)	0.505 (0.0163)
	Attention	**0.612 (0.0008)**	**0.601 (0.0199)**
	HCNN	0.598 (0.0022)	0.596 (0.0182)
	Sample level	0.587 (0.0021)	0.588 (0.0020)
	Sample level SE	0.586 (0.0010)	0.560 (0.0285)
10	FCN	0.624 (0.0137)	0.612 (0.0242)
	Musicnn	0.589 (0.0034)	0.570 (0.0216)
	CRNN	0.529 (0.0037)	0.539 (0.0160)
	Attention	**0.652 (0.0035)**	**0.623 (0.0221)**
	HCNN	0.600 (0.0152)	0.616 (0.0295)
	Sample level	0.587 (0.0112)	0.585 (0.0213)
	Sample level SE	0.620 (0.0064)	0.594 (0.0344)

3.1.1 Error Analysis

It is interesting to analyse the naturally formed clusters as most models produce seven of them. We inspect the misclassifications from the confusion matrix 3 to explicate genre relations. We use the geneology of Western music to explain these confusions with the help of Musicmap [18]. By observing that blues influences almost all the genres except classical, we are able to explain the large spread in the corresponding cluster 1. Also the categorisation of disco as a sub-genre of pop is well illuminated. The majority of the confusion is thus explainable through the eyes of music genealogy.

However, this cannot explain the presence of few rogue points that lie in the heart of clusters of other genre. We inspect these auditorialy and present two interesting cases: jazz.00001.wav which as observed by Sturm [19] is an orchestral piece, reggae.00010.wav which is often conflated as reggae, but as noted by Henke [20] is not.



	metal	hiphop	jazz	disco	country	reggae	classical
blues	11	0	13	6	62	8	0
metal	99	0	0	1	0	0	0
hiphop	3	82	1	13	1	0	0
jazz	0	0	93	3	1	0	3
disco	8	0	0	80	10	2	0
country	7	0	3	1	89	0	0
pop	5	36	0	46	12	0	1
reggae	0	10	2	10	9	69	0
rock	38	0	2	16	43	1	0
classical	0	0	0	1	0	0	99

	blues	metal	hiphop	jazz	disco	country	pop	reggae	rock	classical
blues	40	3	0	2	0	28	0	6	21	0
metal	1	96	0	0	1	0	0	0	2	0
hiphop	3	2	54	0	0	0	41	0	0	0
jazz	8	0	0	88	0	0	1	0	0	3
disco	8	0	0	0	64	0	8	2	18	0
country	2	0	0	3	0	65	0	0	30	0
pop	8	2	15	0	29	3	35	0	7	1
reggae	10	0	8	2	3	5	4	67	1	0
rock	17	20	0	0	5	12	3	0	43	0
classical	2	0	0	0	0	0	0	0	0	98

Figure 3: Confusion matrix

3.2 Model Interpretability

To gain insight into the model and its latent representation space, we resort to methods from interpretability research. The first method aims to find a simple transformation between a song embedding and pitch or tempo shifted version's embedding. Here the goal is to find if there are identifiable directions in the latent space that correspond to human interpretable features like pitch and tempo. This approach is heavily inspired by Word2Vec [21] where the word embedding where shown to obey simple arithmetic operations, e.g., king - man $\approx$ queen - woman. We then turn to a DeepDream [22] like method which aim to perturb the input such that a hidden unit or layer is maximally activated. But instead of images we perturb the music samples, and instead of maximizing a layer activation we minimize the mean square difference between the embedding and mean embedding of a genre. This would correspond to finding an input which according to the model is most representative of a genre.

3.2.1 Pitch and Tempo Transforms

To find simple transformations in the latent space which corresponds to shift in pitch and tempo, we modify the GTZAN dataset and generate two versions - one 50% higher in tempo with no change in pitch and another 50% higher in pitch with no change in tempo with respect to the original. After inference using Harmonic CNN, we assume that pitch or tempo shifted embedding are just a linear transformation of the original embedding, i.e., $\mathbf{X}_{+} = \mathbf{A}\mathbf{X} + \mathbf{b}$. We consider an increasingly complex set of transforms from translation to affine and estimate parameters that minimizes the mean squared error between transformed input and the known pitch or tempo shifted versions. The quality of fit is determined through the coefficient of determination R^2. To get a more tangible measure, we also evaluate how often the nearest neighbor of a transformed input is the corresponding pitch or tempo shifted version. The following table summarises the results.

We observe the quality of all of the above transforms are poor, thus disproving our assumption. We believe this is because the objective of music tagging models only encourages construction of a latent

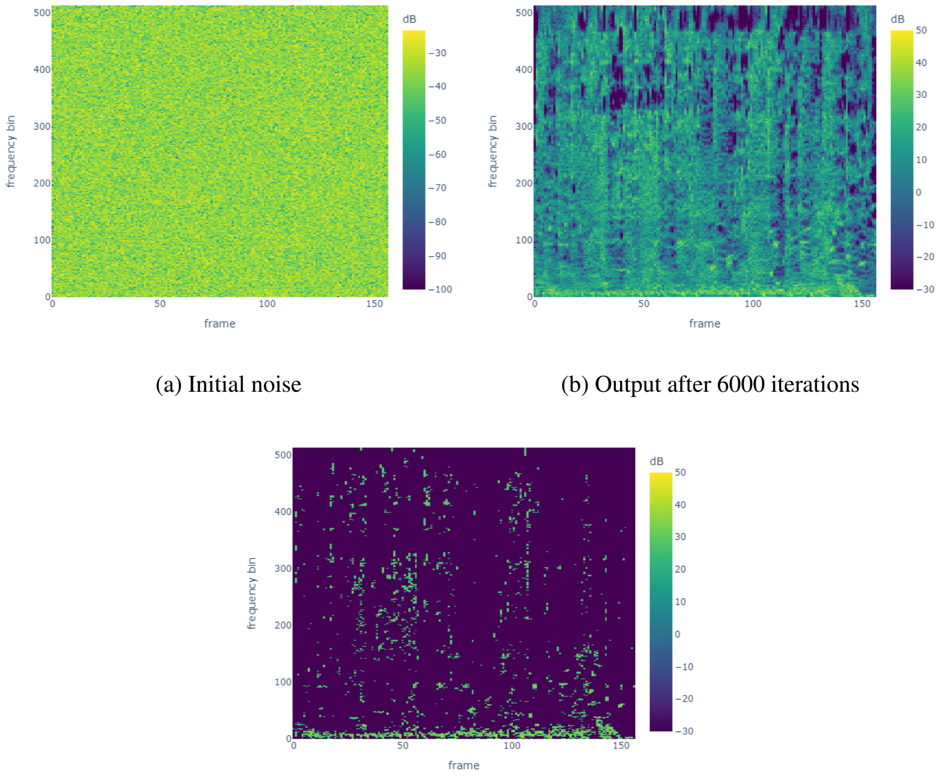
Table 2: Different transformation and their corresponding constraints

Transforms	Constraint	MSE	NN accuracy
Translation	$\mathbf{A} = \mathbb{I}$	6.9039	4.8%
Uniform scaling	$\mathbf{A} = r$	5.1168	2.2%
Non-uniform scaling	$\mathbf{A} = \mathbf{r}_{256\times 1}$	5.0356	2.8%
Scaled orthogonal	$\mathbf{A}$ is orthogonal, $\mathbf{b} = 0$	6.3111	4.8%
Affine	None	8.7669	2.2%

space that captures genre information which are part of the tags and not necessarily disentangled tempo or pitch information.

3.2.2 Deep Dreaming Music

To generate the input which is most representative of a genre, we first obtain the mean embedding of all songs from a genre. We fix the weights of our Harmonic CNN model, and let the input be variable. Starting with Gaussian noise as input, we define the objective function as the mean squared error between the obtained model embedding and the mean genre embedding. Then by computing the gradient of the objective with respective to the input, we gradually modify the input in order to minimize the objective through gradient descent. A spectrogram plot of the final song for the disco genre after 10000 gradient steps is shown in figure 4. A naive thresholding of the energy is used to filter our sample, to make it less harsh to listen.



(a) Initial noise

(b) Output after 6000 iterations

(c) Applying thershold

Figure 4: Spectrogram plot for the disco genre using HCNN model

We note that the song does not resemble anything a human would classify as disco music. But this doesn't come as a complete surprise, given the *disco-est* song according to the music has no reason to be intelligible to humans.

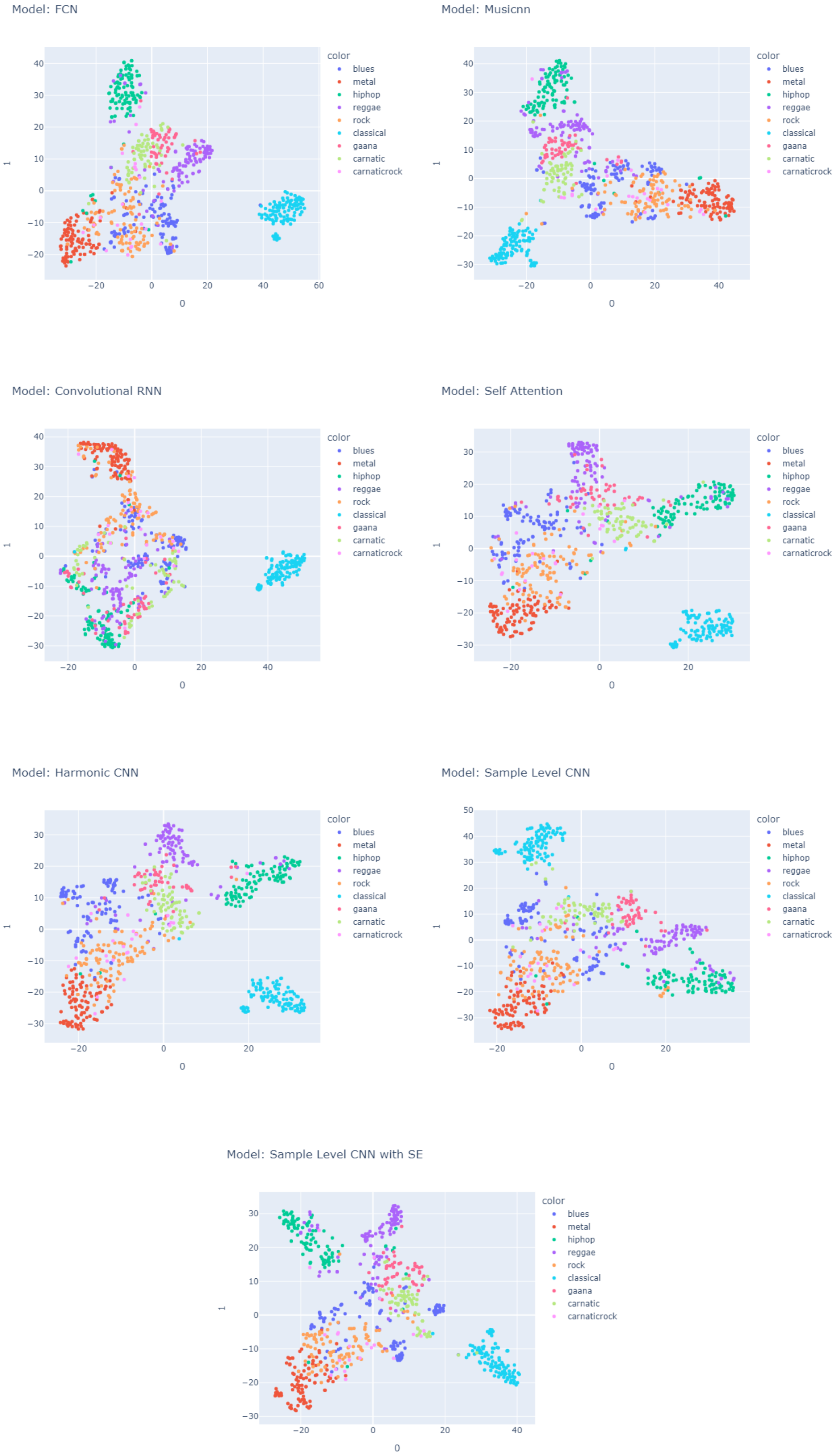
3.3 Performance on Non-Western Music

We inspect if the model is able to provide expressive representation of music that is out of its training domain, which is mostly Western music. We choose two disparate South Indian music genre - Carnatic and Gaana, which feature musical instruments that are unique to them. We manually scrape 30s clips of 50 songs from Carnatic and Gaana each. We also include clips of 23 songs from Carnatic-Rock which as the name suggests is a fusion between Carnatic and Rock. We infer these with our models and visually inspect the two-dimensional t-SNE projections of the embedding 5.

We observe that the Carnatic and Gaana music embedding form surprisingly well defined clusters. Moreover their placement in the embedding space is also sensible with Gaana being close to Reggae and Hip-Hop, given its rap-like resemblance. Carnatic-Rock also gets placed in the space spanned by Rock and Carnatic clusters as we would intuitively expect.

4 Summary and Conclusions

We conclude that the music taggers are able to create a meaningful representation which surprisingly brings out the relationship between genres as observed in the genealogy of music. Further, we observe that majority of the confusion arises solely because of aleatoric uncertainty. However, these models fall short of explaining other characteristics that are not explicitly encoded in the training objective. We believe that having a multi-task training objective might be a better fit for such purposes.



Model: FCN

Model: Musicnn

Model: Convolutional RNN

Model: Self Attention

Model: Harmonic CNN

Model: Sample Level CNN

Model: Sample Level CNN with SE

Figure 5: t-SNE projection of Non-Western music embedding

References

[1] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 2145–2148, 2020.

[2] Òscar Celma and Pedro Cano. From hits to niches? or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 1–8, 2008.

[3] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of cnn-based automatic music tagging models. In *Proc. of 17th Sound and Music Computing*, 2020.

[4] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

[5] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. *arXiv preprint arXiv:1711.02520*, 2017.

[6] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 2392–2396. IEEE, 2017.

[7] Minz Won, Sanghyuk Chun, and Xavier Serra. Toward interpretable music tagging with self-attention. *arXiv preprint arXiv:1906.04972*, 2019.

[8] Minz Won, Sanghyuk Chun, Oriol Nieto, and Xavier Serrc. Data-driven harmonic filters for audio representation learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 536–540. IEEE, 2020.

[9] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789*, 2017.

[10] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 366–370. IEEE, 2018.

[11] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019. URL `http://hdl.handle.net/10230/42015`.

[12] George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals, 2001. URL `http://ismir2001.ismir.net/pdf/tzanetakis.pdf`.

[13] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[14] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[15] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[16] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

[17] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[18] Music map. `https://musicmap.info/`, 2022. Accessed: 2022-11-03.

[19] Bob L. Sturm. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *CoRR*, abs/1306.1461, 2013. URL `http://arxiv.org/abs/1306.1461`.

[20] James Henke. *Marley legend: an illustrated life of Bob Marley*. Chronicle Books, 2006.

[21] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[22] Inceptionism: Going deeper into neural networks. `https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`, 2015. Accessed: 2022-11-03.