# Evaluation of CNN-based Automatic Music Tags Retrieval Models

Chinmayi C. Ramakrishna[1], Dolly Gupta[2], Shraddha Gole[3]

***Abstract*— Deep learning models have been popularly used for Information Retrieval from various types of files like audio, video etc. A sub field of Music Information Retrieval is content based music tagging. Networks like CNNs, FCN and RNNs can be used to achieve content based music labels. music tagging is a multi tagging categorization problem. There is a need to compare the performance of various models implemented in this field. However, these implementations are performed under different experimental setups and conditions. We aim to consistently evaluate the several models with metrics like ROC-AUC and PR-AUC. We conduct performance testing is done on three common datasets Million Song Dataset, MagnaTagATune and MTGJamendo. We also built an UI to show the predictions of music tags made by different models.**

***Keywords:*** **music tagging, CNN, RNN, ROC-AUC, PR-AUC, multi label classification**

## I. INTRODUCTION

music tagging is essential to convey information such as the message of the music, genre and instruments used in the music. These features make the task of music selection and recommendation from a large library of songs easier. Often users search for keywords associated to the music. Retrieval of music through such the keywords is possible through automatic music tagging models. The key step and the challenge is to pick a few keywords from the music that is relevant in its tagging. A particular keyword from the song can represent a chunk of the song. It can convey a collective message from the song. Therefore, two levels of training in a music tagging system is possible: song level and chunk level. There are few music related deformations that can affect the performance of music tagging if not addressed properly. A time stretched track needs extraction of tempo annotations through scaling appropriately. Similarly, pitch-shifting[1] a track should pose transpositions of annotated frequency curves. CNN models justify well its use in automatic tagging. CNN models follow a hierarchical features over multi layers which is similar to music features like chords, beats and pitch. The properties offered by CNNs(Convolutional Neural Networks) such as translation, distortion, and local invariances are helpful in learning musical features. RNNs (Recurrent Neural Networks) through their feed forward neural architecture can handle arbitrary length data. RNNs can also learn patterns to recognise in similar music content.

The researchers of various models have performed their implementation using different software tools, setup environments and variables. Therefore, it is difficult to perform a uniform performance analysis of these models. We implement seven models in a consistent experimental environment and evaluate on metrics (ROC-AUC and PR-AUC) to have insights into the results. The results have been reported on these state-of-the-art music tagging models using three common datasets: Million Song Dataset, MagnaTagATune and MTGJamendo. Additionally, the start-of-the-art model have been tested for their robustness and generalisation capabilites as the audio clips consist perturbed inputs.

Evaluation has been done on 7 models namely: Self-attention [3], CRNN [1], FCN [2], Harmonic CNN[4], Music NN [5], Sample-level CNN [6] and Short Chunk CNN [8], .

## II. LITERATURE SURVEY

The paper [10] uses a hybrid CNN network that consists of two convolutional layers and four fully connected layers. An optimal value for pooling size has been determined. Pooling is explored on frequency since because it is dependant on the rate of input sampling. Brian McFee et al. [9] built a software framework to simulate custom input deformations. One technique is to use a deformation object that turns a single annotated audio example into a sequence of deformed audio samples.

A recent paper on Codified Audio Language Modeling [11] uses a key detection mechanism to predict the pitch class and scale for the underlying key of a song. A deep content user embedding[12] uses two modalities: user-item interaction and audio. These two modalities are jointly trained into a single model and thus embeds it into a hybrid Deep Learning model. This is evaluated for music recommendation and music auto tagging. Edith Law et al. [13] investigate the use of a game called TagATune to use human computation to collect evaluations of algorithm-generated music tags. Fischer's ANOVA determines if the performance differences between algorithms are statistically significant. On the test set, a performance score is computed for each algorithm. This measure of performance is based on how many unique players correctly judged that the clips were the same as those emitted by the TagATune algorithm. The paper on down beat tracking [14] uses two recurrent neural networks at the front end. The first includes modelling rhythmic content on multiple frequency bands. The second one models the rhytmic content of the music. A dynamic Bayesian network takes in output activations as input. This acts as rhythmical language model. F-measure is used to evaluate the downbeat tracking.

A music transfer learning is approached for music classification and regression tasks in [1]. Clips are taken as parts from music sound tracks with different labels. Self Attention model [3] used two input audio types. The evaluations were conducted on two metrics:Area Under Precision Recall curve (AUPR) along with conventional Area Under Receiver Operating Characteristic curve (AUROC).

## III. METHODOLOGY

In this section, we describe the complete methodology of our work along the knowledge of datasets used, the architectures of all models implemented and about evaluation metrics used for performance evaluation of all the models. We have used Essentia library which will load the audio files and Librosa library which extracts the Mel spectrograms [3]. All the audios are re-sampled sample rate of 16 kHz. For the fair comparison, we implemented all the models with equal number of Mel bands. For the final prediction an average over the prediction over 16 chunks was taken.

### A. *Automatic music tagging: A Multi Instance Problem*

Music is characterized semantically by information such as moods, decades, instruments, languages, genres and subgenres. These characteristics can be expressed as music tags. For a music, these tags can be predicted by Automatic content-based music tagging using its acoustic characteristics. But, it is not always the case that the tag correctly defines the relevant characteristics of the whole music audio. For instance, a song with a tag male vocal this does not necessarily have the male voice in every section of the song. As a conclusion, we can say that automatic music-tagging is a is a multiple instance problem [1]. Many acoustic characteristics (instances) can exist for a song, but mostly only a few characteristics are useful for predicting the corresponding tag. To handle this issue of multiple instance problem in music tagging, two approaches are used. The first one is training the model on entire chunk of songs and generate predictions on song-level which is often referred as song -level training. Second Approach is Chunk-level training which includes training the model on the short chunks of the audio. All the predictions based on each chunk are later aggregated during evaluation phase by using methods like majority voting, averaging etc. We have used both these methods of training for various models.

### B. *Datasets*

*1) **MagnaTagATune (MTAT)**:* It is one of the benchmark datasets which is used for automatic music tagging systems. The dataset is formed using 25,863 audio clips. The duration of each is 29 seconds. This audio dataset consists of 188 tags. The MP3 format is used for the audio. There are total 16 folders in the dataset. Among all the tags, only 50 most frequent tags are used for music tagging. We used same split of the dataset in our project but authors of some previous work discarded some of the audio sections which does not contribute to any of the tags. And hence it resulted in better performance of their models.

*2) **Million Song Dataset (MSD)**:* Ten lakh songs' audio characteristics are included in this dataset. This is one of the most extensively used benchmark datasets for the problem of music tagging. It consists of total 241,904 music fragments in all. The audio clips are of different qualities which are encoded as MP3. For our study, we used the same split of dataset that all other authors have used. We used top 50 frequently occurring tags. Unfortunately, in some of the previous studies authors used different dataset split and compared with other models which gave lower performance.

*3) **MTG-Jamendo Dataset**:* This dataset consists of music instances of more than 50 thousands songs. The songs are in Mp3 format. The audios are of duration more than 25 seconds. Hence all the audios in this dataset are large. MTAT and MSD are used for encoding these large audios. There are 692 tags in the dataset and it cover genres, moods, instrumentation and themes. There are multiple splits provided for training, validation and testing data. We have used split0 for this work and considered top 50 most frequent tags.

### C. *Models*

*1) **Fully Convolutional Network (FCN)**:* A fully convolutional network (FCN) is a variation of Convolutional Neural Network which consists of only convolutional layers and does not contain any fully connected layers. When FCN is used for music tagging, it requires input in form of Mel spectrograms. During preprocessing of the dataset, the 29.1s audio segment is converted to a 96 x 1366 Mel spectrogram. After that it is goes through 4 convolutional layers. There are 3 x 3 2D filters in each convolutional layer which is succeeded by a max pooling layer. FCN is trained in song-level in the original paper, but for the MTG-Jamendo Dataset which contains long audio segments, chunk level training is used.

*2) **Musicnn**:* The input is this model is passed in the form of Mel spectrograms as FCN. Musicnn's architecture design decisions are based on some intuition obtained from the music domain. The timbral and temporal features extracted from vertical and horizontal filters are concatenated in the channel. After that, the subsequent one dimensional convolutional layers summarise them in order to forecast relevant labels. During training, the Musicnn model follows chunk level training i.e. it uses only short audio instances of 3s while FCN follows song-wise training.

*3) **Sample-level CNN**:* Sample-level CNN takes on the challenge of automated song labelling from end to end manner. It takes inputs in the form of raw audio wave-forms. This model is straightforward than the approaches which use Mel spectrograms as inputs. This model consists of 10 1D convolutional layers with 1 x 3 filters and 1 x 3 max-pooling layers. Trained front-end filters replicate the process of generating Mel spectrograms, while back-end convolution layers summarise the results. We performed some variations in this model by adding squeeze-and-excitation (SE) blocks.

*4) **Convolutional Recurrent Neural Network (CRNN)**:* This model is a combination of CNNs and RNNs [1]. This also uses Mel spectrogram inputs. The local features

are extracted by the front end of CNN and then back end of RNN summarizes them temporally. Because of the flexible nature of RNNs, these are used for epitomizing progressive information. RNNs can be beneficial to use to prognosticate labels which may be influenced by global structures like moods or themes. The front end consists of four convolutional layers with 3 x 3 2D filters and the backend contains two-layer RNNs with GRU. This model is trained with Long music instances of 29.1s as input.

*5) **Self-attention**:* The self-attention-based music tagging model follows logic of CRNN. It uses CNNs to extract local characteristics and series models for summarizing them. There is only one difference between self-attention models and CRNNs. This model is trained on chunk-level with the 15s-long audio instances.

*6) **Harmonic CNN**:* Harmonic CNN uses harmonically piled time-frequency expression data as input and learnable filters. Filters that can be trained provide the model more flexibility. There are 128 trainable frequency bands and 6 harmonics for stacking. Inputs with 5 seconds duration are used for chunk-level training.

*7) **Short-chunk CNN**:* It is seen earlier research that a basic 2D Convolutional Network consists of 3 x 3 filters can give exceptionally good result when it is trained at chunk-level. It is a vgg-like network but. This model consists of 7 convolutional layers along with one fully-connected layer and has residual blocks (skip connections). The size of max-pooling layer is 2 x 2 which is smaller as compared to FCN model. The input segments are shorter than the song-level inputs since chunk-level training is applied. 7 max-pooling layers are used to summarise 128 Mel bins into a single dimension (27 = 128). The audio inputs of this CNN model is of duration 3.69s, hence this model is called short-chunk CNN.

### *D. Evaluation Metrics*

To perform a our robustness study about the performance of different models , we used several evaluation metrics. All the metrics are helpful to get insights about different automated music-tagging models and then compare them. All the evaluation metrics are listed below.

*1) **ROC-AUC**:* We used the area under the the receiver operating characteristic curve which is a popular statistic for evaluating binary classification issues. It's a graph that depicts the compromise between the true positive rate and the false positive rate. True positive rate and false positive rate are calculated for each threshold and shown on a single chart.

*2) **PR-AUC**:* To plot the Precision-Recall curve, we use PR-AUC curve. This curve combines accuracy and recall in a single depiction, giving a clear image of both.

*3) **Average-Precision (AP)**:* This is the metric which is formed by the combination of precision and recall. When all the relevant documents are retrieved for any information need, their precision scores are calculated and the averaged. This total score is called average precision.

*4) **Matthews Correlation Coefficient (MCC)**:* It is used as a measure of the quality of binary (two-class) classifications. It is a more reliable statistical rate that only yields a high score if the prediction performed well in all four confusion matrix categories (true positives, false negatives, true negatives, and false positives), according to the number of positive and negative items in the dataset.

### *E. Robustness Testing: Input Deformations*

To further investigate the performance and test the robustness of our models, we performed robustness studies. We did it check if models are sensitive against small deformations in input or not. Ideally, any pre-trained model should have good generalization capabilities. We did no deformation in the dataset while training. But during testing, we applied some deformations in the audio. We applied the same 4 types of deformations to the test dataset as in paper [9]. Range of all the deformations was also kept same. Our intention was to determine the generalization abilities of our implemented models. We used existing MUDA Framework to apply all the deformations.

## IV. RESULTS

We implemented the existing models and compared their performance with two more evaluation metric Average Precision and Matthews Correlation Coefficient in addition to ROC-AUC and PR-AUC used in the base paper. We observed that Musicnn, Self-attention, Harmonic CNN and short chunk CNN performed better in comparison to the ones trained with longer audio segments like FCN and CRNN. Among all the models, Short chunk CNN and Harmonic CNN performs comparatively better. Musicnn, on the other hand, shows promising result in MTAT, but sample level and self attention models outperforms it on larger datasets MTG-Jamendo and MSD. From this we can infer that domain knowledge can be more useful in smaller datasets. In case of sequential models, self-attention models outperforms CRNN. For the robustness check, we fed the models with perturbed inputs. Harmonic CNN outperformed all the models in terms of generalising the deformed inputs, except for the white noise addition. We also designed a UI for experimenting with different audio files and comparing the tags generated by different models. In terms of the evaluation metrics used, both Average Precision and Matthews Correlation Coefficient gave similar trends as ROC-AUC and PR-AUC.

The ROC AUC curves for all the 8 models have been plotted as shown in Fig 3.

The PR AUC curves for all the 8 models have been plotted as shown in Fig 4.

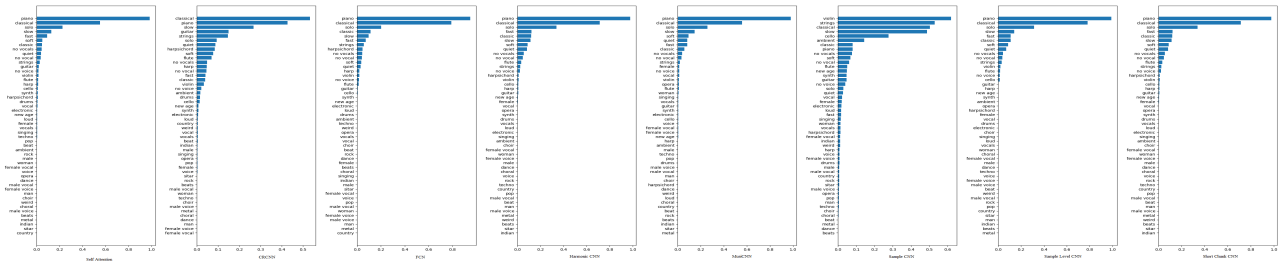

Fig. 1: Music Tags as predicted by the models

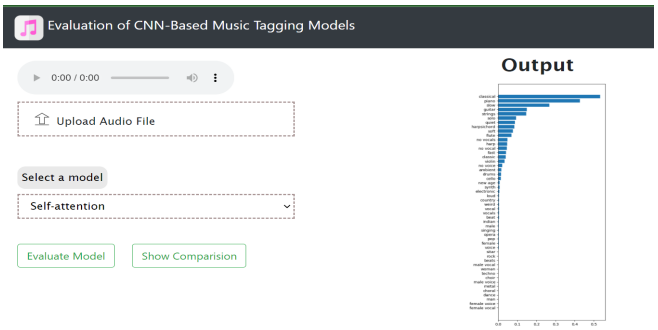


Fig. 2: Screenshot of UI of Music tags prediction using different models

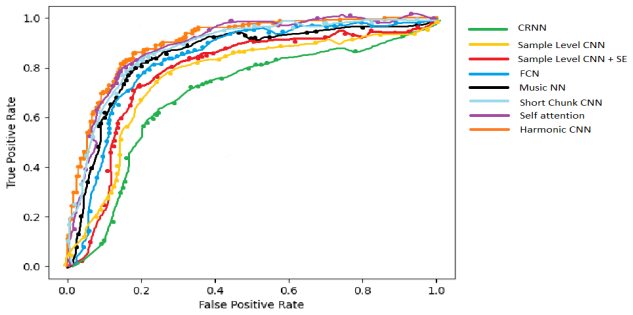


Fig. 3: ROC curves for different models

TABLE I: Performance of models on MTAT dataset

| Model | ROC-AUC | PR-AUC | AP | MCC |
|---|---|---|---|---|
| FCN | 0.9101 | 0.4129 | 0.4221 | 0.8451 |
| MusicNN | 0.9134 | 0.4519 | 0.4617 | 0.8814 |
| Sample-level | 0.9103 | 0.4410 | 0.4534 | 0.8510 |
| Sample-level + SE | 0.9110 | 0.4601 | 0.4790 | 0.8543 |
| CRNN | 0.8899 | 0.3512 | 0.3611 | 0.8371 |
| Self -attention | 0.9098 | 0.4512 | 0.4600 | 0.8734 |
| Harmonic CNN | 0.9156 | 0.4611 | 0.4790 | 0.8991 |
| Short-chunk CNN | 0.9167 | 0.4799 | 0.4876 | 0.8893 |

TABLE II: Performance of models on MSD dataset

| Model | ROC-AUC | PR-AUC | AP | MCC |
|---|---|---|---|---|
| FCN | 0.8712 | 0.3190 | 0.3299 | 0.8350 |
| MusicNN | 0.9021 | 0.2901 | 0.3030 | 0.8789 |
| Sample-level | 0.8911 | 0.3010 | 0.3018 | 0.8421 |
| Sample-level + SE | 0.8923 | 0.2910 | 0.3101 | 0.8400 |
| CRNN | 0.8899 | 0.3112 | 0.3198 | 0.8298 |
| Self -attention | 0.9098 | 0.4113 | 0.4289 | 0.8683 |
| Harmonic CNN | 0.9156 | 0.4314 | 0.4521 | 0.8845 |
| Short-chunk CNN | 0.9157 | 0.4119 | 0.4478 | 0.8811 |

TABLE III: Performance of models on MTG dataset

| Model | ROC-AUC | PR-AUC | AP | MCC |
|---|---|---|---|---|
| FCN | 0.8613 | 0.3098 | 0.3198 | 0.8211 |
| MusicNN | 0.8521 | 0.2791 | 0.2991 | 0.8701 |
| Sample-level | 0.8788 | 0.2999 | 0.3001 | 0.8291 |
| Sample-level + SE | 0.8823 | 0.2888 | 0.2965 | 0.8390 |
| CRNN | 0.8839 | 0.2612 | 0.2790 | 0.8298 |
| Self -attention | 0.9098 | 0.3213 | 0.3391 | 0.8639 |
| Harmonic CNN | 0.9156 | 0.3294 | 0.3411 | 0.8789 |
| Short-chunk CNN | 0.9167 | 0.3211 | 0.3111 | 0.8778 |

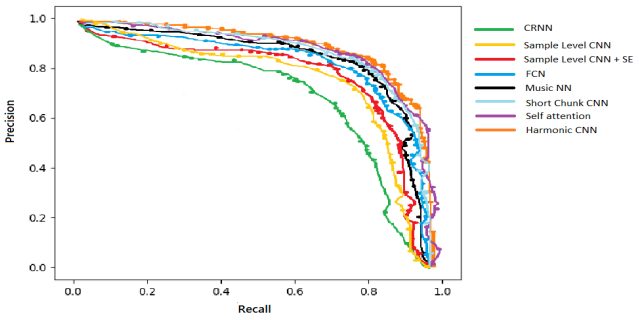


Fig. 4: PR curves for different models

## V. CONCLUSIONS

In this paper, we implemented various music tagging models. From the results obtained, we can infer that short chunk based training model ways gave better results than models which used relatively longer input segments. Short chunk CNN performed better with input without any deformation. In general, Harmonic CNN reports better results with both original and perturbed input. As part of future scope, we can try to come up with newer evaluation metric which can also account for the performance of models under deformed inputs.

## REFERENCES

[1] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 2392–2396.

[2] K. Choi, G. Fazekas, and M. Sandler, “Automatic tagging using deep convolutional neural networks,” In Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR), 2016.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2018.

[4] M. Won, S. Chun, , O. Nieto, and X. Serra, “Datadriven harmonic filters for audio representation learning,” In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

[5] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, “End-to-end learning for music audio tagging at scale,” In Proc. of the 19th International Society for Music Information Retrieval Conference (ISMIR), 2018.

[6] J. Lee, J. Park, K. L. Kim, and J. Nam, “Sample-level deep convolutional neural networks for music autotagging using raw waveforms,” In Proc. of the 14th Sound and music computing (SMC), 2017.

[7] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. of the IEEE conference on computer vision and pattern recognition (CVPR), 2018, pp. 7132–7141.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[9] B. McFee, E. J. Humphrey, and J. P. Bello, “A software framework for musical data augmentation.” in Proc. of the 16th International Society for Music Information Retrieval Conference (ISMIR), vol. 2015, 2015, pp. 248–254.

[10] T. N. Sainath, A. Mohamed, B. Kingsbury and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 8614-8618, doi: 10.1109/ICASSP.2013.6639347.

[11] Castellon, Rodrigo et al. “Codified audio language modeling learns useful representations for music information retrieval.” ArXiv abs/2107.05677 (2021): n. pag.

[12] Lee, Jongpil et al. “Deep Content-User Embedding Model for Music Recommendation.” ArXiv abs/1807.06786 (2018): n. pag.

[13] Law, Edith Ahn, Luis Dannenberg, Roger Crawford, Mike. (2007). TagATune: A Game for Music and Sound Annotation.. 361-364.

[14] Krebs, Florian et al. “Downbeat Tracking Using Beat Synchronous Features with Recurrent Neural Networks.” ISMIR (2016).

# APPENDIX

Add the first page of Plagiarism Report here, after I provide the report to you (must have less than 15% similarity). Each team member should add their signature on the report page

project_report .pdf

ORIGINALITY REPORT

| 7% | 4% | 2% | 5% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Aston University<br>Student Paper | 1% |
| 2 | Submitted to Imperial College of Science, Technology and Medicine<br>Student Paper | 1% |
| 3 | www.dellemc.com<br>Internet Source | 1% |
| 4 | docs.oracle.com<br>Internet Source | 1% |
| 5 | Submitted to CSU, San Jose State University<br>Student Paper | 1% |
| 6 | Submitted to UNITEC Institute of Technology<br>Student Paper | 1% |
| 7 | docplayer.net<br>Internet Source | <1% |
| 8 | www.i-scholar.in<br>Internet Source | <1% |