# On the Automatic Tagging of Music by Joint Time-Frequency Scattering and 3D Convolutions

Nithin Raghavan[1]

## Abstract

*Automatic music tagging is a multi-label classification problem seeing increased use in several areas of music technology that aims to represent audio tracks with useful semantic information. To our knowledge, methods of music tagging generally do not take into account monotonic pitch changes such as portamento and glissando, due to the difficulty of representing them with an informative encoding. Recent advances in signal representations for classification, such as the joint time-frequency scattering transform (jTFST), have allowed for pitch evolution in music to be encoded without the need for learning. Such transformations represent spectro-temporal patterns in an invariant and stable manner. We propose a deep learning model that aims to take advantage of this. By utilizing the jTFST as a feature extractor, we learn higher-level attributes from the inputs with a three-dimensional convolutional neural network in order to perform automatic music tagging.*

## 1. Introduction

Automatic music tagging is a multi-label classification problem that aims to label music with useful semantic information. Machines of this nature continue to see increased use in many modern information filtering systems in music technology such as recommendation systems, music browsing interfaces, music retrieval, and more [13]. Such tags can encompass a wide variety of domains, such as instrumentation, genre, emotion and more [6]. While deep neural networks have shown promise in this area, having been implemented successfully in music service platforms hosted by Apple, Google, Spotify and Amazon [9], end-to-end music tagging systems have usually relied on extracting high-level information using a convolutional neural network (CNN) without taking advantage of the overall spectro-temporal patterns in the time-frequency domain. As such, they lack the ability to predict useful tags that detect musical flourishes, such as slides, glissandi and acciacature. This information could be important in many contexts, such as music education and music retrieval, and so we propose an architecture that intends to rectify this situation.

From [12], we know that the jTFST will be able to output high quality and informative spectro-temporal features that will accurately capture pitch evolution in music, such as slides, glissandi and acciacature. We thus hypothesize that 3D convolutions, especially those of the form mentioned in [7], will be able to accurately learn this feature space and capture high-level information in order to greatly increase the receptive field of the CNN. This will then improve the accuracy of automatic music tagging, and could also potentially allow for more tags to be generated in modern music recommendation and retrieval platforms that utilize this additional information.

## 2. State of the Art

Song, et al. (2020) [10] propose a method for music tagging using convolutional filters that uses the time-scattering transform introducted in Andén, et al. (2014) [3] in combination with learned convolutional filters to extract high-level features from the input raw audio. They also use a modified self-attention mechanism parameterized by a multi-layer perceptron (MLP), first introduced in [4], in order to increase the receptive field size of the convolutional layers. The self-attention enables high-frequency parts of the input raw audio to be weighted more heavily when making the final predictions, which has been empirically found to improve final prediction accuracy. However, there are issues with the time-scattering transform that make it difficult to effectively encode monotonic pitch changes [12].

Andén, et al. (2019) [2] introduced the joint time-frequency scattering transform (jTFST) as a successor to the time-scattering transform, which aims to capture the multi-scale energy distribution of an input audio signal in both the time and frequency domains jointly. This is an improvement over the time-scattering transform, which is restricted to convolutions along the time axis, and so cannot separate signals subjected to time shifts which vary in frequency. This transform is also time-shift-invariant and stable, and is able to capture large-scale temporal structure much better than mel-spectrograms.

A two-dimensional scalogram $X(t, \lambda)$ is generated from

[1] University of California, Berkeley

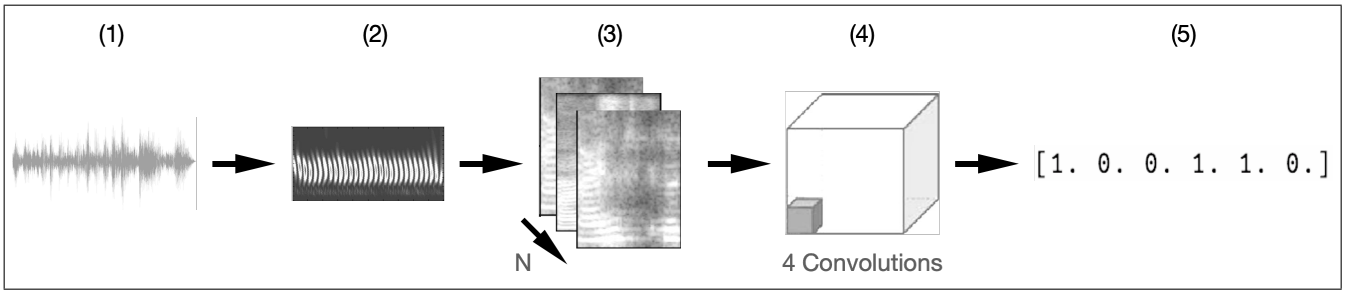


Figure 1. Our proposed architecture. The input (1) is raw audio, which is converted into a scalogram (2). The joint time-frequency scattering transform is then applied twice (one for $\theta = 1$ and $\theta = -1$) and the jTFST with the highest energy is selected, producing a three-dimensional tensor with axes of time, temporal variation and frequency variation, where $N$ is the number of frames (3). These features are then passed into a PackNet-style architecture [7] that uses 3D convolutions to preserve and recover important spatial information without information loss (4). Then, the final multi-label prediction is output from the network (5).

a wavelet transform implemented as a multirate filter bank ($X(t, \lambda) = |x(t) * \psi_\lambda(t)|$, where $x(t)$ is the audio and $\psi_\lambda(t)$ is the temporal filter bank), and is then decomposed using a wavelet transform with a two-dimensional wavelet composed as a product of one-dimensional mother wavelets in time and log-frequency space ($\psi^{(t)}(t)$ and $\psi^{(f)}(\lambda)$). They then dilate by $2^{-\mu}$ along $t$, $2^{-\ell}$ along $\lambda$ and reflect according to $\theta$ to obtain

$$\Psi_{\mu,\ell,\theta}(t, \lambda) = 2^{\mu+\ell}\psi^{(t)}(2^\mu t)\psi^{(f)}(\theta 2^\ell \lambda)$$

where $\mu, \ell \in \mathbb{R}_+$ measure the temporal and frequency variations, and $\theta \in \{-1, +1\}$ is an orientation parameter which represents whether the spectro-temporal pattern oscillates up or down. The first-order scattering coefficients are the same as in the time-scattering transform. However, in order to compute the second-order scattering coefficients, they take the modulus of the two-dimensional wavelet transform of the scalogram and convolved with a lowpass filter in time $\phi_T(t)$:

$$S_2 x = |X * \Psi_{\mu,\ell,\theta}(\cdot, \lambda)| * \phi(t)$$

This ensures time-shift invariance and time-warping stability, and also describes the time-frequency geometry of the input signal much better than [3]. Wang, et al. [12] interprets this transform from a new perspective by considering the temporal and spectral wavelet convolutions in a sequential manner. Namely, instead of using a two-dimensional mother wavelet, they use the dilated one-dimensional mother Morlet wavelets $\psi^{(t)}(t)$ and $\psi^{(f)}(\lambda)$ in the time and frequency domains, and apply a two-dimensional low pass filter $\phi_{T,F}(t)$ in order to obtain the following definition for the second-order scattering coefficients:

$$S_2 x = |X * 2^\mu \psi^{(t)}(2^\mu t) * 2^\ell \psi^{(f)}(\theta 2^\ell \lambda)| * \phi_{T,F}(t)$$

This method provides more explicit information as to what exactly has been captured, and is thus a better feature extractor. We thus intend to follow this method in order to preprocess the data for music tagging.

Currently, many CNNs use striding and pooling of their convolutional layers to increase their receptive field size. However, when fine-grained representations are important, this has been shown to decrease overall accuracy. In order to rectify this, Guizilini, et al. introduces PackNet [7], which uses 3D convolutions and Space2Depth operations to compress detail-preserving representations, which if applied to a scalogram, would allow for the recovery of important spectro-temporal information. This obviates the need for traditional striding and pooling layers, and prevents information loss in the input image. 3D convolutions have been shown to be very good at learning spatio-temporal features [11]. 3D convolutions have also been shown to quickly and accurately learn to recognize overlapping sound events from multichannel features [1]. As such, we believe that using 3D convolutions additionally removes the need for using self-attention layers as well, as the 3D convolutions should be able to learn high frequencies when presented with jTFST coefficients.

We leverage the progress that has been made in these works to produce a neural network architecture designed to perform automatic music tagging using the jTFST combined with packing layers from [7].

## 3. Methodology

Our architecture consists of a wavelet convolution of the raw audio using a temporal multi-rate filterbank. This is then fed into a jTFST feature extraction section, which itself is followed by four PackNet packing layers from [7].

### 3.1. Scalogram

Let the raw audio be represented by $x(t)$. As in [2], we convolve this with a temporal wavelet filter bank and take the modulus. This results in a two-dimensional image $X(t, \lambda)$ parameterized by time and log-frequency. As in

[12], we focus on a spectro-temporal pattern smaller than a box in time and log-frequency restricted by some time scale $T$ and frequency scale $F$. These parameters will be selected in the next section.

### 3.2. jTFST for Feature Extraction

We follow [12] and fix appropriate transform parameters, such as the averaging scale $T$ and the number of filters per octave $Q_1^{(t)}$. The raw audio input is convolved with a temporal filter bank of this size in order to produce a two dimensional scalogram in terms of time and frequency. The maximum scale in this wavelet bank is $J_1^{(t)} = Q_1^{(t)} \log_2(T)$, implying that $\lambda = 1, \cdots, J_1^{(t)}$. For the temporal wavelet bank in the second-order jTFST, set $J_2^{(t)} = Q_2^{(t)} \log_2(T)$. Again as in [12], we let $Q_2^{(t)} = 12$ to properly capture accaciatura and glissando, and let $Q_2^{(t)} = 2$ (so that $\mu = 1, \cdots, J_2^{(t)}$). Repeat this for the frequency wavelet convolution to get that $J_1^{(f)} = Q_1^{(f)} \log_2(Q_1^{(t)} F)$, and set $Q_1^{(f)} = 2$ filters per octave, and $F = 2$ octaves. This further implies that $\ell = 1, \cdots, J_1^{(f)}$. The first-order and second-order scattering coefficients will be calculated in the same manner as [12], which was described in Section 2. $S_2 x$ will be normalized over $S_1 x$ to capture only the temporal variation. We calculate the jTFST over both directions $\theta = 1$ and $\theta = -1$, and select the one with maximum energy. After passing the scalogram through this section, we thus end up with a three-dimensional tensor, where the first dimension refers to the number of discretized frames and is of size $N$. We assume that each frame essentially contains stationary signal behavior. The second and third dimensions are of size $J_2^{(t)}$ and $J_1^{(f)}$, which describe the temporal and frequency variations, respectively. This tensor then contains all the features necessary to predict convolutional weights that will allow for accurate music tagging predictions.

### 3.3. 3D Convolutions

[7] introduces packing layers, which use Space2Depth to fold the spatial dimensions of convolutional feature maps into extra feature channels. 3D convolutions with a cubic kernel then leverage the tiled structure of this feature space in order to expand it into a higher dimensional feature space. This is then flattened using tensor reshaping, wherepon a standard 2D convolution is able to learn to compress it. These layers compress key spatial information, which is important especially in our use case as they also are able to preserve high resolution spectro-temporal interactions given from the jTFST. Starting with a tensor of size $N \times J_2^{(t)} \times J_1^{(f)}$, Space2Depth results in a tensor of size $4N \times \frac{J_2^{(t)}}{2} \times \frac{J_1^{(f)}}{2}$. After the 3D convolutions, the tensor is of size $D \times 4N \times \frac{J_2^{(t)}}{2} \times \frac{J_1^{(f)}}{2}$, which is then reshaped to $4DN \times \frac{J_2^{(t)}}{2} \times \frac{J_1^{(f)}}{2}$. Lastly, after a 2D convolution is applied, the tensor becomes of size $N \times \frac{J_2^{(t)}}{2} \times \frac{J_1^{(f)}}{2}$. After four of these layers are applied, a final 2D convolution is applied that maps the feature maps onto the same dimensions as the multiclass label vectors in the training data, so that they can be directly compared via MSE loss.

## 4. Experiments

There are many datasets available to potentially train this network on. These include the MagnaTagATune dataset [8] and the Million Song Dataset [5]. The results can also be directly compared with other music tagging deep neural networks such as [10] and [6].

## 5. Possible Conclusions

If the experiments are successful, then we not only know that the jTFST is better suited for music tagging than the time-scattering transform, which is what we hypothesized, but also that 3D convolutions are able to directly leverage the joint spectro-temporal space output by the jTFST in order to preserve key details and high-frequency information necessary for prediction. Possible improvements could be combining this back with the self-attention mechanism in [10] to see if that produces additional improvements, as well as potentially adding a recurrent mechanism to better capture time evolution.

If the experiments are not successful, then further ablation studies can be conducted that only attempt to test one out of the two sections (either the jTFST or the 3D convolutions). These can then be directly compared with the networks proposed in [10] and [6] in order to derive further conclusions.

## References


[1] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen. Multichannel sound event detection using 3d convolutional neural networks for learning inter-channel features, 2018. 2

[2] Joakim Anden, Vincent Lostanlen, and Stephane Mallat. Joint time–frequency scattering. *IEEE Transactions on Signal Processing*, 67(14):3704–3718, Jul 2019. 1, 2

[3] Joakim Anden and Stephane Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, Aug 2014. 1, 2

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 1

[5] Thierry Bertin-Mahieux, Daniel Ellis, Brian Whitman, and Paul Lamere. The million song dataset. pages 591–596, 01 2011. 3

[6] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks, 2016. 1, 3

[7] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3

[8] Edith Law, Kris West, Michael I. Mandel, M. Bay, and J. S. Downie. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, 2009. 3

[9] Juhan Nam, Keunwoo Choi, Jongpil Lee, Szu-Yu Chou, and Yi-Hsuan Yang. Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach. *IEEE Signal Processing Magazine*, 36(1):41–51, 2019. 1

[10] Guangxiao Song, Zhijie Wang, Fang Han, Shenyi Ding, and Xiaochun Gu. Music auto-tagging using scattering transform and convolutional neural network with self-attention. *Applied Soft Computing*, 96:106702, 2020. 1, 3

[11] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. 2

[12] Changhong Wang, Vincent Lostanlen, Emmanouil Benetos, and Elaine Chew. Playing technique recognition by joint time–frequency scattering. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 881–885, 2020. 1, 2, 3

[13] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of cnn-based automatic music tagging models, 2020. 1