Enhancing Robustness in Automatic Music Tagging: A Study on Data Augmentation and Vision Transformers

Sadab Hafiz shafiz3@gatech.edu
Robert Sharp rsharp31@gatech.edu
Georgia Institute of Technology
225 North Ave NW, Atlanta, GA 30332-0002

Abstract

In this paper, we address the challenges of automatic music tagging, a critical task within the field of Music Information Retrieval (MIR) that involves classifying audio tracks into multiple labels such as genre and mood. We improve the robustness of music tagging models against real-world audio variability and propose a new a Vision Transformer (ViT) architecture. We leverage the MagnaTagATune dataset, comprising approximately 25,000 music clips, and apply stochastic augmentations including noise addition, time stretching, and pitch shifting during training. Our experiments reveal that while traditional models benefit from these augmentations, the ViT model exhibits unique behaviors, suggesting potential overfitting challenges. Despite these issues, the ViT architecture achieves tagging performance comparable to state-of-the-art models, indicating its promise for future research in music classification. Our findings suggest that effective data augmentation can significantly enhance model resilience, paving the way for more robust applications in MIR.

1. Introduction/Background/Motivation

In the field of Music Information Retrieval (MIR), music tagging involves classifying audio tracks into various labels, such as genre, instruments, and mood. This task presents unique challenges due to the sequential and complex nature of audio data. In this paper, we explore the multi-label classification of a dataset composed of MP3 files. Our goal is to enhance the robustness of existing architectures through the application of audio augmentations. Additionally, we introduce a novel architecture that leverages Vision Transformers (ViTs) to improve tagging accuracy.

Minz Won et al. conducted a survey of the state-of-the-art automatic music tagging architectures [15]. The backbone of these music tagging models is typically a Convolutional Neural Network (CNN), and most of the differences in existing models come from variations in the CNN or the addition of another module on top of the CNN base. Early work showed promising results while using a Fully Convolutional Network (FCN) on Mel spectrogram representation of the music data. The Mel spectrogram is a data processing step that converts audio data into a visual representation. FCNs use CNNs and max-pooling layers on these image representations of the audio to extract features that are used for classification [3]. Another architecture known as Musicnn combines the idea of Mel spectrograms with some insights from music domain knowledge [11]. This architecture uses vertical and horizontal filters. The vertical filters aim to capture "pitch-invariant timbral features", while horizontal filters capture "temporal energy envelope" of the audio [15]. The extracted features are combined and followed by 1D convolution layers. Unlike FCN, which uses the entire track, Musicnn uses short 3 second audio excerpts [15].

Jongpil Lee et al. approached the tagging problem by using raw audio waveforms as input instead of relying on Mel spectrogram representation. The proposed architecture called Sample-level CNN is deeper than the previously described architectures, using ten 1D convolution layers with max pooling and dropout layer for regularization [10]. Finally, a fully connected layer is used for classifying the features learned by the CNNs. Another variant of Sample-level CNN using squeeze-and-excitation(SE) blocks was proposed [8]. SE is a mechanism designed to enhance the representational power of neural networks by adaptively recalibrating channel-wise feature responses [7]. By integrating SE blocks into the Sample-level CNN architecture, the model can effectively focus on the most relevant audio characteristics, potentially improving tagging performance. Similar to Musicnn, Sample CNN and Sample-SE CNN use short 3.69 second audio excerpts for training [15].

As Recurrent Neural Networks (RNNs) became popular due to their ability to handle sequential information, they

found their way to music tagging. Keunwoo Choi et al. introduced the CRNN model, which used CNNs to extract features and run them through RNN to temporally process the features into the final prediction [4]. In addition to Mel spectrogram preprocessing, this architecture uses Gated Recurrent Units (GRU), which are better than vanilla RNNs for processing sequential data such as music. The model uses long 29.1 second excerpts as inputs [15]. This is advantageous because RNNs can leverage temporal and global relationships in long sequence, which is not possible for models relying solely on CNNs. In the similar vein as CRNN, Minz Won et al. proposed an attention-based model which takes inspiration from the success of transformers in natural language processing [5]. Similar to CRNN, the attention based model uses CNNs on Mel spectrograms to extract features that are passed to a transformer encoder, which is a deep stack of attention-layers [14]. The attention model uses 15 second audio excerpts for training [15].

Similar to Musicnn, a proposed architecture known as Harmonic CNN (HCNN) leverages music domain knowledge. HCNN uses trainable CNN based "band-pass filters" and "harmonically stacked time-frequency representation inputs" [13]. These trainable filters learn the harmonic structure of the audio and make the model more flexible. Training is done with 5 second audio excerpts [15]. Based on the success of simple 2D CNN with 3 x 3 filters in HCNN when used with short chunk audio, the authors aimed to create a new CNN based architecture that took inspiration from VGG [15]. This model, referred to as "short-chunk CNN", uses 7-layer CNN, and a fully connected layer to extract features and classify from Mel spectrogram representation. As the name suggests, this model uses short 3.69 second audio excerpts. Two variants of this model are suggested, with and without residual connections [15].

While the approaches discussed show promising results, their robustness is a critical concern, particularly given that real-world audio data is often noisy. Research by Minz Won et al. has demonstrated that the performance of existing models diminishes when faced with perturbed audio samples [15]. To address these challenges, we applied various audio augmentations during training to enhance the resilience of our systems. Additionally, we recognize the potential of recent advancements in the field, particularly the success of Vision Transformers (ViTs) in computer vision [6]. Inspired by these developments, we propose a novel ViT-based architecture for automatic music tagging, aiming to gauge the effectiveness of ViT in audio domain.

The data augmentation we apply in our work are unsupervised, requiring no additional labeling. Our success would show a cost-effective way of improving robustness of automatic music tagging models. Furthermore, the performance of our ViT model may spark an interest in the application of ViTs in the domain of Music Information Retrieval, creating more avenues for research.

Since we are building upon the work of Minz Won et al., we decided to go with one of the datasets used in their work [15]. More specifically, we used the MagnaTagATune dataset, a publicly available dataset containing approximately 25,000 music clips from a variety of genres in MP3 format. Each clip includes tags describing the content of the clip, including characteristics such as genre (*e.g.*, rock, jazz) and instrument information (*e.g.*, drums, guitar) [9]. We used the top 50 most popular tags to be consistent with previous work. Similarly, we used the same train-validation-test split as Minz Won et al[15]. This dataset is attractive to us due to its popularity for automatic music tagging and existing performance benchmarks.

2. Approach

In order to increase the robustness of the existing automatic music tagging models, we stochastically augmented the training data with the addition of noise, time stretch and pitch shift. We evaluated the performance of these models with augmentations and selected the top three for further analysis. We applied the top three most performant models and our ViT model to different noisy test sets to gauge if the augmented training set has an effect in the robustness. We hypothesized that training data augmentation would improve robustness due to its success in other audio classification settings. [12]. Training data augmentation has not been previously utilized in music tagging approaches. Furthermore, although transformer architecture was used in the attention-based model, ViTs were not used before for this task on the MagnaTagATune dataset.

2.1. Tools Used

We utilized the code provided by Minz Won et al. as a starting point [15], ensuring our tools were aligned with theirs. For the neural network architecture and training, we employed PyTorch. Audio preprocessing and data augmentation were carried out using the popular audio manipulation library, librosa. Additionally, we used Matplotlib and Pandas for data visualization and analysis. Training logs were maintained using TensorBoard, which facilitated access to validation and loss curves.

2.2. Data Preprocessing and Augmentations

The data preprocessing code from Minz Won was refactored for initial preprocessing [15]. First, we convert the MP3 files to numpy array format (npy), a format widely used for efficiently storing array data, using librosa. Next, we define a PyTorch DataLoader that incorporates augmentations to enhance the training process. Since some models require short-duration audio, the DataLoader randomly samples inputs to extract excerpts of a specified length if the provided length is shorter than the entire song. This strategy

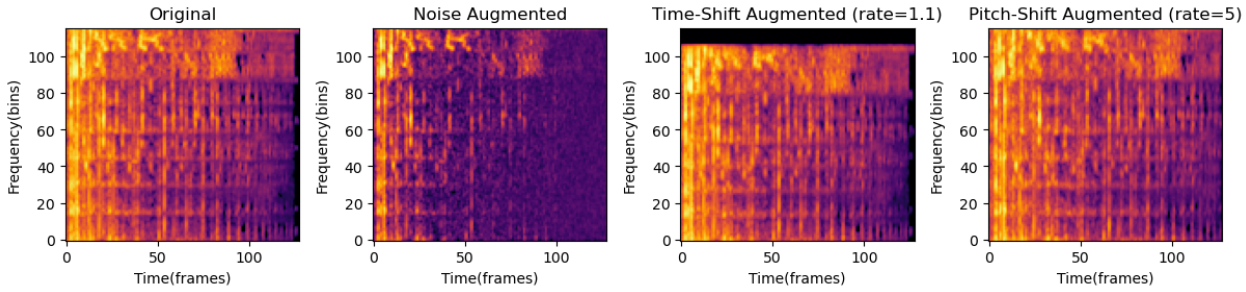

Figure 1. Mel spectrogram of augmented audio showing the variation introduced by the augmentations

introduces variability during each epoch, ultimately enhancing the model's generalization ability. The augmentations applied in the DataLoader are governed by a hyperparameter we refer to as 'aug_prob', which stands for augmentation probability. This hyperparameter determines the percentage of data points that will undergo augmentation. The decision to apply each type of augmentation is made independently. For instance, if 'aug_prob' is set to 0.5, there is a 50 percent chance of applying noise, a 50 percent chance of applying time stretch, and a 50 percent chance of applying pitch shift, in this order. This disjoint nature of the augmentation events is designed to increase variability, resulting in unique combinations that help the model learn to handle diverse audio scenarios. Ultimately, these preprocessing steps and augmentations aim to enhance the model's robustness and performance on unseen data. The effect and variation introduced by these augmentations can be seen in the Mel spectrograms provided in Figure 1.

Noise augmentation adds random noise to the original audio signal. In our code, noise is generated from a standard normal distribution and scaled by a predefined factor, 'noise_factor', which controls the intensity of the noise added. We use a 'noise_factor' of 0.005 for our models. When activated, this augmentation introduces variability in the training data by simulating real-world conditions where audio signals may be subject to background noise. The stochastic nature of noise addition means that each training instance can be slightly different, helping the model to learn to distinguish the primary audio signal from noise. This helps to prevent overfitting and enhances the model's performance in noisy environments.

Time-stretch augmentation involves modifying the speed of an audio signal without altering its pitch. In the provided implementation, the 'librosa.effects.time_stretch' function is employed to stretch or compress the audio signal based on a randomly chosen 'timeshift_rate', which can vary between 0.8 and 1.2. This rate determines how much the audio will be sped up or slowed down. By introducing this variability, the model is exposed to audio signals of different durations while maintaining their original pitch characteristics. Time-stretching mimics potential variations in playback speed and can help the model to generalize better, as it learns to process audio at different tempos and rhythms.

Pitch-shift augmentation alters the pitch of an audio signal while keeping its duration constant. In our implementation, the 'librosa.effects.pitch_shift' function is used, where a random pitch shift value, 'pitchshift_rate', is selected from a range of -5 to 5 semitones. This allows for both upward and downward pitch adjustments. The application of pitch shifting helps to create diverse audio samples that retain the same content but with different tonal qualities. By learning to handle a range of pitches, the model can achieve improved performance on unseen audio data that may differ in tonal characteristics.

2.3. Training Details (Schedule, Epochs, Metrics)

We adopt Binary Cross Entropy Loss (BCELoss) as our loss function, which is standard for architectures applied to automatic music tagging on the MagnaTagATune dataset, consistent with the work of Minz Won et al. [15]. Our training employs the Adam optimizer with a learning rate of 1e-4. Due to time constraints, we limited our training to 10 epochs.. We used a batch size of 16. The model is saved whenever the validation loss improves, ensuring that we retain the version with the lowest validation loss at the end of the 10 epochs.

We evaluate the models using Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) and Precision Recall (PR) AUC metrics. Both ROC AUC and PR AUC have been employed for all existing architectures, establishing them as standard metrics for this task. ROC AUC measures the model's ability to distinguish between classes across various thresholds, providing a scalar value that reflects the trade-off between true positive rates and false positive rates. A ROC AUC of 1 indicates perfect separation, while a value of 0.5 suggests no discrimination capability, akin to random guessing [2]. Conversely, PR AUC focuses on the trade-off between precision (the ratio of true positives to total predicted positives) and recall (the ratio of true positives to total actual positives). This metric is par-

ticularly valuable for imbalanced datasets, where the positive class is rare compared to the negative class. Given that the MagnaTagATune dataset is imbalanced, PR AUC is a suitable choice. A high PR AUC indicates a model that successfully identifies positive instances while minimizing false positives [1]. Together, ROC AUC and PR AUC provide complementary perspectives on model performance.

2.4. Vision Transformer(ViT)

Since ViTs are traditionally used for image processing, the idea of using them for audio analysis is a relatively recent one. We hope that using a ViT would provide a novel perspective on music tagging and potentially be able to identify patterns in the audio that have been missed by existing models. We took inspiration from the model employed for image classification [6]. Our ViT model is structured into several key components, each serving a distinct purpose. Initially, the model transforms raw audio input into a Mel spectrogram, followed by converting the amplitude to decibels, and normalizing the spectrogram by applying batch normalization, which prepares the data for subsequent processing. Next, a series of CNNs extract hierarchical features from the spectrogram, employing ReLU activations and max pooling to reduce dimensionality while enhancing feature representations. The output from the CNN is then reshaped into patches via a convolutional patch embedding layer, facilitating the integration of spatial information into the transformer architecture. A positional encoding is added to these patches to retain spatial context before passing them through a transformer encoder, which consists of multiple layers designed to capture long-range dependencies in the feature representation. Finally, a fully connected layer maps the output of the transformer to class probabilities, utilizing a sigmoid activation function for multi-class classification, thus producing the final output.

2.5. Challenges

We anticipated several challenges related to data augmentation, and long training time due to our limited hardware. Selecting the optimal strength of augmentations was also a concern due to our lack of prior experience in this area. Finding the right balance between effective generalization and learning was crucial; overly strong augmentations could hinder the model's ability to learn. We relied on trial and error to determine the best range for augmentations, ultimately concluding that a better configuration might exist, but time constraints limited our exploration.

While we combated the long training times by training each model for 10 epochs, we did not foresee the lengthy evaluation times, especially when dealing with noisy test sets, which require on-the-fly generation rather than pre-existing data. Our initial attempt with the code from Minz Won et al. was successful for all models except the attention-based model, which encountered dependency issues [15]. Consequently, we opted to exclude this model from our evaluation, given the similarities with the ViT model, which we proceeded to implement.

The first application of our data augmentations yielded promising yet suboptimal results, necessitating further tuning to enhance performance. Similarly, the first implementation of our ViT yielded poor performance in the validation set. We improved the model by adding more CNN layers for feature extraction. We were particularly concerned about the potential for overfitting in the ViT model due to its vast number of parameters. To combat this, we increased the input size and applied random sampling of 15-second segments from audio clips, introducing variety to the dataset and reducing the risk of overfitting.

3. Experiments and Results

3.1. ViT Results

Table 1 shows that our ViT model includes more than 25 million learnable parameters, which is significantly more than the other models and more than double the number in the next most complex model. The large number of parameters reinforced our concern about overfitting.

Model	Learnable Parameters
fcn	447,732
musicnn	784,300
crnn	394,100
sample	1,869,874
se	6,951,858
short	3,687,348
short_res	12,104,244
hcnn	3,631,551
vit	25,066,804

Table 1. Number of learnable parameters for each model

Figure 2 shows the training and validation loss of our ViT model (with no data augmentation) over the span of 10 epochs. Our concerns about our ViT model overfitting the training data were justified, as seen in Figure 3. We saw both training and validation loss decrease through the first 7 epochs. However, after epoch 7, the training loss continued to decrease while the validation loss remained roughly the same. This is also supported by our observations of the validation curves, which showed peak ROC AUC and PR AUC scores at epoch 8. This indicates that there is a tendency for our ViT model to overfit the training data. If the model were run for more than 10 epochs, the divergence between the training and validation loss may become more significant. This outcome could be mitigated through adjustment of the model's hyperparameters and better regularization.

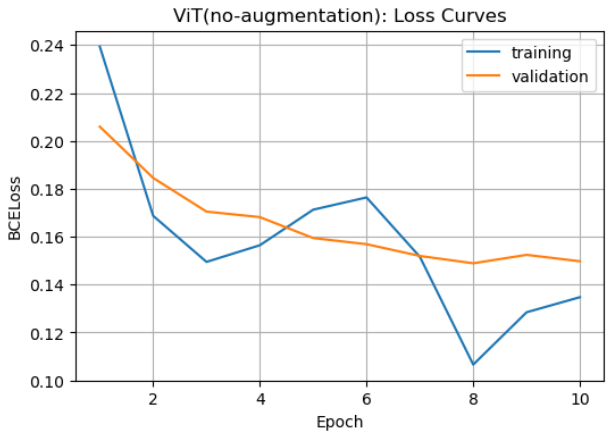

Figure 2. Loss curves for ViT model on base test set

Metric	Train	Validation	Test
ROC AUC	0.9002	0.8906	0.8846
PR AUC	0.4385	0.3995	0.3904

Table 2. ViT model performance on dataset with no augmentation

Table 2 shows that our ViT model achieves comparable results on the MagnaTagATune dataset to the state-of-the-art models. The maximum ROC-AUC score achieved by the state-of-the-art model was 0.9129, while the maximum PR-AUC score was 0.4614 (both achieved by the short-chunk CNN with residual connections) [15]. The fact that our ViT model was able to achieve scores somewhat close to this without extensive hyperparameter tuning and limited training time suggests that ViT models are a promising area of further research for automatic music tagging.

3.2. Augmentation Results

Since we are training our models on augmented training data, we had to make a decision on what percent of augmented data would give us the most optimal results. In order to make this decision, we trained all the models on different 'aug_prob' values and looked at their performance on the base test set. Figure 3 shows that the performance significantly degrades for almost all models for 'aug_prob' higher than 0.5. This trend was also visible in PR AUC. We expected the performance to degrade, since the excessive noise from the disjoint stochastic nature of the augmentations make it difficult for the models to learn effectively due to the drastic changes in the data during each epoch. The top three best performing models are HCNN, short CNN and short CNN with residual connections. We will use these models and our ViT model for the rest of the experiments. It is important to note that the CRNN model would most likely require higher number of epochs to achieve results similar to the other models, since it has an RNN based architecture. Based on the observations from 3, we compare

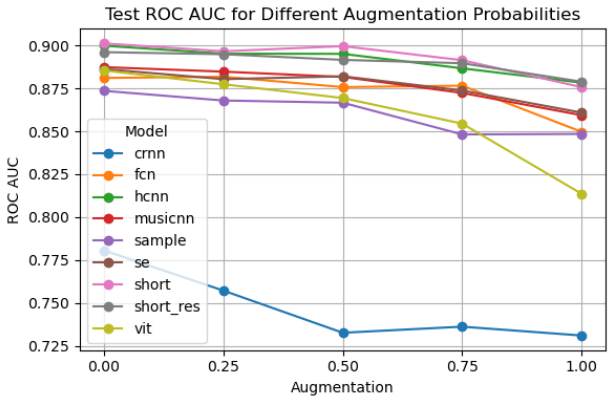

Figure 3. Changing 'aug_prob' for all models in base test set

the selected models with 0.5 'aug_prob' with their default versions that use unaugmented training data. We do these comparisons in noisy test sets to gauge their robustness. We used a total of 6 noisy test sets: added white-noise(0.1,0.4), pitch-shift(1,-1), time-stretch($-2^{\frac{1}{2}}$,$2^{\frac{1}{2}}$). These are the same augmentations that Minz Won et al. tried to test their models for robustness[15]. For the pitch-shift and time-stretch, we use the same process as we did for augmentation. However, the white-noise code is a different process than the one we used for augmenting. This code is borrowed from Minz Won et al. [15] to allow comparison of results.

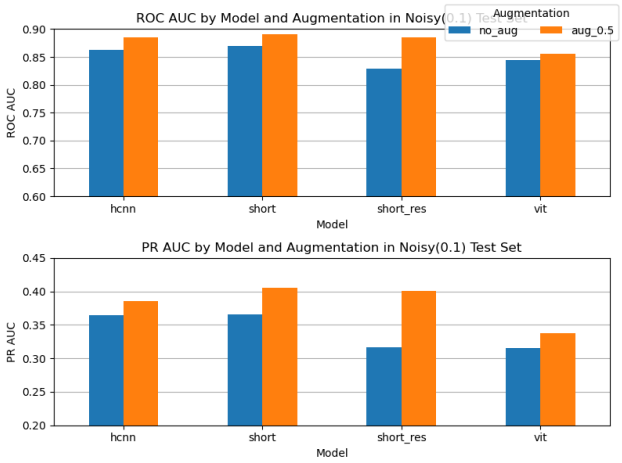

Figure 4. Noisy(0.1) test scores for selected models

Minz Won et al. illustrated that all their models trained on the base dataset showed poor performance on different noisy test sets[15]. Our work shows that augmenting training data could be a good way to improve robustness. As shown in Figure 4, all the models trained on augmented datasets did better on the noisy(0.1) test set compared to

the models trained in the base dataset. We also observed the same trend in the results for both high and low pitch augmentations. However, the results get more interesting when looking at the noisy(0.4) test set. As shown in Figure 5, augmentation improved the performance of all models except the ViT model. As discussed earlier, the ViT model is more likely to be overfitting. If the noise present in the training set is not as high as the noisy(0.4) test set, it is possible that the ViT model overfit to the data in the less noisy training set and thus struggles with the highly noisy test set. While we are inferring this based on our observations, this needs to be further investigated.

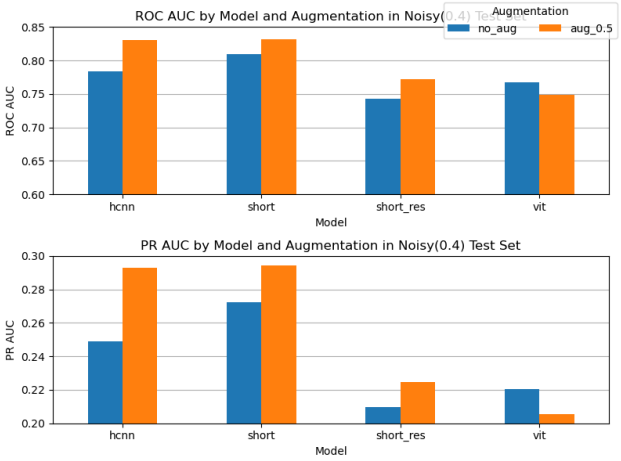

Figure 5. Noisy(0.4) test scores for selected models

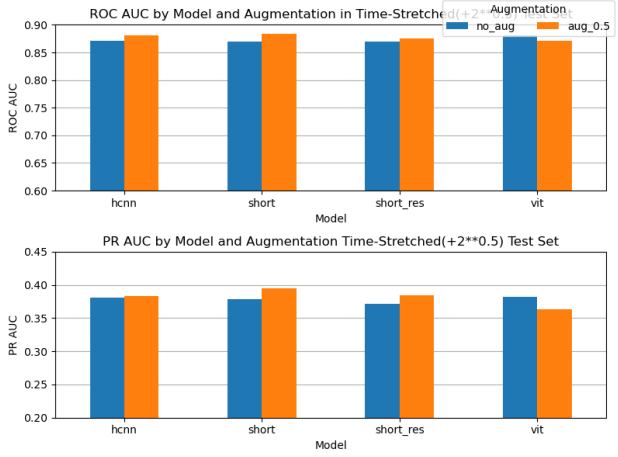

Figure 6. Time-stretch augmented test scores for selected models

Based on time-stretch and time-shrink augmented results in Figure 6 and Figure 7, it is clear that all the models perform better on the time-stretch augmented test set. This intuitively makes sense since reducing shrinking the audio results in less data for the model to predict from since the augmentation maintains the input size by replacing the padded regions with zeros. On the other hand. It is interesting to see the ViT trained with no augmentations doing pretty well on the time-stretch and time-shift augmented datasets. The stochastic nature of the training augmentation makes it difficult to pinpoint exactly why this is happening.

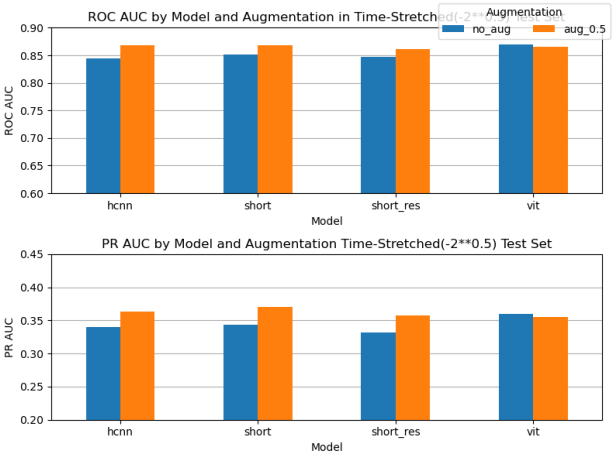

Figure 7. Time-shrink augmented test scores for selected models

Overall, we observed that the augmented training improve the robustness of the models except the ViT model, which showed some interesting behavior that needs to be further investigated in future work.

4. Conclusion

In conclusion, our work shows that different forms of data augmentation can improve robustness of automatic music tagging models. Furthermore, we show that ViT can be successfully used to perform music classification, producing results close to the state-of-the-art models. While our work has exciting implications, there are some shortcomings in our work that can be addressed in the future. Since our work is grounded on the MagnaTagATune dataset, applying the same procedure on a different music tagging dataset may reveal new information. Furthermore, due to time constraints, we had to limit our model training to 10 epochs, and perform limited hyperparameter tuning. With longer training and efficient hyperparameter tuning, it is possible that the ViT model does even better than the results we have presented. Furthermore, the interesting behavior of ViT when applied to time-stretch augmented test set needs to be further investigated for the sake of model interpretability. The effectiveness of augmentations can also be further studied by applying a wider variety of augmentations and testing them independently

5. Work Division

Table 3 shows the contributions from each teammate. Our code can be accessed here.

References

[1] Kendrick Boyd, Kevin H. Eng, and C. David Page. Erratum: Area under the precision-recall curve: Point estimates and confidence intervals. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages E1–E1, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 4

[2] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. 3

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

[4] Keunwoo Choi, George Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification, 2016. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2, 4

[7] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019. 1

[8] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms, 2018. 1

[9] Edith Law, Kris West, Michael Mandel, Mert Bay, and J. Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009*, Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, pages 387–392, 2009. 10th International Society for Music Information Retrieval Conference, ISMIR 2009 ; Conference date: 26-10-2009 Through 30-10-2009. 2

[10] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms, 2017. 1

[11] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale, 2018. 1

[12] Yanjie Sun, Kele Xu, Chaorun Liu, Yong Dou, Huaimin Wang, Bo Ding, and Qinghua Pan. Automated data augmentation for audio classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2716–2728, 2024. 2

[13] 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, 2020. 2

Student Name	Contributed Aspects	Details
Sadab Hafiz	Planning, Implementation, and Analysis	Implemented the data augmentations, trained the models, improved the ViT model, collected results, generated graphs, and provided analysis in final paper.
Robert Sharp	Planning, Implementation, and Analysis	Implemented the ViT model and provided analysis in final paper.

Table 3. Contributions of team members.

[14] Minz Won, Sanghyuk Chun, and Xavier Serra. Toward interpretable music tagging with self-attention, 2019. 2

[15] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of cnn-based automatic music tagging models, 2020. 1, 2, 3, 4, 5

6. Appendix

Layer	Learnable Parameters
Mel-spectrogram	2
CNN	369,664
Patch Embedding Layer	16,777,472
Positional Encoder	16,384
Transformer	7,890,432
Fully Connected Layer	12,850

Table 4. ViT learnable parameters on each layer

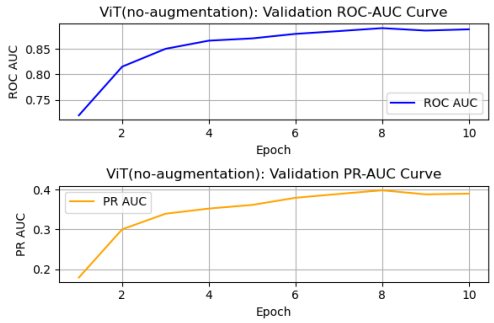

Figure 8. Validation curves for ViT model on base test set

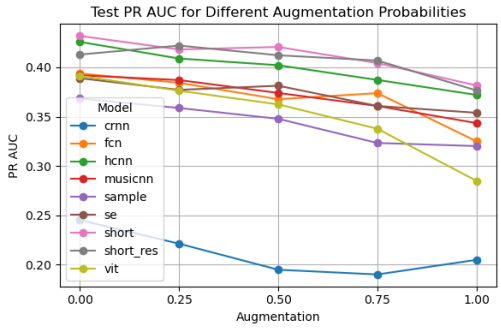

Figure 9. Changing 'aug_prob' for all models in base test set

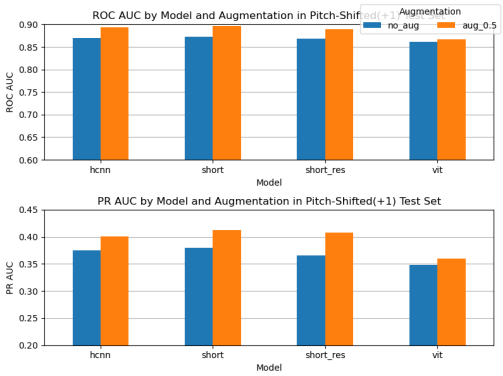

Figure 10. Pitch(+1) augmented test scores for selected models

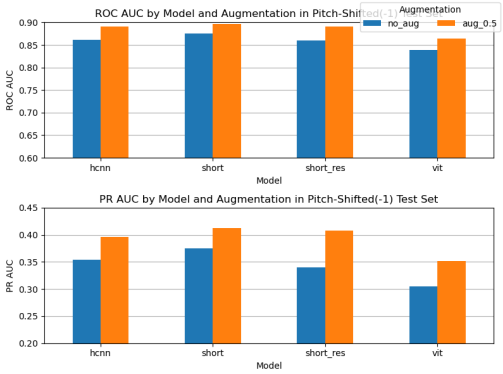

Figure 11. Pitch(-1) augmented test scores for selected models