# Multi-format Contrastive Learning for Music Genre Classification[1]


## ABSTRACT

We study how multiple data formats and contrastive learning algorithms can be applied to learn music representations. This paper presents an experiment where we train two encoders independently by using contrastive learning framework to generate both visual and audio representations of input music data. The approach involves using both audio files and their corresponding Mel Spectrograms as input for the model, and the goal is to learn a combined representation that can capture the characteristics of different music genres. While training both encoders, we adopt stochastic data augmentation methods from SimCLR[1] and CLMR[2]. The result of GTZAN[3] music genre classification task shows that using the combined representations from multiple input format will improve the overall accuracy even though they were trained separately.


## 1 Introduction

Classifying music genres has been a challenging problem in machine learning. Despite the advancement of deep learning models, constructing labeled music datasets still remains an expensive and time-consuming task. In this work, we propose a self-supervised method for categorizing music genre by using both audio files and their Mel Spectrograms. We train encoders for these two formats of data with Contrastive Learning framework - SimCLR[1]. In the end, we discuss the experiment result and how different design decisions and hyperparameters would affect the performance of the network.

## 2 Related Work

### 2.1 SimCLR

SimCLR is a simple framework for contrastive learning of visual representations[1]. The paper uses stochastic data augmentation to get a correlated data pair from a single training data point and learn the representation based on the contrastive loss function that evaluates the similarity of the data pair. The previous work has also shown that the linear classifier trained on top of the learned visual representation can achieve the same performance as the supervised SOTA models. This paper also discusses the critical roles of using data augmentation techniques, learnable nonlinear transformation layers and contrastive loss and larger batch sizes for improving the quality of the trained representation. We adopt the same contrastive setup in our experiement where we generate a pair of corrleated data from input and maximize their agreements by minimizing the noise-contrastive cross-entropy loss of the encoded outputs.

### 2.2 CLMR

CLMR is a self-supervised contrastive learning framework for learning the multimodal representation of raw music audio[2]. CLMR introduces a stochastic data augmentation method that produces positive pairs of same audio fragment, which is essential for improving the quality of the trained representations. It also uses the similar components as SimCLR: Sample CNN encoder network, non-linear (*ReLU*) projector network and *NT-Xent loss* as contrastive loss function. In our experiment, we use the similar data augmentation techniques for training audio encoder.

### 2.3 Multi-Format Contrastive Learning of Audio Representations[4]

The paper also presents a method for training audio representation models using audio and Mel Spectrogram data with contrastive learning approach where they find that this multi-format strategy can lead to significant gains in performance compared to single-format counterparts. The proposed training strategy in this paper was to maximize the similarity between the representations of audio data and its corresponding spectrogram. In contrast, we augment and train each format of input independently and combine the trained representations as a single output in the end. The same data augmentation methods for spectrograms were adopted in this experiment.

[1] https://github.com/fw2399/CLMR

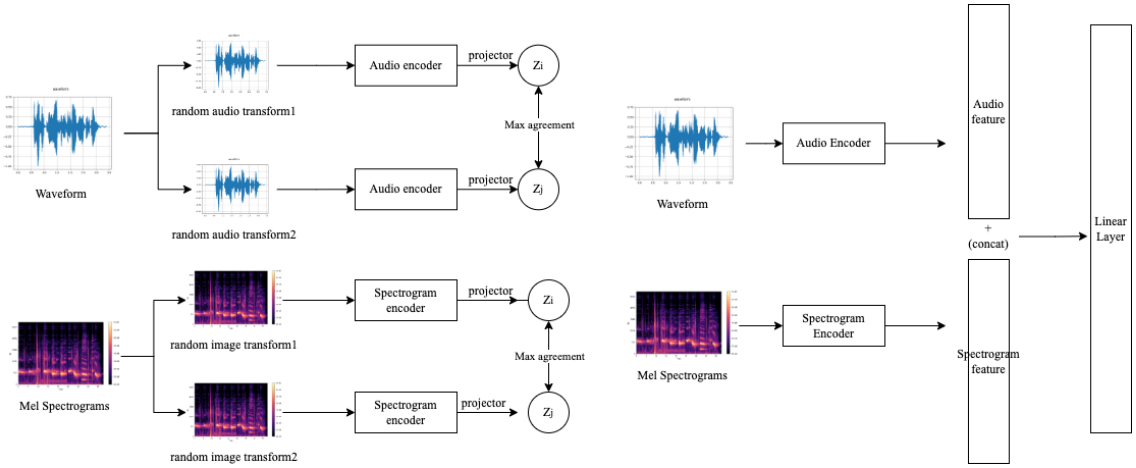


Figure 1: Proposed contrastive framework during training (left) and inference (right)

## 3 Method

This main algorithm is based on SimCLR and CLMR, and we set up the contrastive learning framework for both image data and audio data respectively. Each setup has the following core components: 1) A stochastic composition of data augmentation. 2) An encoder neural network $g_{enc}(\cdot)$ for generating the representation of the input data 3) A non-linear projector neural network $g_{proj}(\cdot)$ that maps encoded representations into the latent space 4) A contrastive loss function that identifies the positive sample $x_j$ from the negative mini batch $\{x_{k\neq i}\}$.

### 3.1 Audio pipeline

Audio encoder consists of 9 1d convolution blocks, a 1d batchnorm (this layer was added to improve the generalization and stabilize the training process, given that the dataset size is relatively small), and a 1d max pooling layer, and it produces a 512-dimensional feature vector as output. For the projector, we use a standard multi layer perceptron with one hidden layer to obtain $z_i = g(h_i) = W_2\sigma(W_1 h_i)$ where $\sigma$ is a ReLU non-linearity (From SimCLR, it shows that using nonlinear projection head improves the representation quality of the layer).

To create positively correlated pairs of audio data, we use the following transformations during training: randomly sample a fixed-length period of the input audio, randomly add white Gaussian noise, reduce gain between $[-6, 0]$, randomly apply frequency filter, and randomly shift pitch in the range of $[-5, 5]$ semitones.

### 3.2 Spectrogram pipeline

Similar as audio encoder, we use the state-of-the-art ResNet-18 as the encoder for processing spectrogram input. Mel Spectrograms is a visual representation of sound signal with the time on the x-axis and frequency on the y-axis. The value of each pixel of the spectrogram represents the intensity of a signal at certain frequency. As mentioned in the milestone document, some of the data augmentation techniques used in SimCLR might not be suitable (for example, horizontal flip, random gray scale) since they do not produce meaningful correlated views of the same input. We are using different set of augmentation methods during training [4]: random frequency masking and random time masking, random mixing audio noise (As shown in figure 2).

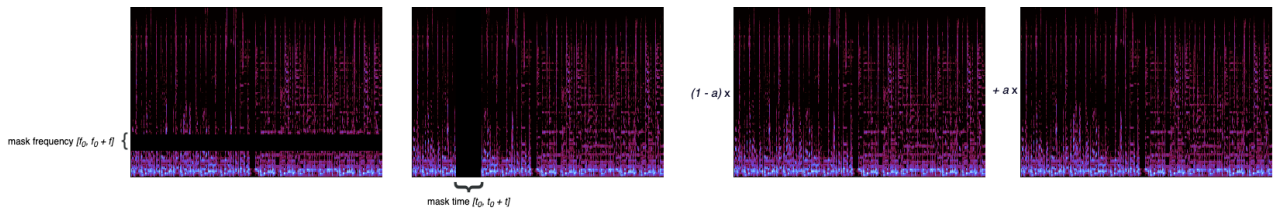


Figure 2: Data augmentations for spectrograms

### 3.3 Loss function and optimizers

In this experiment, we use the same loss function from SimCLR - normalised temperature-scaled ($NT-Xentloss$):

$$l_{ij} = -log\frac{\exp(sim(z_i, z_j)/r)}{\sum_{k=1}^{2N} 1_{k\neq i}\exp(sim(z_i, z_K)/r)}$$

with temperature parameter $r = 0.5$. ($sim$ is cosine similarity) We use Adam optimizer with decay rate $= 10^{-6}$ for training both encoders.

### 3.4 Inference

After obtaining the representation vectors from both visual and audio data, we concatenate two encoded 512-dimension representation vectors into a single 1024-dimension feature vector. To evaluate the quality of the learned representation, we trained a learnable linear classifier on top. To ensure fairness, the linear classifier are trained with the same number of epochs and batch size.

## 4 Experiment

We use GTZAN[3] dataset for our music genre classification task. The dataset contains 10 genres with around 100 30-second audio files and their converted Mel Spectrograms. We keep the default train-test split from torchaudio unchanged (test 290, train 443 and valid 197), each music genre is roughly evenly distributed. Both audio encoder and image encoder are trained from scratch, where we randomly sample 5-second clip from the audio input at 22k Hz.

### 4.1 Result

| Encoder | Input format | mAP | Test accuracy |
|---|---|---|---|
| only enable random sample | Audio | 0.115 | 11.1 % |
| enable random sample + white Gaussian noise | Audio | 0.127 | 10.3% |
| enable all transformations | Audio | 0.219 | 37.7% |
| only enable time masking | Spectrogram | 0.249 | 35.1% |
| enable all transformations | Spectrogram | 0.312 | 48.2% |

Table 1: Accuracy of representations trained with different data augmentation methods enabled.

First, we trained audio and image encoders by enabling different data transformations while constructing positive pairs, then we trained a linear layer (multi-class logistic regression) on top to evaluate the trained representation. We measured the accuracy of the classifier by calculating Mean Average Precision (mAP) on validation dataset. As shown in Table 1, we can see clear accuracy improvement as more data transformation are enabled during training for both audio and image format. (The classifier is trained by using a single feature vector of either audio or spectrogram input.) This finding is reasonable since more data augmentation helps model generalize better to new data and prevent model from overfitting or being too sensitive to small variants of the input. This is more obvious in our experiment since the training and test set are relatively small (a few hundred data points).

While training encoder for spectrograms, we found that the representations learned by using larger size of time or frequency masks (e.g, mask $10\%$ vs $5\%$ of total period) helped model generalize better and improved result in genre classification task. Data shuffling is also critical for contrastive learning as it helps prevent data of the same genre in a batch (so that they will not be used as the negative samples).

| Model | Training data | mAP | Test accuracy |
|---|---|---|---|
| only use audio | GTZAN | 0.269 | 37.7 % |
| only use audio | MagnaTagATune[5] | 0.193 | 19.1 % |
| only use spectrogram | GTZAN | 0.312 | 48.2% |
| use both audio and spectrogram | GTZAN | 0.340 | 53.1% |

Table 2: Accuracy of representations trained with different input representations.

In Table 2, we compare the performance of classifier trained on different music representations. The classifier trained by using only audio feature representation has 0.219 mAP, the classifier trained by using only spectrogram has 0.312

mAP, whereas using both representations results in a better accuracy (0.340 mAP). We believe this might be because spectrogram emphasizes the frequency components of the signal at different points in time, and the linear classifier can potentially utilize that information to better categorize its genre. During the experiment, we often see models have small loss and high validation accuracy; however, the performance is worse on test dataset, and we found that setting an aggressive value on weight decay and early stopping can help improve the generality of the learned representations significantly.

We also evaluate the representations trained from a different dataset - MagnaTagATune[5] (25, 863 music clips with each 29-second long), and its performance was utterly worse than other models that were trained directly from GTZAN, which suggests that the feature learned from MagnaTagATune might not well apply to data from GTZAN.

In the end, we also noticed that adding a non-linear hidden layer (e.g., ReLU) to the linear head improves the accuracy of music genere classification task considerably for any representations (due to time constraints, we did not have the chance to re-train all models listed in Table 1 and 2 by using the new classification head).

### 4.2 Retrospective

As shown in Table 2, the combined representations (from feature vectors of audio and spectrogram) outperform others, but the overall accuracy (53%) for GTZAN genre classification task is not comparable to the result from CLMR (63%) and is substantially worse than the state-of-the-art supervised algorithms. It is also not comparable with the model proposed in Multi-format contrastive learning of audio representations[4] where they adopted early data fusion (a single encoder takes both audio and spectrogram as inputs). In our model, information from different sources is combined at a later stage, after the data has undergone most of the processing, though we believe early fusion could be beneficial in this case since the input sources are highly correlated and can be effectively combined with minimal processing.

As pointed out in the proposal feedback, concatenating representations of visual and audio data is a shallow approach for data fusion (feature-level fusion), and it is worth trying out other fusion techniques, such as decision-level fusion (e.g, ensemble learning, compute a weighted average of the outputs) or model level fusion (training a single model that can handle multiple modalities as input and make predictions based on all of them) to well leverage data from different sources.

## 5 Conclusion

In conclusion, our study found that using multiple data formats and contrastive learning algorithms can be effective for learning music representations. By training two encoders to generate visual and audio representations of music using a contrastive learning framework and combining the resulting representations, we were able to improve the accuracy of a music genre classification task. The experiment result also shows the value of using various data augmentation methods during contrastive learning to construct training pairs.

## References


[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

[2] Janne Spijkervet and John Ashley Burgoyne. Contrastive learning of musical representations. *ArXiv*, 2021.

[3] Bob L. Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *Journal of New Music Research*, 43(2):147–172, apr 2014.

[4] Luyu Wang and Aaron van den Oord. Multi-format contrastive learning of audio representations. *ArXiv*, 2021.

[5] Edith Law, Kris West, Michael Mandel, Mert Bay, and J. Downie. Evaluation of algorithms using games: The case of music tagging. pages 387–392, 01 2009.