Multimodal Music Video Representation

Master's Thesis

Karel Veldkamp, 11251611
University of Amsterdam
Amsterdam, the Netherlands

Mariya Hendriksen, Internal Supervisor
University of Amsterdam
Amsterdam, the Netherlands

Alexander Keijser, External Supervisor
XITE
Amsterdam, the Netherlands

Zoltán Szlávik, External Supervisor
XITE
Amsterdam, the Netherlands

ABSTRACT

Recommending videos that match a user's taste is of key interest to any video platform. This task is inherently multimodal, especially so for music videos, in which the audio is particularly important. One useful way to do this is to learn representations from the audio and video data itself, and use these in a content based recommendation approach. This approach comes with the advantage that it does not rely on historical usage data, and that it is insensitive to popularity bias. Previous research has focused on different supervised tasks to learn content based representations, but this method relies on the availability of a labelled dataset. In this thesis, we use a multimodal contrastive leaning approach. We use separate encoders for the audio and video modalities, and train them using a contrastive loss function to project the two modalities into the same latent space. We use the representations in this latent space on the downstream tasks of genre classification and music tagging, and compare the performance to baselines from the literature. Additionally, we evaluate the quality of representations qualitatively by examining the most similar items to several seed items. We show that representing these two music video modalities in the same space is not trivial. Contrastive learning using a dataset of 600,000 music video segments did not manage to successfully pull embeddings from the two modalities closer together, which was reflected in downstream task performance as well as our qualitative analysis. All our code is made publicly available here.

1 INTRODUCTION

Music videos are some of the most circulated items on the internet, with the most popular videos being watched billions of times [62]. They constitute another medium for musicians to express themselves artistically, and serve as a means to promote albums for record labels. XITE [65] is a company that specialises in music videos, and this project was executed in the context of this company. They provide a smart TV app that offers an arrangement of curated music video channels, as well as some recommendation based channels that are tailored toward the taste of the user.

In the current study, we learn representations for music videos that can be used for recommendation. Given the extensive and diverse catalog of music videos available today, recommending music videos that match a user's taste is of great interest. XITE has over 100,000 music videos published on their platform, so it is key to recommend the right items to the user. Additionally, the use case of music video recommendation requires that items can be recommended as soon as they are released. New items are added to the platform every week, and it is important that these items can be recommended immediately, as music videos are often most relevant in the period shortly after their release. Although the recommendation of music is a topic that has received a lot of attention in the field (for a recent survey, see [37]), there is little research on recommender systems for music videos specifically.

Content based recommendation. One popular approach to recommendation is content based (CB) recommendation [57, 71]. CB methods work by recommending items that are similar to items that a user has liked before. In order to calculate a similarity metric, it is essential to have a good representation of items. In practice, these representations are based on metadata about the item, or directly on the content of the item itself. We believe that the representation of a music video should include information from multiple modalities, because the appeal of a music video depends on more than just audio. One of the core values of XITE is their emphasis on the visual aspect of music videos, reflected in their slogan 'See Music'. In line with this intuition, we learn music video representations based on the audio and video modalities of a music video. Having such a representation is not only important for item recommendation, but can also be useful for item search functionality [31].

Collaborative filtering. A common alternative to CB recommendation is collaborative filtering, which is commonly done using matrix factorization (MF) [26]. Rather than using item representations, MF methods rely on historical patterns of usage data, and are therefore independent of the type of content that is being recommended. They estimate low dimensional vectors representing each user and each item based on the likes or interactions of users with items. This approach generally works well, but it suffers from the cold start problem: when a new item or user gets added to the system, they have no usage history, so there is no data to estimate the model from. This is a big problem for music video recommendation, as waiting for a music video to accumulate usage data is not a viable option, due to the time span of music video relevance. Another downside of MF methods is that they are known to exhibit bias for more popular items, which can hurt user experience for users with a more alternative taste [5]. Therefore, our work focuses on a content based approach to music video recommendation.

Research questions. In this thesis, we devise a method that learns audio-visual representations for music videos based on their content, that can be used for recommendation or other downstream tasks. We aim to answer the following research questions:

(RQ1) Can an audio-visual representation improve the calculation of music video similarity compared to a baseline that uses metadata?

(RQ2) Can an audio-visual representation improve performance on downstream tasks compared to existing music tagging models?
  (a) Can it improve performance on the task of genre classification?
  (b) Can it improve performance on the task of music tagging?

Contributions. The principal contributions of our research are the following:

- The adaptation of a contrastive deep learning method to learn multimodal music video representations.
- The application and evaluation of these learned representations on downstream tasks.
- A qualitative comparison of a model based on representations with a model based on metadata.

The rest of this thesis is structured as follows. First, we provide an overview of some of the important literature on machine learning for the audio and video modalities, and how these can be combined into multimodal models. Then, we discuss our proposed model, as well as our plan for evaluating its performance compared to alternatives. Finally, we share our results, and discuss their implications for music video representation.

2 RELATED WORK

Multimodal methods tend to build on earlier work on modality specific techniques and combine these techniques in a multimodal framework. Therefore, we start by outlining some of the important literature on deep learning based on audio and video data. Then, we discuss several ways in which information from these different modalities can be combined, as well as some of the advantages and weaknesses of these different methods.

2.1 Unimodal learning

Audio. The most popular feature representations for audio are mel frequency cepstral coefficients (MFCCs), which consist of the cosine transform of the log Fourier transform of an audio waveform. However, in deep learning approaches, the cosine transform is often omitted, resulting in log mel spectrum features [39]. These features are popular for general audio tasks [11], as well as music related tasks [30]. Alternatively, some models are trained on the raw audio waveforms [27].

In terms of modelling, convolutional neural networks (CNNs) are the most common for audio data. The convolutions are typically two dimensional when dealing with log mel spectrum features, which contain a time and a frequency dimension. Previous research has shown that using CNNs on mel spectogram data can lead to good performance in various musical tasks like genre classification [10, 49] and music emotion recognition [30].

Long short term memory (LSTM) models are an alternative to CNNs, that model time explicitly [20]. LSTM models have been shown to outperform CNNs on some speech recognition tasks [44], but they are slower, because they can less easily be parallelized [39]. Similar to other fields in machine learning, transformers have been replacing LSTMs in recent years. Transformers have been applied successfully to general audio related tasks [15], as well as some musical tasks, such as genre classification [40], music transcription [19] and music tagging [63], in which they outperformed CNNs. Although transformers have produced promising results in the domain, they need very large amounts of data to train [12]. CNNs can provide good results with much less data because the structure of a CNN can be designed to contain domain specific prior knowledge [64].

Video. The two most covered approaches to using deep learning for video with CNNs are two-stream CNNs and three dimensional CNNs [43]. The two-stream CNN was first proposed in [48], and was originally used for action recognition, but it has since been used for various video related tasks, like near-traffic accident detection [22] and deep fake detection [21]. It contains a spatial stream that performs action recognition from a single image, and a temporal stream which performs action recognition from the optical flow between several consecutive frames. These two streams are then combined into a single output layer using late fusion.

In three dimensional CNNs, as the name suggests, the filters convolve over time as well as the x and y axis of a frame. Although this approach yields very good representations of video data [51], the third dimension increases the complexity exponentially. This complexity can be reduced by separating spatial and temporal convolutions [52]. The idea behind separated convolutions is to have separate two dimensional kernels for the spatial dimension, and one dimensional kernels for the temporal dimension, rather than having each kernel convolve over all three dimensions. The authors show that while reducing the complexity significantly, models with these separable convolutions can still attain state of the art performance on video classification tasks.

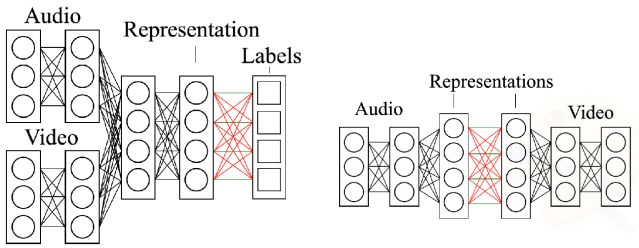
In this thesis, we take music inspired CNNs for audio, and CNNs with separated convolutions for video, and combine these into a multimodal framework. The next section discusses how information from these two modalities can be combined into a single framework.

2.2 Multimodal learning

The main challenge for multimodal learning is that feature vectors for different modalities encode for different types of features, which is referred to as the heterogeneity gap [17]. The most common way to deal with this problem is to project the different modalities into a single subspace, while somehow still retaining information from the different modalities [3]. Where originally, this was often done by combining features from different modalities in clever ways and using statistical learning algorithms on these combined features [2], most recent work on multimodal learning relies on deep neural networks [17]. In this section we discuss supervised, unsupervised and contrastive deep learning methods.

Supervised learning. One approach to project multiple modalities to a single latent space is to create a single deep neural network, with separate input layers for each modality, which are at some point combined into a single fusion layer. This approach is called multimodal fusion. Figure 1a provides a simplified visual representation of such a network inspired by a study where the authors classify human actions based on video and audio clips [24], and there are several applications of these methods in different domains, e.g. [33, 34]. Although these methods have shown good performance in various tasks, they do rely on an explicit supervised learning task, which is not readily at hand in our recommendation use case. One option would be to train a model to predict latent MF scores based on the content of items. Although previous research

Figure 1: Schematic representations of multimodal fusion (a) and multimodal contrastive learning (b). Black lines represent feed forward connections, green and red lines represent similarity that is maximized and minimized respectively.



has shown that this can be a useful way to reduce the cold start problem [54], it requires retraining the model each time that MF scores are recomputed, which is inefficient, since training a multimodal fusion model is computationally much more expensive than fitting a MF model.

Unsupervised learning. Multimodal representations can also be learned in an unsupervised way. This can be done using multimodal autoencoders, which let information from two separate modalities pass though a single hidden layer [32, 47], or by using deep belief nets, by first using separate deep belief nets for each modality, concatenating the low dimensional representations, and running this through a deep belief net again [25, 50]. Although both of these unsupervised methods have been used for various tasks, their applications have remained somewhat limited, and research seems to have moved in other directions in recent years.

Contrastive learning. A different approach to multimodal learning that has been receiving considerable attention recently is based on using contrastive learning. Rather than using multiple modalities to predict a single outcome, these methods rely on the co-occurrence of features in different modalities to come to a single multimodal embedding space.

A recent study in this field that attracted a lot of attention introduced Contrastive Language-Image Pretraining (CLIP) [41]. The authors of this paper were able to create a multimodal embedding space for images and text using contrastive learning on a large dataset of image-caption pairs scraped from the internet. Using their network, they were able to attain very impressive performance on several computer vision tasks that it was not specifically trained for. The basic idea is as follows. Both modalities, image and text in this case, have a separate modality-specific encoder network, which is usually initialised using a pretrained network for some task on the relevant modality. Then, some extra feedforward layers are added to the encoder networks in order to make the output dimensions equal for the different modalities. The network is trained contrastively, by maximizing the similarity of the two output layers for image-text pairs that belong together, and minimizing similarity for pairs that do not. Figure 1b provides a simplified visual representation of what such a network could look like for audio-visual contrastive learning. This seemingly simple training task can lead to very informative representations of the input data, as becomes evident from the high performance on an array of different tasks.

In the wake of this paper, several studies using variations on this method have been published. [18] show that the CLIP model can be improved upon by including audio, resulting in a tri-modal embedding space. In [60], the authors take a similar contrastive learning approach by training a contrastive model on short video segments including audio. This allows them to create very informative audio representations, which are shown to be valuable in several downstream tasks such as detection of instrument, music or speech. Another application of multimodal contrastive learning is given in [1]. The authors train a multimodal representation based on video, audio and text data. Specifically the model consists of three transformers based on audio-video-text triplets of short narrated videoclips, as well as some audio-video pairs of YouTube clips.

An important difference between contrastive learning methods and multimodal fusion is the nature of the multimodal representations. Where multimodal fusion uses input from each modality in order to get to a single representation, the contrastive learning methods calculate a separate representation for each modality, as becomes clear from figure 1b. The training objective requires representation of different modalities of the same item to be close, but the representations are indeed separate. This provides some opportunities that fusion methods do not offer. For example, although representations are learned based on both audio and video, contrastive learning representations can also be used to calculate the similarity between two items based on just a single modality. Of course, the separate modality representations can also be aggregated in order to come to a single multimodal representation.

Connection to our research. Overall, multimodal fusion and contrastive learning methods have been most successful in multimodal representation learning. Contrastive learning methods need more data to train, but can learn representations that are independent of a specific task. Fusion methods can often attain similar performance with much less data, but a clear supervised task is needed to train the model. We believe that the contrastive learning approach is most suited to our use case, because there is no clear supervised learning task at hand. Additionally, a contrastive learning model will probably need to be retrained less often in practice. A fusion model would have to be trained to predict usage data. However, usage data is changing all the time, which calls for expensive retraining of the model. Contrastive learning models are only based on the content of the items, which does not change as frequently.

There are two main ways in which our research differs from previous work. The first important difference is that the modalities are much less closely related in our use case. For example, previous research has shown that an image and its caption [41], or a video of an event and the sound that it produces [60], can be represented in a common latent space. However, a piece of music and its corresponding music video are arguably much less directly related to one another. The video is an artistic addition to the music, and one could imagine a single song fitting with different kinds of videos. Learning a multimodal representation for music videos is, therefore, not straightforward, and in this thesis we investigate whether contrastive learning can indeed be useful for these more indirect and abstract relationships between modalities.

A second way in which our work differs from previous work, is in that we try to use multimodal contrastive learning to learn

representations for CB recommendation. Although there are several applications of contrastive learning to historical usage data [67], the application of contrastive learning to CB recommendation remains limited. Although there is one paper that uses contrastive learning to learn representations for music recommendation [69], it is based strictly on the audio modality. To the best of our knowledge, this is the first application of multimodal contrastive learning to CB recommendation.

3 MODEL

In this thesis, we adapt the multimodal contrastive learning approach discussed in section 2.2 to the use case of music videos. The current section details the architecture of the model as well as the loss function that was used.

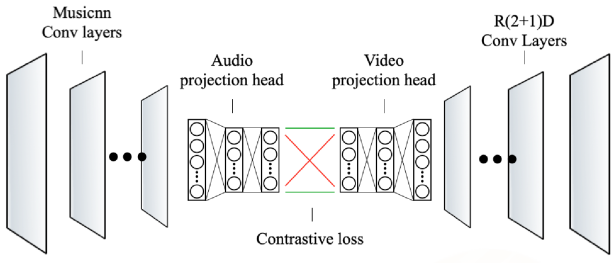
3.1 Architecture

Our model consists of the convolutional layers of two separate pretrained neural networks, with two projection heads added on top, which both consist of two fully connected feedforward layers. Figure 2 presents a schematic overview of this architecture.

Audio encoder. As audio encoder, we use the *musicnn* architecture, which is a convolutional neural network trained on log mel spectograms to predict music tags [38]. The model takes as input a log mel spectogram with 96 bands over three seconds represented as 187 time points. We initialise the network with the pretrained weights, and we do not finetune these weights further during training. A comparative study of several music tagging models showed that the music-specific design choices in *musicnn* allow it to attain better performance than other models for smaller amounts of training data [64]. Since we freeze the *musicnn* weights during training, we think that this music-specific design makes *musicnn* fit for our purpose. On top of the convolutional layers, we added a projection head consisting of two dense feed forward layers. We use a hidden layer size of 512 and an embedding size of 256. These embedding sizes are relatively small compared to some other multimodal networks [13, 41]. We made the decision to keep the embeddings small because we have relatively little training data compared to other studies. The hidden layer uses a ReLU activation function and is trained using a dropout rate of .3. The embedding layer uses a sigmoid activation function.

Video encoder. For the video modality, we use the (2+1)D convolution neural network from [52]. These (2+1)D convolutions separate the convolutions into separate 2D convolutions for space, and 1D convolutions for time. The authors show that the performance of their network is on par with state of the art 3D CNNs while having far fewer parameters. On top of the convolutional layers, we add two dense feedforward layers. Corresponding to the audio encoder, we use a hidden layer size of 512, and experiment with an embedding size 256 and 512. Also, the hidden layer uses a ReLU activation function and a dropout rate of .3, whereas the embedding layer uses a sigmoid activation function. We experiment with several variations to the architecture of these two projection heads, which are discussed in section 4.

Figure 2: Model architecture of our proposed model. It is comprised of the convolutional layers of the pretrained *musicnn* and R(2+1)D networks, and a two layer dense projection head for each modality.



3.2 Loss Function

To train the dense layers of the two encoder networks, we use the contrastive loss function from simCLR [9], adapted to a multimodal scenario. For a batch of N music videos, the audio to video loss for positive pair of audio sample a_i and video sample v_j is defined as:

$$l_{(a,v)} = -log\frac{exp(sim(z_i^a, z_j^v)/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{[k\neq i]} exp(sim(z_i^a, z_k^v)/\tau)} \quad (1)$$

Here $sim(x, y)$ is the dot product of l_2 normalised x and y, z_i^a and z_i^v are the audio and video embeddings for sample i respectively, and $\mathbb{1}_{[k\neq 1]}$ is an indicator function that returns one when $k \neq i$, and zero otherwise. The overall loss of a batch is the sum of $l_{(a,v)}$ and $l_{(v,a)}$ for all positive pairs. In our case, the positive pairs consist of the pairs where audio and video samples are from the same music video, whereas negative pairs consist of pairs where the two modalities stem from different music videos. Minimising this loss function pulls representations of audio and video from the same source closer together, while pushing representations of the audio and video from different sources further apart. The final component of the loss function is τ, which is a temperature parameter that controls the strength of penalties on hard negative samples. When τ is low, negative pairs with high similarity are penalised more heavily, whereas a high temperature parameter results in similar penalties for all negative samples. Research has shown that a low temperature parameter helps to learn separable features, although setting it too low will result in penalising semantically similar negative samples too heavily [58].

4 EXPERIMENTAL SETUP

4.1 Data

Datasets. We use two datasets in this thesis. The main dataset that we use for training and evaluation of our model is a private dataset consisting of 94,656 music videos published to the XITE app. This is excluding videos in the category 'Other/Non-Music'. This category contains videos of live performances as well as non-music videos used for the user interface of the app, which is why we chose to exclude them. In addition to this private dataset, we will also use the million song dataset (MSD) for evaluation, which is a popular public dataset in the field of music processing [4]. For an elaborate description of the XITE dataset and a comparison with the MSD, we refer to appendices A and B. Overall, the main differences are that the music in the XITE dataset is more recent, since the MSD dataset was released in 2010, and has not been updated since. Also, music in the XITE dataset comes from a slightly more diverse set

of origins, although both datasets consist mostly of western music. Finally, the MSD dataset has a larger portion of rock and electronic music, whereas XITE has more hip-Hop and latin tracks.

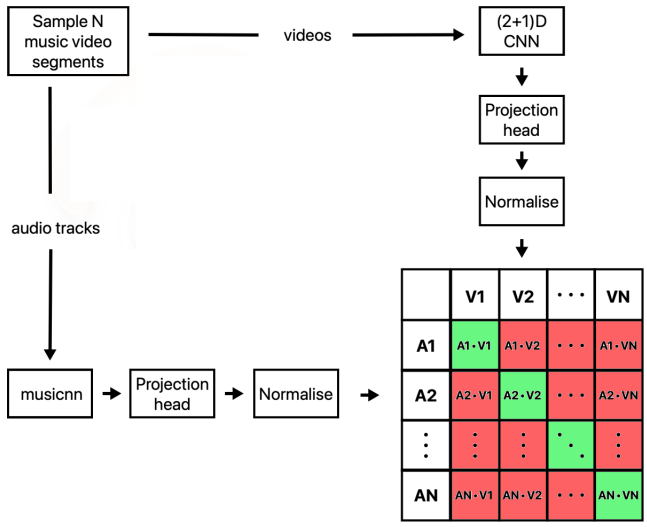
Data processing and storage. Since the storage capacity needed for storing nearly 100k full music videos is not available to us, we decided to store just six five-second non-overlapping segments per music video. We sampled the segments by dividing the track up into six equally sized sections, and sampling a random five second segment from each section. This way segments from as many different parts of the music video as possible are included, and overlap between segments is prevented. Note that model performance might benefit from a more sophisticated sampling strategy. For example, it might be beneficial to extract important segments from music videos using music highlight detection [23] or video summarisation [68, 70]. However, to the best of our knowledge, there is no working open source implementation of these methods that includes functionality for inference on unseen data, and that does not rely on a GPU. One other option that could potentially be interesting is to use chorus detection (e.g. [6, 59]) and to sample segments from the chorus with a higher probability. However, the only working open source implementation of chorus detection that does not rely on lyrics is based on a paper from 2006 [16], and did not show good performance after experimentation on novel data. For these reasons, we decide to stick with our naive sampling method.

It is well known in the video understanding literature that consecutive video frames tend to have high redundancy [61], and researchers have found different ways to sample a subset of frames from a video to process. One of the most simple and common approaches is to sample a random set of frames with equal temporal spacing between them [7, 48]. To achieve this effect, we save five seconds of video to a mp4 file at 432p and 10 frames per second. The lower frame rate allows to sample consecutive frames, which will naturally have an equal temporal spacing of a tenth of a second. The temporal and spatial dimensions of the saved segments (50 frames and 768x432 pixels respectively) are larger than the input sizes of the network, which are 30 frames and 112x112 pixels. This leaves room for temporal and spatial cropping as data augmentations. For the audio modality, we save the corresponding five seconds of audio to a wav file at a sampling rate of 16,000. The stereo channels are converted to mono by taking the average since the *musicnn* architecture is based on a single input channel. For model evaluation purposes, we split the data up into a training split of 78 percent, and a validation and testing split of 11 percent each.

4.2 Training

The suggested pipeline for training the model is represented visually in figure 3. We start by sampling a batch of N music video segments. From each five second segment we randomly sample three seconds of synchronous audio and video. Note that we implement a constraint for music videos to be represented a maximum of once per batch, so that separate segments of the same music video can not appear in the same batch. In addition to this temporal subsampling, we apply random cropping of 112x112 in order to get the correct input shape for the network. To prevent the cropped videos to contain an unreasonably small fraction of the pixels in the original video, we also resized the videos to be smaller, so that the

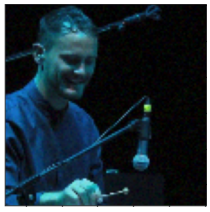
Figure 3: Diagram representing our training pipeline. The audio and video data of a batch of segments is fed into the pretrained networks and projection heads. Cosine similarity between positive pairs (green) is maximized whereas cosine similarity between negative pairs (red) is minimized.



cropped videos still contain a large section of the original frames. Figure 4 shows an example of the effect of this combined resize and crop. Note that in addition to the crop, the processed frame is taken at a slightly different time point than the original, noticeable from the fact that the camera is tilted up slightly. This is due to the temporal subsampling.

Figure 4: Example of the effect of our processing steps on a video frame. These steps consist of a spatial crop and resize, as well as a temporal crop.

(a) First frame from the original clip



(b) First frame from the processed clip

For a batch of N music videos, there are N^2 possible pairs. We take the N matching combinations of audio and video as positive pairs, and the $N^2 - N$ non matching combinations as negative pairs. To make sure there are no 'false negative pairs' we impose the constraint that two segments from the same music video can not appear in the same batch. Since the representations are normalised, the cosine similarity between an audio and a video representation can easily be calculated by taking the dot product. The loss for a pair v_i, a_i is calculated using the loss function in equation 1. In addition to the value of the loss function itself, we track the median rank of the corresponding video segment based on a seed audio segment. This allows us to track whether the training process

is indeed pulling representations from corresponding audio and video segments closer together. As mentioned before, the pretrained convolutional layers are frozen, and we only train the two fully connected layers in the projection head. We use a batch size of 1000, with an initial learning rate of .01 and we use exponential learning decay with a gamma parameter of .95, which has been shown to improve improve training results in previous research [29]. To prevent overfitting, we implement an early stopping rule. Specifically, we stop training when the validation loss does not get any lower over a period of three epochs. In this baseline training setup, we set the temperature parameter to one, which corresponds amounts to using a loss function without a temperature parameter.

Experiments. In addition to our baseline setup, we do several experiments, and track their effect on the training process:

(1) We try using an embedding size of 512 instead of 256.
(2) We test whether using four layers per projection head instead of two improves results.
(3) We experiment with using a single projection head for the video modality, which projects the video embeddings to the dimensions of the *musicnn* embeddings. This way the video representations are moved closer to the audio representations, rather than moving both closer to each other.
(4) We lower the temperature parameter to .3, as suggested in [58]. This penalises hard negatives more heavily, which the authors show helps learning separable embeddings.
(5) We train autoencoders to initialize the weights of the projection heads for the two modalities. Previous research has shown that initializing weights in this way can improve performance on several deep learning tasks [14, 35].

Metrics. To evaluate the outcome of these experiments we look at two separate metrics. Firstly, we look at the value of the loss function defined in section 3.2. Additionally, we track a custom metric in order to check whether corresponding audio and video embeddings are indeed being pulled closer together. For each audio segment in a batch we track the rank of the corresponding video segment in terms of cosine similarity. Similarly, for each video segment in the batch we track the rank of the corresponding audio rank. We aggregate all of these ranks by taking the median, which culminates in a single metric that we will refer to as the *median rank*. Note that a median rank of one would indicate that the corresponding segment is usually also the most similar segment, whereas a value of 500 would mean that corresponding audio and video representations are not more similar than non-corresponding ones.

4.3 Evaluation

We evaluate the model in two ways. First, we use the representations for the two downstream tasks of genre classification and music tagging, and we compare the performance to several baselines from the literature. Then, we investigate the feature representations themselves qualitatively.

Genre classification. For the genre classification task, there is no consensus on a benchmark dataset in the literature. The most used dataset in the field is the GTZAN dataset [53], but subsequent research has pointed out that the quality of the labels in this dataset is poor [36]. This lack of a good common dataset also makes it hard to compare different genre detection models, and to determine what methods are state of the art [42]. In the current study, we perform genre classification on the private XITE dataset, which also allows us to study the effect of the video modality on classification. We fit three different models to predict genre based on our learned representations.

(M1) A model that classifies genre based on the contrastive embeddings calculated from the audio.
(M2) A model that classifies genre based on the contrastive embeddings calculated from the video.
(M3) A model that classifies genre based on the aggregate of the contrastive audio and video embeddings.

To evaluate the effectiveness of our learned representations, we compare these models to three baseline models, which do classification based on the pretrained embeddings.

(B1) A model that classifies genre based on the *musicnn* embeddings.
(B2) A model that classifies genre based on the *(2+1)D CNN* embeddings.
(B3) A model that classifies genre based on the concatenated *musicnn* and *(2+1)D CNN* embeddings.

Each of the models was chosen to be a simple three layer perceptron, with hidden layers sizes of 512 and 256 respectively. This allows us to make several comparisons. Comparing M1 and M2 to B1 and B2 allows us to infer whether fine-tuning models in a contrastive manner improves representation quality. Additionally, comparing M3 to M1 and M2 allows us to infer whether including information from both modalities improves performance on downstream tasks. Finally, M3 is compared to B3 to check whether the effect of having two modalities is greater when both modalities are represented in the same latent space.

Music tagging. For the music tagging task, we use a subset of the Million Song Dataset (MSD) [4], which is a public dataset that consists of a million songs, out of which roughly 240,000 songs include 30 second audio files and tag annotations. For this project we will use the overlap between the private XITE dataset and the public MSD, which allows us to use the video as well as the audio stream of the music videos, while at the same time keeping research as reproducible as possible. The comprehensive list of the 9416 MSD IDs of the overlapping tracks is available on github [56]. The tagging task consists of predicting several tags that were manually annotated based on the audio. These tags contain genres (blues, pop), eras (70s, 80s), and moods (chill, happy) [10, 38]. Since our training set is relatively small, we only use the top ten most common tags. To perform the music tagging task using our representations, we use the same approach as for genre classification. The three baselines do classification using the pretrained embeddings, whereas the three models use as input the embeddings from the model that was fine-tuned using contrastive learning. The model consists of a three layer perceptron with hidden layer sizes 512 and 256 respectively. For both experiments, the metrics of interest are area under the ROC curve (AUC) and F1 score.

Qualitative analysis of representations. In addition to evaluating the quality of feature representations on downstream tasks, we also evaluate them qualitatively. We do this both for the separate modalities and on a multimodal level. Additionally, we look both at individual segments and entire music videos. To come to a representation for an entire music video we aggregate the representations

of the different segments of that video, and to get a multimodal representation, we aggregate the audio and video representations.

We investigate the representations by selecting random seed segments, and retrieving the three most similar segments to this seed segment. By investigating the items that are retrieved for different seeds we are able to get an insight into what characteristics of the video and audio streams of a music video are represented by the learned embeddings. By doing the same thing for aggregated representations for different segments, we test whether aggregation is a valid way of attaining an overall music video representation, which is of course very relevant for music video recommendation.

5 RESULTS

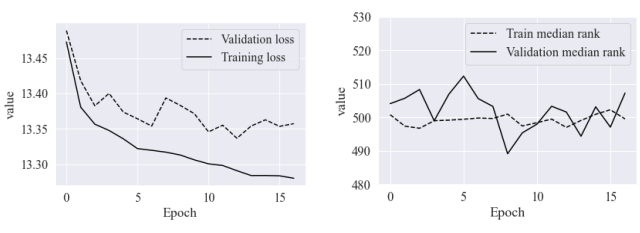
Training. During contrastive training the value of the loss function as well as the median rank of the corresponding video segment was tracked. This allows us to get an insight into whether the training process is succeeding in pulling representations of corresponding segments closer to each other. Figure 5 shows the training and validation metrics during the training process. Although the loss decreases somewhat initially, this is not reflected in the median rank of corresponding segments. The median rank of corresponding segments does not manage to fall below 500 consistently. Taking into account that we are using a batch size of 1000, this indicates that the training process is failing to pull embeddings for corresponding audio and video segments closer to each other, relative to non-corresponding segments. This is a serious concern for our project, as uniting embeddings from different modalities is the fundamental idea of contrastive learning.

Experiments. As discussed in section 4.2 we executed several experiments in order to enhance the training process. Table 1 presents the value for the loss function as well as the median rank of the corresponding segment for all of the experiments. Although both decreasing the temperature parameter to .3 and initializing the projection head weights using autoencoders leads to slightly lower values for the loss function, the median rank stays close to 500 for all models, indicating that none of the models succeed to unite audio and video embeddings.

5.1 Downstream tasks

Music tagging. Table 2 shows the average area under the curve and F1 score for each baseline and model on the downstream task of music tagging. The scores represent macro-averages, indicating that

Figure 5: Training and validation metrics during contrastive training of the baseline model. The loss function shows a slight decrease whereas the median rank does not drop below chance level consistently.



(a) Loss function **(b) Median rank**

Table 1: Effect of experiments 1-5 on the loss function and the median rank.

Type of embeddings	Loss	Median rank
0. Baseline	13.357	507
1. Embedding size 512	13.340	494
2. Four projection layers	13.380	503
3. Single projection head	13.353	499
4. Temperature .3	13.111	497
5. Autoencoder initiated	13.082	501

we first calculate the scores per label, and then take the unweighted mean over tags. We refer to appendix C for the complete table with scores per tag. Overall, these results present two main findings. Firstly, the networks that are trained on embeddings that include information about audio clearly outperform models that are only based on video. This is in line with expectations, as the MSD tags are based solely on the audio of a song, and do not take into account the music video that goes with the audio. Secondly, the performance of the baseline models is better than the performance of the models that were fine-tuned using contrastive learning. We expected the model based on contrastively learned embeddings to outperform the baselines, but given the training issues discussed above, these results are less surprising.

Table 2: Macro average AUC and F1 score for the downstream task of music tagging.

type of embeddings	AUC	F1
(B1) Musicnn	.7762	.2182
(B2) R(2+1)D	.6566	.0881
(B3) Concatenated	.7839	.2390
(M1) Contrastive audio	.7019	.1040
(M2) Contrastive video	.6092	.0934
(M2) Contrastive aggregated	.7082	.1051

Genre classification. The results for the downstream task of genre classification are similar to the results for music tagging. Table 9 presents the same two statistics for the task of genre classification. Similar to the results above, we see that models including audio data outperform models that are solely based on video, and more importantly, that all three baselines outperform their contrastive learning counterparts.

Table 3: Macro average AUC and F1 score for the downstream task of genre classification.

Type of embeddings	AUC	F1
(B1) Musicnn	.7902	.1781
(B2) R(2+1)D	.6789	.0531
(B3) Concatenated	.8199	.2060
(M1) Contrastive audio	.6919	.0754
(M2) Contrastive video	.6293	.0437
(M2) Contrastive aggregated	.6804	.0680

5.2 Qualitative analysis of representations

In addition to the downstream tasks, we also evaluated the learned embeddings qualitatively by comparing several seed music videos to the most similar music videos according to the representations.

We do this both on the level of music videos and on the level of individual segments

Music videos. We started by aggregating the embeddings for different segments and modalities. We first took the average over segments, to come to a single audio and a single video representation, and then took the average to get to a final multimodal representation. To explore these representations, we selected 25 random seed videos, and retrieved the three most similar music videos to each seed video. Originally, we planned to have a group of human judges compare the most similar videos according to the contrastive representations to the most similar music videos according to the metadata. However, it soon became clear that the similarity metric based on the contrastive representations was inferior to the point that such a user study was redundant. In general, the most similar music videos did not appear to be similar to the seed video at all in terms of both audio and video. One group of music videos that formed an exception consisted of live performances, which would often be most similar to other videos of live performances. When taking into account only the video embeddings, rather than the aggregate of audio and video embeddings, some other patterns were present. In addition to videos of live performances, some seed videos that were in black and white, as well as some videos of people dancing or playing instruments, would result in similar videos being retrieved. One thing that these videos have in common, is that they are very consistent in terms of video across different parts of the music video. This could point to the fact that aggregating representations of different segments of the same music video is not a valid way to get a representation of the music video, especially when different parts of the music video are very inconsistent. This is particularly often the case for the video modality, as music videos often consist of a wide variety of shots and scenes.

Segments. In order to investigate this potentially negative effect of aggregating the different segments of a music video, we also examined similarity on the level of segments. We took the same approach as before, this time selecting 20 random seed segments, and retrieving the most similar segments to this seed segment. As expected, the results were somewhat better for single segments. There were more cases in which the retrieved videos were similar to the seed, although in general the retrieved videos still did not seem similar. For example, some seeds for which the retrieved segments were similar were close-ups of women singing, shots of people playing guitar, segments with a dark background with bright neon accents, and shots of men talking. Overall similar segment retrieval was still poor, but there were more cases in which there was some specific concrete similarity between the videos.

6 DISCUSSION AND LIMITATIONS

Our results demonstrate the difficulty of representing the audio and video of music videos in a common space. In this section, we discuss some of the potential reasons for these issues, as well as some limitations to our approach.

Discussion. As mentioned previously, the main challenge for multimodal learning consists of bridging the heterogeneity gap [17]. Different modalities contain different type of features, and the challenge is to project these features into a single space in which both similarity within a modality and similarity between modalities can be meaningfully calculated. Contrastive learning is one approach to overcome this problem, and some notable successful applications involve learning common representations for images and their captions [41], learning common representations for videos and their descriptions [72] and finally, learning common representations for videos and their audio streams [60]. Note that compared to our music video use case, the heterogeneity gap in all of these studies is relatively small. Although an image and its caption are represented in very different ways, they are closely related, as the caption tries to describe exactly what is in the image: the two modalities both refer to the same concept. This is also true for the use case of videos and their descriptions, and a video of an event and the sound of that video. However, for a piece of music and the video that goes with it, the relationship between the two is much less direct. As opposed to the other examples, the two modalities in a music video are not both directly referring to a single concept. Rather than describing the music, the video modality serves as an extra means to entertain the user and to convey emotion. This large heterogeneity gap is reflected in our qualitative analysis of similarity as well as our exploratory data analysis. Whereas the music network mainly encodes for features relating to the types of (digital) instruments used, like distorted guitars or loud electronic bass drums, the video network encodes for features like the color of the video as well as specific actions like singing or dancing. The relationship between these features is not always immediately apparent. Different videos that use a black and white video can be very different in what type of instruments they use, and the fact that a music video contains people dancing, will not tell you much about the instruments that are being used in the song. This is quite different from the video-audio use case from previous research, where different different videos of similar events are likely to have similar audio streams, since the event in the video is often the direct source of the audio [60]. Of course there are certainly correlations between a song and the type of music video that goes with it, and we are not arguing that the different modalities in previous research were always exactly related, and that audio and video of music videos are completely unrelated. However, we do believe that the modalities are considerably less tightly connected in our research, which could be an important cause of the difficulties we encountered trying to unite embeddings from the two modalities.

Limitations. The first important limitation to our approach concerns the input size of the encoder networks. Due to the high dimensionality and complexity of the input data, combined with our limited computational capacity, we were limited to segments of only three seconds during training. This makes the task of retrieving corresponding segments from the other modality significantly harder. Even for a human, retrieving the corresponding piece of music based on a three second videoclip is difficult. Additionally, this short input length makes it harder to compute the similarity between complete music videos, which is necessary for content based recommendation. We tried aggregating representations of different segments to come to a single representation per music video, but our qualitative analysis of similarity revealed that this aggregation was degrading to the performance, especially so when the input data was inconsistent across segments, which is often the case for video data from music videos. It might be more fruitful to use a larger input size, or to extract features that are more consistent across the duration of the music video.

A second limitation concerns the fact that we leave the weights of the pretrained networks frozen during training. Due to the limited computational capacity available to us, we decided not to fine-tune the pretrained networks that we used as feature extractors. It is possible that freezing these layers does not leave enough freedom for the network to learn a feature space in which audio and video embeddings can be united. Although many previous applications of contrastive learning do train all layers of the network [1, 41, 60], there are also various earlier studies that freeze the pretrained layers and come to successful results [66, 72]. It is possible that the extra freedom gained from tuning the entire network would help bridge the large heterogeneity gap, but we believe that it is unlikely this would fix the problem, since fine-tuning generally just leads to a moderate improvement in performance. Of course another option would be to train the entire model form scratch, but this would require a dataset that is several orders of magnitude larger than ours.

7 CONCLUSION

This thesis has established the difficulty of uniting representations for songs and their music videos. Two projection heads trained using a contrastive objective were not able to effectively unite embeddings from corresponding audio and video segments. This failure to bring audio and video embeddings closer together was reflected in performance on downstream tasks as well as a qualitative analysis of embeddings. Models based on the contrastively learned embeddings performed worse at both genre classification and music tagging compared to models that were using pretrained networks without contrastive fine-tuning. Additionally, a qualitative investigation of similarity based on the contrastive representations revealed that music videos with dissimilar content will often have relatively similar representations. Similarity on the level of segments, without aggregation, resulted in more cases in which the similarity makes sense, although in general it still did not work well. Possible explanations for these problems are the discrepancy between songs and their music videos, and the short input size of the encoder networks combined with the inconsistency of the data throughout the duration of a music video. In the next paragraph, we offer two potential directions for future work to deal with these issues.

Future research. Firstly, given the large heterogeneity gap, it would be useful for future research to look into multimodal alternatives to contrastive learning that rely less heavily on the association between features in different modalities. Multimodal fusion might for example be more suitable in our use case, as these models combine features from different modalities based on a supervised task, rather than solely based on the relationship between features in the different modalities [33]. An example of a supervised learning task for a multimodal network used for recommendation could be predicting latent matrix factorisation scores [54].

Secondly, we discussed the small input size of the encoder networks as a potential reason for the poor results. Future research could try to increase this input size in order to learn embeddings that are more representative for entire music videos. However, current research on video processing is based on short videoclips of single events [43], so learning meaningful representations for longer, more diverse videos might be more challenging than just increasing the input length of the network. Ideally, the encoder network for video would learn features relating to the general style and feeling of the video, rather than features related to concrete actions or events, as the style is usually relatively consistent over the music video, as opposed to actions, which differ a lot across shots and scenes. Although most research on deep learning for video data consists of classifying action or events, detecting faces or estimating poses [45], there is also a class of models concerned with style transfer for video data [8, 28]. These methods could be relevant to future research on multimodal music video recommendation, as they aim to learn features related to style rather than content, which is more consistent throughout a music video, and might also be more closely related to the musical content of the song.

REFERENCES

[1] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. 2021. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems* 34 (2021).

[2] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. 2010. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems* 16, 6 (2010), 345–379.

[3] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* 41, 2 (2018), 423–443.

[4] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset. (2011).

[5] Ludovico Boratto, Gianni Fenu, and Mirko Marras. 2021. Connecting user and item perspectives in popularity debiasing for collaborative recommendation. *Information Processing & Management* 58, 1 (2021), 102387.

[6] Zhengyu Cao, Yongwei Gao, and Wei Li. 2020. Chorus Detection Using Music Structure Analysis. In *National Conference on Sound and Music Technology*. Springer, 3–17.

[7] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[8] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*. 1105–1114.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

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

[11] Mittal C Darji. 2017. Audio signal processing: A review of audio signal classification features. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 2, 3 (2017), 227–230.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[13] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* (2017).

[14] Mafalda Falcao Ferreira, Rui Camacho, and Luis F Teixeira. 2020. Autoencoders as weight initialization of deep classification networks for cancer versus cancer studies. *arXiv preprint arXiv:2001.05253* (2020).

[15] Yuan Gong, Yu-An Chung, and James Glass. 2021. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778* (2021).

[16] Masataka Goto. 2006. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing* 14, 5 (2006), 1783–1794.

[17] Wenzhong Guo, Jianwen Wang, and Shiping Wang. 2019. Deep multimodal representation learning: A survey. *IEEE Access* 7 (2019), 63373–63394.

[18] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. 2021. Audioclip: Extending clip to image, text and audio. *arXiv preprint arXiv:2106.13043* (2021).

[19] Curtis Hawthorne, Ian Simon, Rigel Swavely, Ethan Manilow, and Jesse Engel. 2021. Sequence-to-sequence piano transcription with transformers. *arXiv preprint arXiv:2107.09142* (2021).

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] Juan Hu, Xin Liao, Wei Wang, and Zheng Qin. 2021. Detecting Compressed Deepfake Videos in Social Networks Using Frame-Temporality Two-Stream Convolutional Network. *IEEE Transactions on Circuits and Systems for Video Technology* (2021).

[22] Xiaohui Huang, Pan He, Anand Rangarajan, and Sanjay Ranka. 2020. Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6, 2 (2020), 1–28.

[23] Yu-Siang Huang, Szu-Yu Chou, and Yi-Hsuan Yang. 2018. Pop music highlighter: Marking the emotion keypoints. *arXiv preprint arXiv:1802.10495* (2018).

[24] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. 2017. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE transactions on pattern analysis and machine intelligence* 40, 2 (2017), 352–364.

[25] Yelin Kim, Honglak Lee, and Emily Mower Provost. 2013. Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 3687–3691.

[26] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[27] Jongpil Lee, Taejun Kim, Jiyoung Park, and Juhan Nam. 2017. Raw waveform-based audio classification using sample-level CNN architectures. *arXiv preprint arXiv:1712.00866* (2017).

[28] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. 2019. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3809–3817.

[29] Zhiyuan Li and Sanjeev Arora. 2019. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454* (2019).

[30] Xin Liu, Qingcai Chen, Xiangping Wu, Yan Liu, and Yang Liu. 2017. CNN based music emotion classification. *arXiv preprint arXiv:1704.05665* (2017).

[31] Bhaskar Mitra, Nick Craswell, et al. 2018. *An introduction to neural information retrieval*. Now Foundations and Trends Boston, MA.

[32] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *ICML*.

[33] Behnaz Nojavanasghari, Deepak Gopinath, Jayanth Koushik, Tadas Baltrušaitis, and Louis-Philippe Morency. 2016. Deep multimodal fusion for persuasiveness prediction. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*. 284–288.

[34] Juan DS Ortega, Mohammed Senoussaoui, Eric Granger, Marco Pedersoli, Patrick Cardinal, and Alessandro L Koerich. 2019. Multimodal fusion with deep neural networks for audio-video emotion recognition. *arXiv preprint arXiv:1907.03196* (2019).

[35] Tom Le Paine, Pooya Khorrami, Wei Han, and Thomas S Huang. 2014. An analysis of unsupervised pre-training in light of recent advances. *arXiv preprint arXiv:1412.6597* (2014).

[36] Haukur Pálmason, Björn Þór Jónsson, Markus Schedl, and Peter Knees. 2017. Music genre classification revisited: An in-depth examination guided by music experts. In *International Symposium on Computer Music Multidisciplinary Research*. Springer, 49–62.

[37] Dip Paul and Subhradeep Kundu. 2020. A survey of music recommendation systems with a proposed music recommendation system. In *Emerging technology in modelling and graphics*. Springer, 279–285.

[38] Jordi Pons and Xavier Serra. 2019. musicnn: Pre-trained convolutional neural networks for music audio tagging. *arXiv preprint arXiv:1909.06654* (2019).

[39] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-Yiin Chang, and Tara Sainath. 2019. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing* 13, 2 (2019), 206–219.

[40] Lvyang Qiu, Shuyu Li, and Yunsick Sung. 2021. DBTMPE: Deep bidirectional transformers-based masked predictive encoder approach for music genre classification. *Mathematics* 9, 5 (2021), 530.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

[42] Jaime Ramírez and M Julia Flores. 2020. Machine learning for music genre: multifaceted review and experimentation with audioset. *Journal of Intelligent Information Systems* 55, 3 (2020), 469–499.

[43] Qiuyu Ren, Liang Bai, Haoran Wang, Zhihong Deng, Xiaoming Zhu, Han Li, and Can Luo. 2019. A Survey on Video Classification Methods Based on Deep Learning. *DEStech Transactions on Computer Science and Engineering* cisnrc (2019).

[44] Tara N Sainath and Bo Li. 2016. Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks. (2016).

[45] Gabriel NP dos Santos, Pedro VA de Freitas, Antonio José G Busson, Álan LV Guedes, Ruy Milidiú, and Sérgio Colcher. 2019. Deep learning methods for video understanding. In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*. 21–23.

[46] Alexander Schindler and Andreas Rauber. 2015. An audio-visual approach to music genre classification through affective color features. In *European conference on information retrieval*. Springer, 61–67.

[47] Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 721–732.

[48] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems* 27 (2014).

[49] Janne Spijkervet and John Ashley Burgoyne. 2021. Contrastive learning of musical representations. *arXiv preprint arXiv:2103.09410* (2021).

[50] Nitish Srivastava and Ruslan Salakhutdinov. 2012. Learning representations for multimodal data with deep belief nets. In *International conference on machine learning workshop*, Vol. 79. 3.

[51] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[52] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 6450–6459.

[53] George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing* 10, 5 (2002), 293–302.

[54] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. *Advances in neural information processing systems* 26 (2013).

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

[56] K.A. Veldkamp. 2022. Multimodal music video representation. https://github.com/KarelVeldkamp/Multimodal-Musicvideo-Representation.

[57] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. Content-based neighbor models for cold start in recommender systems. In *Proceedings of the Recommender Systems Challenge 2017*. 1–6.

[58] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2495–2504.

[59] Ju-Chiang Wang, Jordan BL Smith, Jitong Chen, Xuchen Song, and Yuxuan Wang. 2021. Supervised chorus detection for popular music using convolutional neural network and multi-task learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 566–570.

[60] Luyu Wang, Pauline Luc, Adria Recasens, Jean-Baptiste Alayrac, and Aaron van den Oord. 2021. Multimodal self-supervised learning of general audio representations. *arXiv preprint arXiv:2104.12807* (2021).

[61] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*. Springer, 20–36.

[62] Wikipedia contributors. 2022. List of most-viewed YouTube videos — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos [Online; accessed 2-February-2022].

[63] Minz Won, Keunwoo Choi, and Xavier Serra. 2021. Semi-Supervised Music Tagging Transformer. *arXiv preprint arXiv:2111.13457* (2021).

[64] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. 2020. Evaluation of CNN-based automatic music tagging models. *arXiv preprint arXiv:2006.00751* (2020).

[65] Xite. [n.d.]. Xite Home page. https://xite.nl/. Accessed: 08-02-2022.

[66] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084* (2021).

[67] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-Supervised Learning for Recommender Systems: A Survey. *arXiv preprint arXiv:2203.15876* (2022).

[68] Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[69] Hongyuan Zhu, Ye Niu, Di Fu, and Hao Wang. 2021. MusicBERT: A Self-supervised Learning of Music Representation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3955–3963.

[70] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. 2020. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing* 30 (2020), 948–962.

[71] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2019. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering* 32, 4 (2019), 631–644.

[72] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. 2021. Crossclr: Cross-modal contrastive learning for multi-modal video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1450–1459.

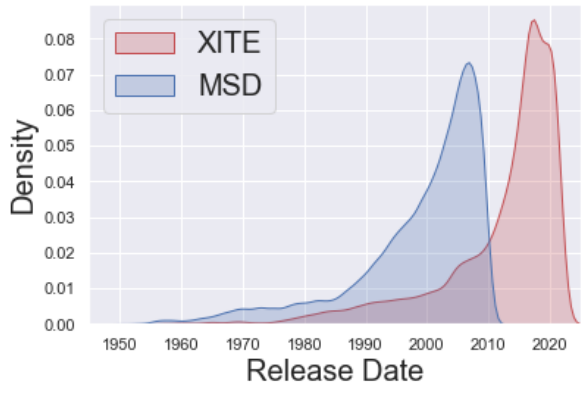
A EXPLORING THE METADATA

This appendix is meant to characterize the metadata of the XITE dataset, as it is a private dataset. It aims to describe some of the features in the XITE dataset, and compare it to a publicly available dataset.

XITE and MSD. There are currently 97.241 videos published to the XITE app. In addition to the audio and video data itself, XITE has metadata concerning the artist, genre, sub-genre, release date, origin and duration of each music video. Additionally, XITE keeps data on three more subjective, manually tagged features: mood, energy and urgency. Since this a private dataset, we will describe some of the characteristics of the dataset, and compare it to a public dataset that it popular in the field of music information retrieval. Specifically, we will compare the XITE dataset to the MSD [4]. This dataset consists of roughly one million songs with metadata and audio features. Additionally, roughly 240.000 of the songs include manually labelled tags, which include features like genre ('Pop', 'Rock'), decade (70s, 90s), origin ('British', 'Dutch') and general descriptions ('Female', 'Bass', 'Acoustic'). Of course one main important difference between the datasets is that the XITE dataset concerns music videos, whereas the MSD just contains songs. Although there is one paper that introduces a music video dataset [46], this just contains features based on the audio and video data, as opposed to the actual signals themselves. To the best of our knowledge, there is no publicly available academic music video dataset.

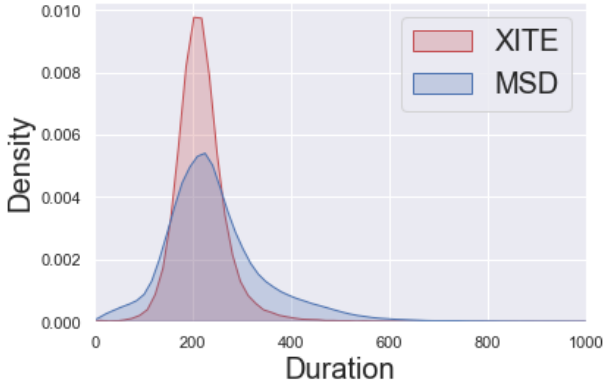
Release date. Figure 6 provides the empirical distributions of release date in both datasets. The distribution of release dates in the XITE dataset is similar to the distribution in the MSD, but shifted to the right. This is due to the fact that the XITE app is constantly getting updated with new music videos, whereas the MSD has remained the same since its release in 2011.

Figure 6: Density plot of release dates for the MSD and the XITE dataset.



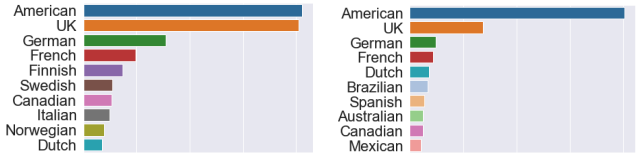
Duration. In terms of the average duration of songs, the two datasets are similar, although figure 7 shows that the MSD has more variation in song duration, whereas songs in the XITE dataset show less variation in their duration. This could potentially be because of the fact that the XITE dataset only consists of music videos, as opposed to songs. A possible explanation is that music videos generally have less variation in their length than songs, but we did not test this hypothesis.

Figure 7: Density plot of duration for the MSD and the XITE dataset.



Origin. We also compare the datasets in terms of the music origin. For the XITE dataset, getting a distribution of origin is easy, as it is one of the variables in the dataset, but for the MSD it is slightly more complicated. The MSD does contain information about origin, but it is part of the MusicBrainz tags, so there is no separate category for origin. Note that tags are on artist level, and not on song level. To get an estimate of the distribution of origin in the MSD, we selected the top 10 MusicBrainz tags that were concerned with countries. However, there were various duplicates in the tags. For example, there are separate tags for 'UK' and 'British', and for 'Canada' and 'Canadian'. In these cases, we decided to count only the most occurring tag. Note that aggregating two tags with the same meaning would not have the desired effect, as a single artist often has both tags. Therefore, this would lead to a distribution that is biased towards countries with multiple tags. Figure 8 shows that the majority of songs in both datasets comes from either the United States or Britain. The remaining countries that appear often are mostly western countries, although Brazil and Mexico do appear in the top ten countries of the XITE dataset.

Figure 8: Empirical distribution of the top ten origin countries in the MSD and XITE dataset.

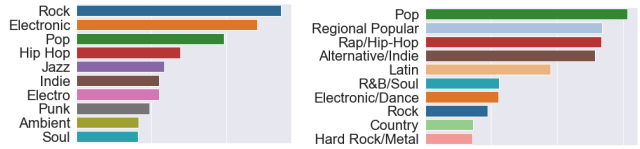


(a) MSD (b) XITE

Genre. To get an estimate of the distribution of genre, we take a similar approach. For the XITE dataset, we simply use the genre feature, and for the MSD, we look at the Echo Nest tags, filtering out tags that are not related to genre. As we did for origin, we filter out tags that overlap in meaning, such as 'Rock' and 'Alternative Rock', because there are no 'Alternative Rock' songs without the 'Rock' tag. In these cases we kept the most common tag. Figure 9

shows that Hip Hop and Indie are more common on XITE, whereas Electronic and Jazz are much more frequent in the MSD.

Figure 9: Empirical distribution of the top ten genres in the MSD and XITE dataset.



(a) MSD **(b) XITE**

B EXPLORING FEATURE REPRESENTATIONS

In this appendix we discuss our exploration of the audio and video data through the pretrained networks that we propose to use for the project: the *Musicnn*, and the *R(2+1)D CNN*. We sampled 3000 random music videos to explore, and fed them into the networks. We start by discussing our exploration of audio representations. Then we move on to the video representations, and finally we take a look at the relationship between the two.

B.1 Audio

For the audio modality, we sampled segments from the audio by splitting the track into six equally long subsections, and sampling a random thee second segment from each section. These segments were fed into the musicnn network. We used only the first seven layers of the network, omitting the two dense classification layers. The output are vectors of size 1506. To calculate a feature representation for a video, we aggregated the feature representations of the six segments by taking the element wise mean. To attempt to understand this high dimensional representation of the audio segments better, we used t-SNE dimensionality reduction [55], which is a useful tool for visualizing high dimensional data. t-SNE tries to map points in a high dimensional space to a two or three dimensional space, where similar items are closer together in the lower dimensional space. We ran t-SNE with two components using a perplexity of 800 and we plotted the first two components in figure 10. Based on the plot, there was no clear clustering of the different audio segments. Also, there was no clear relationship between the two t-SNE scores and genre. We investigated the t-SNE scores further by creating an interactive plot that plays the audio of a song when a point in the plot is clicked. This revealed that t-SNE scores are mostly related to certain sounds or instruments. For example, songs with a high score on the second component were very melodic, whereas songs with a low score on the second component were more percussion based, oftentimes with loud electronic drums. Out of the more melodic songs, the ones based on piano or vocals scores lower on the first t-SNE score, whereas songs with distorted guitars had a higher first t-SNE score.

Figure 10: t-SNE plot of musicnn feature representations

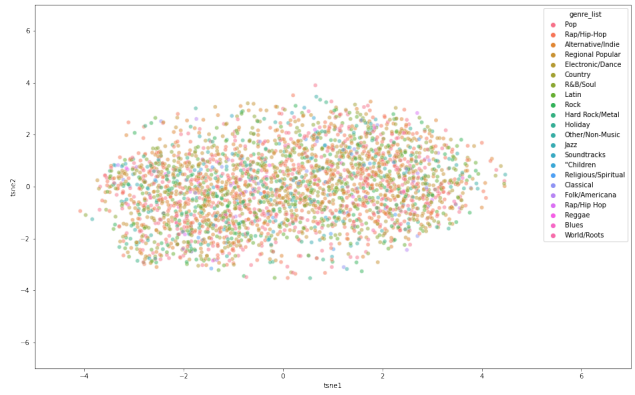
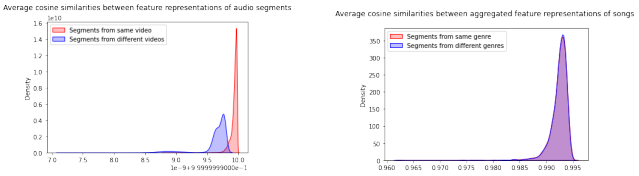


Figure 11: Average cosine similarities between audio feature vectors



(a) Same video vs different video **(b) Same genre vs different genre**

In addition to looking at the t-SNE scores, we looked at the cosine similarities between feature representations of different audio segments. Figure 11a shows that the average cosine similarity between different segments of the same song is higher than the average cosine similarity between segments of different songs. We also explored whether the feature representations of songs of the same genre are more similar than representations for songs of different genres. To do this, we took the same audio segments as before, and calculated a song representation by averaging the feature representations of the different segments of the song. as can be seen in figure 11b feature representations for songs of the same genre are not more similar on average than feature representations of songs of two different genres. This corresponds to the t-SNE decomposition, which did not show a clear relationship between feature representations and genre.

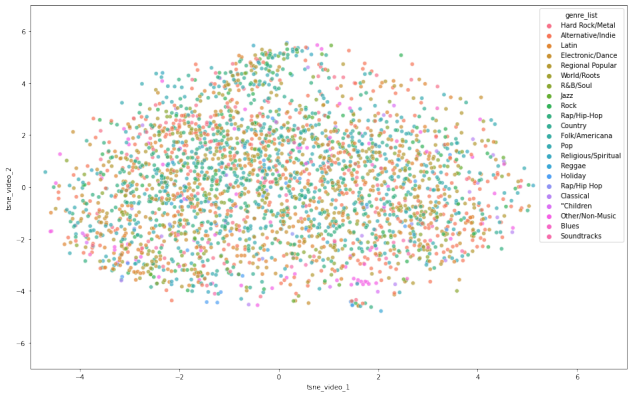
Overall, these results show that although the feature representations do not seem to directly relate to metadata features like genre, they do clearly represent different acoustical features of the music, mostly based on the specific types of sounds or instruments that are used.

B.2 Video

Our approach to exploring video data was similar to our exploration of audio data. We sampled six three second segments from the same 3000 music videos used for the audio part. The segments were fed into the 3D ResNet model, using only the convolutional layers, and omitting the classification layers. This led to output vectors of size 512. The high dimensional representations were analysed with t-SNE dimensionality reduction, and the component scores for the two factors are plotted in fig 12 The graph does not show a clear relationship between genre and t-SNE scores, and there are no obvious big clusters in the t-SNE scores. as with the audio data, we explored the t-SNE scores further using an interactive plot. Compared to the t-SNE scores for audio, it was less clear to spot a large scale pattern in the t-SNE scores. However, on a lower level, similar videos were definitely close together. For example, there is

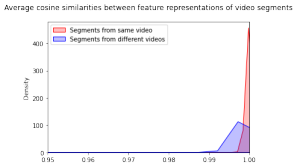
a small compact cluster at (1.75, -4). Exploring these videos shows that these are almost all animated/cartoon style music videos. Right above it, at (1.75, -3.5) there is another small cluster of videos that are in the category 'Other/Non-Music'. Inspection showed that nine out of the ten music videos in this cluster are live performances of songs at the VEVO studios. There is another larger but less compact/well defined cluster around (0, 5). Most videos in this region are black and white videos. In the bottom left of the graph, around the area of (-3, -3), there is no clear cluster in the t-SNE scores, but most videos in this area are videos of live performances on big stages for a large crowd. Overall, we can conclude that although there is no clear clustering structure available in the t-SNE scores of the video data, videos with similar visual content are often grouped together, indicating that the feature representations from the ResNet 3D model contain useful information about the video component of the music videos.

Figure 12: t-SNE plot of ResNet 3D feature representations

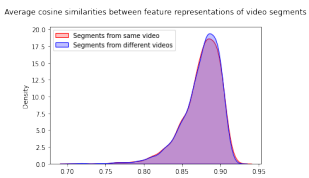


As with the audio representations, we investigate the cosine similarities between audio segments of different songs and different segments of the same song, as well as the cosine similarities of aggregated representations of songs from the same genre and videos from different genres. The results are plotted in figure 13. Again, different segments of the same video seem to be more similar that segments of different videos, although this distinction was more clear for audio than for video. There was no clear relationship between genre and cosine similarity, corresponding to our findings for audio.

Figure 13: Average cosine similarities between video feature vectors



(a) Same video vs different video

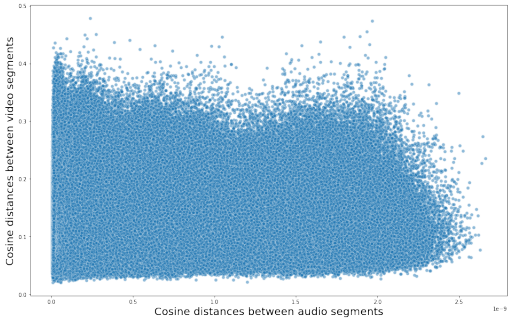


(b) Same genre vs different genre

B.3 Relationship between audio and video

In addition to investigating the feature representations of audio and video separately, we also look into the relationship between the two. We did this by calculating the cosine similarities of all pairs of audio representations and the cosine similarities of all pairs of video representations. This resulted in two square, size 3000 matrices. To look at the relationship between the two, we calculated the Pearson correlation , only taking into account the off-diagonal and skipping duplicates. The correlation coefficient which was .02 ($p < .001$), indicating that there is no strong linear relationship between the between-video similarities between video representations and the cosine distances between audio representations. The visualisation in figure 14 also showed no clear non linear relationship between the between-video similarities between representations of the two modalities. This shows that currently, representations in the audio and video modalities are not closely related. The aim of this project is to bring these representations closer to each other, hoping that this will improve the representation quality.

Figure 14: Relationship between the cosine similarities of audio representations and cosine similarities of video representations



C ADDITIONAL RESULTS

Table 4: Area under the curve per tag for the downstream task of music tagging.

Type of embeddings	Country	Alt Rock	Dance	RnB	indie	Female vocals	Alternative	80s	Pop	Rock
(B1) Musicnn	.8371	.7845	.8476	.8837	.6980	.7317	.7423	.7861	.6494	.8026
(B2) R(2+1)D	.6293	.7299	.7156	.7527	.5687	.6754	.5811	.7299	.5983	.6745
(B3) Concatenated	.8305	.7915	.8431	.8934	.6858	.7526	.7566	.8142	.6626	.8090
(M1) Contr. audio	.6325	.7719	.7696	.8468	.6551	.5743	.7364	.6037	.6462	.7827
(M2) Contr. video	.5371	.6170	.6264	.6842	.5308	.6520	.5781	.5928	.5955	.6789
(M2) Contr. aggregated	.6180	.7718	.7595	.8424	.6448	.6009	.7311	.6931	.6340	.7860