POLITECNICO DI TORINO

MASTER's Degree in DATA SCIENCE AND ENGINEERING



MASTER's Degree Thesis

Artist Similarity Assessment Using Instagram Video Content

Supervisors

Prof. Cristina ROTTONDI

Prof. Massimiliano ZANONI

Prof. Andrea BIANCO

Candidate

Xiangbo GONG

SEPTEMBER 2026

Artist Similarity Assessment Using Instagram Video Content

Xiangbo Gong

Abstract

This thesis examines whether visual information from artist-related videos can support artist similarity learning. It addresses the video branch of a broader multimodal artist-similarity project and evaluates this modality independently before combining it with audio, captions, or static images. Frame-level Contrastive Language-Image Pre-training (CLIP) embeddings provide the visual input.

Each artist is represented by up to ten videos, with thirty sampled frames per video and a 768-dimensional CLIP embedding per frame. The main model, TripletNet1, preserves this structure through a hierarchical Transformer. Its first level summarises the frames within each video; its second aggregates the resulting video tokens into one artist representation. Training uses triplet loss, cosine distance, and similarity relations derived from AllMusic and Spotify. Three one-dimensional convolutional (Conv1D) baselines instead average the video dimension before learning.

Artist-disjoint cross-validation reduces leakage between the training and validation sets. The evaluation uses Mean Reciprocal Rank (MRR), Precision@K, and Hit Rate at K (HitRate@K) for retrieval. The best TripletNet1 configuration reaches a validation MRR of 0.3114, compared with 0.2787 for the strongest convolutional baseline. In full-catalogue retrieval, TripletNet1 obtains an MRR of 0.4055, Precision@1 of 0.2759, and HitRate@5 of 0.5657. Pairwise and latent-space analyses indicate that the representation separates positive and negative artist pairs, although the results remain constrained by the incomplete and subjective ground truth.

The results indicate that video-derived visual cues provide useful information for artist similarity modelling. Because the model does not use the audio track and relies on incomplete platform relations, its learned space is not a complete measure of musical similarity. It is instead a video-based representation of artist identity that can later be combined with audio, text, and static-image branches in a multimodal system.

ACKNOWLEDGMENTS

I am sincerely grateful to my main supervisor, Professor Cristina Rottondi, for her guidance, patience, and thoughtful feedback throughout this thesis. Her support shaped the project from its initial direction to the work presented here. I also thank Professors Andrea Bianco and Massimiliano Zanoni for their advice and the time they devoted to my research.

I thank Davide Lista and Giacomo Sansoni for working alongside me and for the discussions, ideas, and practical help we shared during the project. I am equally grateful to my family and friends for their patience and encouragement throughout my studies.

Table of Contents

1 Introduction **1**

2 State of the Art **4**

2.1 MIR and Artist Similarity 4

2.2 Techniques for Artist Similarity 4

2.2.1 Content-Based Techniques 4

2.2.2 Context-Based Techniques 5

2.2.3 Multimodal Techniques 6

2.3 Visual Representation Learning for Artist Identity 6

2.4 CLIP and Transferable Visual Embeddings 7

2.5 Recent Artist Similarity Models 7

3 Theoretical Background **10**

3.1 Artificial Intelligence and Machine Learning 10

3.2 Feature Extraction and Representation Learning 11

3.3 Feature Normalisation 11

3.4 Dimensionality Reduction 12

3.5 Deep Learning 12

3.6 Artificial Neural Networks 13

3.6.1 Feed-Forward Neural Networks 13

3.6.2 Activation Functions 14

3.7 CNNs 14

3.7.1 One-Dimensional CNNs 15

3.8 Transformers 16

3.9 Vision Transformers and CLIP 17

3.10 Video-Based Artist Representation 17

3.11 Hierarchical Video-to-Artist Encoding 20

3.12 Regularisation 21

3.13 Siamese Neural Networks 22

3.14 Triplet Loss 23

3.15 Cosine Similarity and Cosine Distance 23

3.16 Fixed-Negative Triplet Training 23

3.17 Artist Similarity as Retrieval 24

3.18 Evaluation Metrics 24

3.18.1 Triplet Ranking Accuracy 25
3.18.2 Precision, Recall, and F1 Score 25
3.18.3 Retrieval Metrics 25
3.19 Artist-Disjoint Evaluation 26
3.20 Latent Space Visualisation 26

4 Methodology **28**
4.1 Scope of the Proposed Method 28
4.2 Problem Formulation 29
4.3 System Pipeline 29
4.4 Video Frame Embedding Extraction 30
4.5 Artist Tensor Construction 31
4.5.1 Stack Aggregation for the Main Model 32
4.5.2 Mean Aggregation for the Baselines 32
4.5.3 Information retained by artist aggregation 32
4.6 Triplet Dataset Construction 33
4.6.1 What the triplets supervise 33
4.7 Artist-Disjoint Splitting 35
4.8 TripletNet1: Hierarchical Transformer Encoder 36
4.9 Conv1D Baseline Models 37
4.10 Margin Sweep and Checkpoint Selection 37
4.11 Methodological Design Choices and Trade-offs 38
4.12 Implementation and Reproducibility Details 40
4.12.1 Reproducible decision points 40

5 Experiments and Results **41**
5.1 Chapter Overview 41
5.2 Experimental Data 41
5.2.1 Coverage and representativeness 42
5.3 Artist-Disjoint Cross-Validation Protocol 42
5.3.1 Fold balance and effective sample size 43
5.4 Training Configuration 44
5.4.1 Fairness of the optimisation comparison 44
5.5 Evaluation Metrics 44
5.6 Cross-Validation Model Selection 45
5.7 Margin Sensitivity Results 46
5.8 Fold-Level Stability 48
5.9 Out-of-Fold Triplet Evaluation 49
5.10 Threshold-Based Pair Classification 51
5.11 Full-Catalogue Retrieval Evaluation 52
5.11.1 Retrieval setup 52
5.11.2 Retrieval Results and Interpretation 53
5.12 Latent Space and Metadata Diagnostics 56

5.12.1 Country- and Genre-Level Similarities 56
5.12.2 t-SNE Visualizations 58
5.12.3 Limits of the Metadata Diagnostics 58
5.13 Error Analysis and Computational Considerations 61
5.13.1 Error Analysis 61
5.13.2 Computational Considerations 61

6 Conclusion **62**
6.1 Summary of the Thesis and Main Findings 62
6.2 Limitations and Future Work 63
6.2.1 Relation labels and evaluation 63
6.2.2 Video coverage and representation 64
6.2.3 Architecture and experimental scope 64
6.2.4 From a video branch to multimodal retrieval 65
6.3 Final Remarks 65

Bibliography **66**

List of Figures

2.1 Progression of artist-similarity representations. 5
2.2 Audio content, contextual relations, and social-media signals used in artist-similarity research. Original schematic based on prior surveys and multimodal studies [1, 21, 5]. 7

3.1 Low-dimensional projection of artist embeddings. 13
3.2 Comparison of common activation functions. Original plot based on their standard definitions [31, 35]. 15
3.3 One-dimensional convolution over a sequence of frame-level embeddings. Original schematic based on the standard convolutional operation [14]. 16
3.4 Main components of a Transformer encoder block. Original schematic based on Vaswani et al. [13]. 18
3.5 Frame sampling and CLIP image encoding used to obtain video descriptors. Original schematic of the proposed preprocessing pipeline, using the CLIP encoder from Radford et al. [6]. 19
3.6 Static-image and video-based artist representations. 20
3.7 Hierarchical encoding from frame descriptors to video representations and then to an artist embedding. Original schematic of the proposed architecture, based on Transformer encoders [13]. 21
3.8 Triplet-based Siamese architecture in which anchor, positive, and negative artists share one encoder. Original schematic based on Siamese metric learning [9, 10]. 22
3.9 Cosine similarity interpreted through the angle between embedding vectors. Original schematic based on the cosine definition [12]. . . . 24
3.10 Two-dimensional diagnostic view of neighbourhood structure in an artist embedding space. Original schematic based on dimensionality-reduction analysis [17, 18]. 27

4.1 Video-based artist-similarity pipeline from frame extraction and CLIP encoding to triplet learning and retrieval. Original schematic developed for this thesis, using CLIP [6]. 30

4.2 Construction of the fixed-size artist tensor, including zero padding and a validity mask for missing videos. Original schematic developed for this thesis. 32
4.3 Triplet construction from AllMusic and Spotify relations. Original schematic based on the broader project's published procedure [5]. . . 34
4.4 Artist-disjoint splitting, which prevents an artist from appearing in both training and validation triplets. Original schematic developed for this thesis. 35
4.5 TripletNet1 hierarchical Transformer encoder. Original schematic of the proposed architecture, based on Transformer encoders [13]. . . . 36

5.1 Training and validation triplet counts after artist-disjoint splitting. Generated from the experiment folds reported in Table 5.2. 43
5.2 Best mean validation MRR for the tested models. Error bars indicate the standard deviation across the five artist-disjoint folds. Generated from the cross-validation summaries in Table 5.4. 46
5.3 TripletNet1 margin sensitivity across the tested values. Error bars indicate the standard deviation across the five artist-disjoint folds. Generated from the cross-validation outputs in Table 5.5. 47
5.4 Mean out-of-fold positive and negative cosine similarities and their mean gap. Generated from the pair scores summarised in Table 5.8. 50
5.5 Out-of-fold positive and negative similarity distributions for Triplet-Net1. Generated from the 1,132 out-of-fold triplets. 51
5.6 Threshold-based pair classification for TripletNet1. Generated from the out-of-fold pair scores. 52
5.7 Full-catalogue retrieval metrics across the tested models. Generated from the results in Table 5.10. 54
5.8 t-SNE projection of the TripletNet1 artist space coloured by broad genre. Generated from the learned embeddings and post-hoc metadata. 59
5.9 t-SNE projection of the TripletNet1 artist space coloured by country. Generated from the learned embeddings and post-hoc metadata. . . 59

List of Tables

4.1 Model configurations used in the comparison. 37
4.2 Methodological design choices and their implications. 39

5.1 Dataset quantities used in the experiments. Values are obtained after matching the project relation table to the available video descriptors. 41
5.2 Artist-disjoint cross-validation folds shared by all models and margins. Artist counts refer to retained triplets, whereas partition counts include artists assigned before row filtering. 42
5.3 Common training hyperparameters used in the experiments. 44
5.4 Best cross-validation results by model. The selection metric is mean validation MRR across the five artist-disjoint folds. 45
5.5 TripletNet1 margin sensitivity. Values are mean and standard deviation over the five artist-disjoint folds. 46
5.6 Mean validation MRR across the margin grid for all models. 47
5.7 Fold-level validation results at the selected margin $m = 0.1$. 48
5.8 Out-of-fold triplet and pairwise metrics at the best margin. Each model is evaluated on 1,132 validation triplets aggregated across folds. 49
5.9 Best threshold-based pair classification on out-of-fold positive and negative pairs. 51
5.10 Full-catalogue retrieval metrics. All 3,892 encoded artists are used as candidates, and 2,795 artists with at least one known positive are used as queries. 53
5.11 Country-level similarity diagnostics for TripletNet1. The model is not trained with country labels; the table is used only for interpretation. 57
5.12 Broad-genre similarity diagnostics for TripletNet1. 57
5.13 Examples of fine-grained genre groups with strong intra-group similarity in the TripletNet1 latent space. 58
5.14 Silhouette scores for TripletNet1 metadata labels. Scores are computed with cosine distance on a sampled set of 2,500 artists. 58

Chapter 1

Introduction

Streaming platforms and social media have changed how listeners encounter artists. Albums, radio, concerts, and editorial catalogues remain important, but recommendation feeds, short videos, visual posts, and online interactions now mediate many first encounters. Artist similarity is therefore both a theoretical question and a practical requirement for recommendation, playlist generation, catalogue organisation, and the discovery of emerging artists [1, 2]. Yet similarity has no single definition. Artists may sound alike because of genre, timbre, rhythm, instrumentation, or vocal style, while cultural background, audience perception, visual identity, and public presentation create other forms of affinity.

Music Information Retrieval (MIR) has traditionally estimated artist similarity from audio descriptors or metadata relations [1, 3, 4]. The musical signal directly describes an artist's sound, but it does not exhaust artist identity. Social media, promotional videos, live-performance clips, visual aesthetics, fashion, locations, gestures, and recurring narrative choices also shape an artist's public image. Recent work showed that Instagram images and captions can recover artist relations at a level comparable with an audio-based benchmark [5].

Within that line of research, this thesis focuses on video content for artist similarity assessment. It forms part of the multimodal project described by Zanoni et al., which examines audio and Instagram image-text representations [5]. Here, video-based visual representations are isolated so that their contribution can be assessed before multimodal fusion. The research question is narrower: can frame-level visual embeddings from artist-related videos produce an artist similarity space that agrees with known relations from music platforms [6, 7, 8]?

Each artist is represented by a set of videos rather than a single image, caption, or audio track. The visual sequence may convey performance style, branding, stage presence, editing choices, and other aspects of the artist's online identity. The pipeline uses precomputed CLIP frame embeddings instead of training a visual model directly from pixels [6]. The model can therefore focus on artist-level similarity while drawing on a pretrained visual representation.

The task is formulated as metric learning rather than classification. A Siamese network applies the same encoder to several inputs so that their representations can be

compared directly [9]. The triplet form used here receives an anchor artist, a related positive artist, and an unlinked negative artist; training places the positive closer to the anchor than the negative [10]. The ground truth combines human-curated AllMusic relations [7, 11] with algorithmic related-artist relations from Spotify [8], providing editorial and platform-based views of similarity.

Training uses a triplet network, triplet loss, and cosine distance [9, 10, 12]. Nearest-neighbour search in the resulting latent space identifies artists that the model regards as similar. The same representation supports binary similarity decisions and recommendation-style rankings [2, 12].

Aggregating video-level information into one artist representation is the main modelling challenge. The proposed architecture addresses it with a hierarchical Transformer that first summarises the frames in each video and then aggregates the resulting video representations at artist level [13]. The pipeline is benchmarked against Conv1D baselines that reduce an artist's videos to one sequence before applying convolution [14]. This comparison tests the value of preserving the multi-video structure.

Artist-disjoint cross-validation prevents the same artist from occurring in both training and validation and thereby reduces leakage [15]. Evaluation considers several properties of the learned space. Triplet-level metrics test whether the positive artist is closer to the anchor than the negative, while pair-level metrics assess positive-negative separation at different similarity thresholds [16]. Precision@1, HitRate@5, and MRR evaluate recommendation-oriented retrieval [12, 2]. Dimensionality reduction and metadata-aware analyses show whether artists with related genres or countries occupy nearby regions [17, 18].

In summary, the thesis makes the following contributions:

- an assessment of artist similarity based on video-derived visual representations and precomputed CLIP frame embeddings [6];
- an independent evaluation of the video modality before its future integration with audio, static-image, and text representations;
- a triplet-based metric-learning formulation supervised by Spotify and AllMusic similarity relations [10, 19, 7, 8];
- a hierarchical Transformer encoder that preserves the multi-video structure of each artist [13];
- a controlled comparison with Conv1D baselines under a shared training and evaluation framework [14];
- an artist-disjoint cross-validation protocol for evaluating generalisation without artist leakage [15];
- an evaluation combining triplet accuracy, pair-level measures, retrieval metrics, and visual analysis of the latent space [16, 12, 18].

The remainder of the thesis is organised as follows. Chapter 2 reviews work on artist similarity, MIR, and multimodal representation learning. Chapter 3 introduces the technical foundations needed for the proposed method. Chapter 4 describes dataset preparation, video embeddings, triplet construction, and the model architectures. Chapter 5 presents the experiments and numerical results. Chapter 6 presents the conclusions, limitations, and future work.

Chapter 2

State of the Art

This chapter places the video-based system within research on artist similarity. It begins with audio and contextual methods, then considers learned and multimodal representations. The final sections focus on visual social-media content and explain how the proposed video branch differs from earlier work.

2.1 MIR and Artist Similarity

MIR develops methods for analysing, organising, retrieving, and recommending music-related information [1]. Artist similarity supports catalogue navigation, discovery, and recommendation, but it is difficult to define objectively. Genre, instrumentation, rhythm, vocal style, period, audience, geographic origin, and visual identity may each change a listener's judgement [20, 1].

This ambiguity has shaped the field. Early methods emphasised acoustic descriptors, while later work added contextual data, graph relations, and learned embedding spaces. Recent multimodal studies also use images, text, and social-media behaviour to represent aspects of an artist that are not contained in the audio signal [21, 5].

2.2 Techniques for Artist Similarity

2.2.1 Content-Based Techniques

Content-based methods analyse the musical signal and place artists with similar acoustic properties near one another [1]. Common features describe timbre, rhythm, melody, harmony, tempo, and spectral shape.

Early systems often used handcrafted descriptors such as Mel-Frequency Cepstral Coefficients (MFCCs). Statistical models and distance functions then compared tracks or artists through distributions of these features [3, 4]. Such descriptors capture low-level acoustic properties but may miss cultural or audience-based relations. Two artists can sound different and still belong to the same scene, influence one another, or attract similar listeners.

Representation learning moved the field beyond direct distances between selected

descriptors. Neural models trained on genres, artists, or tags can supply intermediate embeddings for similarity estimation [22]. The similarity function is then learned from data rather than fixed in advance.

Deep models learn features from waveforms, spectrograms, and log-mel spectrograms [23]. Convolutional Neural Networks (CNNs) commonly process time-frequency representations, while Recurrent Neural Networks (RNNs) and Convolutional Recurrent Neural Networks (CRNNs) model temporal dependencies [24, 25].

For artist-related tasks, these architectures have been used for classification, genre recognition, tagging, and similarity learning. CRNN experiments show that longer temporal context can capture stylistic evidence not visible in short excerpts [25]. Graph Neural Networks (GNNs) offer a different route: artists form the nodes of a relation graph, and a model learns from both content features and graph connections [26].

Figure 2.1 summarises this progression. Handcrafted audio descriptors remain useful, but current systems increasingly learn representations and combine several evidence sources. The proposed video method belongs to the last stage of this development.

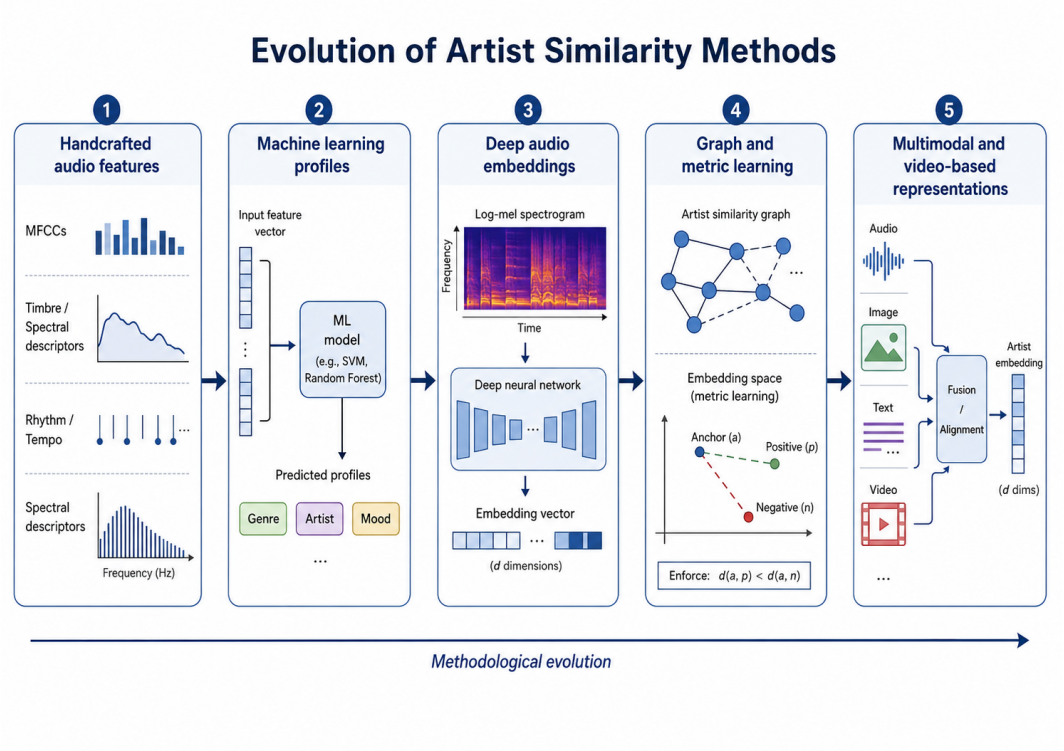


Figure 2.1: Progression from handcrafted audio descriptors to learned, multimodal, and video-based artist representations. Original schematic based on the literature reviewed in this chapter [3, 22, 26, 5].

2.2.2 Context-Based Techniques

Where content-based techniques derive similarity from the signal itself, context-based methods draw on tags, biographies, playlists, listening histories, social networks, and

platform-provided related-artist relations [1, 2]. These sources treat music similarity as partly social and cultural.

Co-listening, playlist co-occurrence, shared tags, and participation in the same scene can reveal relations that audio alone misses. They are also vulnerable to popularity bias, community bias, platform effects, and data sparsity [1]. The ground truth used in this thesis is contextual: AllMusic provides expert-curated editorial relations [7, 11], while Spotify supplies algorithmic related-artist links [8]. At this point, these sources define known artist relations; Chapter 4 explains how they are converted into triplets.

2.2.3 Multimodal Techniques

Multimodal techniques combine content and context with further evidence from the ways artists present themselves beyond recorded audio.

Online platforms expose dimensions of artist identity beyond recorded music. Images, captions, videos, interviews, and promotional posts affect public perception and provide additional evidence for similarity [21, 5]. Audio conveys musical style, text captures semantic and communicative content, and visual media document presentation and performance.

Multimodal systems can combine these sources before learning or merge the outputs of separate models. Early fusion gives a model access to cross-modal relations at feature level, whereas late fusion preserves separate modality-specific branches [21]. The choice depends on whether the study aims to optimise a complete multimodal system or measure the contribution of one modality.

Zanoni et al. combined CLIP image features and Sentence-BERT (SBERT)-based caption embeddings from Instagram, then trained a Siamese network with triplet loss [5]. Their results show that social-media content can recover artist relations at a level comparable with an audio-based benchmark. The current thesis extends the broader project to video but evaluates the video branch separately, so its contribution remains measurable.

2.3 Visual Representation Learning for Artist Identity

Album covers, photographs, stage design, clothing, lighting, editing, and recurring symbols contribute to an artist's public presentation. Their relation to musical similarity is indirect: visual codes may closely follow genre conventions in some scenes and instead reflect promotion, audience, or cultural positioning in others [21, 5].

Traditional image analysis used descriptors such as colour histograms, texture measures, and local features. CNNs and Transformer-based image encoders now learn semantic visual representations from large datasets [14, 27]. These embeddings make it possible to compare visual material without defining every relevant object or style in advance.

Figure 2.2 distinguishes the three evidence families used in artist-similarity research. The categories overlap in practice: platform relations can supervise a model whose input is audio or visual content. In this thesis, AllMusic and Spotify provide contextual supervision while video frames provide the input representation.

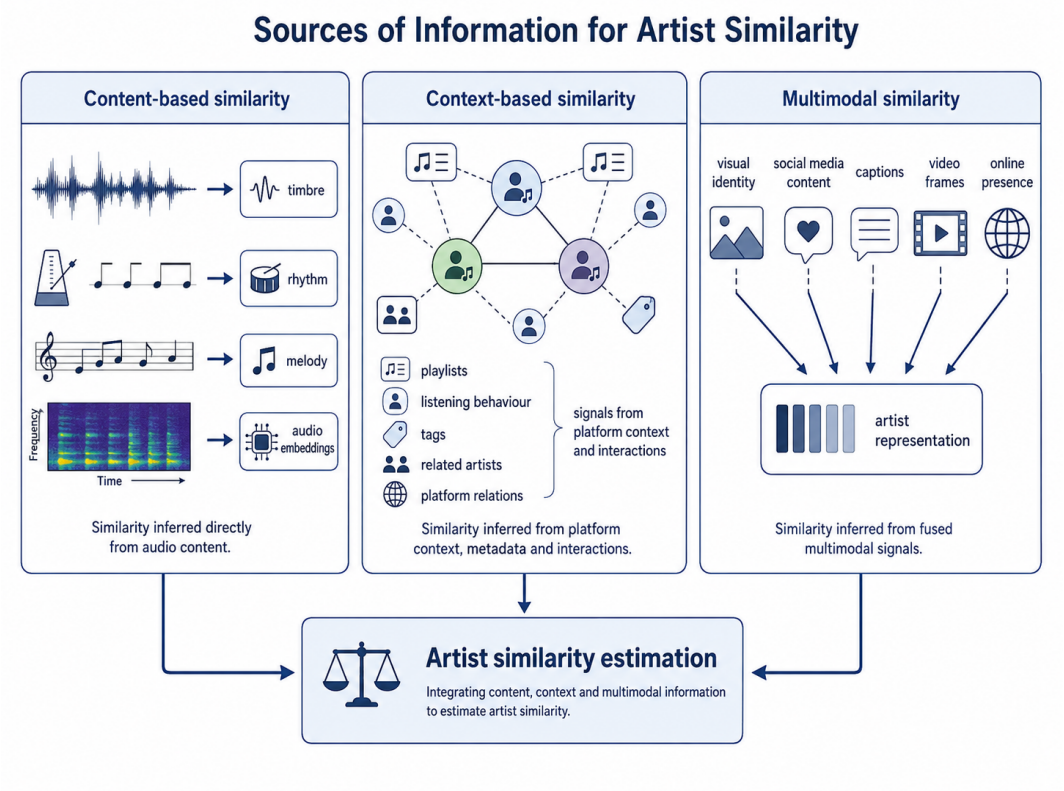


Figure 2.2: Audio content, contextual relations, and social-media signals used in artist-similarity research. Original schematic based on prior surveys and multimodal studies [1, 21, 5].

2.4 CLIP and Transferable Visual Embeddings

CLIP aligns image and text representations through large-scale paired pretraining [6]. Its image encoder can therefore serve as a general-purpose feature extractor without task-specific fine-tuning. Beyond colour and texture, the resulting embeddings contain information about scenes, objects, clothing, instruments, and performance settings.

The architecture proposed in [5] used CLIP to encode static Instagram post images. The present work applies the image encoder to sampled video frames and then learns how to aggregate several frames and videos for each artist. The visual backbone remains fixed; the trainable part focuses on the artist representation.

2.5 Recent Artist Similarity Models

Several studies address artist similarity more directly than general music-similarity research. Ellis et al. showed that artist similarity lacks a single reliable ground truth

because human judgements depend on the task and the information available to the listener [20]. McFee and Lanckriet treated this subjectivity as an embedding problem and learned from heterogeneous similarity judgements rather than assuming one universal metric [28].

More recent graph-based methods combine artist features with relational structure. Korzeniowski et al. use GNNs to integrate content and artist connections [26]; da Silva et al. extend this direction with heterogeneous graphs that represent several relation types [29]. These models show that an artist's neighbourhood can carry information that isolated audio features do not capture.

The Instagram study by Zanoni et al. provides the closest comparison for this thesis because it uses the same broader project, the same AllMusic and Spotify relation sources, and social-media content as input [5]. Its image-caption representation is static and multimodal. The method developed here is visual-only and explicitly models several frames within several videos.

Across these studies, both the evidence used and the definition of a correct neighbour vary. Schedl et al. aggregate music content, music context, and user context from tracks to artists [1]. This design recognises that an artist is represented by a catalogue and by listener activity, not by one recording. It also makes the resulting similarity dependent on the available listening data and on the platform from which that data was collected.

McFee and Lanckriet address another source of variation: similarity judgements obtained from different sources need not agree [28]. Their heterogeneous embedding formulation does not force every judgement into one metric. The point is relevant here because AllMusic and Spotify relations are produced by different processes. Agreement between the sources is useful evidence, but disagreement does not imply that either source is wrong.

Graph-based models make the relation structure part of the representation itself. Korzeniowski et al. combine audio content with a graph of artist relations and train the model with a triplet objective [26]. The graph supplies contextual information that cannot be recovered from an isolated recording. Da Silva et al. use a heterogeneous graph so that several types of nodes and relations can contribute to the artist representation [29]. Such models are well suited to relation-rich catalogues, although part of their predictive signal comes from the same relational structure that later defines relevance.

The Instagram study takes a different route. Its trainable similarity model receives visual and textual content, while AllMusic and Spotify relations provide external supervision [5]. The separation between input content and relation labels makes it possible to ask whether social-media posts contain information that generalises beyond the relations used during training. This thesis follows the same separation but replaces static images and captions with sequences of video frames.

These choices change the meaning of a successful result. An audio model can recover acoustic proximity, a graph model can reconstruct relational neighbourhoods, and a social-media model can recover similarities expressed through public

presentation. None is a complete account of artist similarity. Their outputs are better understood as views of a relation that depends on the input modality and the reference data [20].

Unlike the architecture proposed in [5], which uses static Instagram images and captions, the system proposed in this thesis uses sampled video frames and explicitly preserves the organisation of frames within videos and videos within an artist profile. The video branch is evaluated separately so that its independent contribution can be measured without immediate fusion with audio or other social-media representations.

The next chapter introduces the theoretical concepts needed for the video-specific methodology.

Chapter 3

Theoretical Background

This chapter introduces the technical foundations of the video-based artist similarity system. After outlining general learning concepts, it covers the neural architectures and metric-learning methods used in the project [30, 23, 19]. The later sections address the components specific to this thesis: CLIP visual embeddings, video-level artist representations, hierarchical Transformer encoding, masks for padded videos, video dropout, fixed-negative training, and retrieval-oriented evaluation [6, 13, 12].

3.1 Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) concerns computational systems that perform tasks associated with human intelligence, including reasoning, perception, pattern recognition, decision making, language understanding, and learning from experience [30, 31]. For artist similarity, such systems can analyse collections too large for manual comparison and identify relations within them.

Machine Learning (ML) is a subfield of AI in which systems learn patterns from data [32, 30]. Rather than following an exhaustive set of fixed rules, a model estimates a function from examples. Given a dataset

$$\mathcal{D} = \{x_1, x_2, \ldots, x_N\}, \tag{3.1}$$

the algorithm seeks a function

$$f_\theta(x), \tag{3.2}$$

where θ denotes the model parameters, which are adjusted during training for the target task.

ML methods are commonly divided into three main paradigms [30, 31]:

- **Supervised learning**: the model is trained using examples paired with target labels or values. Classification and regression are typical supervised learning tasks.
- **Unsupervised learning**: the model receives data without explicit labels and tries to discover hidden structures, such as clusters or low-dimensional

representations.

- **Reinforcement learning**: the model learns by interacting with an environment and receiving rewards or penalties according to its actions.

The artist similarity task uses supervision without adopting a standard classification formulation. Instead of predicting a class label, the model learns a space in which similar artists are close and dissimilar artists are farther apart. This approach is known as metric learning [19].

3.2 Feature Extraction and Representation Learning

Machine-learning algorithms generally require numerical representations of images, text, audio, or video. Feature extraction converts raw data into descriptors that preserve information relevant to the task [22, 31]. A feature vector can be written as

$$\mathbf{x} = [x_1, x_2, \ldots, x_d] \in \mathbb{R}^d, \tag{3.3}$$

where each component represents one numerical descriptor of the input.

Traditional machine-learning pipelines often relied on manually designed features, such as spectral descriptors for audio or colour histograms and edge descriptors for images [3, 4]. Their design requires domain knowledge, and the resulting descriptors may omit complex patterns.

Deep learning allows a neural network to learn representations directly from data rather than depend entirely on handcrafted descriptors [22, 23]. An embedding is a dense vector representation of an object, such as a word, image, frame, video, or artist. It is useful when distances in the embedding space preserve relationships relevant to the task.

This thesis uses precomputed CLIP frame embeddings rather than raw video pixels. A pretrained CLIP image encoder maps each sampled frame to a visual descriptor [6], and the trainable network aggregates these descriptors into artist-level representations.

3.3 Feature Normalisation

Feature normalisation limits the influence of numerical scale. Without it, large-valued features may dominate learning even when they are no more informative than smaller-valued features [31, 33, 34].

One common technique is min-max normalisation:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{3.4}$$

This transformation maps values into the range $[0, 1]$. Another common technique is

standardisation:

$$x' = \frac{x - \mu}{\sigma}, \tag{3.5}$$

where μ is the mean and σ is the standard deviation.

Normalisation can also be applied within a neural network. The implemented architectures use Layer Normalisation and Batch Normalisation to stabilise training and reduce sensitivity to scale [34, 33]. The artist encoder then uses ℓ_2 normalisation to project its output onto the unit hypersphere:

$$\hat{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}. \tag{3.6}$$

This matters here because the model uses cosine distance to compare artists.

3.4 Dimensionality Reduction

Modern machine-learning representations are often high-dimensional. Although effective for learning, these representations are difficult to visualise and may contain redundant information [31, 17].

Dimensionality reduction maps these data to fewer dimensions while retaining selected aspects of their structure. Principal Component Analysis (PCA), a widely used linear method, identifies orthogonal directions of maximum variance and projects the data onto a subset of them [17].

If X is a data matrix, PCA computes a new representation:

$$Y = XW, \tag{3.7}$$

where W contains the selected principal directions.

t-distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear technique used mainly for visualisation [18]. It maps high-dimensional points to two or three dimensions while seeking to preserve local neighbourhoods. Here, t-SNE helps inspect whether artists with related genres, countries, or visual identities occupy nearby regions of the latent space. Figure 3.1 illustrates the change from a high-dimensional artist space to a two-dimensional diagnostic view. Distances in the projection are interpreted cautiously because the mapping does not preserve the full geometry.

3.5 Deep Learning

Deep learning uses multilayer neural networks to learn complex nonlinear functions by composing simpler transformations [23, 31]. A deep network can be expressed as the sequence

$$f(x) = f^{(L)}(f^{(L-1)}(\dots f^{(1)}(x))), \tag{3.8}$$

where each function $f^{(i)}$ corresponds to one layer or block of the network.

For complex inputs such as images, video, text, and audio, neural networks can learn hierarchical representations that are difficult to specify manually [23, 22].

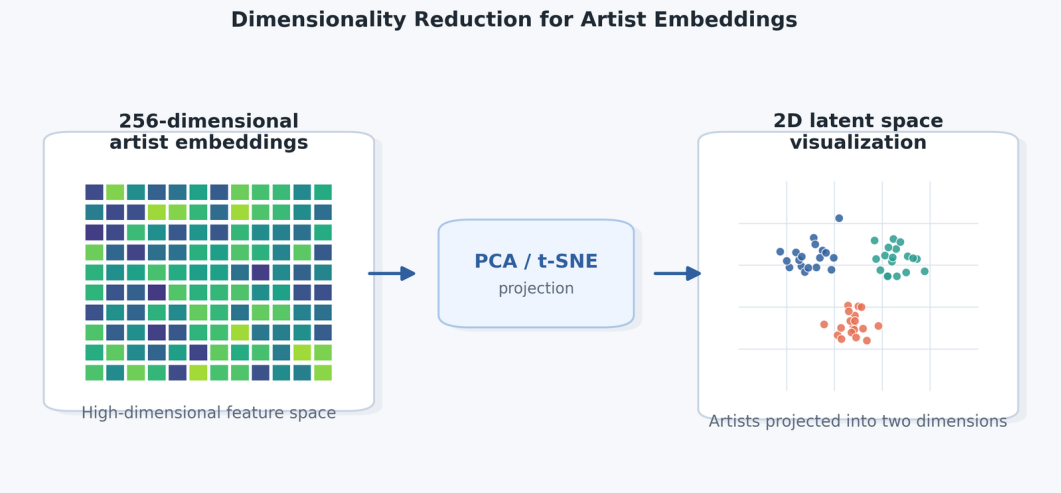


Figure 3.1: Projection of high-dimensional artist embeddings into a low-dimensional space for exploratory visualisation. Original schematic based on PCA and t-SNE [17, 18].

Earlier layers tend to capture simple patterns, while deeper layers combine them into more abstract features.

Deep learning enters the pipeline at two stages. A CLIP image encoder pretrained on large-scale image-text data extracts frame embeddings [6]; a trainable triplet network then maps the resulting artist inputs into a similarity space [10, 19].

3.6 Artificial Neural Networks

An Artificial Neural Network (ANN) consists of interconnected computational units, conventionally called neurons. A neuron combines its inputs with trainable weights, adds a bias, and applies an activation function [31].

For an input vector $\mathbf{x}$, the output of a neuron can be written as:

$$y = \phi(\mathbf{w}^T\mathbf{x} + b), \tag{3.9}$$

where $\mathbf{w}$ is the weight vector, b is the bias, and ϕ is the activation function.

A neural network is usually organised into layers [31]:

- **Input layer**: receives the input features.
- **Hidden layers**: transform the data into increasingly abstract representations.
- **Output layer**: produces the final prediction or embedding.

Training compares the network output with an objective through a loss function and updates the parameters by gradient-based optimisation [31].

3.6.1 Feed-Forward Neural Networks

In feed-forward neural networks, information passes from the input through one or more hidden layers to the output, without cycles or recurrent connections [31].

A fully connected layer applies the following transformation:

$$\mathbf{h} = \phi(W\mathbf{x} + \mathbf{b}), \tag{3.10}$$

where W is a matrix of weights, $\mathbf{b}$ is a bias vector, and ϕ is a nonlinear activation function.

Feed-forward layers occur throughout deep-learning architectures [31, 13]. In this project, the projection head uses them to map the encoded artist representation to 256 dimensions.

3.6.2 Activation Functions

Nonlinear activation functions allow neural networks to approximate complex functions. Without a nonlinear activation, a stack of linear layers remains equivalent to a single linear transformation [31].

Common activation functions include:

- **Sigmoid**:
$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{3.11}$$
- **Hyperbolic tangent**:
$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{3.12}$$
- **Rectified Linear Unit (ReLU)**:
$$\mathrm{ReLU}(x) = \max(0, x). \tag{3.13}$$
- **Gaussian Error Linear Unit (GELU)**:
$$\mathrm{GELU}(x) = x\Phi(x), \tag{3.14}$$
where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution.

The proposed Transformer-based model mainly uses GELU activations, while the convolutional baselines use ReLU-based blocks [35, 31]. Figure 3.2 compares their response curves with sigmoid, hyperbolic tangent, and the linear reference shown in the plot. ReLU suppresses negative inputs, whereas GELU weights them smoothly rather than applying a hard cutoff.

3.7 CNNs

CNNs process structured data such as images, signals, and sequences by applying learnable filters to local input regions [14]. Successive layers can combine the detected local patterns into higher-level representations.

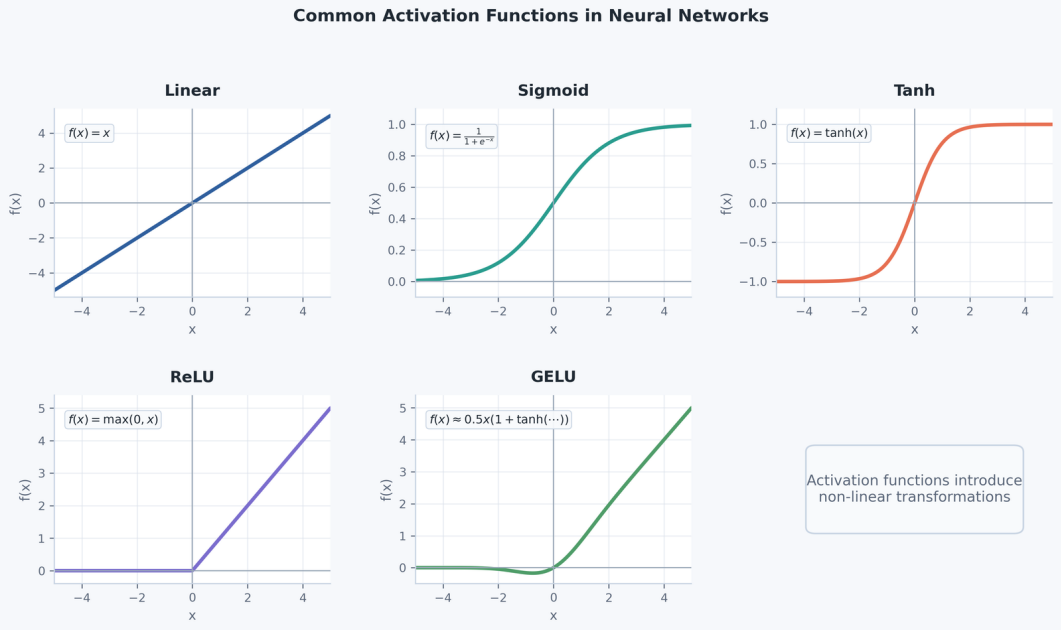


Figure 3.2: Comparison of common activation functions. Original plot based on their standard definitions [31, 35].

In the one-dimensional case, convolution over a sequence can be written as:

$$y_i = \sum_{k=-r}^{r} w_k x_{i+k}, \tag{3.15}$$

where x is the input sequence, w_k are the kernel weights, and $2r+1$ is the kernel size.

CNNs are based on three important ideas [14]:

- **Local connectivity**: each filter observes only a local region of the input.
- **Parameter sharing**: the same filter is applied at different positions.
- **Hierarchical feature extraction**: deeper layers combine simple patterns into more complex features.

A CNN block commonly combines convolution with activation, pooling, normalisation, or dropout [36]. Pooling reduces sequence length or spatial size and therefore lowers computational cost.

3.7.1 One-Dimensional CNNs

One-dimensional CNNs operate on sequential data [14]. The convolutional baselines considered in this thesis receive one sequence of frame-level features for each artist:

$$X \in \mathbb{R}^{T \times D}, \tag{3.16}$$

where T is the sequence length and D the feature dimension.

Convolutional filters move along the temporal axis and capture local patterns. Pooling or strided convolution reduces the sequence before a projection maps it to an

artist embedding. These models are efficient, but they do not explicitly distinguish frames within videos from videos within an artist profile.

Figure 3.3 shows the local receptive field used by a one-dimensional convolution. Stacking layers expands the portion of the sequence that influences each output, while the input remains a single sequence.

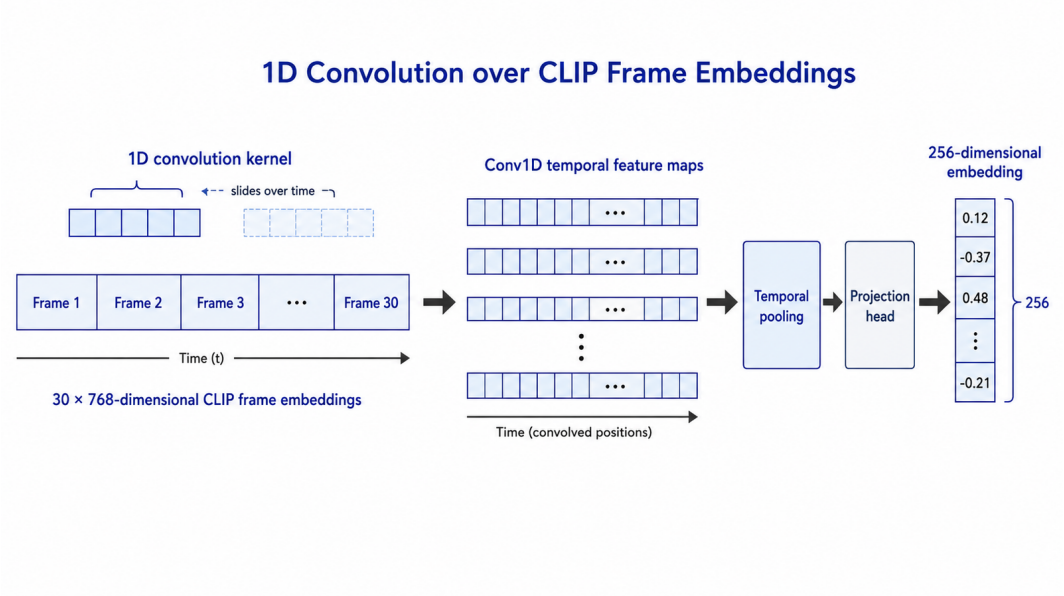


Figure 3.3: One-dimensional convolution over a sequence of frame-level embeddings. Original schematic based on the standard convolutional operation [14].

3.8 Transformers

Transformers are attention-based neural architectures introduced for sequence modelling and now widely used in language, vision, audio, and multimodal learning [13].

A Transformer's central operation is self-attention, through which every sequence element can interact with every other element [13]. For an input sequence X, the model computes three matrices:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \tag{3.17}$$

where Q is the query matrix, K is the key matrix, and V is the value matrix.

The attention operation is defined as [13]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{3.18}$$

where d_k is the dimensionality of the key vectors. The term QK^T measures the similarity between sequence elements. The softmax converts these similarities into attention weights.

Multi-head attention applies several attention mechanisms in parallel, allowing individual heads to capture different sequence relations. Their outputs are concatenated and projected into the resulting representation.

A Transformer encoder layer usually includes [13]:

- multi-head self-attention;
- a feed forward network;
- residual connections;
- normalisation layers;
- dropout.

The input in this thesis is both sequential and hierarchical: frames form a temporal sequence within each video, and a collection of videos describes each artist. The proposed architecture accordingly uses one Transformer within videos and another across an artist's video representations [13]. Figure 3.4 shows the attention, residual, normalisation, and feed-forward components repeated in each encoder stage.

3.9 Vision Transformers and CLIP

A Vision Transformer (ViT) divides an image into patches, projects each patch to a token, and processes the token sequence with a Transformer encoder [27]. For an image of height H, width W, and patch size $P \times P$, the number of patches is

$$N = \frac{HW}{P^2}. \tag{3.19}$$

CLIP contains image and text encoders trained to align both modalities in a shared embedding space [6]. Matched image-text pairs receive high similarity during pretraining, so the image encoder learns transferable visual descriptors. Only this image encoder is used in the proposed pipeline. For frame $I_{i,t}$, it produces

$$\mathbf{x}_{i,t} = f_{\text{CLIP}}(I_{i,t}) \in \mathbb{R}^D. \tag{3.20}$$

Artist videos may contain performance settings, clothing, instruments, lighting, gestures, and recurring visual motifs. CLIP descriptors encode this material before the trainable artist model is applied. Figure 3.5 separates the fixed visual extraction stage from the later artist-level learning stage.

3.10 Video-Based Artist Representation

A static image records one view of an artist, while video also contains changes in pose, scene, camera position, and performance over time. Direct spatiotemporal models can learn from raw frames but require substantial data and computation [37,

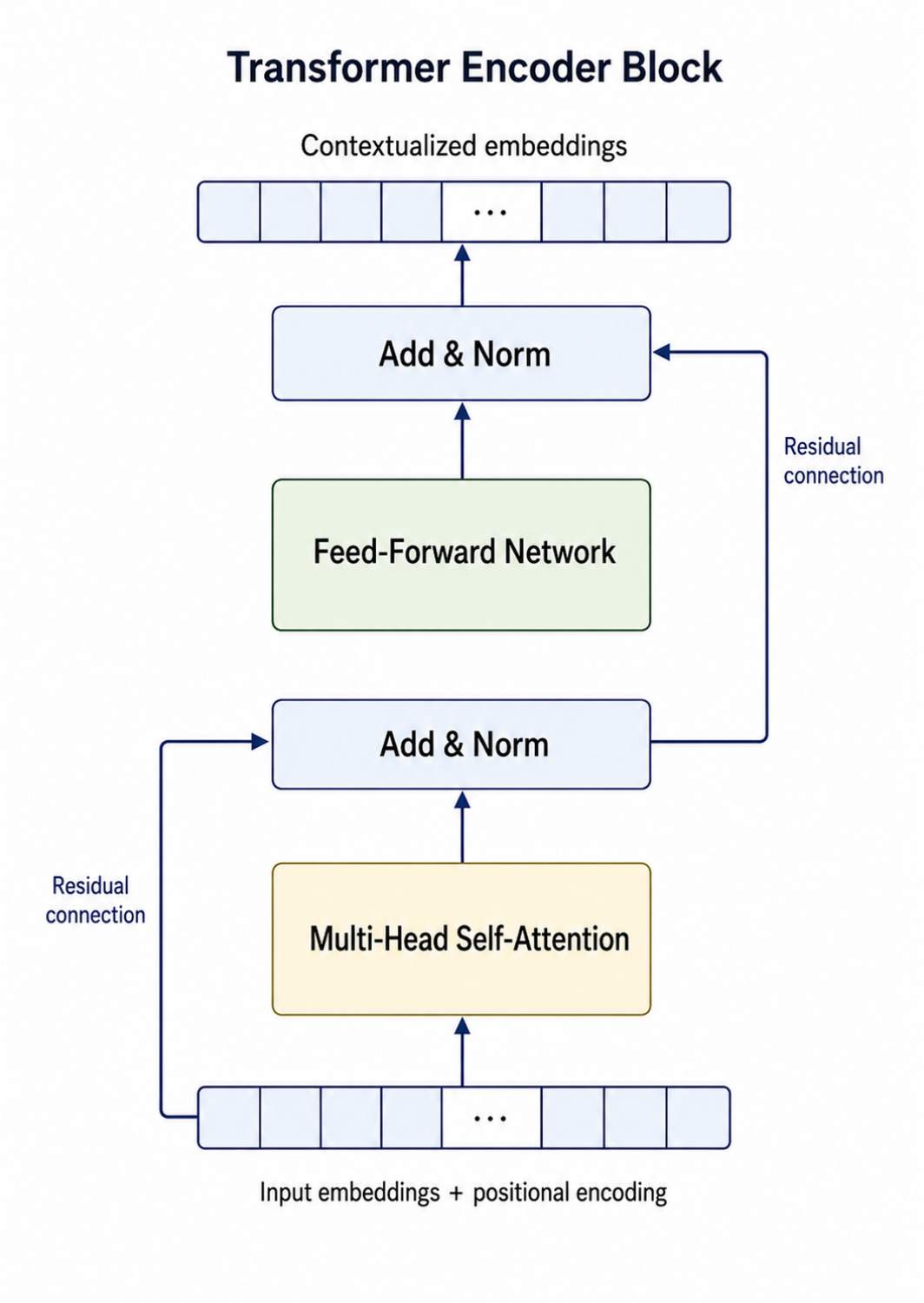


Figure 3.4: Main components of a Transformer encoder block. Original schematic based on Vaswani et al. [13].

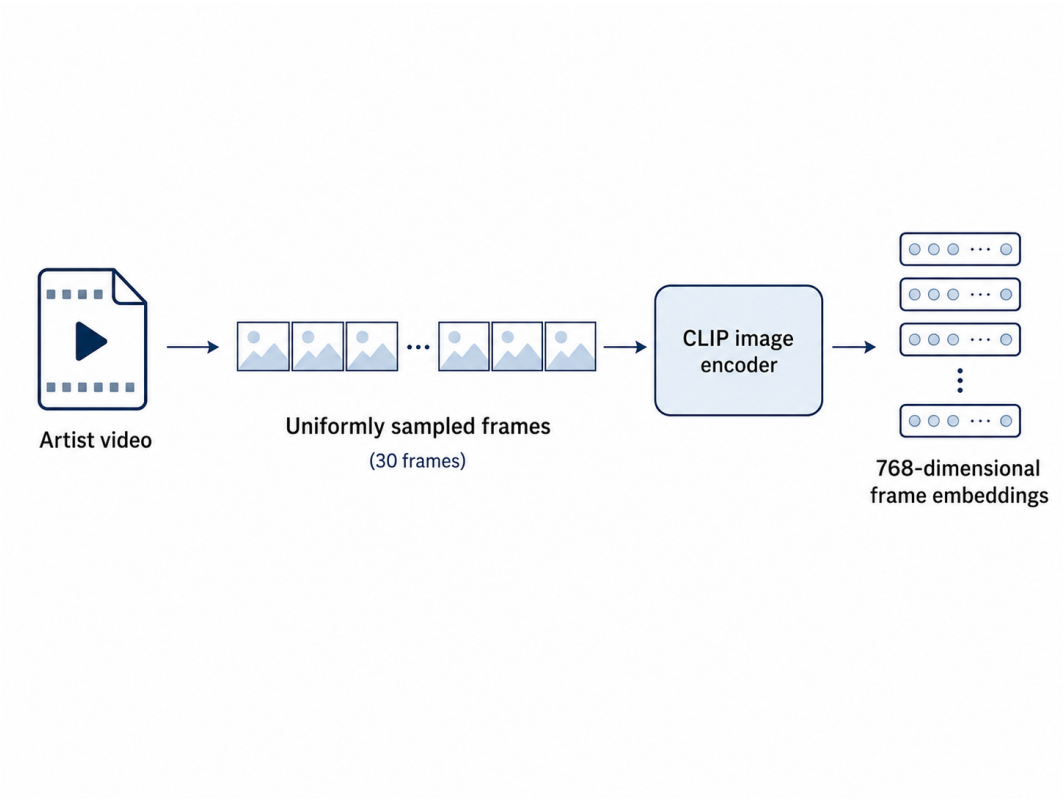


Figure 3.5: Frame sampling and CLIP image encoding used to obtain video descriptors. Original schematic of the proposed preprocessing pipeline, using the CLIP encoder from Radford et al. [6].

38]. An alternative is to extract each frame with a pretrained image model and learn only the aggregation stage. The proposed pipeline follows this second approach.

Let T sampled frames describe video i, each with a D-dimensional descriptor. The video representation is

$$V_i = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,T}] \in \mathbb{R}^{T \times D}. \tag{3.21}$$

An artist with up to M videos is then represented by

$$X_a \in \mathbb{R}^{M \times T \times D}. \tag{3.22}$$

Unused video slots are padded, and a mask prevents them from contributing to the artist representation.

Figure 3.6 contrasts this structure with a static-image representation. The additional frame and video axes preserve evidence that immediate averaging would remove.

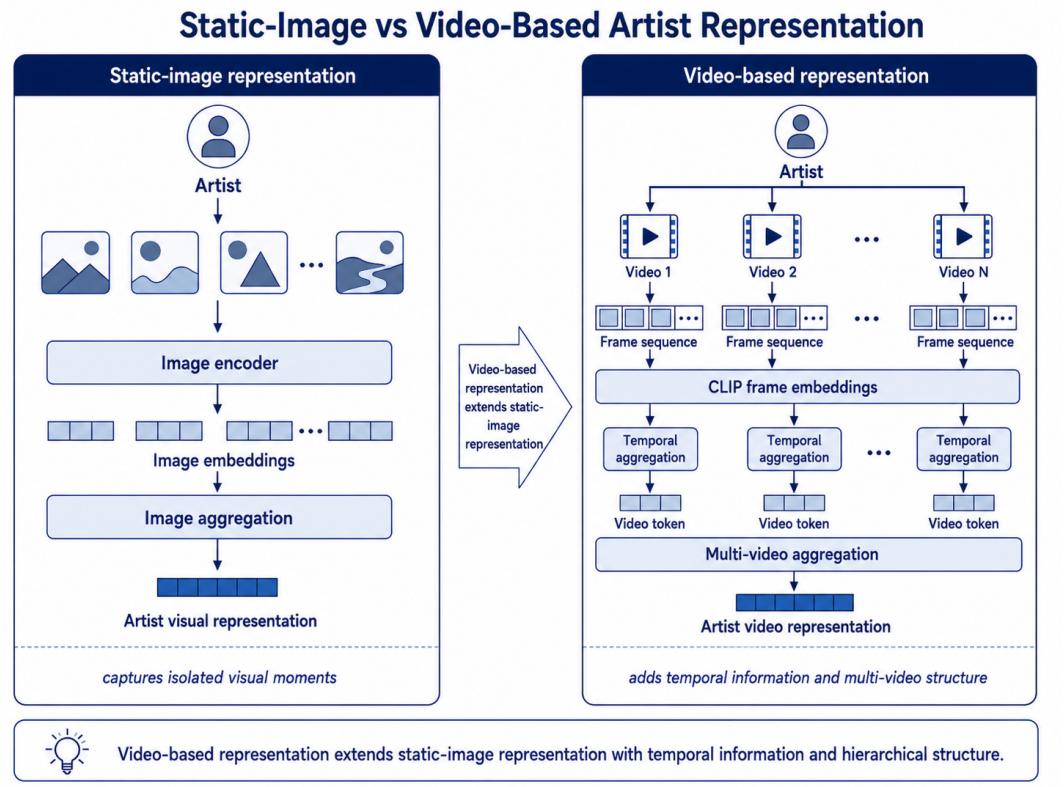


Figure 3.6: Static-image and video-based artist representations. Original schematic developed for this thesis.

3.11 Hierarchical Video-to-Artist Encoding

The proposed architecture follows the data hierarchy

$$\text{artist} \rightarrow \text{videos} \rightarrow \text{frames}. \tag{3.23}$$

A learned projection maps each CLIP descriptor to the internal model dimension, and positional embeddings retain frame order. A frame-level Transformer then contextualises the descriptors within each video [13].

The encoded frames are pooled into video representations:

$$\mathbf{v}_{a,i} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{h}_{a,i,t}. \tag{3.24}$$

An artist-level Transformer compares the valid video representations. Masked mean pooling excludes padded slots:

$$\mathbf{r}_a = \frac{\sum_{i=1}^{M} m_{a,i} \mathbf{u}_{a,i}}{\sum_{i=1}^{M} m_{a,i}}, \tag{3.25}$$

where $m_{a,i}$ indicates whether video i is valid.

A projection head produces the final normalised artist embedding. Figure 3.7 shows the two aggregation stages. The first summarises temporal evidence within a

video; the second compares evidence across the artist's videos.

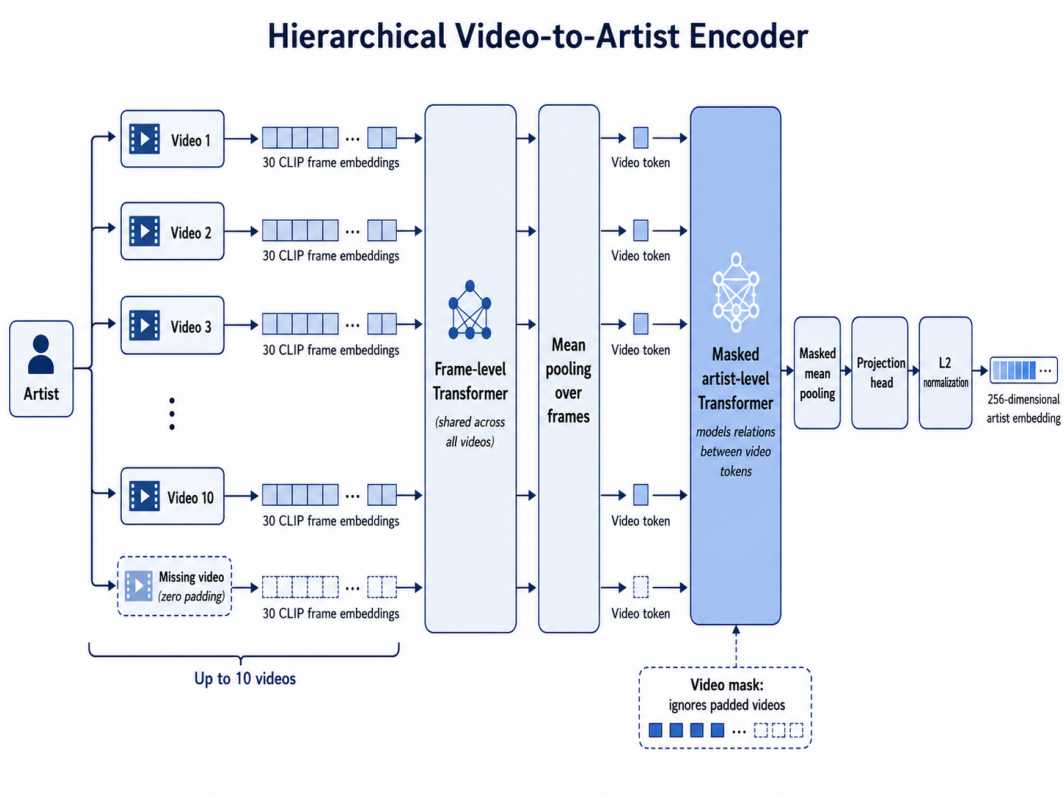


Figure 3.7: Hierarchical encoding from frame descriptors to video representations and then to an artist embedding. Original schematic of the proposed architecture, based on Transformer encoders [13].

3.12 Regularisation

Regularisation reduces overfitting, which occurs when performance on the training data does not transfer to unseen data [31].

Dropout is a common form of regularisation that randomly removes part of a representation during training [36]:

$$\tilde{\mathbf{h}} = \mathbf{m} \odot \mathbf{h}, \tag{3.26}$$

where $\mathbf{m}$ is a random binary mask and $\odot$ denotes element-wise multiplication.

The proposed architecture adapts this standard dropout [36] to the video-token level. Randomly removing valid video tokens discourages reliance on any single video; retaining at least one valid token for each artist keeps the representation defined.

Because the number of available videos differs among artists, this regularisation also reduces sensitivity to that imbalance.

3.13 Siamese Neural Networks

A Siamese Neural Network applies a shared encoder to two or more inputs so that their embeddings are directly comparable [9].

A standard Siamese network compares two inputs encoded with shared parameters. This thesis uses the triplet-based extension [9, 10], in which each sample contains

- an **anchor artist**;
- a **positive artist**, known to be similar to the anchor;
- a **negative artist**, not known to be similar to the anchor.

The shared encoder maps the three artists to embeddings:

$$\hat{\mathbf{z}}_a = f_\theta(X_a), \quad \hat{\mathbf{z}}_p = f_\theta(X_p), \quad \hat{\mathbf{z}}_n = f_\theta(X_n). \tag{3.27}$$

The objective places the positive artist closer to the anchor than the negative artist. This relative ordering expresses artist similarity without fixed categories [19, 10]. The three branches in Figure 3.8 share one encoder; only their inputs and roles differ.

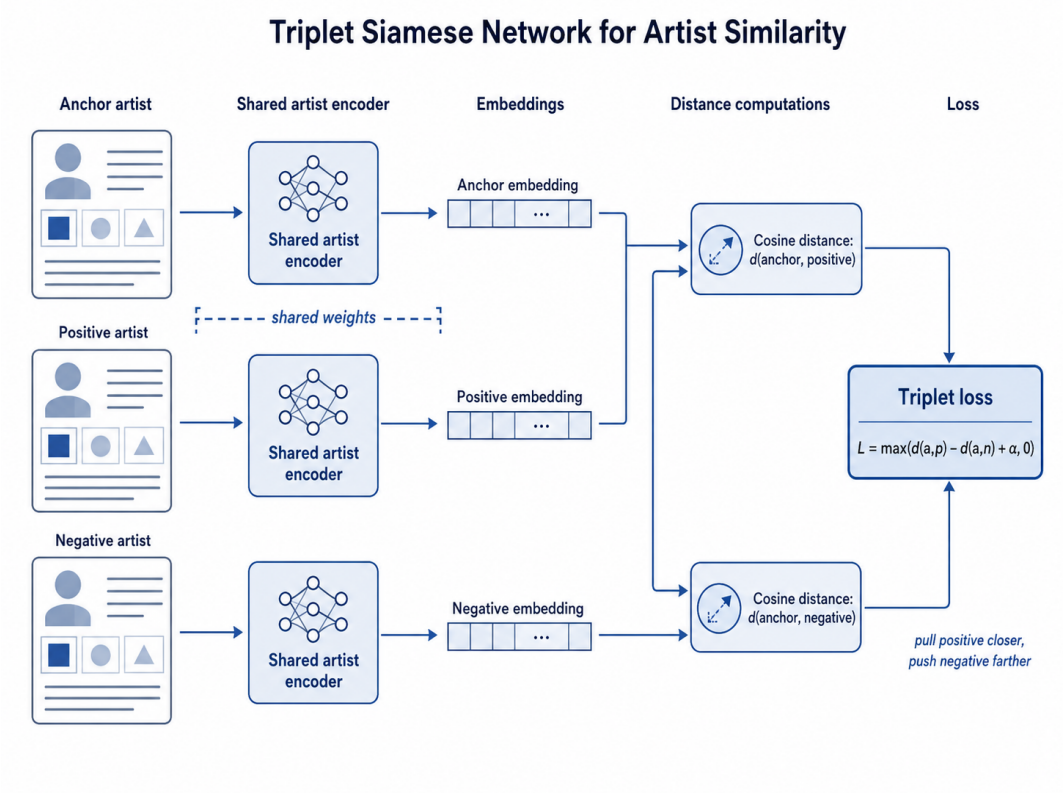


Figure 3.8: Triplet-based Siamese architecture in which anchor, positive, and negative artists share one encoder. Original schematic based on Siamese metric learning [9, 10].

3.14 Triplet Loss

Triplet loss trains an embedding space from an anchor artist a, positive artist p, and negative artist n [19, 10]:

$$\mathcal{L}(a, p, n) = \max\left(d(\hat{\mathbf{z}}_a, \hat{\mathbf{z}}_p) - d(\hat{\mathbf{z}}_a, \hat{\mathbf{z}}_n) + m, 0\right), \tag{3.28}$$

where $d(\cdot, \cdot)$ is a distance function and m is the margin.

The margin defines how much farther the negative artist should be compared with the positive artist. The loss becomes zero when:

$$d(\hat{\mathbf{z}}_a, \hat{\mathbf{z}}_p) + m < d(\hat{\mathbf{z}}_a, \hat{\mathbf{z}}_n). \tag{3.29}$$

Under this condition, the positive artist is closer to the anchor than the negative by at least the margin. Otherwise, training reduces the positive distance and increases the negative distance.

This relative comparison suits artist similarity: the model learns that one artist should be closer to another than to a third, rather than assigning each artist to a fixed class [19, 10].

3.15 Cosine Similarity and Cosine Distance

Cosine similarity measures the angle between two vectors and is widely used in vector-space retrieval models [12]. For two non-zero vectors $\mathbf{u}$ and $\mathbf{v}$, cosine similarity is defined as:

$$s_{\cos}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}. \tag{3.30}$$

Cosine distance is commonly defined as [12]:

$$d_{\cos}(\mathbf{u}, \mathbf{v}) = 1 - s_{\cos}(\mathbf{u}, \mathbf{v}). \tag{3.31}$$

Cosine similarity compares vector direction rather than magnitude, a common choice for high-dimensional semantic embeddings [12]. Because the final artist embeddings are ℓ_2-normalised, their cosine similarity equals their dot product:

$$s_{\cos}(\hat{\mathbf{z}}_i, \hat{\mathbf{z}}_j) = \hat{\mathbf{z}}_i^T \hat{\mathbf{z}}_j. \tag{3.32}$$

In this project, cosine distance is used for training, validation, and retrieval. Figure 3.9 shows why magnitude does not affect this comparison: two vectors with the same direction have maximum cosine similarity even if their lengths differ.

3.16 Fixed-Negative Triplet Training

The quality of the training triplets strongly affects triplet loss [10]. An easily distinguished negative soon yields little learning signal, whereas an unannotated but

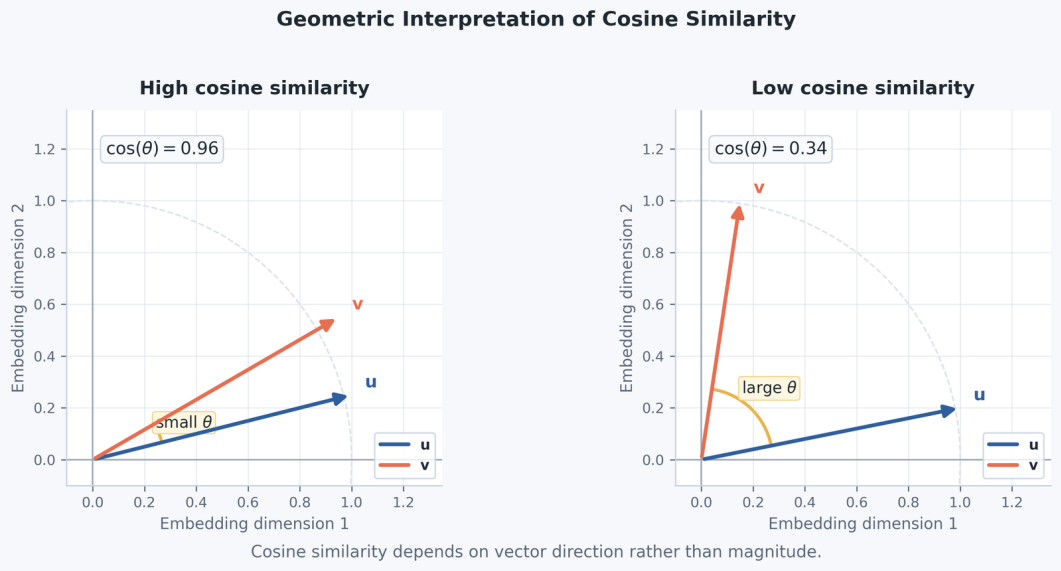


Figure 3.9: Cosine similarity interpreted through the angle between embedding vectors. Original schematic based on the cosine definition [12].

genuinely related artist introduces label noise. Online hard-negative and semi-hard methods choose negatives during training, but they also add decisions that can interact with the model architecture.

This thesis instead uses a fixed triplet set. Every tested architecture receives the same anchor-positive-negative relations, so differences are not caused by model-specific negative selection. Chapter 4 describes how those triplets are generated from the project relation data.

3.17 Artist Similarity as Retrieval

Once each artist has an embedding, comparing a query with all candidates turns similarity estimation into a retrieval problem [12, 2]. For a query artist q, candidates are ranked by cosine similarity:

$$\text{score}(q, c) = s_{\cos}(\hat{\mathbf{z}}_q, \hat{\mathbf{z}}_c). \tag{3.33}$$

The highest-ranked candidates form the related-artist list returned by a recommendation system [12, 2].

Unlike classification, which chooses a label from a fixed set, retrieval produces a ranking. It is therefore better suited to artist similarity as a continuous relation [19, 12].

3.18 Evaluation Metrics

The evaluation covers three related properties. Triplet-level metrics test whether the positive artist is closer than the negative; pair-level metrics measure the separation

of similar and non-similar pairs; retrieval metrics determine whether known similar artists occur near the top of a ranking [12, 2].

3.18.1 Triplet Ranking Accuracy

Triplet ranking accuracy measures whether [10]:

$$d(\hat{\mathbf{z}}_a, \hat{\mathbf{z}}_p) < d(\hat{\mathbf{z}}_a, \hat{\mathbf{z}}_n). \tag{3.34}$$

For N triplets, it is computed as:

$$\mathrm{Acc}_{\mathrm{rank}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[d(\hat{\mathbf{z}}_{a_i}, \hat{\mathbf{z}}_{p_i}) < d(\hat{\mathbf{z}}_{a_i}, \hat{\mathbf{z}}_{n_i})\right], \tag{3.35}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

The numerical assessment in Chapter 5 also reports margin accuracy:

$$\mathrm{Acc}_{\mathrm{margin}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[d(\hat{\mathbf{z}}_{a_i}, \hat{\mathbf{z}}_{p_i}) + m < d(\hat{\mathbf{z}}_{a_i}, \hat{\mathbf{z}}_{n_i})\right]. \tag{3.36}$$

3.18.2 Precision, Recall, and F1 Score

Pair-level evaluation can be performed by selecting a similarity threshold [16]. Two artists are predicted as similar if:

$$s_{\cos}(\hat{\mathbf{z}}_i, \hat{\mathbf{z}}_j) \geq \tau. \tag{3.37}$$

Let TP denote true positives, FP false positives, FN false negatives, and TN true negatives. Precision measures the proportion of predicted similar pairs that are correct:

$$\mathrm{Precision} = \frac{TP}{TP + FP}. \tag{3.38}$$

Recall measures the proportion of true similar pairs that are recovered:

$$\mathrm{Recall} = \frac{TP}{TP + FN}. \tag{3.39}$$

The F1 score combines precision and recall:

$$F1 = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{3.40}$$

3.18.3 Retrieval Metrics

Retrieval metrics directly address artist recommendation [12, 2]. Precision@K measures the proportion of relevant artists among the first K results:

$$\mathrm{Precision@}K = \frac{\text{number of relevant artists in the top } K}{K}. \tag{3.41}$$

Recall@K measures the fraction of the query's known relevant artists retrieved in the first K positions:

$$\text{Recall@}K = \frac{\text{number of relevant artists in the top } K}{\text{number of known relevant artists for the query}}. \tag{3.42}$$

HitRate@K checks whether at least one relevant artist appears in the first K positions:

$$\text{HitRate@}K = \mathbb{I}[\text{at least one relevant artist appears in the top } K]. \tag{3.43}$$

MRR measures the position of the first relevant result:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{r_q}, \tag{3.44}$$

where Q is the set of query artists and r_q is the rank of the first relevant artist for query q.

MRR is useful when the first correct related artist should appear as high as possible in the ranking [12].

3.19 Artist-Disjoint Evaluation

An artist may occur as an anchor, positive, or negative across many triplets. A random row-level split can therefore place the same artist in both training and validation, creating leakage because the validation artist is no longer unseen.

The cross-validation procedure instead uses artist-disjoint splits, so validation artists never appear in training [15]. This stricter protocol tests generalisation to unseen artists rather than merely to unseen combinations of known artists.

This condition reflects the recommendation setting, where the encoder should produce informative representations for artists outside the training set [15, 2].

3.20 Latent Space Visualisation

After training, quantitative retrieval metrics and qualitative visualisations can be applied to the artist space [12, 17, 18].

PCA and t-SNE project the embeddings into two dimensions, with each point representing an artist [17, 18]. Colouring points by genre or country can reveal local stylistic or cultural patterns in the learned space.

Because dimensionality reduction can distort distances, these plots are diagnostic rather than conclusive. They nevertheless help identify local patterns, clusters, and possible errors [17, 18]. Figure 3.10 illustrates how metadata colours can expose local neighbourhoods without turning the projection into a quantitative result.

The following chapter describes the dataset pipeline and model architectures used in this thesis.

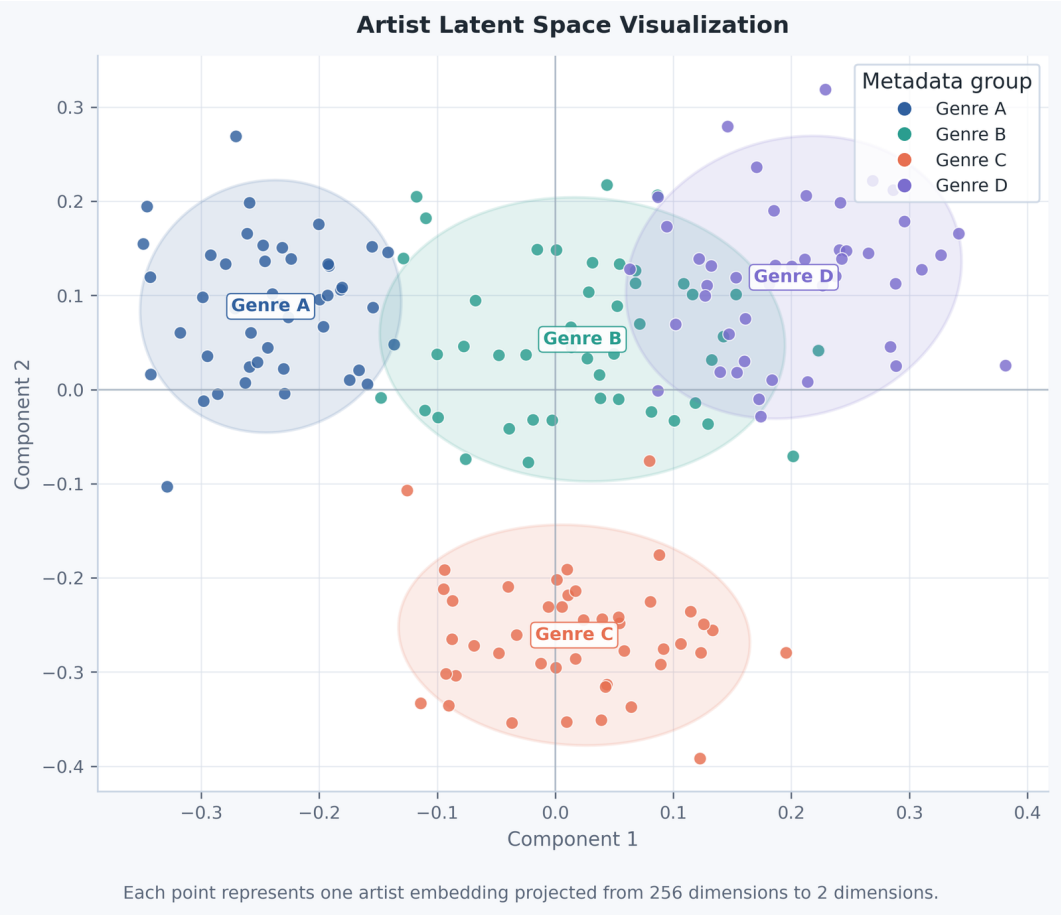


Figure 3.10: Two-dimensional diagnostic view of neighbourhood structure in an artist embedding space. Original schematic based on dimensionality-reduction analysis [17, 18].

Chapter 4

Methodology

This chapter presents the methodology used to learn and evaluate video-based artist representations. It defines the data pipeline, input construction, triplet supervision, artist-disjoint validation, proposed hierarchical encoder, Conv1D baselines, operational loss, checkpoint selection, and implementation choices.

Section 4.1 defines the scope of the method, and Sections 4.2 and 4.3 present the learning problem and system pipeline. Sections 4.4 through 4.7 describe video and artist representations, triplet construction, and artist-disjoint validation. Sections 4.8 and 4.9 introduce the proposed encoder and the comparison models. The final sections specify the operational margin loss and checkpoint rule, consolidate the principal trade-offs, and document reproducibility decisions.

4.1 Scope of the Proposed Method

Earlier work in the project used audio descriptors, while the architecture proposed in [5] used static Instagram images and captions. The present thesis retains the relational artist-similarity objective but replaces those inputs with video-derived visual representations. It evaluates the video branch separately so that its contribution can be measured without immediate fusion with audio or other social-media representations.

Video differs from the earlier inputs in content and structure. Its frames can convey recurring aesthetics, stage performance, fashion, lighting, gestures, visual narratives, and public image even when the audio track is omitted. It also adds a structural level: a collection of videos represents each artist, and a sequence of frame embeddings represents each video. Aggregation must therefore occur from frames to videos and from videos to artists.

This methodological choice leads to the following input representation for the main model:

$$X_a \in \mathbb{R}^{10\times30\times768}, \tag{4.1}$$

where a is an artist, 10 is the maximum number of videos retained for that artist, 30 is the number of sampled frames per video, and 768 is the dimension of each CLIP frame embedding [6]. Unlike the one-dimensional artist vectors used in Instagram image-

caption approaches or audio embeddings derived from log-mel-spectrogram-based models [24], this tensor explicitly represents the video hierarchy.

The method learns a continuous embedding space rather than predicting genre, country, or a fixed artist label. Because similarity is relational, the relevant question is how close two artists are under a given notion of similarity. Metric learning addresses this question directly by optimising distances between representations [19].

4.2 Problem Formulation

Let $\mathcal{A}$ be the set of artists for which video embeddings are available. For each artist $a \in \mathcal{A}$, the input data consist of an artist tensor X_a. The encoder f_θ, parameterised by θ, should learn the following mapping [22, 19]:

$$f_\theta : X_a \mapsto \mathbf{z}_a, \qquad \mathbf{z}_a \in \mathbb{R}^{256}, \tag{4.2}$$

where $\mathbf{z}_a$ is the final latent representation of artist a. The resulting representation $\mathbf{z}_a$ is ℓ_2-normalised so that

$$\|\mathbf{z}_a\|_2 = 1. \tag{4.3}$$

The experiments use the cosine distance defined in Section 3.15 throughout training and retrieval.

The supervision is provided through triplets [10]:

$$(a, p, n), \tag{4.4}$$

where a is the anchor artist, p is a positive artist considered similar to the anchor, and n is a negative artist not considered similar to the anchor. The expected ordering in the learned space is

$$d\left(f_\theta(X_a), f_\theta(X_p)\right) < d\left(f_\theta(X_a), f_\theta(X_n)\right), \tag{4.5}$$

where d is the cosine distance defined in Section 3.15.

The triplet formulation follows Siamese metric learning [9, 10, 19] and matches the available ground truth. AllMusic [7, 11] and Spotify [8] provide anchor-positive relations but no closed set of artist classes. Adding a negative artist converts each relation into a trainable comparison.

A successful embedding ranks known related artists near the top for a query artist a [12]. The validation procedure consequently selects checkpoints with retrieval-style MRR, as described in Section 4.10.

4.3 System Pipeline

The proposed pipeline converts raw videos into reusable sequences of CLIP frame descriptors [6]. It groups these descriptors by artist, retains only triplets whose three artists have valid representations, and creates artist-disjoint training and validation

sets [15]. A shared encoder then maps every artist to a normalised vector for cosine-based retrieval.

Figure 4.1 shows the complete path from video collection to related-artist ranking. The visual extraction stage is executed once per video; later training runs load the stored frame descriptors, which keeps the inputs identical across architectures. The proposed hierarchical encoder receives separate video sequences, whereas the convolutional baselines receive a video-averaged sequence.

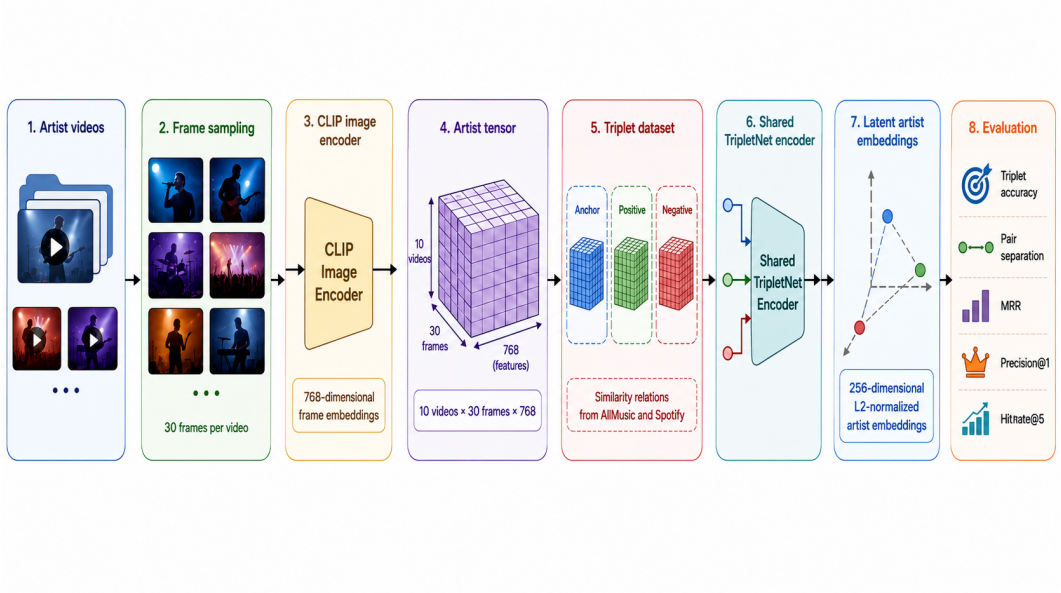


Figure 4.1: Video-based artist-similarity pipeline from frame extraction and CLIP encoding to triplet learning and retrieval. Original schematic developed for this thesis, using CLIP [6].

The system learns an artist representation rather than a video classifier. A training item therefore contains the available visual evidence for an artist, and the loss compares artists through their relation labels. This distinction affects the whole pipeline. Treating each video as an independent sample would give artists with many videos more influence and would assign the artist relation to clips that may be unrepresentative on their own.

4.4 Video Frame Embedding Extraction

A preprocessing routine extracts descriptors from each raw video once and stores them for later experiments. It samples $T = 30$ positions from the central 10% to 90% of a video. For a video with L frames, the candidate interval begins at

$$s = \lfloor 0.1L \rfloor \tag{4.6}$$

and ends before

$$e = \max\left(s + 1, \lfloor 0.9L \rfloor\right). \tag{4.7}$$

Thirty linearly spaced indices are selected between s and $e-1$. Excluding the ends reduces the influence of logos, credits, and blank transitions.

Each selected frame is converted to the red-green-blue (RGB) colour space and processed by the CLIP ViT-L/14@336px image encoder [6, 27]. If $I_{i,t}$ is frame t of video i, the descriptor is

$$\mathbf{x}_{i,t} = g_{\text{CLIP}}(I_{i,t}) \in \mathbb{R}^{768}. \tag{4.8}$$

The video tensor is therefore

$$V_i = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,30}] \in \mathbb{R}^{30\times 768}. \tag{4.9}$$

Unreadable selected frames are skipped. If fewer than thirty valid frames remain, the last valid frame is repeated; a video with no readable frame is discarded. The fixed tensor in Eq. (4.9) is saved once per video and reused during training.

Uniform sampling provides coverage of the selected interval without requiring shot boundaries or motion estimates. Thirty frames can represent several parts of a short clip while keeping the stored descriptor set small enough for repeated training. The procedure is deterministic for a given video, so changes across experiments come from the trainable artist encoder rather than from a new draw of frames.

This sampling rule is only an approximation to video content. A long static shot contributes more sampled frames than a short but informative scene. Fast cuts may be represented by a single frame, and a visual cue outside the central interval is absent. The representation therefore describes regularly spaced appearances, not every event in the video.

The decision to store frame descriptors rather than compressed video-level features leaves temporal aggregation to the artist model. Consecutive descriptors can still express changes in setting, camera position, or performance context. At the same time, CLIP processes each frame independently, so motion is represented only indirectly through differences between descriptors [6]. A model that encodes optical flow or short clips could capture information that this pipeline cannot.

Fixed-length tensors make batching possible but create a difference between observed and repeated evidence when a video has unreadable frames. The validity checks ensure that a completely unreadable video is excluded. Repetition of the last readable descriptor preserves shape but does not add information.

The preprocessing design favours comparability over exhaustive coverage. Every model sees descriptors extracted with the same rule, and the extraction is not altered after inspecting validation results. Consequently, model differences cannot be attributed to architecture-specific frame selection.

4.5 Artist Tensor Construction

Artists have different numbers of usable videos, so the pipeline converts their video tensors to fixed shapes before batching. The proposed model keeps the video axis, whereas the baselines average it.

4.5.1 Stack Aggregation for the Main Model

Up to ten valid videos are retained per artist. Artists with fewer videos receive all-zero tensors in the unused slots, yielding the representation defined in Eq. (4.1). A validity mask prevents padded slots from contributing to attention or pooling.

Figure 4.2 shows this construction. The fixed shape supports mini-batches, but the mask preserves the actual amount of visual evidence available for each artist.

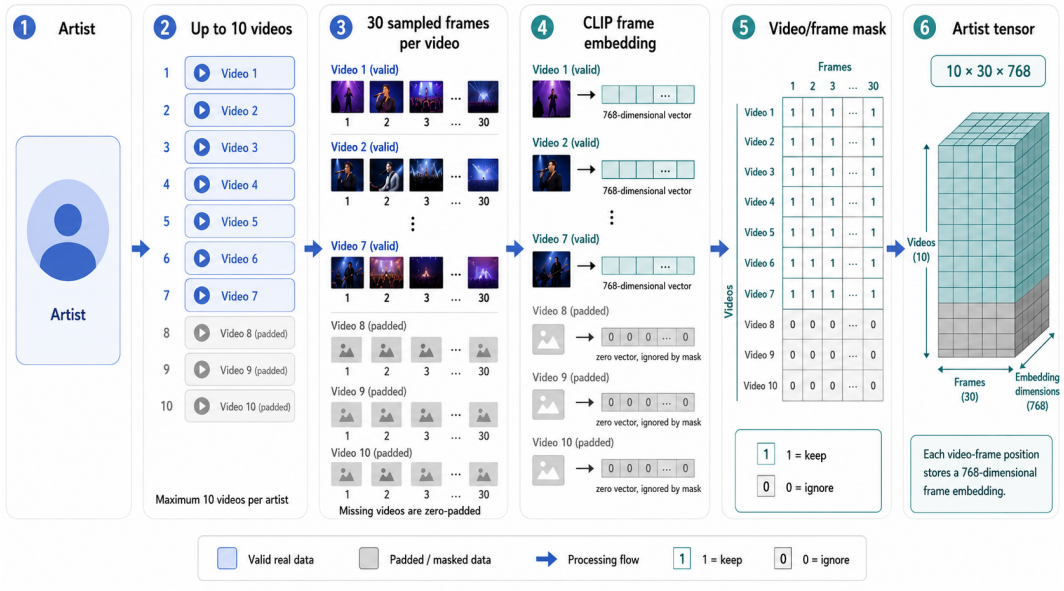


Figure 4.2: Construction of the fixed-size artist tensor, including zero padding and a validity mask for missing videos. Original schematic developed for this thesis.

Let N_a be the number of valid videos for artist a, and let $M = 10$. Stack aggregation is

$$X_a[j] = \begin{cases} V_{a,j}, & 1 \le j \le \min(N_a, M), \\ \mathbf{0}, & \min(N_a, M) < j \le M. \end{cases} \tag{4.10}$$

4.5.2 Mean Aggregation for the Baselines

The convolutional baselines use one sequence per artist. Their input is the mean of the valid video tensors:

$$\bar{X}_a = \frac{1}{N_a} \sum_{i=1}^{N_a} V_{a,i} \in \mathbb{R}^{30\times 768}. \tag{4.11}$$

This representation is compact but removes variation among videos before the encoder can learn from it.

4.5.3 Information retained by artist aggregation

Stack and mean aggregation answer different questions about the available videos. Mean aggregation asks which frame-level properties are common on average. It suppresses the identity of individual videos before learning, which can be useful when clips are noisy or redundant. The main limitation of mean aggregation is that it is

irreversible: after averaging, the model can no longer recover which video contributed a particular feature.

The stacked representation keeps this distinction. The artist-level encoder can assign different roles to a concert recording, an interview, and a promotional clip. It can also reduce the influence of an isolated video when the other tokens provide different evidence. The validity mask is needed because the number of videos varies by artist and an all-zero slot must not be treated as a real observation.

Artists with many available videos still undergo truncation at the fixed maximum. The model therefore receives a bounded sample of their material rather than a complete archive. Artists with only one or two videos provide less varied evidence, even though padding gives them the same tensor shape. Masked pooling prevents padded slots from changing the mean, but it cannot compensate for missing content.

This difference in evidence volume is a property of the dataset, not a class label. The model is not told how popular an artist is or how many videos were originally available. Nevertheless, availability can affect representation quality. Later results should therefore be interpreted as performance for the collected video set, not for an ideal collection with equal coverage per artist.

4.6 Triplet Dataset Construction

Triplets follow the procedure used in the broader project [5]. For each anchor artist, related-artist lists from AllMusic and Spotify identify positive artists. The sources can be considered separately or through their union. A negative artist is sampled at random from artists absent from both relation lists. One triplet is generated for each known positive, and missing reciprocal relations are added so that a recorded relation can supervise both anchor directions.

The combined AllMusic and Spotify table contains 27,367 triplets [5]. AllMusic contributes expert-curated editorial associations [7, 11], while Spotify contributes algorithmic platform relations [8]. The table stores the anchor, positive, and sampled negative artist identifiers. Before training, duplicate rows and triplets containing an artist without a valid video representation are removed.

Figure 4.3 illustrates the selection rule. The negative is not guaranteed to be perceptually dissimilar; it is simply absent from the available relation lists. This distinction matters when interpreting reported errors and large triplet margins.

The sampled negatives remain fixed in the stored table, so every architecture and margin receives the same supervision. This controls one source of variation in the model comparison.

4.6.1 What the triplets supervise

The relation sources do not define the same concept in exactly the same way. AllMusic relations are expert-curated [7, 11], whereas Spotify provides platform-derived related-artist relations [8]. Their union increases coverage but also combines judgements

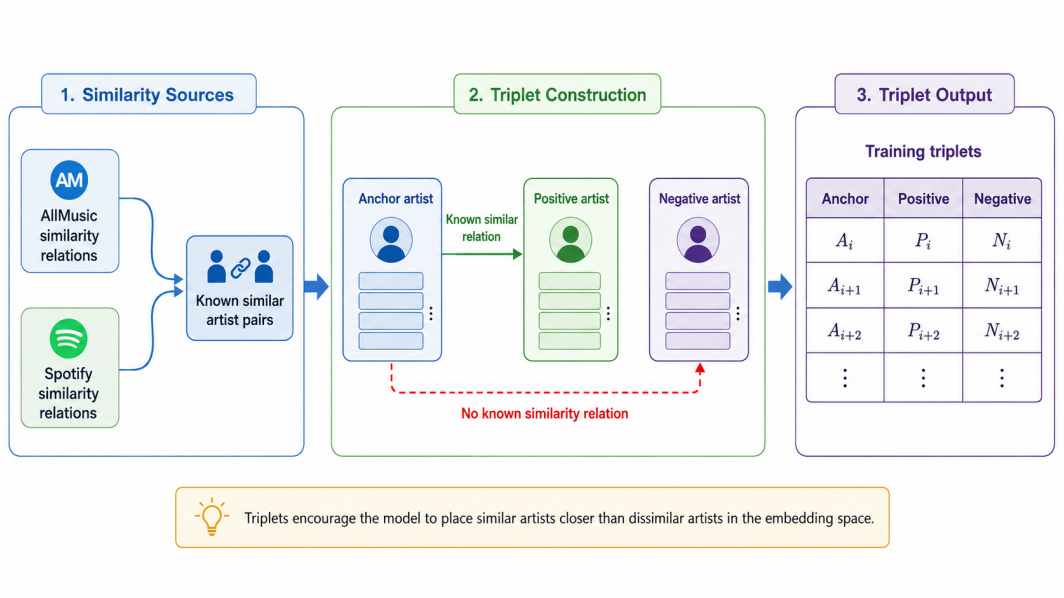


Figure 4.3: Triplet construction from AllMusic and Spotify relations. Original schematic based on the broader project's published procedure [5].

made under different criteria. A learned embedding is therefore expected to recover a shared neighbourhood structure, not a single perceptual rule.

The positive relation has a clearer interpretation than the negative. A positive states that at least one source recorded a relation. A sampled negative states only that neither source recorded that relation for the anchor. The negative may still share genre, audience, geographic context, or visual style with the anchor. The margin experiment tests how strongly the model can enforce these relative labels without assuming that every unlinked pair is dissimilar.

Adding reciprocal relations treats the reference graph as undirected for supervision. This is appropriate for a symmetric embedding distance, although the original services may construct or display their lists asymmetrically. The choice means that retrieval evaluation asks whether a recorded neighbour appears near the query in either direction.

Keeping the sampled negatives fixed removes variation that would otherwise come from drawing a different comparison for each model. It also limits the diversity of training constraints. A model can learn to order the stored comparisons without seeing every difficult unlinked neighbour. Full-catalogue retrieval is included later because it tests the embedding against all available candidates rather than only the sampled negatives.

All models receive the same retained triplets and share one encoder across the anchor, positive, and negative branches. For triplet (X_a, X_p, X_n), the network computes

$$\mathbf{z}_a = f_\theta(X_a), \qquad \mathbf{z}_p = f_\theta(X_p), \qquad \mathbf{z}_n = f_\theta(X_n). \tag{4.12}$$

Sharing the weights θ maps all three inputs into the same latent space, as required by Siamese metric learning [9, 10]. The same encoder is then used for retrieval.

4.7 Artist-Disjoint Splitting

Artist-disjoint validation prevents the same artist from occurring in both training and validation, even in different triplet roles [15]. The protocol partitions artists rather than rows. Let $\mathcal{A}_{\text{train}}$ and $\mathcal{A}_{\text{val}}$ be disjoint sets:

$$\mathcal{A}_{\text{train}} \cap \mathcal{A}_{\text{val}} = \emptyset. \tag{4.13}$$

A triplet belongs to training only when

$$a, p, n \in \mathcal{A}_{\text{train}}, \tag{4.14}$$

and to validation only when

$$a, p, n \in \mathcal{A}_{\text{val}}. \tag{4.15}$$

Triplets crossing the boundary are dropped.

Five folds are constructed by repeatedly shuffling the artist partition and retaining valid splits with non-empty subsets. Candidate splits are ranked by the number of retained validation rows, total retained rows, and closeness to the requested partition ratio. This high-level procedure is applied identically to every architecture.

Figure 4.4 makes the constraint explicit: an artist assigned to validation cannot reappear as an anchor, positive, or negative in training. The resulting folds are smaller than row-level splits but measure generalisation to unseen artists.

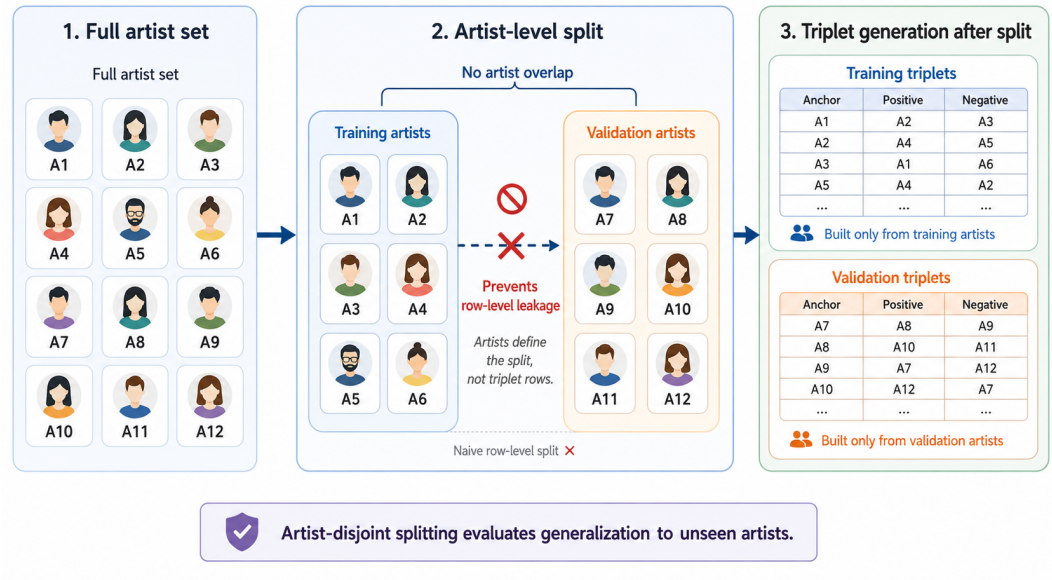


Figure 4.4: Artist-disjoint splitting, which prevents an artist from appearing in both training and validation triplets. Original schematic developed for this thesis.

4.8 TripletNet1: Hierarchical Transformer Encoder

The main architecture proposed in this thesis is TripletNet1, a hierarchical Transformer encoder that preserves the organisation of frames within videos and videos within an artist profile. Its stacked input follows the hierarchy

$$\text{artist} \rightarrow \text{videos} \rightarrow \text{frames}. \tag{4.16}$$

Figure 4.5 shows the two encoder levels. The frame-level Transformer forms one token per video, while the artist-level Transformer compares the valid video tokens before pooling.

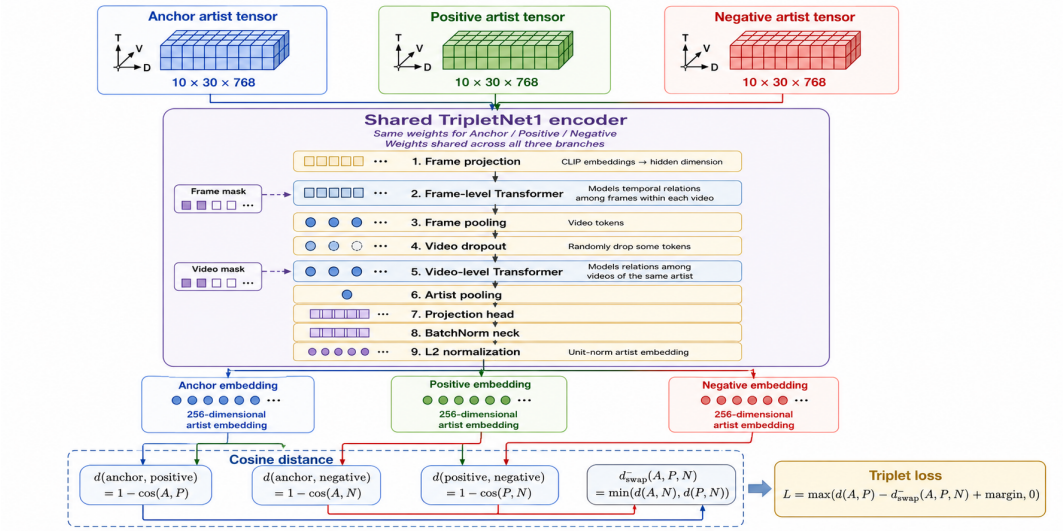


Figure 4.5: TripletNet1 hierarchical Transformer encoder. Original schematic of the proposed architecture, based on Transformer encoders [13].

A video slot is valid when it contains nonzero frame descriptors. The resulting mask excludes padded videos from attention and artist-level pooling. Inspired by dropout regularisation [36], the training procedure applies dropout at the video level: valid video tokens are randomly removed with probability 0.15, while at least one token is retained for every artist.

Each frame descriptor is projected to the internal dimension and receives a learned positional embedding. A one-layer Transformer with eight attention heads contextualises the frame sequence [13]. Mean pooling produces video token

$$\mathbf{v}_{a,j} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{h}'_{a,j,t}. \tag{4.17}$$

The resulting video tokens form the input to the artist-level encoder. A second one-layer Transformer contextualises the valid video tokens. Masked mean pooling forms artist representation

$$\mathbf{r}_a = \frac{\sum_{j=1}^{M} m_{a,j} \mathbf{u}_{a,j}}{\sum_{j=1}^{M} m_{a,j}}. \tag{4.18}$$

A feed-forward projection and batch-normalisation neck map this vector to the final normalised artist representation [33].

4.9 Conv1D Baseline Models

To benchmark the proposed TripletNet1 architecture, three Conv1D baselines are considered: TripletNet2, TripletNet3, and TripletNet4. They receive the mean-aggregated artist sequence and apply one-dimensional convolutions along the frame axis [14]. They differ in depth and downsampling, but each returns an ℓ_2-normalised artist embedding. Table 4.1 places the proposed architecture and baselines side by side.

Table 4.1: Model configurations used in the comparison.

Model	Input	Encoder configuration	Aggregation	Output
TripletNet1	$10 \times 30 \times 768$	Two 1-layer Transformers, 8 heads, hidden size 384	Masked video pooling	256
TripletNet2	30×768	Conv channels 256, 256, 128; kernels 3, 3, 3	Max pooling	256
TripletNet3	30×768	Conv channels 256, 256, 128, 128; kernels 5, 5, 3, 3	Max pooling	256
TripletNet4	30×768	Conv channels 256, 256, 128, 128; kernels 3, 3, 3, 3	Strided convolution	256

TripletNet1 differs from the Conv1D baselines in both input aggregation and encoder design. Consequently, the comparison concerns complete model configurations, and its results cannot be interpreted as a strict single-factor ablation.

4.10 Margin Sweep and Checkpoint Selection

Training uses the operational distance-swap triplet margin loss. For two embeddings $\mathbf{u}$ and $\mathbf{v}$, cosine distance is

$$d(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}. \tag{4.19}$$

Because the artist embeddings are ℓ_2-normalised, this reduces to $d(\mathbf{u}, \mathbf{v}) = 1 - \mathbf{u}^\top \mathbf{v}$.

For a triplet (A, P, N), the swapped negative distance is

$$d^-_{\text{swap}}(A, P, N) = \min\left(d(A, N), d(P, N)\right), \tag{4.20}$$

and the loss used in the experiments is

$$\mathcal{L}(A, P, N) = \max\Big(d(A, P) - d^-_{\text{swap}}(A, P, N) + m, 0\Big). \tag{4.21}$$

Distance swap therefore uses the closer of the anchor-negative and positive-negative distances. A triplet remains difficult when the negative approaches either member of the positive pair. The same formulation is used for every model and fold.

For every model, fold, and tested margin, the best checkpoint is selected by validation MRR [12]. This retrieval criterion is used because the intended application ranks artists rather than evaluating one stored comparison in isolation.

Validation artists are encoded and compared through cosine similarity:

$$S_{ij} = \hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j. \tag{4.22}$$

Self-similarities are excluded, and the relation table is treated symmetrically for retrieval. Ties in validation MRR are broken by ranking accuracy and then validation loss.

4.11 Methodological Design Choices and Trade-offs

The preceding sections define the individual stages of the method. Table 4.2 brings together the principal choices that determine what the comparison controls and what its results can establish.

Table 4.2: Methodological design choices and their implications.

Design choice	Considered alternative	Rationale	Implication for interpretation
Fixed CLIP frame descriptors	End-to-end visual fine-tuning	Provides identical visual inputs across architectures, lowers data and computational requirements, and isolates artist-level aggregation.	The visual backbone cannot adapt specifically to the artist-similarity objective.
Uniform sampling from the central 10%-90% of each video	Shot-aware, motion-aware, or learned frame selection	Provides deterministic and computationally manageable temporal coverage.	Short scenes, fast transitions, boundary content, and explicit motion information may be missed.
Up to ten videos and thirty frames per video	All available videos or variable-length artist inputs	Provides bounded memory cost and fixed-shape mini-batches.	Artists with extensive video collections are truncated, while artists with few videos provide less varied evidence.
Stacked multi-video representation for TripletNet1	Averaging videos before encoding	Preserves the identity and variation of individual videos.	The comparison with the Conv1D baselines changes both input aggregation and encoder architecture, so it is not a single-factor ablation.
Fixed sampled negatives	Online hard-negative or semi-hard-negative mining	Ensures that all architectures receive the same triplet supervision.	Training does not expose every difficult or plausible unlinked neighbour.
Artist-disjoint cross-validation	Random triplet-row splitting	Prevents the same artist from appearing in both training and validation.	Many cross-partition triplets are discarded, reducing the effective sample size.
MRR-based checkpoint selection	Validation-loss-based checkpoint selection	Aligns checkpoint selection with the intended retrieval task.	The checkpoint with the lowest triplet loss may not give the best neighbourhood ranking.

Taken together, these choices give priority to comparability, interpretability, and reproducibility. They are not presented as theoretically optimal for every artist-similarity setting. Several alternatives in Table 4.2 could retain more visual information or provide harder supervision, but they would also change the data requirements or the object being compared.

Fixed visual inputs, fixed triplets, shared folds, and common optimisation settings make the complete systems comparable under one protocol. This consistency does not remove differences in model capacity, aggregation, or inductive bias. The observed ranking therefore concerns the tested configurations, each of which combines an input construction with an encoder, rather than architectures made equivalent in every

respect.

The reported results can support selection of the best complete configuration under the current experimental setting. They cannot assign its advantage to self-attention, hierarchy, parameter count, video stacking, or any other single component. Establishing such causal attribution would require controlled, parameter-matched ablations that are not part of this thesis.

4.12 Implementation and Reproducibility Details

All models are implemented in PyTorch [39]. A fixed random seed and explicit model and optimisation parameters make repeated runs traceable. The inputs and supervision listed in Table 4.2 are retained exactly. Input checks reject missing or malformed artist representations before training, and experiment summaries record the selected fold, margin, epoch, and checkpoint.

4.12.1 Reproducible decision points

Reproduction also depends on the relation-table version, the set of artists with valid descriptors, and the artist assignment for each fold. These decisions determine which triplets reach training and validation. Preserving them is necessary because a small change in artist coverage can remove many cross-boundary rows.

Deterministic inputs do not imply bitwise identical neural-network training on every device. Parallel numerical kernels and hardware can introduce small differences. The five-fold results are therefore more informative than one checkpoint value: they show whether the ordering among models persists when the unseen artists change.

The experiment summaries retain the selected checkpoint and its validation metrics for each fold and margin. This traceability links every out-of-fold prediction to a checkpoint selected without using that artist for training. Section 5.11.1 separately states which fold checkpoint supplies each full-catalogue result.

Chapter 5

Experiments and Results

5.1 Chapter Overview

This chapter reports the numerical assessment of the video-based artist-similarity system. It first describes the available data, retained cross-validation folds, optimisation settings, and metrics. The results then compare margin values and model families through out-of-fold triplet evaluation, pair classification, full-catalogue retrieval, and metadata diagnostics.

The main questions are whether video-derived CLIP descriptors contain a usable artist-similarity signal, whether retaining separate videos improves on pre-encoding averaging, and whether performance on stored triplets translates to catalogue-level retrieval.

5.2 Experimental Data

The experiments contain video descriptors for 3,892 artists. Filtering the 27,367 combined AllMusic and Spotify triplets removes only 17 rows with at least one unavailable artist, leaving 27,350 triplets. Table 5.1 reports the resulting data quantities. The small descriptor-matching reduction preserves essentially the full relation table before artist-disjoint partitioning; all architectures begin from the same filtered rows and artist coverage.

Table 5.1: Dataset quantities used in the experiments. Values are obtained after matching the project relation table to the available video descriptors.

Quantity	Value
Artists with available video descriptors	3,892
Raw AllMusic and Spotify triplets	27,367
Filtered triplets	27,350
Discarded triplets	17

5.2.1 Coverage and representativeness

The loss of only 17 triplets shows that descriptor availability matches the relation table closely. It does not show that the artist sample is representative of all musicians. An artist needs both a relation entry and usable video material to appear in the experiments. Artists with limited online presence, restricted content, or videos that could not be processed are less likely to be represented.

Coverage also varies within the retained set. Some artists contribute several distinct videos, while others contribute only one usable clip. The hierarchical model masks absent slots, so padding does not create false observations, but the amount and variety of visual evidence still differ. The numerical assessment averages over this variation.

The relation table has its own selection effects. Established artists and those with active platform profiles are more likely to have catalogue relations. Coverage can vary with genre, country, career stage, and popularity [1, 7, 8]. The experiment measures generalisation within this collected relation network.

These limitations do not invalidate the comparison among architectures because every model uses the same artists and triplets. They limit external interpretation. A model that performs well here has learned useful structure for the available catalogue, but the result does not establish equal performance for less documented artists or for video platforms with different content.

5.3 Artist-Disjoint Cross-Validation Protocol

The artist-disjoint construction defined in Section 4.7 produces five shared folds. Table 5.2 lists the retained training and validation rows. Cross-boundary triplets are excluded, so every validation artist is unseen in the corresponding training fold.

Table 5.2: Artist-disjoint cross-validation folds shared by all models and margins. Artist counts refer to retained triplets, whereas partition counts include artists assigned before row filtering.

Fold	Train rows	Val. rows	Dropped	Kept ratio	Train artists	Val. artists	Train partition	Val. partition
1	14,117	228	13,005	0.5245	2,404	375	2,405	602
2	13,790	235	13,325	0.5128	2,405	378	2,405	602
3	14,337	218	12,795	0.5322	2,405	365	2,406	601
4	14,060	216	13,074	0.5220	2,406	366	2,406	601
5	13,837	235	13,278	0.5145	2,406	384	2,406	601

Figure 5.1 compares the retained row counts. Training sizes are stable across folds, while validation contains between 216 and 235 triplets. The five validation subsets form an out-of-fold set of 1,132 triplets.

Roughly half of the raw triplets cross an artist partition and are dropped. This is the numerical consequence of retaining only triplets contained within one artist partition under the protocol in Section 4.7.

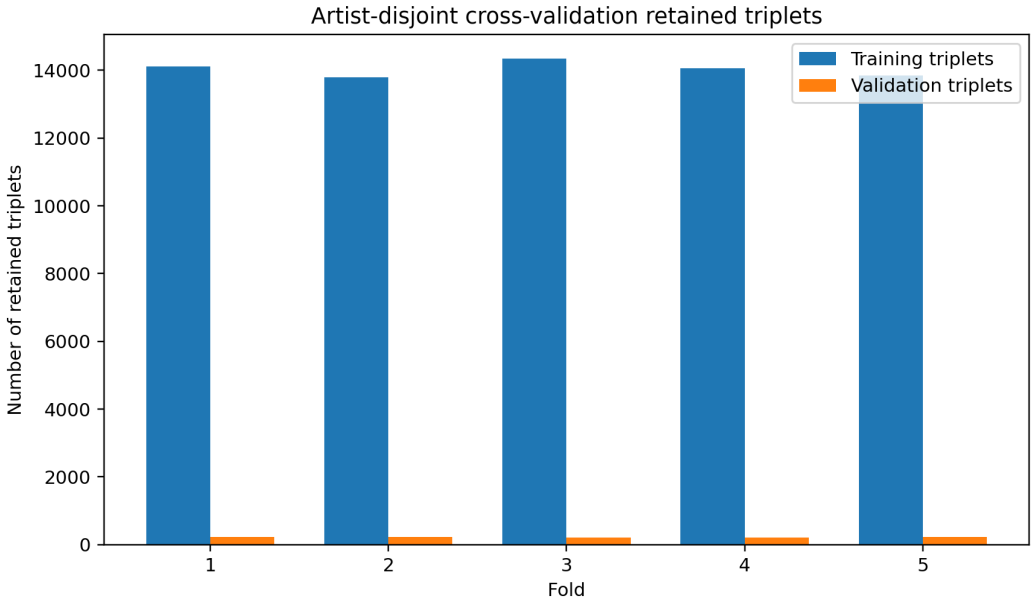


Figure 5.1: Training and validation triplet counts after artist-disjoint splitting. Generated from the experiment folds reported in Table 5.2.

5.3.1 Fold balance and effective sample size

The retained training sets range from 13,790 to 14,337 triplets, a difference of 547 rows. Validation sets range from 216 to 235 triplets. This narrow range reduces the chance that a model benefits simply from a much larger fold, although the identity of the artists can still make one fold harder than another.

Artist counts show the same pattern. Between 365 and 384 artists occur in retained validation triplets. These counts are lower than the artist partitions because many assigned artists do not form a triplet entirely within the partition. The distinction matters: the partition controls leakage, while retained rows determine the numerical sample used for a fold metric.

The five validation sets are disjoint in artist identity, but the five training sets overlap. Fold averages therefore describe sensitivity to the unseen-artist partition; they are not five independent samples from a population of all artists. Standard deviations reported below should be read as variation across these constructed folds rather than confidence intervals.

Dropping roughly half of the rows reduces the number of observed constraints, especially relations that join artists assigned to different partitions. This changes the relation graph seen during training. Since the folds are shared, the reduction affects every model equally, but it can make small differences in fold composition visible in MRR.

5.4 Training Configuration

Table 5.3 collects the common optimisation settings. All model families use these values, the same folds, and the same fixed triplets.

Table 5.3: Common training hyperparameters used in the experiments.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	1×10^{-5}
Batch size	128
Maximum epochs	30
Early stopping patience	8 epochs
Scheduler	ReduceLROnPlateau
Scheduler factor	0.7
Scheduler patience	3 epochs
Gradient clipping	1.0
Random seed	3407
Fold count	5
Distance function	Cosine distance
Distance-swap option	Enabled

Training uses the operational distance-swap triplet margin loss defined in Section 4.10. The tested margins are $m \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

5.4.1 Fairness of the optimisation comparison

All architectures use one optimisation schedule. This avoids model-specific tuning on the validation folds and keeps the comparison manageable. It may not be optimal for every model. Convolutional encoders and Transformers can respond differently to learning rate, regularisation, and batch size, so the reported ranking concerns the shared configuration.

Early stopping uses the retrieval criterion defined in Section 4.10. A model is not required to reach its minimum validation loss if an earlier checkpoint gives a better first-positive rank. This is consistent with the intended use, but it explains why loss and MRR do not always select the same epoch.

The margin grid is evaluated separately for every model and fold. Model selection then uses the mean validation MRR rather than the best single fold. This avoids choosing an architecture because of one unusually favourable partition. The grid is still finite, so the selected value is the best among the tested margins, not an estimate of a continuous optimum.

5.5 Evaluation Metrics

Section 3.18 defines triplet ranking accuracy, margin accuracy, thresholded precision, recall, F1, Precision@K, Recall@K, HitRate@K, and MRR. This chapter reports the

triplet measures for stored comparisons, pair measures for out-of-fold score separation, and retrieval measures for catalogue ranking.

The pair analysis also reports the area under the receiver operating characteristic curve (ROC-AUC) and average precision, together with TP, FP, FN, and TN counts at the selected threshold [16]. Full-catalogue evaluation ranks every other encoded artist for each query and also reports average precision over all known positives [12, 2].

Country and genre labels are used only after training. Mean within-group and between-group similarities, t-SNE plots, and silhouette scores diagnose local and global structure without treating metadata as supervision [18, 40].

5.6 Cross-Validation Model Selection

Table 5.4 gives each model's highest mean validation MRR across the margin grid. Every model selects 0.1. TripletNet1 reaches 0.3114, approximately 11.8% above TripletNet2 at 0.2787, 13.3% above TripletNet3, and 15.0% above TripletNet4.

Table 5.4: Best cross-validation results by model. The selection metric is mean validation MRR across the five artist-disjoint folds.

Model	Best margin	Mean val. MRR	Mean val. acc.	Mean val. loss
TripletNet1	0.1	0.3114	0.9050	0.0429
TripletNet2	0.1	0.2787	0.9161	0.0366
TripletNet3	0.1	0.2749	0.9206	0.0390
TripletNet4	0.1	0.2708	0.9046	0.0422

Figure 5.2 makes the same comparison visually. The hierarchical model has the highest retrieval score, but the gap is moderate on these small validation folds.

The fold-level values in Table 5.7 show that TripletNet1 has the highest MRR in Folds 1, 2, 3, and 5. TripletNet4 is higher in Fold 4 by 0.0089, so TripletNet1 leads on four artist partitions rather than relying on one exceptional fold.

The paired difference between TripletNet1 and TripletNet2 ranges from 0.0137 to 0.0624 and is positive in every fold. Against TripletNet3, it ranges from 0.0220 to 0.0426 and is again positive throughout. With only five folds, these differences are descriptive, but their direction is consistent.

TripletNet3 has the highest mean validation triplet accuracy even though its MRR is lower. Ranking accuracy evaluates one positive-negative comparison at a time, whereas MRR evaluates the resulting neighbourhood ordering among validation artists. A model can perform well on the first task without producing the best retrieval order.

The evidence supports TripletNet1 as the strongest complete configuration tested here. It does not attribute the difference separately to attention, parameter count, hierarchy, or input aggregation.

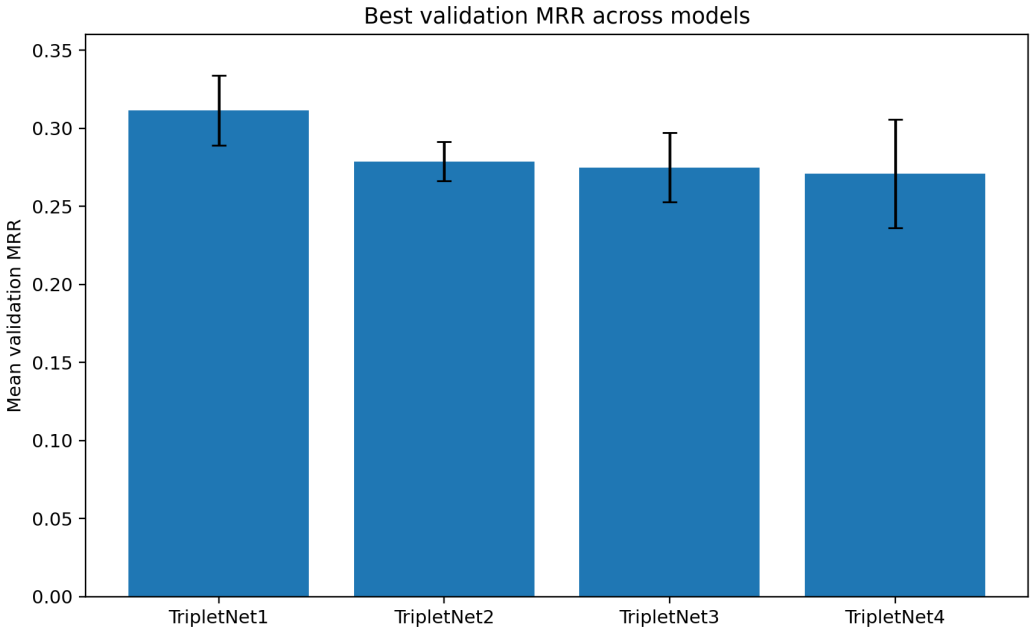


Figure 5.2: Best mean validation MRR for the tested models. Error bars indicate the standard deviation across the five artist-disjoint folds. Generated from the cross-validation summaries in Table 5.4.

5.7 Margin Sensitivity Results

The five margins listed in Section 5.4 are first evaluated for TripletNet1. Table 5.5 reports its sensitivity across this grid. The highest MRR occurs at $m = 0.1$; it falls from 0.3114 to 0.2607 across the tested range. Ranking accuracy remains high, whereas margin accuracy declines as the constraint becomes stricter.

Table 5.5: TripletNet1 margin sensitivity. Values are mean and standard deviation over the five artist-disjoint folds.

Margin	Val. MRR	Std.	Val. acc.	Std.	Margin acc.	Std.	Val. loss	Std.	Best epoch
0.1	0.3114	0.0223	0.9050	0.0133	0.8795	0.0219	0.0429	0.0065	8.0
0.3	0.3108	0.0269	0.9153	0.0161	0.7878	0.0432	0.0917	0.0144	6.0
0.5	0.2984	0.0182	0.9102	0.0122	0.6744	0.0550	0.1612	0.0226	6.8
0.7	0.2716	0.0213	0.9050	0.0200	0.5548	0.0658	0.2608	0.0242	8.4
0.9	0.2607	0.0148	0.8970	0.0169	0.4457	0.0260	0.3688	0.0278	6.6

The lower MRR at larger margins indicates that a softer separation constraint is better for this dataset. Because the external relation tables are incomplete, a large margin may push apart unlinked artists that still share visual or stylistic characteristics [2]. A margin of 0.1 separates the annotated pairs without imposing as large a distortion on the latent space.

Figure 5.3 shows that the MRR decline becomes marked beyond $m = 0.3$. Ranking accuracy changes less, which confirms that margin satisfaction and retrieval quality respond differently to the constraint.

The convolutional baselines follow the same pattern. In Table 5.6, every model

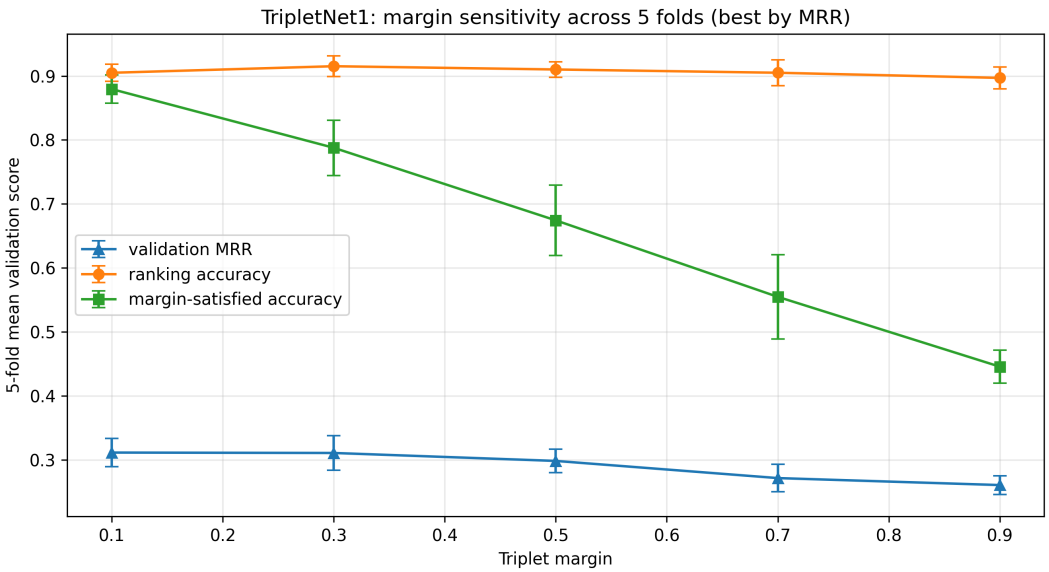


Figure 5.3: TripletNet1 margin sensitivity across the tested values. Error bars indicate the standard deviation across the five artist-disjoint folds. Generated from the cross-validation outputs in Table 5.5.

reaches its highest mean MRR at $m = 0.1$ and declines at larger margins. The selected value is therefore not specific to TripletNet1.

Table 5.6: Mean validation MRR across the margin grid for all models.

Model	$m = 0.1$	$m = 0.3$	$m = 0.5$	$m = 0.7$	$m = 0.9$
TripletNet1	0.3114	0.3108	0.2984	0.2716	0.2607
TripletNet2	0.2787	0.2742	0.2647	0.2354	0.2220
TripletNet3	0.2749	0.2680	0.2561	0.2458	0.2042
TripletNet4	0.2708	0.2598	0.2451	0.2259	0.2070

The margin trend is consistent with the supervision. Artist similarity is not a clean class-membership problem, and the absence of an AllMusic [7, 11] or Spotify [8] link does not establish that two artists are musically or visually unrelated [2]. Strict margins may amplify this label incompleteness by separating plausible neighbours.

A smaller margin requires the annotated positive to be closer without demanding an extreme gap [19, 10]. The space can therefore retain local neighbourhoods not recorded in the triplet file, including retrieved artists that may be semantically plausible despite their absence from the ground truth [2].

For TripletNet1, $m = 0.1$ and $m = 0.3$ produce nearly identical mean MRR values, 0.3114 and 0.3108. The result supports a low-margin region more strongly than one exact value. The larger decline begins at $m = 0.5$ and continues through $m = 0.9$.

The smallest tested value lies at the boundary of the grid. It is therefore not possible to conclude that 0.1 is the global optimum. Margins below 0.1, or a formulation without a fixed margin, could perform similarly or better. The present

experiment establishes that the larger tested constraints are harmful under this supervision.

Validation loss increases with the margin because the loss contains the requested gap. This rise is partly mechanical and should not be compared across margins as if every row used the same objective. MRR and ranking accuracy remain comparable because their definitions do not change with the training margin.

Margin accuracy declines from 0.8795 to 0.4457 for TripletNet1. At the same time, ordinary ranking accuracy changes much less. Many triplets still have the correct order at large margins but fail the wider separation constraint. The extra pressure does not translate into better neighbourhood ranking.

5.8 Fold-Level Stability

After selecting $m = 0.1$, Table 5.7 reports each model and fold's best epoch, total epochs, validation MRR, ranking accuracy, and validation loss.

Table 5.7: Fold-level validation results at the selected margin $m = 0.1$.

Model	Fold	Best epoch	Epochs run	Val. MRR	Val. acc.	Val. loss
TripletNet1	1	13	21	0.3393	0.8947	0.0433
TripletNet1	2	5	13	0.2968	0.8936	0.0406
TripletNet1	3	5	13	0.3270	0.9174	0.0489
TripletNet1	4	12	20	0.3102	0.9213	0.0331
TripletNet1	5	5	13	0.2840	0.8979	0.0485
TripletNet2	1	9	17	0.2769	0.9167	0.0413
TripletNet2	2	4	12	0.2767	0.9319	0.0303
TripletNet2	3	2	10	0.2817	0.8991	0.0382
TripletNet2	4	5	13	0.2965	0.9306	0.0321
TripletNet2	5	1	9	0.2615	0.9021	0.0413
TripletNet3	1	1	9	0.2967	0.9386	0.0306
TripletNet3	2	5	13	0.2604	0.9277	0.0363
TripletNet3	3	12	20	0.2859	0.9220	0.0470
TripletNet3	4	5	13	0.2882	0.9213	0.0375
TripletNet3	5	5	13	0.2434	0.8936	0.0437
TripletNet4	1	10	18	0.2736	0.8947	0.0448
TripletNet4	2	2	10	0.2315	0.9149	0.0375
TripletNet4	3	8	16	0.2855	0.8991	0.0480
TripletNet4	4	9	17	0.3191	0.9120	0.0382
TripletNet4	5	5	13	0.2444	0.9021	0.0425

TripletNet1 has the highest validation MRR in four folds and the highest mean. TripletNet4 reaches 0.3191 in Fold 4 but remains lower on average, while TripletNet3's higher ranking accuracy in some folds does not yield a higher MRR.

Training usually stops before thirty epochs. TripletNet1 reaches its best checkpoint between epochs 5 and 13; some convolutional baselines peak earlier, notably TripletNet3 in Fold 1 and TripletNet2 in Fold 5. These results show that validation

MRR often peaks well before the maximum epoch count.

The artist-disjoint protocol makes validation noisier than a random row split because each fold contains only a few hundred triplets from unseen artists. This variance is the cost of avoiding a smoother but leaked estimate [15].

TripletNet1 varies from 0.2840 to 0.3393 in fold MRR. The range is large enough that one fold cannot summarise the model. It also shows why the common folds matter: comparing models trained on different partitions could confuse an architectural difference with a change in validation artists.

The strongest convolutional result occurs for TripletNet4 in Fold 4, while that model is substantially weaker in Folds 2 and 5. This suggests that its averaged representation works well for some artist groups but is less stable across the available partitions. TripletNet1 is not best on every fold, yet its lower results remain closer to its mean.

Best epochs vary both across folds and models. Early checkpoints are not necessarily evidence of underfitting. Since selection uses MRR, continued reduction of triplet loss can stop improving the order of validation neighbours. The patience setting allows several epochs for that order to recover before training ends.

No formal significance test is reported for the five folds. Such a test would have limited power and would need to account for overlapping training sets. The fold table instead supports a narrower statement: the mean advantage of TripletNet1 has the same sign against TripletNet2 and TripletNet3 on every tested artist partition.

5.9 Out-of-Fold Triplet Evaluation

Each fold's best checkpoint encodes its validation triplets, and predictions from all five folds are concatenated into an out-of-fold set of 1,132 triplets. Every prediction therefore concerns artists unseen by its corresponding training fold [15].

Aggregating every validation fold avoids selecting a favourable split and supplies all positive and negative pairs for ROC-AUC and average-precision estimates [16].

Table 5.8 shows identical ranking accuracy of 0.9046 for TripletNet1 and TripletNet4, with slightly higher values for TripletNet2 and TripletNet3. TripletNet1 nevertheless has the highest margin accuracy, positive-negative similarity gap, ROC-AUC, and average precision. It separates the pair classes more clearly despite ordering fewer individual triplets correctly than two baselines.

Table 5.8: Out-of-fold triplet and pairwise metrics at the best margin. Each model is evaluated on 1,132 validation triplets aggregated across folds.

Model	Ranking acc.	Margin acc.	Pos. sim.	Neg. sim.	Gap	ROC-AUC	AP	Best F1
TripletNet1	0.9046	0.8790	0.5820	0.0051	0.5769	0.9254	0.9127	0.8574
TripletNet2	0.9161	0.8445	0.7746	0.3434	0.4311	0.9127	0.9003	0.8432
TripletNet3	0.9205	0.8578	0.7932	0.3290	0.4642	0.9113	0.8895	0.8449
TripletNet4	0.9046	0.8463	0.7835	0.3141	0.4693	0.9129	0.8998	0.8409

Figure 5.4 visualises the positive and negative means in Table 5.8 together with

their mean difference. TripletNet1 assigns lower similarities overall but creates the widest separation between the two pair classes.

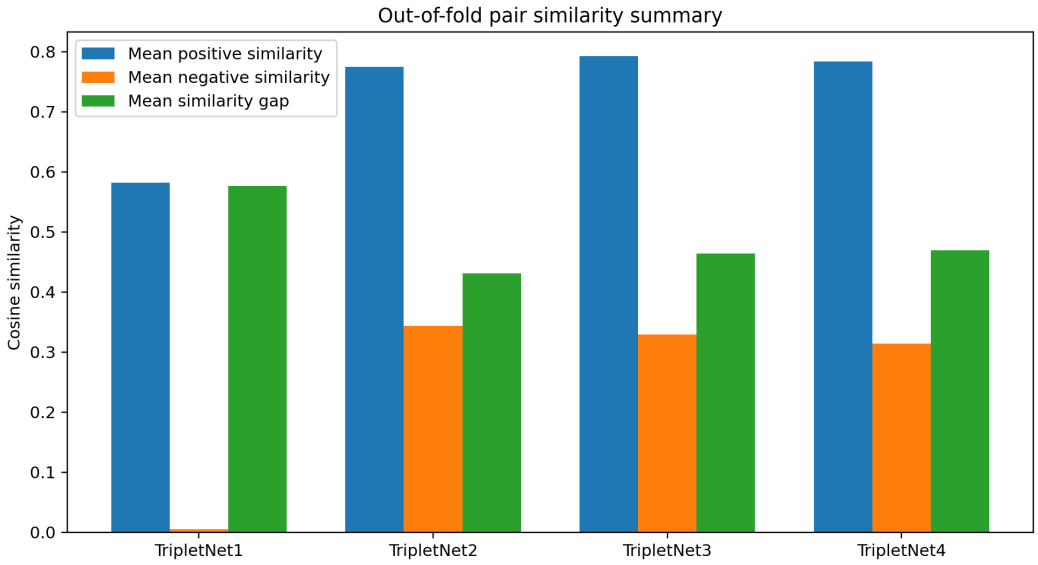


Figure 5.4: Mean out-of-fold positive and negative cosine similarities and their mean gap. Generated from the pair scores summarised in Table 5.8.

TripletNet1 has lower mean similarities for both pair types, but its negative mean lies near zero and produces the largest gap. The baselines assign high similarity to positive and negative pairs alike, yielding denser spaces that are harder to separate with a threshold.

The distributions reported in Figure 5.5 confirm this summary: TripletNet1's positive scores are shifted above its negative scores, although substantial overlap remains. A single threshold cannot separate every pair, which agrees with the ROC-AUC and F1 values in Table 5.8.

The mean similarity gap is reported in Figure 5.4. For each triplet, the corresponding margin gap is defined as

$$g_i = d(a_i, n_i) - d(a_i, p_i). \tag{5.1}$$

A positive gap places the negative farther than the positive, and a gap above the selected margin satisfies the full constraint. TripletNet1 has the highest margin accuracy, so it meets that constraint for the largest share of validation triplets. Margin accuracy and validation MRR thus both favour the hierarchical model.

Ranking accuracy depends on the sign of each triplet gap, ROC-AUC orders all positive and negative pair scores, and the mean similarity gap depends on their absolute locations. These measures can therefore give different model orderings without a calculation error.

TripletNet1 has a lower positive mean than the baselines, but its negative mean is much closer to zero, producing the largest separation. TripletNet3 assigns high similarity to many positive and negative pairs, which supports strong within-triplet

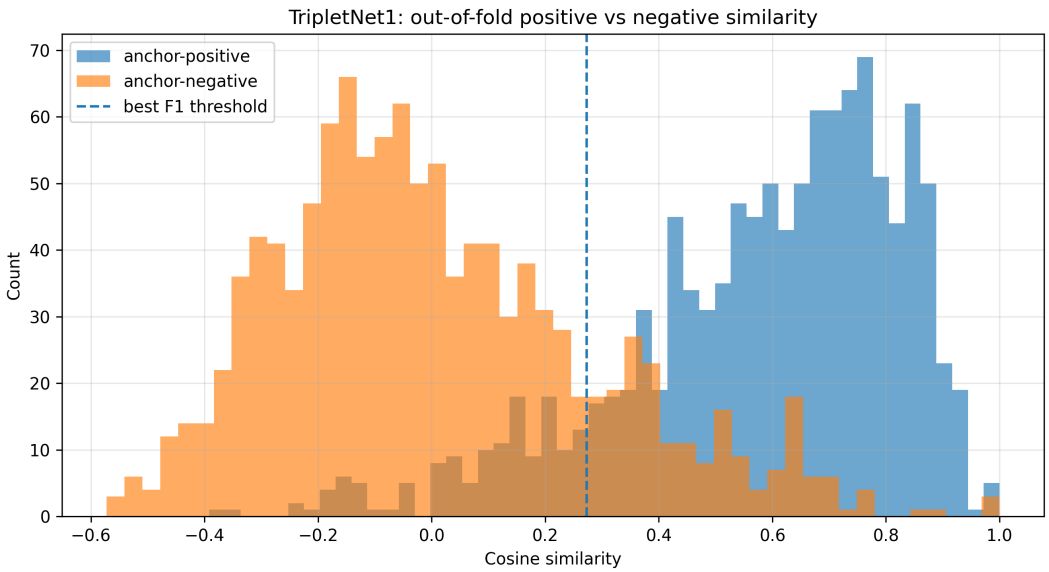


Figure 5.5: Out-of-fold positive and negative similarity distributions for TripletNet1. Generated from the 1,132 out-of-fold triplets.

ordering but weaker global pair separation.

The out-of-fold data contain one positive and one sampled negative per triplet, so the pair classes are balanced by design. A deployment candidate set would be highly imbalanced, and precision on these out-of-fold pairs should not be interpreted as deployment precision.

5.10 Threshold-Based Pair Classification

A cosine-similarity threshold converts the pair score into a binary decision [16]. The threshold that maximises F1 on out-of-fold pairs is selected; Table 5.9 reports that operating point and its confusion matrix.

Table 5.9: Best threshold-based pair classification on out-of-fold positive and negative pairs.

Model	Threshold	Precision	Recall	Specificity	F1	TP	FP	FN	TN
TripletNet1	0.2732	0.8368	0.8790	0.8286	0.8574	995	194	137	938
TripletNet2	0.6139	0.8341	0.8525	0.8304	0.8432	965	192	167	940
TripletNet3	0.5977	0.8141	0.8781	0.7995	0.8449	994	227	138	905
TripletNet4	0.5168	0.7778	0.9152	0.7385	0.8409	1036	296	96	836

TripletNet1 has the highest F1, with precision 0.8368 and recall 0.8790. TripletNet4 recovers more positives but has lower precision and specificity, accepting more negative pairs. The selected TripletNet1 threshold provides the best measured balance.

Figure 5.6 shows how precision, recall, and F1 change with the operating point. A discovery system may favour recall, whereas a curated interface may require higher precision [16, 2].

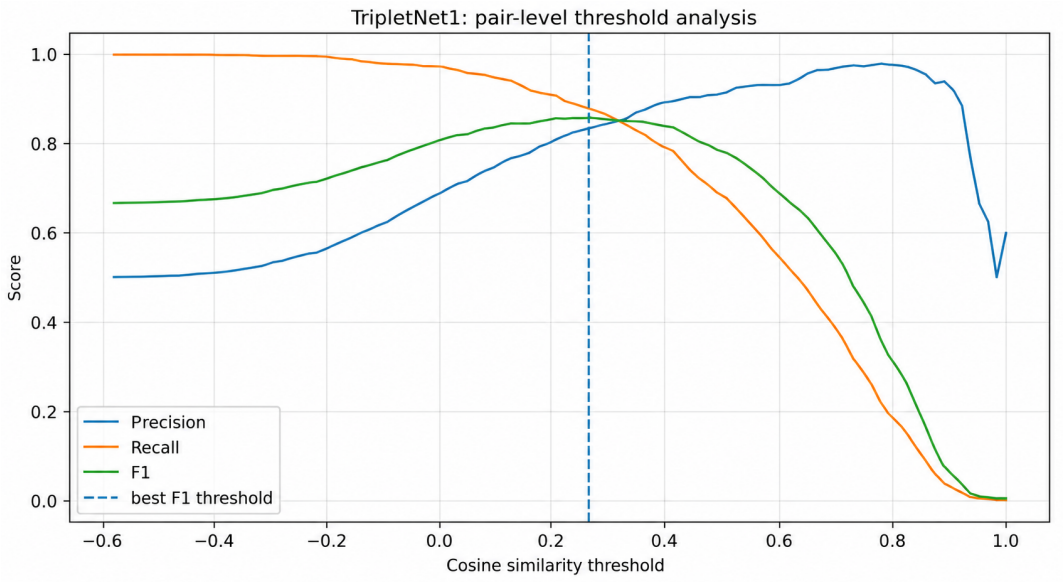


Figure 5.6: Threshold-based pair classification for TripletNet1. Generated from the out-of-fold pair scores.

TripletNet1 uses a lower threshold because its similarity scale differs from those of the baselines. Absolute thresholds are not comparable across models; ROC-AUC and average precision assess separation on each model's own scale.

The same out-of-fold pairs are used to select the F1-maximising threshold and to report the resulting F1. The value therefore describes the best operating point on these data and may be optimistic for a new catalogue.

Thresholds also differ substantially across models because their similarity distributions occupy different ranges. A fixed cross-model threshold is therefore inappropriate, and the reported operating points are specific to each model and to the positive prevalence in these data.

The confusion matrices are still useful for diagnosing error type. TripletNet4 favours recall and produces 296 FPs, whereas TripletNet1 produces 194. This difference is visible even though both models have the same out-of-fold ranking accuracy. The threshold analysis therefore adds information that the triplet order alone cannot provide.

Nearest-neighbour recommendation does not require a binary threshold. A system can return the top K candidates for every query and allow a later stage to rerank them. The full-catalogue analysis is the more direct evaluation of that use case.

5.11 Full-Catalogue Retrieval Evaluation

5.11.1 Retrieval setup

Full-catalogue evaluation encodes all 3,892 available artists. Of these, 2,795 have at least one known positive and serve as queries, with approximately 9.79 positives each on average. The known relation set provides the relevance labels [12, 7, 8].

The results in Table 5.10 use one cross-validation checkpoint for each model. They do not use an ensemble, aggregate embeddings or scores across folds, or retrain a model on the complete triplet table. Cross-validation first selects $m = 0.1$ from the highest mean validation MRR across the five folds. At that margin, the checkpoint with the highest fold-level validation MRR is chosen; ties are resolved by ranking accuracy and then validation loss. This selects Fold 1 at epoch 13 for TripletNet1, Fold 4 at epoch 5 for TripletNet2, Fold 1 at epoch 1 for TripletNet3, and Fold 4 at epoch 9 for TripletNet4.

The Fold 1 checkpoints were trained on 14,117 retained fixed triplets involving 2,404 artists from an assigned training partition of 2,405 artists. The Fold 4 checkpoints used 14,060 triplets involving all 2,406 artists in that training partition. These are rows of the filtered relation table for which the anchor, positive, and stored negative all belong to the selected fold's training partition. Thus, no selected checkpoint was trained on all 27,350 triplets or all 3,892 artists.

Each selected checkpoint encodes all 3,892 artist tensors once, and the same embedding matrix supplies the queries and candidates. The 256-dimensional vectors are ℓ_2-normalised, so their dot products give cosine similarity. The query's diagonal entry is excluded from its ranking. Known positives are the anchor-positive edges in the filtered relation graph, made symmetric and restricted to encoded artists. This is a descriptive catalogue analysis of the selected fold checkpoints, not an additional artist-disjoint test.

Unlike validation on one positive and one negative, full-catalogue retrieval ranks thousands of candidates. It is closer to the intended application and naturally yields lower absolute Precision@K [12, 2].

5.11.2 Retrieval Results and Interpretation

Table 5.10 reports metrics at $K = 1$, 5, 10, and 20, together with MRR and average precision. P@K and R@K denote Precision@K and Recall@K, respectively. H@K denotes HitRate@K, while AP denotes average precision. TripletNet1 leads every measure. Its Precision@1 of 0.2759 places a known positive first for about 27.6% of queries; HitRate@5 of 0.5657 places one within the first five for more than half, and HitRate@20 reaches 0.7828.

Table 5.10: Full-catalogue retrieval metrics. All 3,892 encoded artists are used as candidates, and 2,795 artists with at least one known positive are used as queries.

Model	P@1	R@1	H@1	P@5	R@5	H@5	P@10	R@10	H@10	P@20	R@20	H@20	MRR	AP
TripletNet1	0.2759	0.0310	0.2759	0.2112	0.1145	0.5657	0.1798	0.1893	0.6844	0.1456	0.2987	0.7828	0.4055	0.1826
TripletNet2	0.2021	0.0236	0.2021	0.1540	0.0821	0.4587	0.1354	0.1394	0.5961	0.1125	0.2257	0.7091	0.3263	0.1361
TripletNet3	0.1506	0.0183	0.1506	0.1189	0.0623	0.3828	0.0994	0.1023	0.4962	0.0828	0.1638	0.6111	0.2627	0.0987
TripletNet4	0.2093	0.0240	0.2093	0.1628	0.0871	0.4701	0.1414	0.1459	0.6050	0.1188	0.2396	0.7206	0.3335	0.1437

Figure 5.7 compares the principal retrieval measures. The hierarchical model's advantage is visible at both the first position and the longer cutoffs, rather than coming from one metric alone.

Relative to TripletNet2, TripletNet3, and TripletNet4, TripletNet1 improves

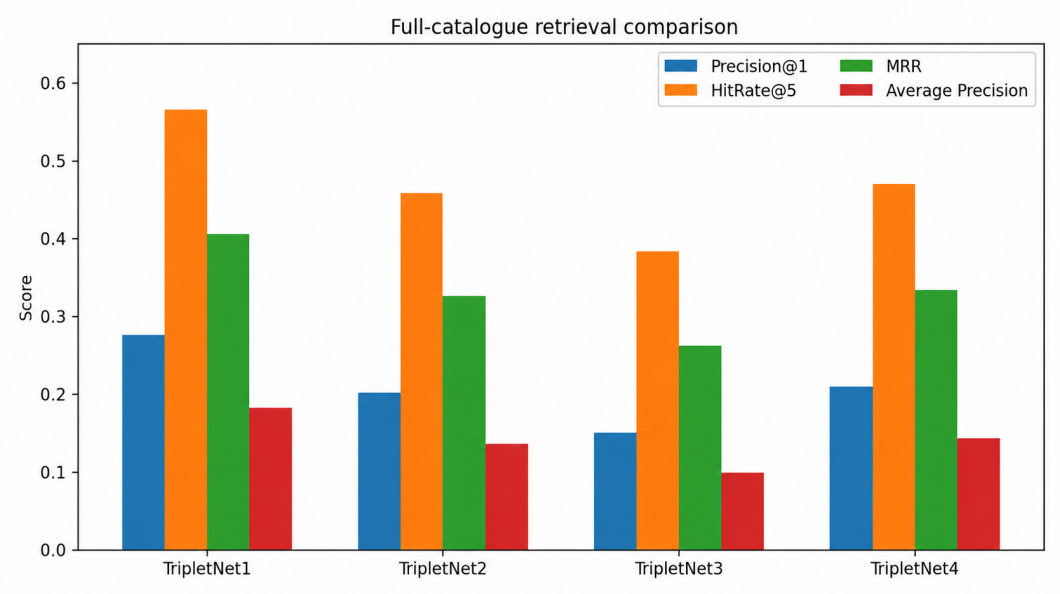


Figure 5.7: Full-catalogue retrieval metrics across the tested models. Generated from the results in Table 5.10.

MRR by approximately 24.3%, 54.4%, and 21.6%, respectively. Its Precision@1 improvements are 36.5%, 83.1%, and 31.8%. These gaps are substantially larger than the differences in triplet accuracy, showing that the metrics capture different properties.

Precision@1 of 0.2759 is measured against nearly four thousand candidates and an incomplete ground truth. Only AllMusic and Spotify relations count as positives, so a plausible but unlisted neighbour is scored as an error. The reported Precision@1 may therefore underestimate relevance under human judgement [2].

TripletNet1 reaches HitRate@5 of 0.5657, HitRate@10 of 0.6844, and HitRate@20 of 0.7828. These values show how frequently at least one known related artist occurs within progressively longer recommendation lists.

The gap between Precision@1 and HitRate@5 is consistent with several plausible neighbours competing for the first position. Artist similarity can reflect genre, era, visual identity, audience, and cultural context, so no single relation need dominate every query [1, 2].

An analytical random-ranking reference puts this scale in context. It is calculated from the reported quantities, not from an additional experiment. Excluding the query artist leaves $N = 3{,}891$ candidates, with an average of $R = 9.79$ known positives per query. The approximate expected precision of a random first position is

$$\mathbb{E}[\text{Precision@1}] \approx \frac{R}{N} = \frac{9.79}{3{,}891} \approx 0.0025.$$

For a particular query with an integer known-positive count R_q, the probability that

at least one positive occurs in the first K positions of a random ordering is

$$P(\text{HitRate@}K) = 1 - \frac{\binom{N-R_q}{K}}{\binom{N}{K}}.$$

Because the reported 9.79 is an average rather than a list of the query-specific values R_q, only a rough catalogue-level approximation can be formed:

$$P(\text{HitRate@}K) \approx 1 - \left(1 - \frac{R}{N}\right)^K.$$

This approximate chance reference gives HitRate@5 0.0125, HitRate@10 0.0249, and HitRate@20 0.0491. The observed Precision@1 and HitRate@K values are well above the corresponding approximate random-ranking scale, which indicates that the model learned non-random ordering associated with the relation labels. This reference is not a strict per-query random baseline, does not establish statistical significance, and applies only to the incomplete known-positive graph used for evaluation.

The cross-validation mean MRR of 0.3114 and the full-catalogue MRR of 0.4055 are not successive measurements of the same retrieval problem. The former is averaged over artist-disjoint validation folds, each with its own held-out query and candidate set. The latter uses a different query population and ranks against all other artists in the available catalogue. Since the candidate sets, query sets, and purposes differ, the higher value of 0.4055 cannot be interpreted as an improvement over 0.3114, nor does it show that full-catalogue retrieval is easier. Cross-validation MRR is used to compare architectures and margins and to select checkpoints. Full-catalogue MRR instead describes how the selected Fold 1 TripletNet1 checkpoint specified in Section 5.11.1 ranks known relations across the complete available catalogue.

Precision@K declines as K increases because longer lists contain more candidates, while Recall@K and HitRate@K rise because the model has more opportunities to retrieve a known relation. This expected trade-off appears for all four models. TripletNet1 remains first at every reported cutoff, so its advantage is not confined to the top position.

The ordering among the baselines changes between validation and full-catalogue retrieval. TripletNet2 is the strongest convolutional model in mean validation MRR, but TripletNet4 has the higher catalogue MRR and average precision. The validation candidate sets are much smaller and contain different artist subsets. This reversal is another reason to treat validation MRR as a selection tool rather than the final performance claim.

At $K = 1$, Precision@1 and HitRate@1 are identical because one returned item is either a known relation or not. At larger cutoffs, HitRate@K records whether the list contains a known relation, whereas Precision@K measures how much of the list is known to be relevant. A list can have a high HitRate@K and still include many unlabelled candidates.

Recall@20 remains below 0.30 for TripletNet1. Each query has about 9.79 recorded

positives on average, but the count varies by artist. Queries with many relations need a longer list to attain high Recall@K. Queries with one known relation reach full Recall@K as soon as that artist appears. Aggregate Recall@K therefore mixes different levels of reference-set difficulty.

Average precision evaluates the positions of the complete known-positive set, whereas MRR stops at the first. TripletNet1's average precision of 0.1826 is lower than its MRR of 0.4055, indicating that retrieving one known neighbour is easier than ranking the complete recorded set near the top. This matters for catalogue exploration: the embedding is more convincing as a candidate generator than as a complete reconstruction of the AllMusic and Spotify relation graph.

The catalogue evaluation supplies one descriptive result per trained configuration rather than a distribution over independent catalogues. Relative improvements are useful for comparing the tested models, but they are not confidence bounds for another service or data collection.

The aggregate retrieval values combine queries with different reference-set sizes. Some artists have one recorded relation, while others have many. MRR gives both queries equal weight and considers the first known positive. Recall@K is more demanding for the artist with many positives because a short list can recover only a fraction of them.

Relation count can also correlate with popularity and catalogue coverage. A well-documented artist offers more labelled opportunities for a hit, but also has a larger set to recover. A less documented artist may have one label that fails to capture several plausible neighbours. The direction of the resulting bias depends on the metric.

Video coverage creates another source of query variation. Artists represented by several distinct clips give the hierarchical encoder more evidence than artists represented by one clip. The current aggregate answers whether the configuration works across the collected set. It does not show whether its advantage is concentrated among artists with richer video coverage.

The full-catalogue results support the video branch as a candidate feature or complementary signal in a recommendation pipeline. Its neighbourhoods reflect visual presentation and public identity rather than complete musical similarity.

The video branch should not be interpreted as a replacement for an audio-based recommender because it does not model listener preferences or the full musical signal. The present experiment evaluates only the video-derived contribution.

5.12 Latent Space and Metadata Diagnostics

5.12.1 Country- and Genre-Level Similarities

The metadata analysis compares intra-country and inter-country similarity in the TripletNet1 space. Table 5.11 reports both means and their difference for countries with sufficient representation.

Table 5.11: Country-level similarity diagnostics for TripletNet1. The model is not trained with country labels; the table is used only for interpretation.

Country	Artists	Mean intra	Std. intra	Mean inter	Intra minus inter
Spain	93	0.2909	0.2285	0.0628	0.2281
France	247	0.2862	0.2057	0.0839	0.2022
Italy	99	0.2306	0.2165	0.0837	0.1468
Germany	437	0.1774	0.2255	0.0394	0.1380
UK	770	0.1639	0.2138	0.0605	0.1034
USA	2,212	0.0401	0.2289	0.0332	0.0070

Spain and France have the largest intra-minus-inter differences. For the United States, which dominates the dataset and spans many genres and scenes, the two means are almost equal. The broad country label therefore provides little group separation for American artists; it does not indicate poor representation quality.

Broad genres show uneven group separation in Table 5.12. Classical, country, metal, hip hop, and jazz have relatively large intra-minus-inter values, whereas pop and rock have smaller differences and encompass more varied subgenres and visual identities.

Table 5.12: Broad-genre similarity diagnostics for TripletNet1.

Broad genre	Artists	Mean intra	Std. intra	Mean inter	Intra minus inter
classic	36	0.3713	0.1846	0.0902	0.2811
country	131	0.2609	0.3343	-0.0134	0.2742
metal	440	0.2193	0.2852	-0.0225	0.2419
hip hop	323	0.2631	0.2195	0.0258	0.2373
jazz	159	0.3039	0.2093	0.0699	0.2341
others	1,444	0.1293	0.1985	0.0527	0.0766
pop	788	0.1010	0.2224	0.0405	0.0605
rock	571	0.0803	0.2521	0.0305	0.0498

The results are consistent with genre-related visual structure in the video representation. Metal and hip hop, for example, may share performance aesthetics, styling, and production conventions. Because genre labels are absent from training, any such coherence arises indirectly from pairwise artist supervision.

Fine-grained genres provide a local diagnostic. Among groups of at least five artists, Table 5.13 lists the largest intra-minus-inter gaps, including Neue Deutsche Harte, progressive house, jam band, chiptune, and brostep.

Several groups contain only five or six artists, so their large gaps are uncertain. They nevertheless illustrate local neighbourhoods. Such local structure is more directly relevant to recommendation than complete global clustering [2, 12].

Table 5.13: Examples of fine-grained genre groups with strong intra-group similarity in the TripletNet1 latent space.

Fine genre	Artists	Mean intra	Mean inter	Intra minus inter
neue deutsche harte	5	0.7862	-0.0818	0.8680
progressive house	5	0.7621	0.0222	0.7398
jam band	8	0.7075	-0.0005	0.7080
talent show	5	0.6493	-0.0213	0.6706
chiptune	5	0.8055	0.1426	0.6630
nouvelle chanson francaise	6	0.7308	0.0840	0.6468
acid house	5	0.6442	0.0163	0.6279
emo	5	0.7265	0.1047	0.6218
brostep	6	0.7238	0.1212	0.6026
deep euro house	5	0.7118	0.1102	0.6016

5.12.2 t-SNE Visualizations

TripletNet1 embeddings are projected to two dimensions with t-SNE [18]. Figures 5.8 and 5.9 support qualitative inspection, but dimensionality reduction can distort high-dimensional distances. Table 5.14 therefore supplements them with silhouette scores [40].

Table 5.14: Silhouette scores for TripletNet1 metadata labels. Scores are computed with cosine distance on a sampled set of 2,500 artists.

Label type	Artists	Labels	Silhouette score
Country	2,500	6	-0.0233
Broad genre	2,500	8	-0.0201
Fine genre	2,500	60	-0.2662

The negative silhouette scores in Table 5.14 show that country and genre do not form clean global clusters. The two-dimensional figures are used to inspect possible local patterns; the evidential limits of comparing those patterns with the original embedding space are discussed in Section 5.12.3.

5.12.3 Limits of the Metadata Diagnostics

Country groups are highly imbalanced. The United States accounts for 2,212 artists, while Spain and Italy have fewer than one hundred each. A broad label for the largest group includes many regional scenes, genres, and periods. Its small intra-minus-inter difference may reflect heterogeneity within the label rather than an absence of cultural information.

Country can also correlate with data collection. Language, platform use, video availability, and editorial coverage differ across markets. The analysis cannot separate a visual country pattern from these associated factors. Since the labels are not used in training, the table is descriptive and does not support a claim that the model infers nationality.

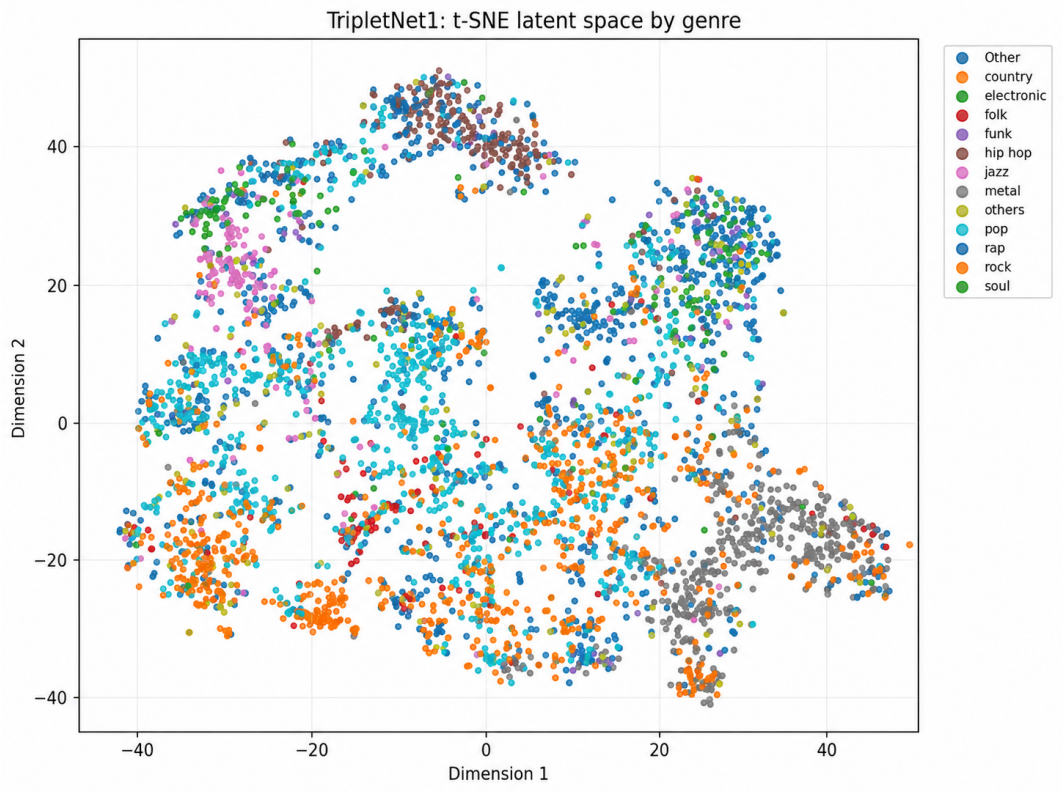


Figure 5.8: t-SNE projection of the TripletNet1 artist space coloured by broad genre. Generated from the learned embeddings and post-hoc metadata.

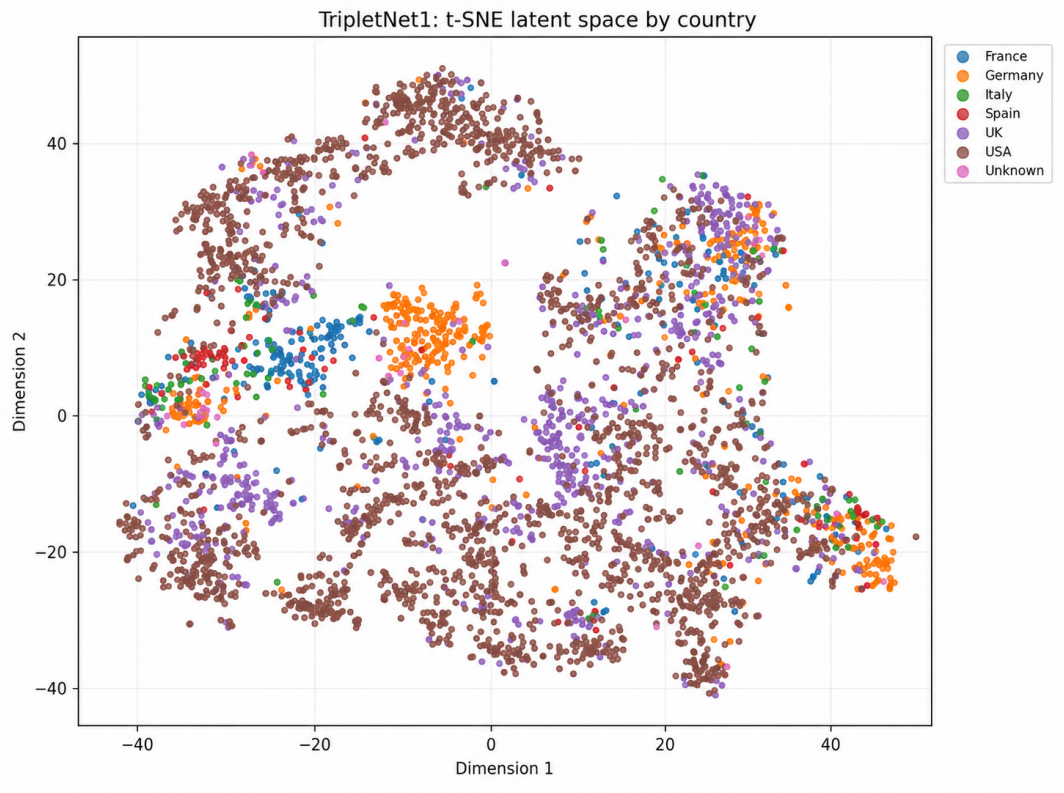


Figure 5.9: t-SNE projection of the TripletNet1 artist space coloured by country. Generated from the learned embeddings and post-hoc metadata.

Local genre-consistent neighbourhoods and weak or negative global silhouette scores are not contradictory. A subset of artists can occupy a locally coherent region even when artists carrying the same broad label are dispersed elsewhere and neighbouring labels overlap. Silhouette analysis evaluates whether each labelled group is compact and separated from its nearest alternative across the sampled space. It therefore penalises interleaved boundaries and dispersed subgroups that do not prevent some local neighbourhoods from being interpretable.

Genre is also not naturally a single, mutually exclusive artist property. One artist may be associated with several genres, combine styles across releases, or receive different metadata labels across services. Broad categories such as pop and rock cover many subgenres and frequently overlap with other categories. Reducing this structure to one label for a silhouette calculation imposes a stricter partition than the underlying musical categories warrant. Classical and metal may have more consistent visual conventions in parts of the catalogue, but the differences in Table 5.12 do not identify the cause of that consistency.

The visual input should not be treated as genre supervision. Stage design, clothing, lighting, performance setting, and an artist's public image may recur within some genres and produce local regularities. The same cues can cross genre boundaries, change across videos, or reflect a particular production context rather than the music. Neither genre nor country metadata enters training. The group comparisons are consequently post-hoc interpretability diagnostics, not evidence that the model performs genre or country classification.

The fine-grained table is intentionally selective: it lists groups with the largest observed gaps among those meeting the minimum size. Selection by the observed statistic makes the displayed values larger than a typical genre gap. Groups of five artists are especially sensitive to one unusually similar pair. These rows are examples for inspection, not estimates expected to generalise to every artist carrying the label.

t-SNE retains only part of the local structure when mapping the 256-dimensional representation into two dimensions [18]. Apparent islands in Figures 5.8 and 5.9 should therefore not be measured from the plotted coordinates or treated as proof of separation. Visual separation in a projection cannot replace quantitative evidence from the original space. Here, the negative silhouette scores computed with the learned embeddings are the stronger evidence against globally separated metadata classes.

The lack of global metadata clusters is not necessarily a failure under the retrieval objective. Training asks the embedding to rank relation positives, and those relations can cross genre and country boundaries. The most cautious reading is that the learned representation contains locally interpretable stylistic and cultural structure, but it does not organise the complete artist catalogue into globally separated genre or country classes.

5.13 Error Analysis and Computational Considerations

5.13.1 Error Analysis

At the best F1 threshold, TripletNet1 produces 194 FPs among 1,132 negative pairs. Here, a negative label denotes only the absence of a recorded relation, so these cases cannot all be assumed to be unreasonable recommendations [2].

The limitation originates in the incomplete AllMusic [7, 11] and Spotify [8] relations, which may emphasise different dimensions of similarity. Video-based retrieval can identify visual or cultural relations absent from the table; the metric penalises them even when they may aid discovery.

False negatives expose a different limitation. TripletNet1 has 137 FNs at the best F1 threshold. Their videos may not express the dimension recorded by the ground truth: artists related by genre or historical influence can differ in production style, era, and visual identity. Sparse, low-quality, or unrepresentative videos create the same problem.

Sampling thirty frames from the central portion also omits cues outside that range and makes the available evidence dependent on the selected positions [37, 38].

The relation between triplet accuracy and retrieval quality provides a further diagnostic. TripletNet3 combines the highest out-of-fold ranking accuracy with the weakest full-catalogue retrieval. A stored triplet contains one negative, whereas catalogue evaluation ranks every candidate [12]. The leading model under validation MRR also leads final retrieval, although this agreement does not constitute an ablation of the checkpoint criterion.

The error categories above are interpretive hypotheses based on the observed numerical patterns, retrieval results, and data limitations rather than independently verified causes. Confirmation would require human relevance judgements and controlled analyses beyond the scope of the present experiments.

5.13.2 Computational Considerations

TripletNet1 processes up to $10 \times 30 = 300$ frame embeddings per artist, compared with the baselines' averaged sequence of 30. Its cost is therefore higher, especially at the frame level, although a moderate hidden dimension, shallow Transformers, early stopping, and mini-batches keep training feasible [13, 31].

The simpler convolutional models remain relevant under strict deployment constraints, but their retrieval scores are lower. Model choice therefore depends on the trade-off between recommendation quality and computational efficiency [14, 41].

Offline CLIP extraction avoids end-to-end training of the visual encoder [6]. The saved frame embeddings make repeated margin and architecture experiments possible without reprocessing the videos.

Chapter 6

Conclusion

6.1 Summary of the Thesis and Main Findings

This thesis examined whether sampled visual information from artist-related videos can support artist-similarity retrieval. Unlike the static Instagram image-caption architecture proposed in [5], it treats video as a separate branch and evaluates that branch on its own. This design measures what the video representation contributes without conflating it with audio or other social-media inputs.

AllMusic and Spotify relations provide the supervision, while fixed CLIP frame descriptors provide the visual input [6, 5]. TripletNet1 preserves frames within videos and videos within an artist profile; three Conv1D baselines instead receive video-averaged inputs. All four complete configurations use the common protocol summarised in Table 4.2. The assessment combines artist-disjoint validation, out-of-fold pair analysis, full-catalogue retrieval, and post-hoc metadata diagnostics.

Based on the numerical assessment, it emerges that video frames contain a measurable artist-relation signal. TripletNet1 reaches the highest mean validation MRR, 0.3114, compared with 0.2787 for the strongest convolutional validation baseline. It leads four of the five artist-disjoint folds, and its paired fold differences against TripletNet2 and TripletNet3 remain positive on every partition.

In full-catalogue retrieval, TripletNet1 obtains MRR 0.4055, Precision@1 0.2759, and HitRate@5 0.5657. It leads at every reported retrieval cutoff. Section 6.2.3 explains why this configuration-level result cannot be attributed to attention, hierarchy, or input aggregation in isolation.

The margin results favour a low-margin region. Every model performs best at the smallest tested value, 0.1, and performance declines as the requested separation increases. For TripletNet1, 0.1 and 0.3 are almost tied, while the clearer decline begins at 0.5. Because values below 0.1 were not tested, the experiment does not establish a global optimum.

The metadata diagnostics support only a local interpretation. Several country and genre groups have higher within-group than between-group similarity, but every global silhouette score is negative. Country and genre labels are not training supervision, and the embedding does not reproduce a country or genre classifier.

Triplet accuracy and catalogue retrieval measure different behaviour. TripletNet3 has the highest out-of-fold ranking accuracy but the lowest catalogue MRR. Correctly ordering one stored positive-negative comparison is therefore insufficient evidence for catalogue recommendation. Validation MRR selects the model that also leads the catalogue assessment, which supports its use as the checkpoint criterion in this experiment.

6.2 Limitations and Future Work

The following limitations constrain the scope and interpretation of the conclusions. They do not undermine the consistency of the comparison under the shared protocol summarised in Table 4.2.

6.2.1 Relation labels and evaluation

AllMusic and Spotify provide two external judgements of artist similarity rather than a complete definition of the construct [7, 8]. The editorial and platform relations do not exhaust musical, visual, cultural, or listener-dependent similarity, and their union does not turn a subjective relation into an objective binary label. This thesis therefore measures recovery of the available reference relations, not agreement with a unique true ordering of artists.

This construct limitation affects both training and evaluation. A sampled negative is only a pair absent from the relation graph, not a verified dissimilar pair. Full-catalogue metrics likewise count plausible but unlisted neighbours as non-relevant. The reported Precision@K and error counts are consequently conditional on the current reference tables rather than direct measurements of human relevance.

There is a further evaluation constraint at the pair threshold. The F1 threshold is selected and reported on the same out-of-fold pair set, so its value may be mildly optimistic when transferred to a new catalogue with different score distributions or positive prevalence. Full-catalogue relevance remains incomplete for the same reason as the training graph. No independent calibration set, human relevance study, or source-specific test is claimed in the present work; each is a future validation step.

Human assessment should sample high-scoring unlabelled pairs and low-scoring recorded pairs from all tested configurations, conceal model scores and relation sources, and rate musical, visual, and general artist similarity separately. This would test whether apparent errors reflect missing labels or a mismatch between visual and musical similarity.

AllMusic and Spotify relations should also be evaluated separately, together with their intersection and source-specific subsets. Agreement between sources can define a stricter positive set, while the subset comparisons can show which relation type the video representation captures more easily.

An independent calibration set or additional annotations would test threshold transfer without reusing the evaluation pairs. Query-level resampling could estimate

uncertainty within the catalogue, while evaluation on a second collection would be needed for external generalisation.

6.2.2 Video coverage and representation

The model sees sampled visual frames, not the complete video. Uniform sampling can miss short scenes, and framewise CLIP descriptors do not encode motion directly. The soundtrack is absent. Artists related through timbre, composition, or historical influence may remain distant when their video presentation differs.

Video availability is uneven. Padding and masks prevent absent clips from becoming false evidence, but an artist represented by one video offers less information than an artist represented by ten. The collected set may also favour artists with stronger platform presence. Performance for poorly documented or emerging artists remains uncertain.

The conclusions therefore apply to the collected artist catalogue and its observed distribution of videos. They are not evidence of unchanged performance for artists with very few videos, material drawn from another social platform, other geographic markets or career stages, or substantially different styles of visual content. Differences in video count, quality, and representativeness may affect the resulting artist vector. These factors limit external validity beyond the collected catalogue.

Retrieval and error rates should be stratified by the number of valid videos, usable-frame coverage, and variation among an artist's video tokens. This would show whether false negatives and the hierarchical model's advantage are concentrated among artists with sparse or rich visual evidence.

Shot-aware sampling is a direct extension. Frames could be selected at scene changes or by visual diversity rather than at fixed positions. Short video encoders could add motion, while audio extracted from the same clips could measure whether the soundtrack contributes information beyond the images [37, 38]. These changes should be tested separately so that gains can be assigned to sampling, motion, or audio.

CLIP is trained on general image-text pairs rather than specifically on music videos [6]. Fine-tuning on music-related visual data may improve domain sensitivity, but it also increases the risk of fitting the available artists. A useful comparison would keep the artist encoder fixed and vary only the visual backbone.

6.2.3 Architecture and experimental scope

TripletNet1 and the Conv1D baselines differ simultaneously in input aggregation, encoder family, capacity, and inductive bias. The experiment therefore supports comparison of complete configurations. It does not establish that attention or hierarchy alone caused the improvement; parameter-matched component ablations remain future work.

Future parameter-matched ablations should vary the encoder family, mean versus stacked aggregation, and artist-level attention one factor at a time. Applying the

video-coverage strata described in Section 6.2.2 would then distinguish architectural effects from gains associated with richer input evidence. Parameter counts, training time, and retrieval latency would complete the comparison.

Only one shared optimisation configuration and one random seed are reported. Architecture-specific tuning or repeated seeds could change the model ordering.

6.2.4 From a video branch to multimodal retrieval

Video should complement rather than replace musical evidence. A future system can combine this branch with the audio, static-image, and caption representations of the broader project while retaining separate branch outputs for controlled comparison [5]. Fusion weights should account for unequal modality coverage.

6.3 Final Remarks

Artist-related videos contain enough structure to support relation-based retrieval. Under the reported protocol, the hierarchical configuration ranks known related artists more effectively than the video-averaged convolutional alternatives. The advantage appears most clearly when every catalogue artist becomes a candidate.

The learned space is one view of artist similarity. It reflects sampled visual content and the available relation sources, not a complete definition of musical relatedness. The reported results define where this video representation is useful and where additional evidence is required.

Bibliography

[1] Markus Schedl, Emilia Gómez, and Julián Urbano. "Music Information Retrieval: Recent Developments and Applications". In: *Foundations and Trends in Information Retrieval* 8.2–3 (2014), pp. 127–261. DOI: 10.1561/1500000042 (cit. on pp. 1, 4, 6–8, 42, 54).

[2] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. "Evaluating Collaborative Filtering Recommender Systems". In: *ACM Transactions on Information Systems* 22.1 (2004), pp. 5–53. DOI: 10.1145/963770.963772 (cit. on pp. 1, 2, 6, 24–26, 45–47, 51, 53, 54, 57, 61).

[3] Beth Logan and Ariel Salomon. "A Music Similarity Function Based on Signal Analysis". In: *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.* Tokyo, Japan, 2001, pp. 745–748. DOI: 10.1109/ICME.2001.1237829 (cit. on pp. 1, 4, 5, 11).

[4] Jean-Julien Aucouturier and Francois Pachet. "Music Similarity Measures: What's the Use?" In: *Proceedings of the 3rd International Conference on Music Information Retrieval*. 2002, pp. 157–163. URL: https://ismir2002.ismir.net/proceedings/02-FP05-2.pdf (cit. on pp. 1, 4, 11).

[5] Massimiliano Zanoni, Giacomo Sansoni, Davide Lista, Cristina Rottondi, and Andrea Bianco. "Inferring Artist Similarity From Social Media Content: The Instagram Case". In: *Journal of the Audio Engineering Society* 74.6 (2026), pp. 404–416 (cit. on pp. 1, 4–9, 28, 33, 34, 62, 65).

[6] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763. URL: https://proceedings.mlr.press/v139/radford21a.html (cit. on pp. 1, 2, 7, 10, 11, 13, 17, 19, 28–31, 61, 62, 64).

[7] AllMusic. *AllMusic*. URL: https://www.allmusic.com/ (visited on 06/17/2026) (cit. on pp. 1, 2, 6, 29, 33, 42, 47, 52, 61, 63).

[8] Spotify. *Get Artist's Related Artists*. URL: https://developer.spotify.com/documentation/web-api/reference/get-an-artists-related-artists (visited on 06/17/2026) (cit. on pp. 1, 2, 6, 29, 33, 42, 47, 52, 61, 63).

[9] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. "Signature Verification Using a "Siamese" Time Delay Neural Network". In: *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann, 1994, pp. 737–744. URL: https://proceedings.neurips.cc/paper/1993/hash/288cc0ff022877bd3df94bc9360b9c5d-Abstract.html (cit. on pp. 2, 22, 29, 34).

[10] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA, 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682. arXiv: 1503.03832 (cit. on pp. 2, 13, 22, 23, 25, 29, 34, 47).

[11] Juyong Park, Oscar Celma, Markus Koppenberger, Pedro Cano, and Javier M. Buldú. "The Social Network of Contemporary Popular Musicians". In: *International Journal of Bifurcation and Chaos* 17.7 (2007), pp. 2281–2288. DOI: 10.1142/S0218127407018385. arXiv: physics/0609229 (cit. on pp. 2, 6, 29, 33, 47, 61).

[12] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN: 978-0-521-86571-5. URL: https://nlp.stanford.edu/IR-book/ (cit. on pp. 2, 10, 23–26, 29, 37, 45, 52, 53, 57, 61).

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems 30*. 2017, pp. 5998–6008. arXiv: 1706.03762. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (cit. on pp. 2, 10, 14, 16–18, 20, 21, 36, 61).

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791 (cit. on pp. 2, 6, 14–16, 37, 61).

[15] David R. Roberts et al. "Cross-Validation Strategies for Data with Temporal, Spatial, Hierarchical, or Phylogenetic Structure". In: *Ecography* 40.8 (2017), pp. 913–929. DOI: 10.1111/ecog.02881 (cit. on pp. 2, 26, 30, 35, 49).

[16] Tom Fawcett. "An Introduction to ROC Analysis". In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010 (cit. on pp. 2, 25, 45, 49, 51).

[17] I. T. Jolliffe. *Principal Component Analysis*. 2nd ed. New York: Springer, 2002. DOI: 10.1007/b98835 (cit. on pp. 2, 12, 13, 26, 27).

[18] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data Using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: https://www.jmlr.org/papers/v9/vandermaaten08a.html (cit. on pp. 2, 12, 13, 26, 27, 45, 58, 60).

[19] Brian Kulis. "Metric Learning: A Survey". In: *Foundations and Trends in Machine Learning* 5.4 (2013), pp. 287–364. DOI: 10.1561/2200000019 (cit. on pp. 2, 10, 11, 13, 22–24, 29, 47).

[20] Daniel P. W. Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. "The Quest for Ground Truth in Musical Artist Similarity". In: *Proceedings of the 3rd International Conference on Music Information Retrieval*. 2002 (cit. on pp. 4, 8, 9).

[21] Sergio Oramas, Oriol Nieto, Mohamed Sordo, and Xavier Serra. "A Deep Multimodal Approach for Cold-Start Music Recommendation". In: *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*. ACM, 2017, pp. 32–37. DOI: 10.1145/3125486.3125492. arXiv: 1706.09739 (cit. on pp. 4, 6, 7).

[22] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50 (cit. on pp. 5, 11, 12, 29).

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539 (cit. on pp. 5, 10–12).

[24] Shawn Hershey et al. "CNN Architectures for Large-Scale Audio Classification". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2017, pp. 131–135. DOI: 10.1109/ICASSP.2017.7952132 (cit. on pp. 5, 29).

[25] Zain Nasrullah and Yue Zhao. "Music Artist Classification with Convolutional Recurrent Neural Networks". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. Budapest, Hungary, 2019, pp. 1–8. DOI: 10.1109/IJCNN.2019.8851988. arXiv: 1901.04555 (cit. on p. 5).

[26] Filip Korzeniowski, Sergio Oramas, and Fabien Gouyon. "Artist Similarity with Graph Neural Networks". In: *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. 2021. DOI: 10.48550/arXiv.2107.14541. arXiv: 2107.14541. URL: https://arxiv.org/abs/2107.14541 (cit. on pp. 5, 8).

[27] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. arXiv: 2010.11929. URL: https://openreview.net/forum?id=YicbFdNTTy (cit. on pp. 6, 17, 31).

[28] Brian McFee and Gert R. G. Lanckriet. "Heterogeneous Embedding for Subjective Artist Similarity". In: *Proceedings of the 10th International Society for Music Information Retrieval Conference*. 2009, pp. 513–518 (cit. on p. 8).

[29] Antonio C. M. da Silva, Diego F. Silva, and Ricardo M. Marcacini. "Artist Similarity Based on Heterogeneous Graph Neural Networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 32 (2024), pp. 3717–3729. DOI: 10.1109/TASLP.2024.3437170 (cit. on p. 8).

[30] Tom M. Mitchell. *Machine Learning*. New York: McGraw-Hill, 1997. ISBN: 0-07-042807-7 (cit. on p. 10).

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: https://www.deeplearningbook.org/ (cit. on pp. 10–15, 21, 61).

[32] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (July 1959), pp. 210–229. DOI: 10.1147/rd.33.0210 (cit. on p. 10).

[33] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 448–456. URL: https://proceedings.mlr.press/v37/ioffe15.html (cit. on pp. 11, 12, 37).

[34] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. DOI: 10.48550/arXiv.1607.06450. arXiv: 1607.06450 (cit. on pp. 11, 12).

[35] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. 2016. DOI: 10.48550/arXiv.1606.08415. arXiv: 1606.08415 (cit. on pp. 14, 15).

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: https://www.jmlr.org/papers/v15/srivastava14a.html (cit. on pp. 15, 21, 36).

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning Spatiotemporal Features with 3D Convolutional Networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4489–4497. DOI: 10.1109/ICCV.2015.510. URL: https://openaccess.thecvf.com/content_iccv_2015/html/Tran_Learning_Spatiotemporal_Features_ICCV_2015_paper.html (cit. on pp. 17, 61, 64).

[38] Joao Carreira and Andrew Zisserman. "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4724–4733. DOI: 10.1109/CVPR.2017.502. URL: https://openaccess.thecvf.com/content_cvpr_2017/html/Carreira_Quo_Vadis_Action_CVPR_2017_paper.html (cit. on pp. 19, 61, 64).

[39] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 8024–8035. arXiv: 1912.01703. URL: https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html (cit. on p. 40).

[40] Peter J. Rousseeuw. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7 (cit. on pp. 45, 58).

[41] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-Scale Similarity Search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2021), pp. 535–547. DOI: 10.1109/TBDATA.2019.2921572. arXiv: 1702.08734 (cit. on p. 61).