JOHANNES KEPLER
UNIVERSITY LINZ

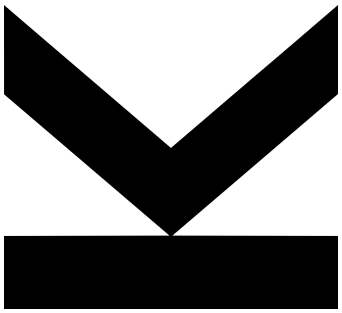
Author
Youniss Kandah
k12204692

Submission
Institute of Computational Perception

Thesis Supervisor
Dr. **Shah Nawaz**

October 2025

Vision-Mediated Learning for Audio–Text Retrieval



Bachelor Thesis

to obtain the academic degree of

Bachelor of Science

in the Bachelor's Program

Artificial Intelligence

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Austria
www.jku.at
DVR 0093696

Abstract

Current state-of-the-art language-based audio retrieval systems rely on fine-tuning audio and Text embedding models, which are compared using contrastive loss. In our approach, we first generate embeddings for audio and text, then create images from these embeddings, and finally train a simple Vision Transformer to make the retrieval decisions. We evaluate on Clotho and report retrieval metrics Recall@{1, 5, 10} and mean Average Precision (mAP@10) for both directions (audio $\rightarrow$ text, text $\rightarrow$ audio). Compared to a baseline, the visual-proxy variant underperforms on all metrics. To support reproducibility, we provide a structured codebase with very simple, clear instructions. Overall, our findings indicate that the dual-encoder baseline remains stronger under modest data and compute.

Contents

List of Figures

List of Tables

1 Introduction

Ultimately, the goal of Language-Based Audio Retrieval, is to search for Audios through a text query. Meaning, use cases could include specialized Search engines. The main challenge in this is the cross modal learning of similarities between Audio and Text content. To combat this problem, previous works have used Audio and Text embedding models to convert the raw text and audio data into a numerical representation, which would then allow for cosine similarity retrieval.

The learning of such a model fine-tunes both the audio and text embedding models and then projects it. A Contrastive loss function is used to learn these similarities.

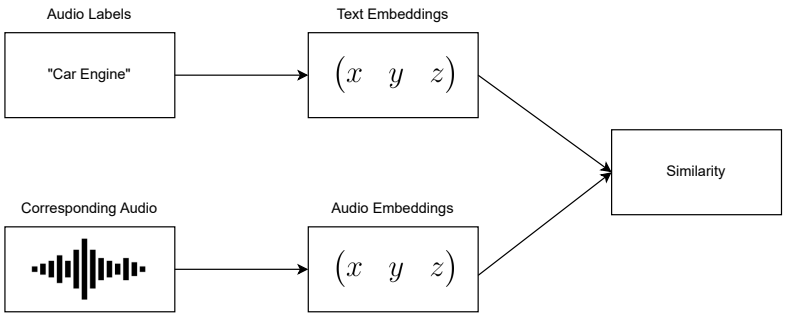


Figure 1.1: Language-Based Audio Retrieval Baseline system

Our solution to this issue was to convert both the Audio and Text embeddings into images, and using a vision transformer to create new, rich numeric representations of both the Audio and Text image representations, which is then ultimately fed into a contrastive loss function.

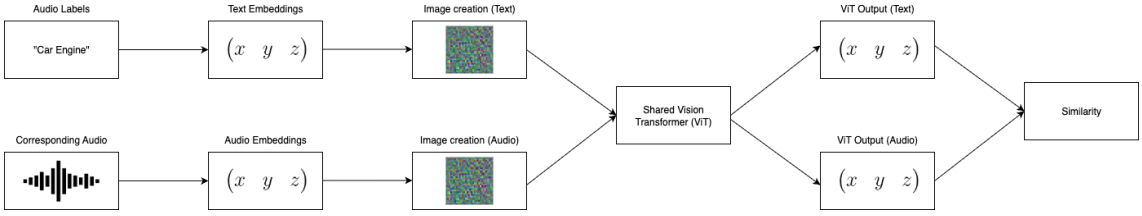


Figure 1.2: Vision Proxy system

The resulting models after training of both the baseline and the Vision Proxy solution are then evaluated on the Clotho v2.1 **?** Dataset on R@1, R@5, R@10 and mAP@10.

2 Related Work

2.1 DCASE language-based audio retrieval

Language-based audio retrieval (LBaR) has been organized by DCASE since 2022 and currently uses the Clotho dataset as the primary benchmark. The 2025 setup retains Clotho v2.1 and introduces multiple relevant audios per query, reporting R@$\{1, 5, 10\}$ and mAP@10. Baselines and most top systems follow a dual-encoder alignment with contrastive objectives, often with task-specific augmentation and distillation (**?**).

2.2 PaSST

Patchout faSt Spectrogram Transformer (PaSST) is an audio transformer that reduces training cost via patch dropping while preserving performance on large audio-tagging benchmarks. PaSST has become a common audio backbone in Language-based audio retrieval systems because it produces robust clip-level embeddings and scales well. (**?**).

2.3 RoBERTa

RoBERTa is a robustly optimized variant of BERT that uses more data, dynamic masking, and longer training to improve generalization across many NLP tasks. **?**

2.4 Vision Transformers

Vision Transformers (ViTs) treat images as sequences of patches and, when sufficiently pretrained, match or surpass strong CNNs. ViTs are widely used as image backbones and have inspired proxy-image ideas for non-image inputs. **?**

2.5 Evaluation protocols (R@k, mAP@k)

We follow standard information-retrieval definitions: Recall@k is the fraction of relevant items found in the top-k results; mean Average Precision at k (mAP@k) averages the precision at ranks where relevant items appear, clipped at k, across queries. DCASE adopts these metrics for LBaR and publishes official baselines and splits (**?**).

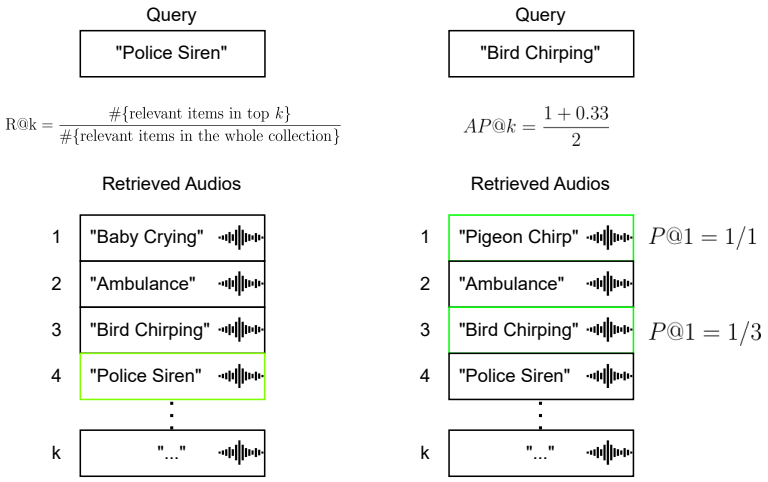


Figure 2.1: Recall@K and Average Precision@k

3 Method

3.1 Task

The Task is connected to the DCASE Task 6 of 2025 on Language-Based Audio Retrieval. It focuses on retrieving audio clips based on their corresponding captions. These captions will serve as queries. For each of them, the objective is to retrieve a set of audio files from the Clotho **?** dataset and rank them according to their relevance to the query.

The trained model should precompute the Audio dataset and store each embedding, and then the text query is computed on the fly. The similarity per Clip is then rated and they are then Ranked and evaluated for the according metric.

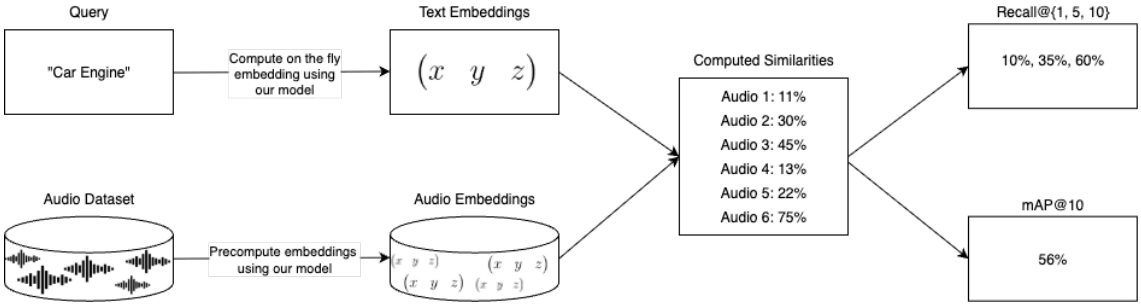


Figure 3.1: Language Based Audio Retrieval Pipeline

3.2 Dataset

We use Clotho v2.1 **?**, which contains 15–30 s audio clips sourced from Freesound, each paired with five crowd-sourced captions (typically 8–20 words). We refer readers to the original Clotho paper for collection and curation details. **?**

For the thesis experiments we load only the Clotho v2.1 **?** development split, create a 10% internal validation subset for model selection, and (for final thesis numbers) evaluate on the entire development split. These results are intended for internal comparison between our baseline and visual-proxy variants and are not directly comparable to DCASE leaderboard numbers that use the official development-testing split.

However, the same architecture was used during the DCASE challenge, which used the official split. See section 4.3

3.3 Rendering Vector Embeddings as Pseudo-Images

Our rendering function takes a vector of numbers (an embedding) and turns it into a small three-channel image so that ViT can process it.

Pipeline:

1. We plot the numbers from the embedding into a nearly square grid.
2. We either split the embedding into three equal parts (one per channel) or create a single-channel image and duplicate it to three channels. Three channels make the output compatible with common vision backbones.
3. A tiny convolutional stem cleans up blocky artifacts from the reshape and doubles the image size so there is more spatial detail.
4. We resize the image to a fixed height and width. We choose this size so it divides evenly into the ViT patch size.
5. Finally, we scale each image to the 0–1 range.

This process imposes an arbitrary 2D layout on a 1D vector, so nearby pixels are not guaranteed to be meaningfully related in the original embedding.

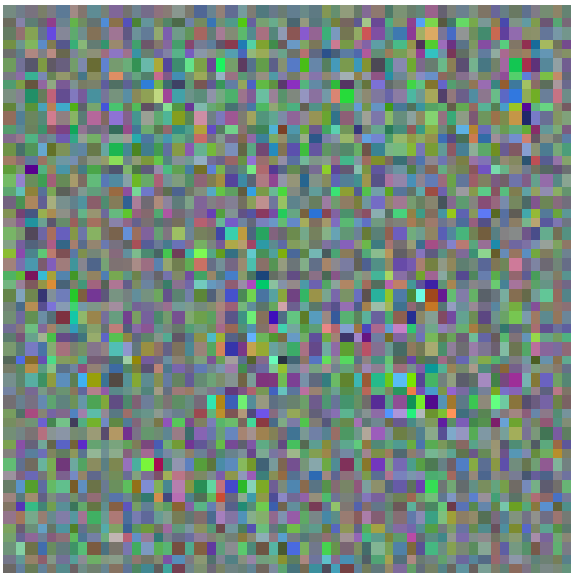


Figure 3.2: Example of an outputted image

3.4 Embedding models

The audio and text encoders are frozen and only the projection layers and temperature parameter are trained. This follows a fixed-feature retrieval setting for clarity and reproducibility.

We use PaSST **?** as the audio backbone, operating on log-mel spectrograms. In our configuration, a small PaSST variant is loaded in embedding-only mode, and we extract a single scene-level vector per clip via the model's global pooling head. The resulting audio embedding has dimensionality 768 by default. The backbone weights remain fixed during training.

For text embeddings, we use a RoBERTa **?**. Captions are tokenized and passed through the pretrained model without the classification pooling layer. We derive a sentence-level vector by attention-mask-aware mean pooling over the last hidden states. The default representation has dimensionality 768. As with audio, the text encoder is frozen.

Note: Other Models were tried and tested without significant differences. See the Technical report for more details. **?**

3.5 Contrastive objective and scoring.

Let z_a and z_t denote the projected, L2-normalized audio and text vectors in a mini-batch. Training uses a symmetric contrastive objective with a learnable temperature: matched audio–text pairs are pulled together and mismatches are pushed apart using in-batch negatives. At evaluation, cosine similarity scores between a query and all candidates are used to rank items; we report Recall@$\{1, 5, 10\}$ and mAP@10.

3.6 Vision Transformer (ViT)

The Vision Transformer (ViT) **?** applies the Transformer encoder architecture to images by treating fixed-size image patches as a sequence of tokens. Each patch is linearly embedded, augmented with positional information, and processed by stacked self-attention blocks. The result is an image-level embedding obtained by pooling token representations. The ViT output serves as the visual-proxy embedding.

4 Experiments

Code to reproduce all experiments is available at: `https://github.com/younissk/embed2image-contrastive-retrieval` For reproduction, See Makefile and README.md.

4.1 Settings

We train for 20 epochs with mini-batches of 8 and gradient accumulation of 4, yielding an effective batch size of 32 per device. Audio clips are capped at 10 seconds during preprocessing. We use bfloat16 mixed precision for throughput and memory efficiency and apply global gradient norm clipping at 1.0. The learning-rate schedule includes a warm-up phase of 1 epoch up to a maximum LR of 3×10^{-6}; for the visual-proxy runs we additionally specify a minimum LR of 1×10^{-7} for the decay phase. Encoders remain frozen, and only the projection heads and the contrastive loss temperature are trained.

For the ViT-based projection we render each embedding to a 224×224 three-channel proxy image and use a ViT-Small (patch16, 224) backbone with CLS pooling and 0.05 dropout in the vision head. Baseline runs use the MLP projection head with the same optimization schedule as above.

Table 4.1: Training configuration (defaults). Vision-specific settings apply only to the visual-proxy head.

Common settings	**Value**
Epochs	20
Mini-batch / Accumulation	8 / 4 (effective 32 per device)
Max audio duration	10 s
Precision	bfloat16 (mixed)
LR schedule	warm-up 1.0 epoch; max LR $= 3\times10^{-6}$
Gradient clipping	global norm = 1.0
Trainable parameters	projection head(s) + temperature (encoders frozen)
Vision-proxy head (ViT)	**Value**
Proxy image size	224×224
Backbone	ViT-S/16 (patch size 16, input 224)
Feature pooling	`cls`
Dropout (vision head)	0.05
LR schedule (extra)	min LR $= 1\times10^{-7}$

Both the Baseline and the Vision-proxy were trained only on Clotho v2.1. on a Nvidia A100.

4.2 Results

The visual proxy underperforms on all recall measures in both directions (Text to Audio, Audio to Text).

Table 4.2: Validation retrieval on dev(all). Best in bold. Absolute values.

	Audio → Text				Text → Audio			
Model	R@1	R@5	R@10	mAP@10	R@1	R@5	R@10	mAP@10
Baseline (MLP)	**0.135**	**0.363**	**0.518**	**0.412**	**0.119**	**0.119**	**0.195**	**0.749**
Visual proxy (ViT)	0.054	0.188	0.304	0.349	0.046	0.046	0.083	0.711

The proxy retrieves a relevant item less often (large drops in R@K). In Text → Audio R@1=R@5 was the same for both models, which is unusual.

4.3 External DCASE evaluation

In addition to the within-thesis experiments, we evaluated the same architecture under the official DCASE protocol in separate challenge runs **?**. Because the split and scoring protocol differ, these numbers are not directly comparable to our results; they are reported here solely as external validation of the architecture and training recipe.

Table 4.3: Top runs ranked by R@10 fields compared to the Baseline with the same Data and Epochs.

Audio Encoder	Text Encoder	Projection	R@1	R@5	R@10	mAP@10
PaSST	RoBERTa-large **?**	MLP	0.182	0.447	0.592	0.296
PaSST	RoBERTa-large **?**	Vision	0.144	0.377	0.524	0.245
PaSST	BGE-base **?**	Vision	0.122	0.345	0.485	0.218

While the results were much closer to the baseline with sufficient training data, the architecture still underperformed the baseline with very similar settings.

5 Discussion

Our main findings are that replacing a projection head with a Vision proxy leads to underwhelming results and leads to a loss in retrieval accuracy. This is likely due to information loss during the conversion from embedding to image. Unlike natural images, spatial neighbors in the proxy image need not be semantically related, so our method can fragment meaningful correlations that the dual encoder had already organized in its latent space.

Proxy images differ drastically from natural images. If we initialized from ImageNet, those priors may be mismatched; if we trained from scratch, the data scale (Clotho) is too small for ViTs to shine. This could explain the better results when training on more data for the DCASE competition **?**, where the same architecture with the vision proxy was trained on both Clotho **?** and Audiocaps **?**.

The ViT head adds parameters and optimization complexity without adding supervision. In contrast, the standard dual encoder approach focuses on alignment under contrastive loss and scales predictably with the data.

Using better methods for generating images, while preserving information, and running the same experiment on a larger dataset, could drastically change the results.

6 Conclusion

We introduced a vision-mediated proxy for audio–text retrieval that renders pre-trained audio / text embeddings as images and feeds them to a ViT head trained with symmetric contrastive loss. In Clotho v2.1 **?**, the approach underperformed a strong frozen encoder + MLP baseline across R@K and mAP@10 in both retrieval directions. Given the negative result, the contribution is primarily diagnostic: under modest data and compute, adding a vision head on top of already meaningful embeddings can hurt alignment, likely due to information loss when converting embeddings to images and capacity–data imbalance.

Practically, our study supports keeping the simplest effective head for dual encoders in this regime. Promising next steps include information-preserving mappings and increased training data, so that Vision Transformer **?** can pick up closer similarities between images. We release a clean codebase to facilitate reproduction and future investigation.

Acknowledgments

I thank Prof. Shah Nawaz (supervisor) for guidance and feedback, and Prof.Paul Primus for helpful discussions and ideas. I also appreciate the support of my colleages Giovanni Filomeno and Florian Spiessberger (DCASE Teammates). Any errors are my own.

7 Appendix

7.1 Code

The Code is available under `https://github.com/younissk/embed2image-contrastive-retrieval`. It is a Python project which uses uv as its project manager. See the README.md and Makefile for reproducibility and getting started.

7.2 LLM-Agent Use Disclosure

This thesis is my own work. I used the following commercial tools in limited ways:

- **Grammar check and typo fixes:** `https://www.grammarly.com/`.
- **Rewording, Limited Research:** ChatGPT - GPT-5 (All text was edited or approved by me)
- **Code suggestions and bug-fixes:** Cursor and Codex (All code was edited or approved by me)