Video Clip Retrieval

Petros Kafkas

July 3, 2023

Table of Contents

Project Description

- The purpose of this project is to implement a content retrieval framework (Database + query)
- To this end, we experiment using an unsupervised learning model (VAE) in order to represent our Database
- We utilized a "distance" metric as a similarity value between videos

Project Pipeline

1. We encoded a number of popular songs from spotify in order to act as our Database

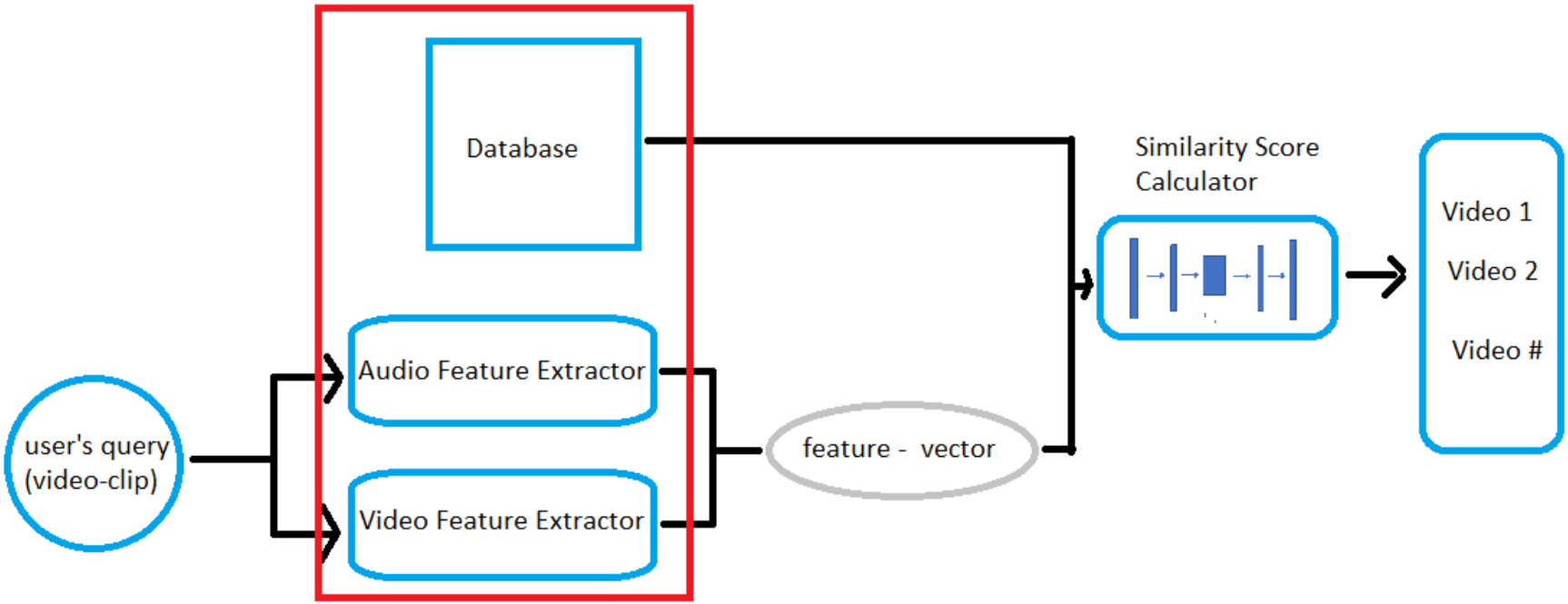


Figure: Project Pipeline

Project Pipeline

2. A user video-query in encoded using the same architecture as the Database

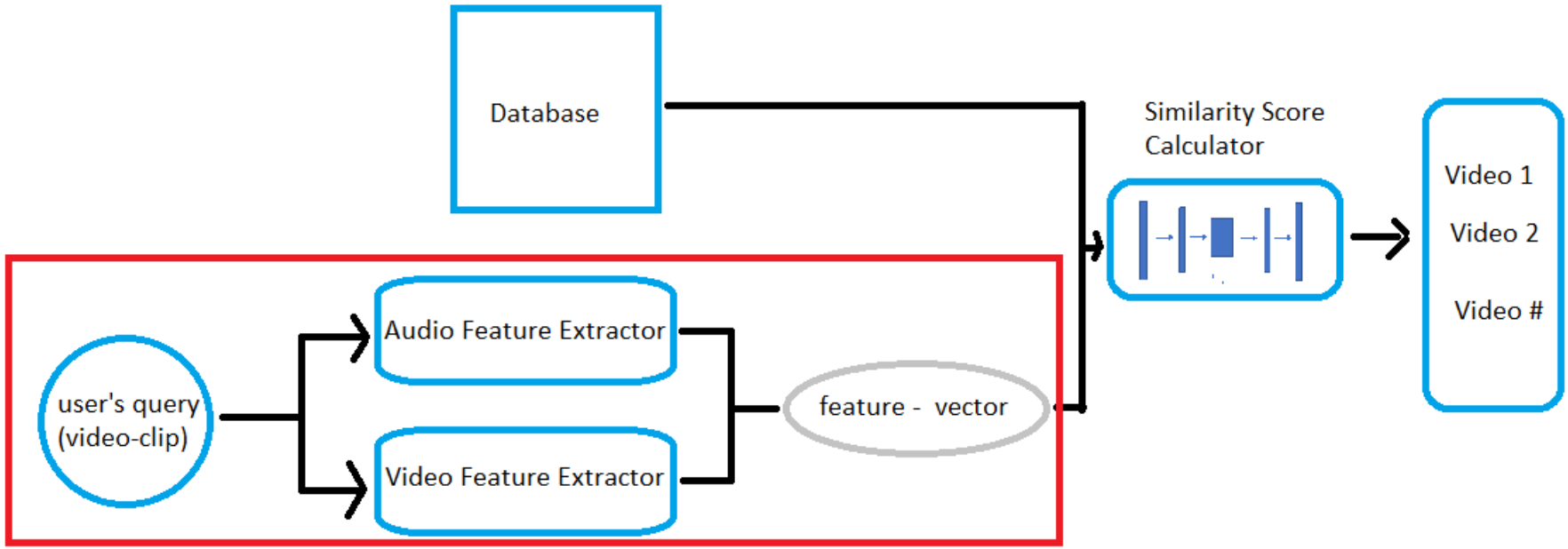


Figure: Project Pipeline

Project Pipeline

3. The query-feature vector is compared to our database and we get N closest matches

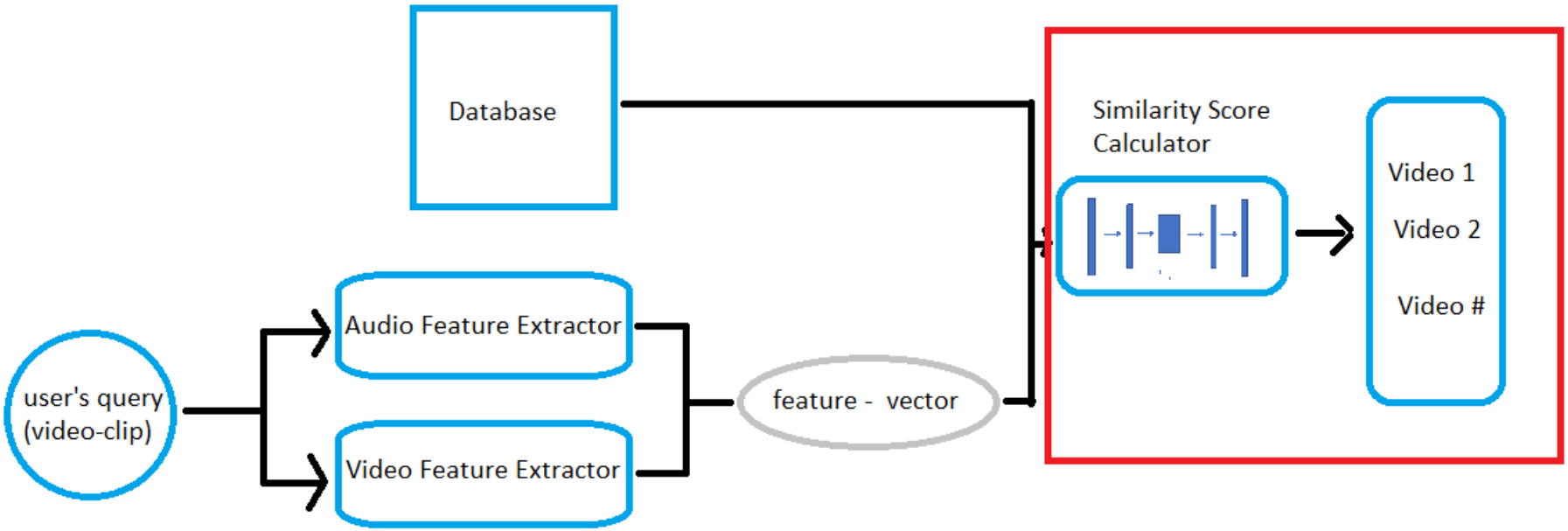
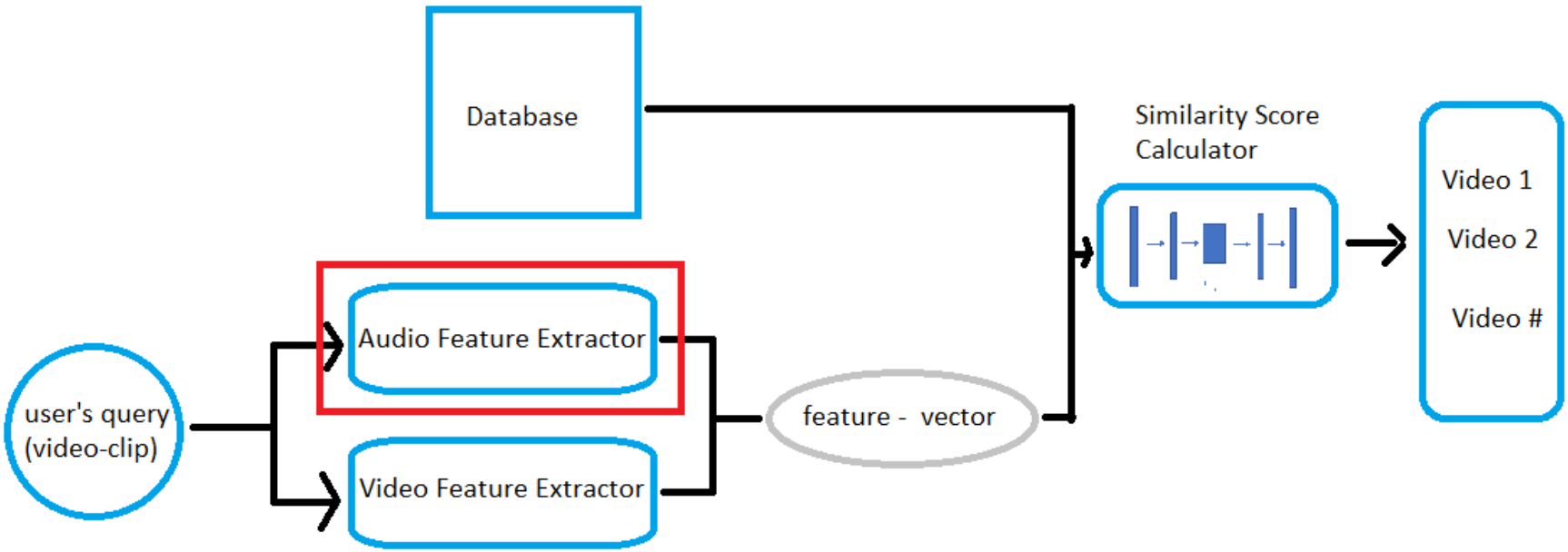


Figure: Project Pipeline

Tools

For the audio extractor we used the pyAudioAnalysis library in order to extract feature vectors:

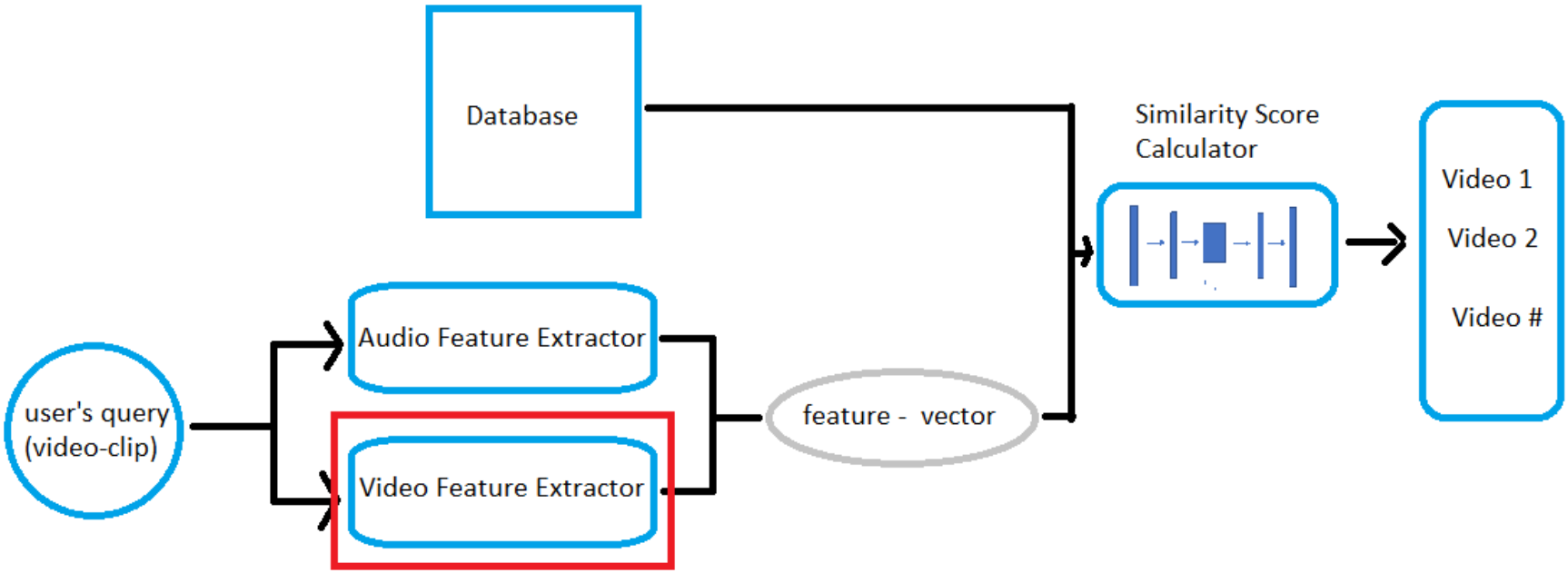
- From each video clip was extracted the equivalent raw audio (.wav) file
- From pyAudioAnalysis we utilized the function for mid-term (average) feature generation (138 values per song)



Tools

For the video extractor we used the openAi CLIP model:

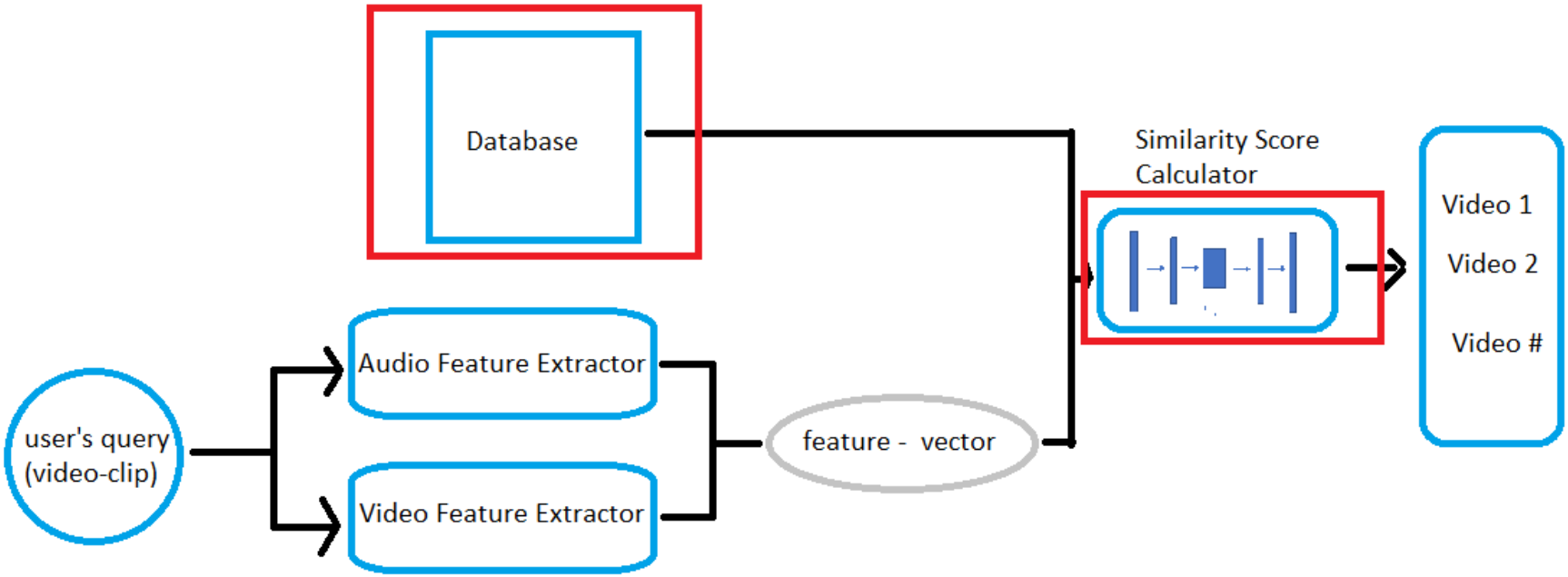
- We used the image encoder of the CLIP model to generate feature vectors
- The total feature matrix had a dimension of *(frames (/1s) * 512)*
- We average across each of the 512 features, final f. vector of *dim(1x512)*



Tools

For the feature matrix creation

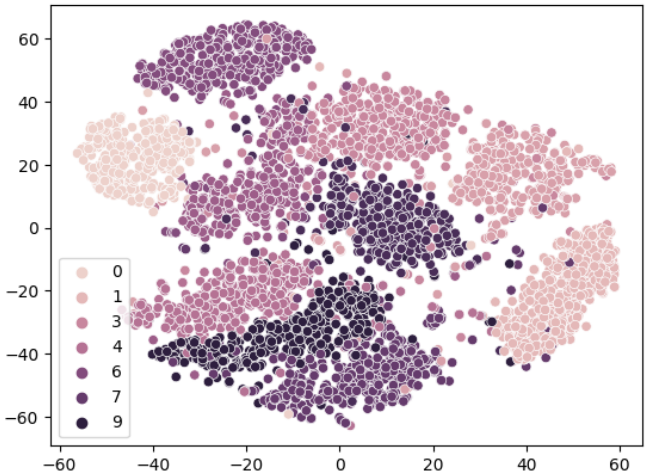
- We encoded the concatenated feature vectors and passed them through a Variational Auto Encoder
- The VAE "learned" the representation of the feature vectors
- This representation in the latent space is our Database



Why VAE

We chose to use a VAE for the encoding process because:

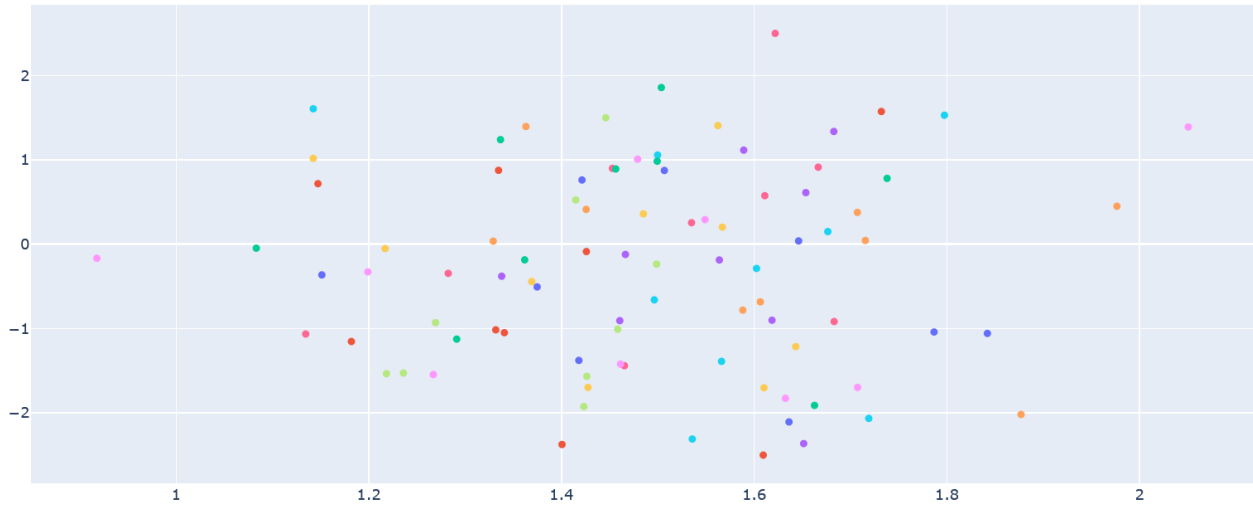
- A variational auto encoder produce a *well regulated* latent space
- This means that distances within this space have a direct correlation to the type of data encoded/decoded
- *Key Idea: identify most similar content by the proximity of the representations on the latent space*



Why VAE

We chose to use a VAE for the encoding process because:

- A variational auto encoder produce a *well regulated* latent space
- This means that distances within this space have a direct correlation to the type of data encoded/decoded
- *Key Idea: identify most similar content by the proximity of the representations on the latent space*



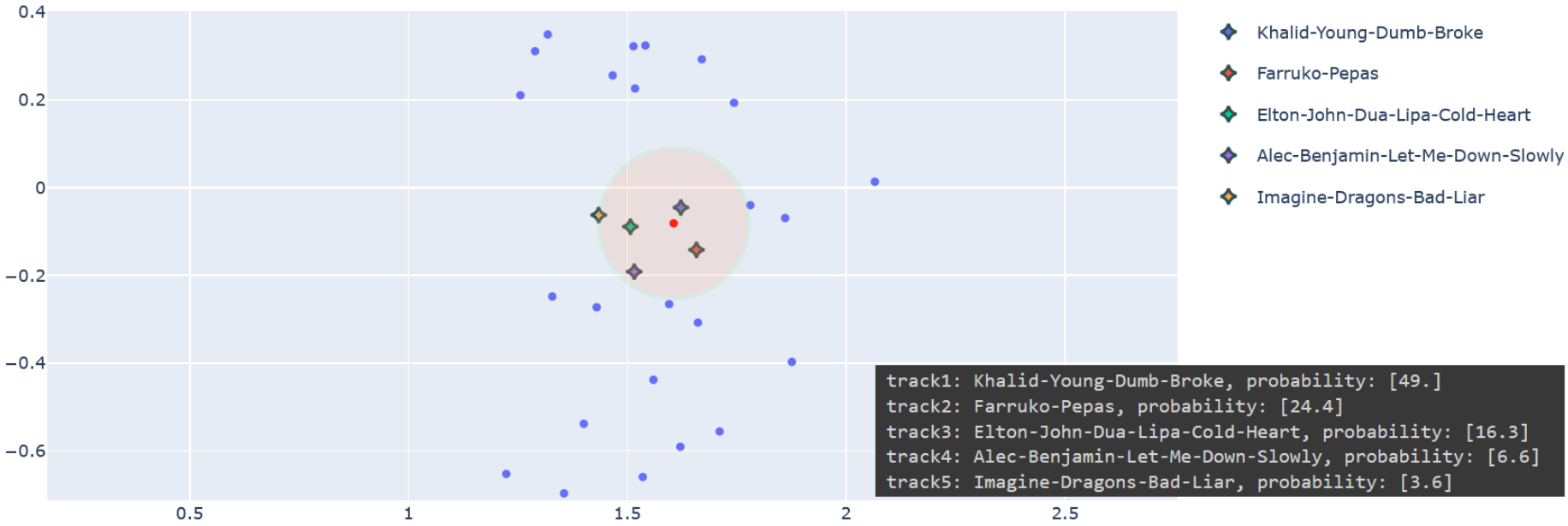
How to measure similarity

We chose then to find the most similar content to our query in this way:

- Pass the query to the encoder function and project its representation to the latent space of the Database
- Measure the distance of the data point to its K-nearest neighbors: *in principle* these would be the most similar to it
- Normalize the distances and apply a softmax function in order to get "probability of similarity"

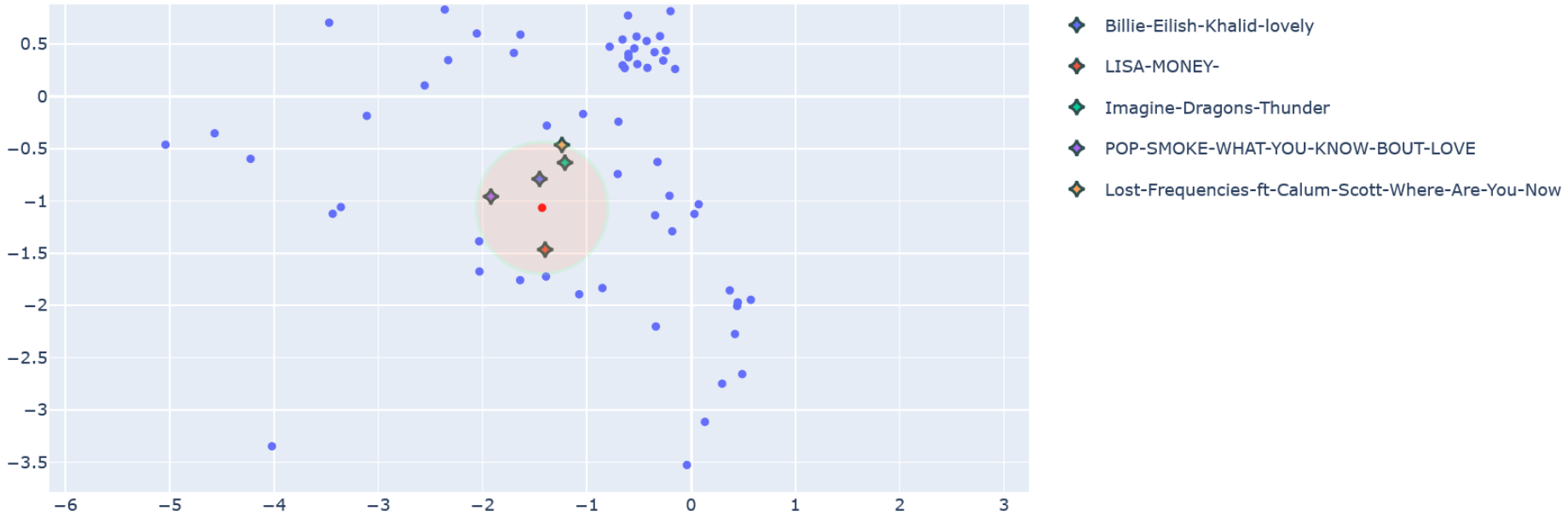
Experiment Results

Some of our experiment outcomes:



Experiment Results

Some of our experiment outcomes:



Our algorithm has the following advantages:

- It is easy to create visualizations of the data and the data proximity
- Relatively easy to combine with other modalities in an *early fusion* model
- Easily scalable

And the following disadvantages:

- Not possible to visualize for separate classifiers: VAE creates different latent space representations for different modalities
- Therefore, you need to have different database representations for separate modalities

Things we would like to have implemented:

- Text based identification and matching
- Higher dimension feature vectors (retain more information about each video)

Thank you!