Movie Summarization for the automatic generation of Movie Tributes

Ricardo Espadinha
Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal
ricardo.espadinha@tecnico.ulisboa.pt

Abstract

The rapid growth of video content on platforms like YouTube has led to an influx of multimedia data, posing computational challenges for storage, browsing, indexing, retrieval, and sharing. Beyond that, the study of emotions in multimedia content has promising applications in various industries, including advertising and personalized multimedia content. The research also advances AI-driven creative processes, particularly in the film industry. By combining the expressive power of movies and music, the proposed approach aims to capture the essence of a movie and evoke nostalgic sentiments through the automatic generation of a movie tribute. The traditional process of creating these tributes is time-consuming and manual, requiring meticulous editing and selection of scenes, music, and visual effects. Automating the generation of movie tributes democratizes the ability to pay homage to cinematic masterpieces, fosters a deeper appreciation of film culture, and facilitates the preservation and celebration of influential movies for future generations. This work combines audio and video segmentation algorithms, various deep learning models, and natural language processing to propose a tool to generate emotionally resonant movie tributes automatically.

Keywords: Computational Creativity, Audio Segmentation,, Video Segmentation, Generation of Movie Tributes, Video Summarization, BERT for Text Extractive Summarization, Affective Audio-visual Correspondence Learning, Music Segmentation, Audio and Video data processing, Transformers, Attention Mechanisms

1. Introduction

In filmmaking, tributes play a significant role in honoring the legacy of influential movies and their creators. A tribute is a heartfelt homage, allowing audiences to revisit beloved films' remarkable moments, themes, and artistic achievements. Traditionally, creating such tributes required extensive manual effort, often involving meticulous editing and selection of scenes, music, and visual effects. This work aims to propose a pipeline for the automatic generation of compelling and emotionally resonant movie tributes.

Over the years, through the advancements of the Internet and social networking sites, Humankind has been experiencing a massive influx of multimedia, particularly video content. Considering YouTube as an example, over 500 hours of video content are currently uploaded to the platform every minute. Entertainment, education, sports, news, and general users or consumer videos are some of the different domains contributing to the fast-growing video content. Nevertheless, while big video data is an excellent source of information discovery, the computational challenges are becoming more demanding and unparalleled. This type of information is typically large, and storing, browsing, indexing, retrieving, and sharing large amounts of data are not trivial tasks. Furthermore, processing such large amounts of data requires significant time and hardware storage. As a result, efficient techniques are needed to deliver the content in a compact format while maintaining the context and the most critical aspects. Intelligent algorithms for different kinds of video processing, e.g., summarization, retrieval, recognition, and others, (re-)emerge as a pressing need.

Being a movie tribute a summary of a movie, we take advantage of the recent discoveries to present an approach that combines computer vision, audio analysis, natural language processing, and deep learning techniques to automatically generate tributes that capture the essence of a movie and evoke nostalgic sentiments.

1.1. Motivation

Usually, the way humans naturally communicate and express emotions is multimodal (Morency et al., 2011). That means we can express and apprehend emotions in various ways (verbally, visually, and others). What is inspiring is that Humankind developed the capability of creating tools that help us perform or improve the performance of a given task. Because of our need and thrive on connecting and expressing ourselves amongst each other, we used this proficiency to create tools to communicate more efficiently at a larger scale (bigger audiences) through creating pieces of art, music, movies, and others. This phenomenon has been the concern and the research

focus of many psychologists (Fernández-Aguilar et al., 2019). Also, more broadly, over the last three decades, interest in the study of emotions has increased notably, focusing both on the construct itself and its interaction with other concepts such as cognition, behavior, personality, and physiology (Kreibig et al., 2013; Kuo et al., 2014; Vianna et al., 2006).

The research on emotions in multimedia content is also very promising to the industry. For instance, recognizing the continuous dynamic emotion evoked by movies can be used to build better multimedia intelligent applications, such as computational affective video-in-video advertising and personalized multimedia content (Yadati et al., 2013; Aditya et al., 2021), to create automatic summaries and adaptive playback speed adjustment for long videos.

This work explores these concepts, resorting to the conjunction of movies and music, specifically through the automatic generation of movie tributes.

1.2. Problem Formulation & Objectives

As described in a previous iteration of this topic by Aparício (2015), *a movie tribute consists of a short music video containing essential parts of the movie playing along with the specified song*. The length of the video corresponds to the song's length. By coordinating two very powerful tools for human expression (music and movies), the purpose of a movie tribute consists of *reliving emotions from the movie quickly and effectively*. *There are videos of this kind on YouTube, made manually by people who long to gather their most meaningful scenes to remember later a movie that they have seen and enjoyed*.

This work will focus on creating a framework to generate a movie tribute taking advantage of recent developments. The primary objective is to develop a robust framework to automatically select and assemble scenes from a given movie into a given musical composition, generating a cohesive and emotionally engaging movie tribute.

In order to achieve this, a set of milestones were defined as follows: **1)** Exploration of the recent advancements in general video summarization techniques; **2)** Exploration of the recent advancements in extractive text summarization techniques; **3)** Exploration of the recent advancements in affective music-video retrieval techniques; **4)** Implementation.

Ultimately, this research aims to contribute to advancing AI-driven creative processes, specifically in the film industry. By automating the generation of movie tributes, we strive to democratize the ability to pay homage to cinematic masterpieces, fostering a deeper appreciation of film culture and facilitating the preservation and celebration of influential movies for future generations. Unlocking an intuitive and efficient way to summarize a movie accordingly to a specific soundtrack would have an enormous impact on the creation of movie trailers, for example, which are even more impactful in the industry.

1.3. Outline

This work is organized as follows:

- **Background and Related Work:** presents a *Conceptual Approach on Video Summarization* for a better understanding of the critical concepts related to general video summarization, and the existing research on the fields used in the proposed approach for this work: general video summarization, extractive text summarization, and affective music-video retrieval;
- **Generation Of A Movie Tribute:** presents our proposed solution, encompassing a detailed explanation of the movie and music streams' data processing, content selection, emotional coherence, and post-production concerns;
- **Experiments:** provides an overview of the dataset utilized in our experiments, as well as the respective results and further discussion and conclusion;
- **Conclusions and Future Work:** presents our overall conclusions and directions for further research and development.

2. Background and Related Work

This section presents a *Conceptual Approach on Video Summarization* for a better understanding of the critical concepts related to general video summarization, as well as existing research on the fields used in the proposed approach for this work (section 1.2): general video summarization, extractive text summarization, and affective music-video retrieval.

2.1. Conceptual Approach on Video Summarization

The evolution of technology and the way it blends into our day-to-day basis led to extreme ease in creating video content. However, with the growth of the generation and availability of this media, there is a direct correlation between the need for new and more efficient ways to handle this type of data. That is why research on this topic has seen exponential growth over the years. Efficient video summarization techniques can facilitate efficient storage, quick browsing, indexing, fast retrieval, and quick content sharing (Tiwari and Bhatnagar, 2021).

2.1.1. Hierarchical Structure of a Video

A video stores spatiotemporal information. In order to be able to manipulate and rearrange this information, we need to define some key concepts in the video structure. We followed the concepts presented by Tiwari and Bhatnagar (2021):

- **Frame:** The elementary unit of a video that is displayed in sequential order (individual images of a video stream).

- **Shot:** Collection of frames captured in an uninterrupted time interval with a single camera.
- **Scene:** Several shots representing a part of an event's complete sequence.

The final video comprises an appropriate combination of the scenes in a timeline, producing a story with its context.

Figure 1: Hierarchical structure of a video (Tiwari and Bhatnagar, 2021).

2.1.2. Structure of a Video Summary

Video summarization techniques aim to produce a compact representation of a given video, keeping the most relevant information intact. How this compact information is presented to the user can vary in different ways depending on the purpose of the summarization task. Money and Agius (2008) identified the audio-visual cues as follows. **Keyframe cues** are the most representative frames extracted from a video sequence and must be presented in temporal order to preserve context. **Video Segment cues** are the essential continuous parts of the original video, being considered an extension of keyframe cues that generally preserve both the motion and audio elements. **Graphical Cues** use visual elements and syntax as a supplement to other cues, presenting an additional level of detail (e.g., the *in-video* text identification of the action that is being presented at that specific moment). **Textual Cues** summarize the content of the video via textual descriptors.

There are multiple types of video summaries. Nonetheless, if the generated summary maintains the video format, a general mathematical representation of the problem can be formulated as follows:

Let V be the video with n frames in sequence. Then, the summarization video, S, is a collection of m keyframes, not necessarily consecutive, but in temporal order.

$$V = \{F_1, F_2, F_3, ..., F_n\} \tag{1}$$

$$S = \{F_{x_1}, F_{x_2}, ..., F_{x_m} \mid 1 \leq x_i \leq n \text{ and } m \ll n\} \tag{2}$$

It is essential to mention here that there is no specified need to present the video segments in the temporal order for our specific problem of generating movie tributes. Music/video synchronization and coherence must be the top priority.

2.1.3. Main Challenges

Summarizing a video stream presents several challenges. The sequential nature of a video makes it a more complex type of content than singular images. The spatiotemporal dependencies of this type of data must be taken into account to produce a summary that properly maintains the context of the original video.

A video agglomerates a wide range of semantics in various ways, such as still and moving images, sound, music, and text. This multimodal nature makes this task much more complex than analyzing text documents or single images (Money and Agius, 2008).

Video summarization is a subjective task. Different users may have different opinions and preferences over the summaries, all being valid (Otani et al., 2019). This hinders the comparison of a generated summary with well-defined ground truth.

In order to identify which frames or parts of a video will be present in the summary, an importance score must be generated. This score depends on the type of summary, user requirements, and genre of the videos. Since what defines importance may vary for different persons, determining what is essential is also a highly subjective task (Otani et al., 2019).

2.1.4. General Framework for Video Summarization

Truong and Venkatesh (2007) analyzed the concept of video abstraction as *a mechanism for generating a summary of a video, which can either be a sequence of stationary images (keyframes) or moving images (video skims)*, corresponding to Static and Dynamic Summaries.

The type of summary we aim to produce with this work is a Dynamic Summary. The general framework to generate this type of summary out of the existing video sequence is shown in Fig. 2.

The process consists of three phases, as follows: **1) Video Segmentation** divides the original video stream into smaller parts that can be comprehended and processed independently. Each of these parts is a composition of sequential frames defining a specific activity or moment and correctly carrying its meaning. For structured videos, where the different segments are easy to locate, this process identifies well-defined shots and scenes from the video stream; **2) Importance Score Prediction** is considered a crucial step as it aims to attribute a score to each segmented unit of the video that defines what will be present in the summary. The challenge of this step is due to the subjectivity related to defining what is essential, as this criterion is not always the same. The formulation of importance may vary depending on the application domain, user preferences, or specific requirements. Some

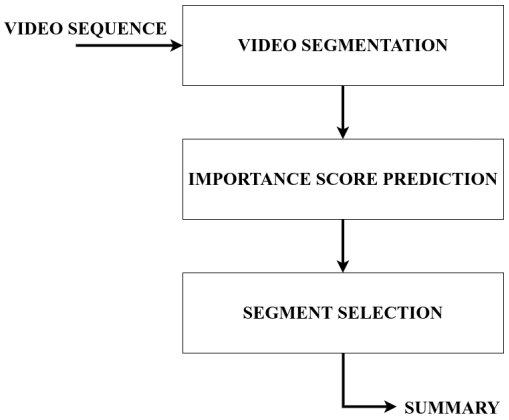

Figure 2: General framework for Dynamic Summary generation (Tiwari and Bhatnagar, 2021).

previous works have focused on visual "interestingness", compactness, and diversity to produce this score (Gygli et al., 2014, 2015; Zhang et al., 2016a; Zhao and Xing, 2014; Zhou et al., 2018; Atencio et al., 2019); **3) Segment Selection** removes the redundant frames and selects the segments in the intended summary based on the computed importance scores.

2.2. Related Work

By looking at the research community, we can find significant contributions related to the fields we need to consider to generate a movie tribute. In this section, we compiled an overview of the current state-of-the-art related to the topics of *Video Summarization Techniques*, *Extractive Text Summarization Techniques*, and *Music - Video Retrival*.

2.2.1. Video Summarization Techniques

Over the years, the video summarization process and techniques have seen many advancements and a broad paradigm shift (Sharghi et al., 2016). As previously said, selecting keyframes is regarded as the most crucial stage in any approach, to obtain a summary based on the video's semantic information. The earlier techniques mainly used low-level appearance cues, motion cues, and graph modeling to identify the key frames from a video sequence such that the identified key frames are important, diverse, and representative (Tiwari and Bhatnagar, 2021). It is up to the programmer to measure the importance and diversity of the key frames through the cues.

Tiwari and Bhatnagar (2021) establish that the summarization process based on user preferences is more prevalent and that Machine Learning and, more specifically, Deep Learning techniques can be used to accomplish it. Machine Learning has proved to be beneficial for the summarization process to yield better results compared to the traditional methods.

Movies and music are sequences. The context in each segment of these sequences depends on the previously gathered knowledge. For example, it is unclear how a traditional neural network could reason about previous events in the film to classify what is happening at a given time. Because of this, the deep learning architectures that show better results in video summarization are the ones dedicated to processing sequential data, such as Long Short-Term Memory (LSTM) (Zhang et al., 2016b; Zhao et al., 2018), Generative Adversarial Networks (GAN) (Mahasseni et al., 2017; Zhou et al., 2018).

Attention Mechanism

In the book *The Principles of Psychology* by James (2007), the author wrote that *Attention is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalization, concentration, of consciousness are of its essence*. Attention is a behavioral and cognitive process of selectively concentrating on specific parts of information while ignoring others. For instance, if we want to guess which person is the oldest in an image of a group of people, intuitively, we will not analyze all the aspects of the image. Instead, perhaps we would look at the faces in the image for some specific features. It was based on this notion that Bahdanau et al. (2014) introduced the fundamental concept of attention mechanism for neural networks. His work focused on Natural Language Processing (NLP); therefore, the attention mechanism appeared as an improvement over the encoder-decoder architecture for neural machine translation systems.

The use of Attention provides a way to build an architecture that consumes all hidden states of the LSTM instead of using just the last hidden state as a proxy for the entire sentence. This way, the context will not weaken through the sequence.

There are two main distinct attention algorithms, hard and soft. Xu et al. (2015) describe both as follows. Soft Attention is when the context vector is computed as a weighted sum of the encoder's hidden states. Oppositely, Hard Attention uses attention scores to select a single hidden state for the context vector. The challenge with this latest algorithm is that the function that selects the hidden state is not differentiable (e.g., argmax), leading to more complex techniques than back-propagation to update the model's weights.

Ji et al. (2019) utilized this mechanism to treat video summarization as a sequential encoder-decoder problem and formulate it with an attention-based LSTM framework, the Attentive encoder-decoder networks for Video Summarization (AVS).

Fajtl et al. (2019) argued that the use of encoder-decoder architectures has a significant problem because of the fixed size of the latent space, completely independent from a possible variation in the input's length. This "bottleneck" implies a higher information loss for longer sequences. To address this problem, the authors proposed a supervised keyshot-based architecture that completely replaces the LSTM encoder-decoder network with soft self-attention and a two-layer, fully connected network for regression of the frame importance score.

Transformers

The Transformer architecture was introduced by Vaswani et al. (2017) in a paper called *Attention is all you need* to improve the performance of deep learning NLP translation models using attention mechanisms.

At its core, the Transformer is composed by a stack of encoders and decoders (Fig. 3). The original architecture uses six of each.

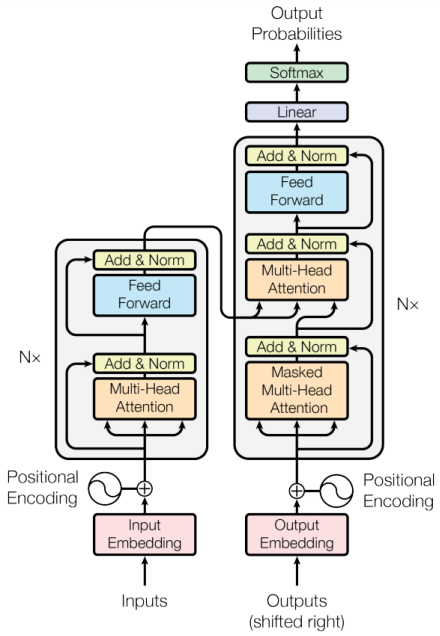

Figure 3: The Transformer - model architecture (Vaswani et al., 2017).

In training, the input enters the Encoder consisting of a self-attention layer to compute the relationship between each word of the input sequence and a Feed-forward layer. By stacking multiple encoders on top of each other, the model can further encode the information where each layer has the opportunity of learning different attention representations, potentially improving the predictive power of the Transformer. The sequence that we are trying to obtain enters the Decoder consisting of a first self-attention layer to compute the relationship between each word of the output sequence, a second encoder-decoder attention layer that aims to compute the relation between the output of the self-attention layer below it and the encoded input sequence, and a Feed-forward layer as well. Each stacked Decoder takes as inputs the multiple outputs from the layers of encoders, allowing the model to focus on different combinations of attention, once again, potentially improving the predictive power.

The Multi-Head Attention is a way to compute the Attention with a greater power of discrimination by combining several similar Attention calculations performed in parallel.

Both inputs of the encoder and decoder stacks pass through a Positional Embedding layer to obtain the position information of the inputs since the sequence is processed simultaneously, contrarily to Recurrent Neural Networks (RNN) architectures (LSTM and Gated Recurrent Units (GRU)).

The way these models are trained is as follows. The input sequence passes through a positional embedding layer and enters the Encoder's stack, which outputs an encoded input representation. The target sequence also passes through a positional embedding layer and enters the Decoder stack alongside the encoded representations of the input. The output will be a sequence to be compared, via a Loss function, to the target one, generating gradients to train the Transformer during backpropagation. As stated before, the complete target sequence is fed to the Decoder, allowing it to access all the single inputs (past and future ones). However, the goal is for the Decoder to predict the word based only on the past ones. Therefore, it uses an attention mask in the self-attention layer to prevent the Decoder from "peaking" ahead at the rest of the target sentence when predicting the next word.

The steps are identical to the training process during inference, but we do not have a complete sequence for the decoder. Therefore, at the start of the prediction, we use an empty sequence with only a start-of-sentence token; the decoder will predict a word at a time with the concatenation of the previously predicted words until it predicts an end-of-sentence token.

Narasimhan et al. (2021) have proposed a multimodal summarization model, which takes a video and a natural language text as inputs to generate a summary of the video conditioned by the text. Note that, for generic video summarization, the text input is a system-generated video description.

The authors formulate the video summarization task as a per-frame binary classification problem. The image and text embeddings are extracted using pre-trained networks for each purpose, respectively.

They modified the Multi-Head Attention described by Vaswani et al. (2017) to a Language-Guided Multi-Head Attention to fuse information across the video and language modalities efficiently and infer long-term dependencies across both. Besides, because the objective is for all sentences of the text description to attend to all frames in the video, a single attention layer is not sufficient.

After that, a Frame-Scoring Transformer uses the fused image-text representations as input and outputs scores to individual frames in the video. The image-text embeddings are fed to the bottom of both the encoder and decoder stacks. Similar to Vaswani et al. (2017), the authors add positional encoding to the input embeddings at the bottom of the encoder and decoder stacks to insert information about the relative positions of the tokens in the sequence.

Finally, the frame-level scores are converted into shot-level scores, and these shots are selected and arranged to produce the final summary.

2.2.2. Extractive Text Summarization Techniques

In the work of Aparício (2015), they considered five text-based summarization approaches: LexRank (Erkan and Radev, 2004) and Support Sets (Ribeiro and de Matos, 2011), which are centrality-based, MMR (Carbonell and Goldstein, 1998), and GRASSHOPPER (Zhu et al., 2007), which are diversity-based, and LSA (Gong and Liu, 2001), which is a mathematical technique based on Singular Value Decomposition (SVD). Centrality-based algorithms consider that the most important content of an input is the most central, considering its representation as a graph, spatial, etc. On the other hand, Diversity-based algorithms focus on maintaining diversity in the summary.

Aparício (2015) summarized the movie's subtitles using the centrality-based LexRank (Erkan and Radev, 2004) algorithm to determine the film's most important content and avoid diversity (to guarantee coherence). The LexRank algorithm is a graph-based algorithm in which the text is converted into a graph representation, where sentences, represented by TF-IDF score vectors, are nodes, and edges between sentences are weighted based on their cosine similarity.

An adapted version of Google's PageRank algorithm for ranking web pages (Brin and Page, 1998) is applied to determine the centrality of each sentence. It measures the importance of a sentence based on its connections to other sentences.

Bidirectional Encoder Representations from Transformers (BERT) was introduced by Devlin et al. (2018) to improve language understanding by pre-training the bidirectional transformer architecture on a large text corpus.

When it was presented, BERT generated significant excitement within the Machine Learning community due to its ability to achieve state-of-the-art performance in various NLP tasks. These tasks include Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and several others.

The primary technical breakthrough of BERT involves utilizing bidirectional training from the Transformer (explained previously in section 2.2.1). Unlike previous approaches focused on sequential or combined left-to-right and right-to-left training, BERT trains the model bidirectionally. Devlin et al. (2018) findings also demonstrate that bidirectional training enables the model to better understand language context and coherence compared to unidirectional models. In BERT's case, as it aims to build a language model, only the encoder part of the Transformer is needed.

The input to BERT consists of a token sequence. These tokens are initially transformed into vector embeddings and subsequently processed within the neural network. The network's output is a sequence of vectors, each representing an input token with a corresponding index, and all vectors have a size of n. The main objective of implementing bidirectional training is to capture contextual relationships between words or sub-words in text. This way, to be able to leverage this quality, the authors defined two training strategies:Masked-Language Modeling (MLM) (inspired by the *Cloze* task in Taylor (1953)) and Next Sentence Prediction (NSP).

During the training of the BERT model, both MLM and NSP strategies are trained simultaneously. The objective is to minimize the combined loss function that incorporates both training objectives.

The pre-training process in BERT aligns with the established practices in language model pre-training. The authors utilized the BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words) as the pre-training corpus. In the case of Wikipedia, only the text passages were extracted (disregarding lists, tables, and headers). To extract longer contiguous text sequences, utilizing a corpus at the document level rather than a shuffled corpus at the sentence level, like the Billion Word Benchmark (Chelba et al., 2014), is crucial.

The pre-trained BERT model can be used to obtain sentence embeddings to be applied in many different ways. (Miller, 2019) leverage BERT for Extractive Text Summarization in Lectures. By visual examinations of clusters, the authors determined that the second to last averaged layer produced the best embeddings for representations of words for their use-case. One hypothesis for this, which the authors also mention, is that the *final layer was biased by the classification tasks in the original training of the model*.

After extracting the embeddings for each sentence of the complete text corpus, the authors selected the K-Means algorithm for clustering the embeddings. They defined, as k, the final desired number of sentences in the produced summary. The sentences closest to the clusters' centroids were selected for the final summary.

The K-Means algorithm gained significant popularity in MacQueen (1967), where the author expanded on it, formalizing its principles and demonstrating its effectiveness for clustering analysis. The algorithm is an unsupervised machine-learning technique for partitioning a dataset into distinct groups or clusters. The algorithm aims to group similar data points based on their embeddings similarity. It starts by randomly initializing k cluster centroids in the embeddings space and assigning each data point to the nearest centroid based on their distance (usually Euclidean distance). From here, in each iteration, a new centroid position for each cluster is computed by taking the mean of all data points assigned to it and reassigning all of them to the new centroids' positions until the centroids stabilize and there is minimal change in the data point assignments. This procedure results in obtaining k clusters, with each data point belonging to the cluster defined by its nearest centroid.

2.2.3. Music - Video Retrival

- **Available Datasets**

There are limited datasets for *affective* audio-visual correspondence learning, and they are derived from combinations of existing ones.

Some other studies involving action recognition assignments or audio signal categorization produced the datasets used in most audio-visual correspondence learning tasks (Gemmeke et al., 2017; Aytar et al., 2016; Kay et al., 2017; Chung et al., 2017; Parkhi et al., 2015; Nagrani et al., 2017). The IMEMNet (Zhao et al., 2020) and IMAC (Verma et al., 2019) datasets are created from music and images. The two datasets in Li and Kumar (2019) are both dedicated to affective correspondence learning between music and video. However, they are not released. The authors constructed music-video pairs involving a crowd-sourcing approach, where annotators provided feedback on the prevalence of emotions in both streams. The dataset comprises 3,000 music-video pairs, evenly divided into matched and mismatched pairs, covering 140 emotions.

The video streams for the first dataset were gathered from *Cowen's* dataset (Cowen and Keltner, 2017), and the music segments were chosen at random from the Unbalanced Train set of the *Music Mood* dataset, which is part of the *AudioSet* ontology (Gemmeke et al., 2017).

For the second dataset, the music was collected from Spotify and the videos from Instagram and from *Moments in Time* dataset (Monfort et al., 2019).

- **EmoMV - Datasets and Proposed Model**

The limitation on available datasets with affective audio-visual correspondence learning benchmarks motivated Thao et al. (2023) to create a collection of three datasets (EmoMV) for affective correspondence learning between music and video modalities. Furthermore, alongside creating three novel datasets, a benchmark deep neural network model is introduced for binary classification of affective music-video correspondence. Subsequently, this model undergoes modifications to accommodate affective music-video retrieval.

EmoMV Datasets

The authors utilize emotion categories to construct matched and mismatched music-video pairs, as discrete representations of emotions are generally more comprehensible for non-experts. Three datasets were produced:

- **EmoMV-A Dataset:** Makes use of the music video segments from the MVED dataset Pandeya et al. (2021);
- **EmoMV-B Dataset:** Makes use of the Music Mood dataset (Olah, 2015) of the AudioSet ontology;
- **EmoMV-C Dataset:** Comprises of self-collected music videos of songs featured in movies (soundtrack music videos).

A summary of the three datasets in the EmoMV collection is presented in Table 1.

EmoMV Model for Affective Music-video Retrieval

In addition to the dataset Collection, the EmoMV authors also proposed two models for distinct tasks: binary affective music-video correspondence classification and affective music-video retrieval. In the scope of this work, we are presenting the affective music-video retrieval model, although they are both very similar.

The proposed model utilizes pre-trained deep neural networks, initially designed for action recognition and audio classification, to extract visual and audio features from video and music streams. To achieve a shared representation, video and music projection heads are employed to embed the visual and audio features into a common representation space where the distance between the visual and audio embeddings is computed. Following the multi-task learning approach, the authors appended the music and video branches (for music and video emotion classification) to the music and video subnetworks, respectively. A highlighted representation of the proposed model is presented in Fig. 4.

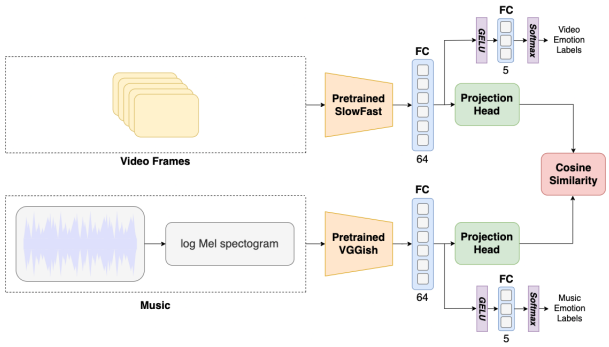

Figure 4: EmoMV Model for Affective Music-video Retrieval.

Video subnetwork

To extract the embeddings of the video stream, the authors of Thao et al. (2023) made use of the SlowFast (Feichtenhofer et al., 2019) network pre-trained on the Kinetics human action video dataset (Kay et al., 2017).

The SlowFast (Feichtenhofer et al., 2019) network is a popular video classification model that leverages spatial and temporal information for better performance. It is typically pre-trained on large-scale video datasets like Kinetics, which contains various human action videos.

During pretraining on the Kinetics human action video dataset (Kay et al., 2017), the SlowFast network is trained to predict the correct action labels for many video clips. This process enables the model to learn meaningful video representations that can generalize to other video-related tasks.

After pretraining, the SlowFast network can be used as a feature extractor for video streams by removing its last classification layer. Given a video, the network processes it frame by frame, and the output embeddings capture high-level semantic information about the video content.

As a result, the authors of Thao et al. (2023) end up

Table 1: The EmoMV Dataset Collection.

Dataset	MusicVideos	Train Set	Validation Set	Test Set
EmoMV-A	Matched	2208	310	124
	Mismatched	1902	246	124
EmoMV-B	Matched	248	60	-
	Mismatched	248	60	-
EmoMV-C	Matched	180	48	-
	Mismatched	180	48	-

with a $2,304$ dimensional feature vector from each video stream, which goes through a fully-connected layer with 64 neurons for dimensionality reduction and passes to the video projection head.

Music subnetwork

To extract the embeddings of the music stream, the authors of Thao et al. (2023) made use of the VGGish (Hershey et al., 2017) network pre-trained on the AudioSet ontology (Gemmeke et al., 2017) to extract a 128-dimensional feature vector from the log-mel spectrogram computed with each 0.98-second music segment (which is at a sampling rate of 16 kHz with signed 16-bit Pulse-code Modulation (PCM) encoding and a mono channel).

The feature vectors extracted from all 0.98-second (*hop size*) music segments were averaged element-wise to obtain a condensed representation of the music stream within each segment within each music segment. This averaging process resulted in a 128-dimensional vector that captures the essence of the music in the segment. Similar to the video subnetwork, this 128-dimensional vector is then passed through a fully-connected layer with 64 neurons to reduce its dimensionality further and passed to the music projection head.

The AudioSet ontology (Gemmeke et al., 2017) is a large-scale dataset that contains a wide range of audio clips, each labeled with a specific sound event. The VGGish (Hershey et al., 2017) network is trained on this dataset to learn discriminative audio representations.

Emotion classification branches

To enable multi-task learning, Thao et al. (2023) extend the video and music subnetworks by adding video and music branches. As shown in Fig. The dimensionally reduced visual and audio feature vectors, obtained by the fully-connected layers of 64 neurons in their respective subnetworks, are directed to the newly introduced video and music emotion classification branches.

Both the video and music emotion classification branches share a common structure. They consist of a Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel, 2016) activation function, followed by a fully-connected layer comprising five neurons (corresponding to the number of emotion categories). A *softmax* layer is then applied to obtain the emotion classification output for each modality.

These emotion classification branches are jointly trained with the main branch during training. The training process involves using three cross-entropy loss functions, where each loss function carries equal weight.

By incorporating these multi-task branches and jointly training them, the authors aimed to leverage the shared information between video and music modalities, enhancing the model's ability to classify emotions accurately.

Projection Heads

Inspired by the projection heads originally applied to the textual and visual features in the Contrastive Language-Image Pretraining (CLIP) model (Radford et al., 2021), Thao et al. (2023) also applied this technique for music and visual features instead, with the objective of embed the visual and audio features into a common representation space.

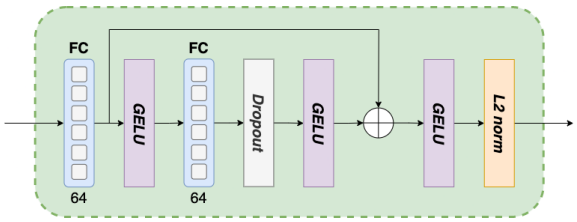

Figure 5: Structure of the Projection Heads used to project visual and audio features into a common representation space.

Both video and music projection heads follow the same structure: fully-connected layers of 64 neurons each, the GELU (Hendrycks and Gimpel, 2016), a dropout ratio of 0.5, a residual connection, and L2-normalization, as described in Fig. 5.

Cosine Similarity

Similar to the works of Zhao et al. (2020); Li and Kumar (2019); Wang et al. (2012), to perform the affective music-video retrieval task, the main branch of the model computes the cosine distance $d_{\cos}(f_v, f_m)$ between visual (f_v) and audio embeddings (f_m) as follows:

$$d_{\cos}(f_v, f_m) = 1 - S_{\cos}(f_v, f_m) \tag{3}$$

Where $S_{\cos}(f_v, f_m)$ is the cosine similarity between the visual and audio embeddings:

$$S_{\cos}(f_v, f_m) = \frac{f_v . f_m}{\|f_v\| \times \|f_m\|} \tag{4}$$

Where $\|f_v\|$ and $\|f_m\|$ are the Euclidean norm of the vectors f_v and f_m, respectively.

Experimental Setup

The three networks are trained simultaneously using three equal weighted loss functions, including two cross-entropy loss functions (Ackley et al., 1985) for the video and music subnetworks, and the contrastive loss (Hadsell et al., 2006) on the cosine distance between the visual and audio embeddings.

During the inference process, when a music query is provided, the model computes the cosine similarity score between the audio embedding of the queried music and the visual embeddings of all video segments in the provided database. This similarity score quantifies the similarity between the audio and visual features. The video segments are ranked by leveraging these similarity scores, and the top-ranked results are identified as the best matches to the music query. This ranking process enables us to find the video segments that align closely with the audio characteristics of the music query.

2.3. Summary

We verify that many recent advances in the Deep Learning field related to video summarization come from NLP architectures. This happens because of the similarities in the structure of both data types.

We chose to use BERT for extractive text summarization of subtitles and the EmoMV model for Affective music-video retrieval to generate a movie tribute automatically.

3. Generation Of A Movie Tribute

This chapter presents our approach to the generation of a movie tribute. An overview of the proposed pipeline is presented in Fig. 8.

3.1. Proposed Architecture

In the Music Branch, we segment the music based on its tempo (Fig. 6) and extract audio features from each segment using the VGGish (Hershey et al., 2017) network pre-trained on the AudioSet ontology (Gemmeke et al., 2017).

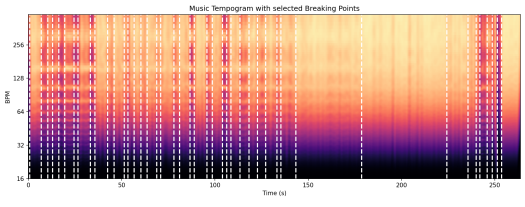

Figure 6: Computed tempogram and respective breaking points for the music "Chevaliers De Sangreal" by Hans Zimmer.

For the Movie Branch, we run extractive summarization using BERT on the subtitles corpus to select highlights corresponding to movie scenes with subtitles. We also implemented K-means clustering to select movie highlights without subtitles. After selecting the movie highlights, the embeddings for each one are extracted using the SlowFast (Feichtenhofer et al., 2019) network pre-trained on the Kinetics human action video dataset (Kay et al., 2017).

We selected the EmoMV model pretrained on the EmoMV-C dataset presented before (section 2.2.3) to compute the similarity matrix between the selected movie highlights and the music segments. We also add a weight related to the movie scenes' temporal order (7) before matching each music segment to the corresponding highlight.

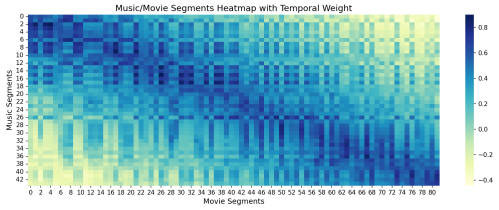

Figure 7: Example of a heatmap representation of the final score matrix to be used for the music segments/highlights matching.

The Post-Production corresponds to loudness adjustment for each soundtrack (music and collected movie segments) and an offset that is added to all the breaking points' values in case there is a need to adjust the movie scenes timestamps (in the majority of the cases, this is not needed).

On top of the model, we also created a simple User Interface (UI) to monitor the whole process and intuitively adjust the parameters to generate the movie tribute.

3.1.1. Setup

In our study, the training phase was done in a local machine using *Python 3.8.12* on a *2.3 GHz Quad-Core Intel Core i5 processor*. All the training parameters are the same as in the original work (Adam Optimizer, $1,000$ Maximum number of epochs, 256 Batch size, 0.0001 of Learning Rate and 20 of Early stopping patience).

The **Optimizer** algorithm adjusts a model's parameters during training. This model uses the Adam (Kingma and Ba, 2014) optimizer. The **Maximum number of epochs** is the maximum number of complete iterations through the training dataset. The **Batch size** is the number of feature vectors loaded per batch. The **Learning Rate** is the magnitude of the parameter update at each iteration. The **Early stopping patience** is how long to wait after the last time validation loss improved.

The model stopped training at **387 epochs during 86.07 seconds** of running.

The **Mean Average Precision (mAP) score** and the **top-K retrieval accuracy** metrics obtained are presented in Table 2.

3.2. Architecture and User Interface

In the process of building the proposed pipeline, the following architectural designs were taken into consider-

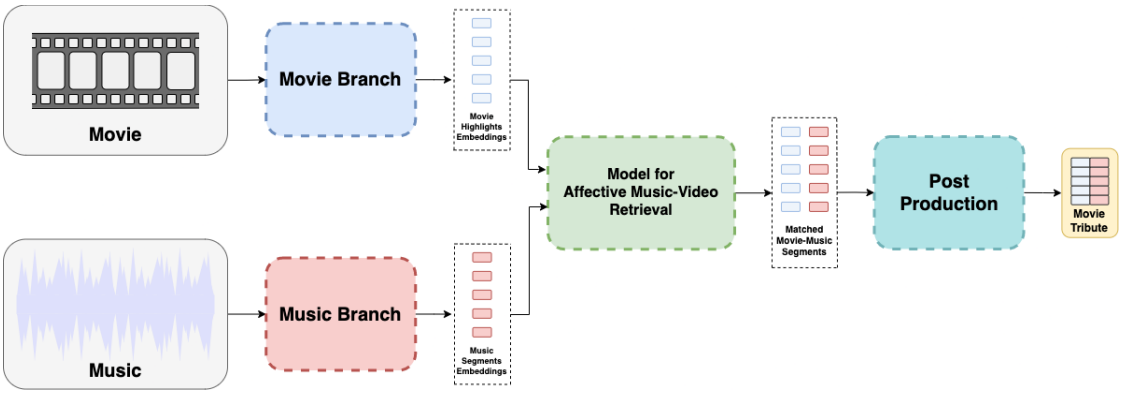

Figure 8: Proposed Architecture Overview.

Table 2: Model's metrics obtained in the EmoMV-C dataset.

Top-1	Top-3	Top-5	mAP
48.95%	80.20%	88.54%	41.34%

ation: The overall architecture is built with a clear distinction between different modules, making it easier to change models and algorithms being used for the multiple tasks in the pipeline; Each module has its own internal *Jupyter Notebook*, where the related methods were tested; There is a module dedicated to storage management of the saved and temporary files generated in the process; In the project's root folder is a *Jupyter Notebook* with the "Main Pipeline" where it is possible to generate a movie tribute selecting from available music and movies and introducing the aforementioned *user inputs*; It is possible to run a *Straemlit* App that provides the user with an intuitive UI to generate movie tributes, monitor the whole process and adjust the post-production parameters.

Straemlit is a Python library that allows developers to create, as Python scripts, interactive web applications, and dashboards for data science and machine learning, accessible through a browser.

The homepage of the app (Fig. 9) presents the user with some basic information regarding the difference between *Main Inputs* and *Export Inputs*:

Main Inputs: Movie, Music, Desired Average Music Segment Time (seconds), Desired Minimum Music Segment Time (seconds), Music Hop Size (seconds), Summarization Ratio Multiplyer, Ratio of no Subtitled Highlights to consider and Device.

Export Inputs: Temporal Weight, Loudness adjustment for the music soundtrack, Loudness adjustment for the movie segments soundtrack and Movie Scenes Breaking-points Offset (seconds).

4. Experiments

This chapter provides an overview of the dataset utilized in our experiments, along with details about the individuals who assessed the final tributes. Subsequently,

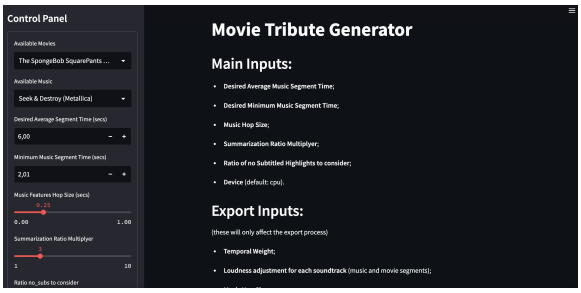

Figure 9: Home page of the *streamlit* UI.

we delve into the evaluation process and engage in a comprehensive discussion of the obtained results.

4.1. Dataset

Using the *The Movie Tribute Generator* App, seven Movie Tributes were created using seven different movies and songs (presented in Table 3). "To The Edge", "See You Again", "The Last Goodbye", and "About You" contain vocals, while the remaining do not. "Howls Moving Castle" is the only animated movie in the dataset.

4.2. Setup

Based on the same evaluation process described by the previous work regarding the "Automatic Generation of Movie Tributes" (Aparício, 2015), we collected opinions related to the generated movie tributes from 32 participants. Most people were males (59.9%), had between 18 and 25 years (58.6%), had a Master's degree (53.1%), and were Computer Engineers (20.7%) or related to other areas of Engineering (20.7%) or related to Tourism (20.7%). Most people watch movies once a week (43.8%), and 7% didn't know what a movie tribute was, while 37.5% watches one once a year, at least.

4.3. Results

The tributes that had the best scores were "Atonement" and "Interstellar", with an average of 8 points in all three

Table 3: Generated Movie Tributes.

Movie	Music	Duration
"Atonement" (2007)	"La Plage" by Yann Tiersen	$01:57'00$
"300" (2006)	"To The Edge" by Lacuna Coil	$03:19'00$
"Furious 7" (2015)	"See You Again" by Wiz Khalifa	$03:46'00$
"The Curious Case of Benjamin Button" (2008)	"The Last Goodbye" by Billy Boyd	$04:05'00$
"Interstellar" (2014)	"Chevallers De Sangreal" by Hans Zimmer	$04:23'00$
"Howls Moving Castle" (2004)	"Merry-Go-Round of Life" by Joe Hisaishi	$02:44'00$
"Before Sunset" (2004)	"About You" by The 1975	$05:24'00$

criteria (content selection, emotional coherence, and overall evaluation), except in the overall evaluation of "Atonement", reaching an average score of 7 points, on a scale from 1 to 10. "Furious 7" obtained the worst scores, with 6.25 points on overall evaluation, 6.5 on content selection criteria, and 5.6 on emotional coherence criteria.

On average, our method led to average scores of 7.2, 6.8, and 6.9 on content selection, emotional coherence criteria, and overall evaluation, respectively (Fig. 10).

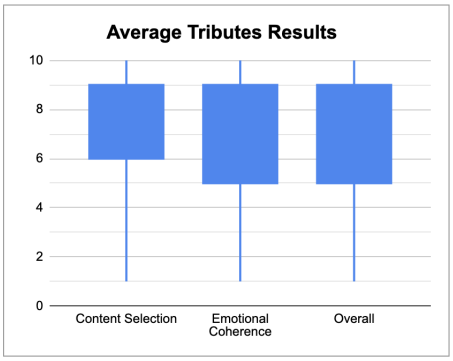

Figure 10: Average tributes' evaluation results.

Concerning Content Selection, some evaluators mentioned the existence of some unnecessary short clips and cuts in the middle of the dialogue in the tributes corresponding to "The Curious Case of Benjamin Button", "Atonement", "Furious 7", and "Before Sunset".

Regarding Post-Production, it was considered that some parts of the tribute have a big discrepancy between the background sound of the movie highlight and the correspondent music segment (shooting scenes, car sounds, low dialogue sound, movie background music playing at the same time as the tribute music). This was mainly identified in the following tributes: "The Curious Case of Benjamin Button", "Furious 7", "300", "Interstellar", and "Before Sunset"

In terms of emotional coherence, the majority of the negative critics evolve around a discrepancy between the overall tone of the song against the overall tone of the music. For instance, it was mentioned that in "Furious 7" the clips are majorly action-based and the song has more of an overall sad tone. In the case of "300", "most of the clips were pure action which felt right with the song", but there were a few clips where a bigger emotional dissonance was felt, majorly corresponding to more calm clips, with dialogues, for example.

We also computed Spearman's ranks for all metrics combinations and concluded that all of them have a strong positive correlation with each other. The ones that appear to be more strongly related are Content Selection and Emotional Coherence, and Content Selection and Overall Evaluation.

5. Conclusions and Future Work

5.1. Conclusions

As described in a previous iteration of this topic by Aparício (2015), *a movie tribute consists of a short music video containing essential parts of the movie playing along with the specified song*.

The focus of this work was on the creation of multimedia artifacts, particularly movie tributes. We implemented various methods for selecting content and ensuring emotional coherence in the generation process.

All this process can be triggered and monitored in real-time through a simple UI created using *Streamlite*.

Seven tributes were generated, and their overall human evaluation was positive. On average, our method led to average scores of 7.2, 6.8, and 6.9 on content selection, emotional coherence, and overall evaluation on a scale from 1 to 10. The tributes that had the best scores were "Atonement" and "Interstellar," with an average of 8 points in all three criteria (content selection, emotional coherence, and overall evaluation), except in the overall evaluation of "Atonement," reaching an average score of 7 points. "Furious 7" obtained the worst scores, with 6.25 points on overall evaluation, 6.5 on content selection criteria, and 5.6 on emotional coherence criteria.

The more strongly correlated metrics are Content Selection and Emotional Coherence, and Content Selection and Overall Evaluation. However, all metrics present a strong correlation with each other.

5.2. Future Work

The majority of the negative critics of the generated tributes, seem to be deeply related to the tribute's edition quality: bad quality and/or balance between the audio

from the movie highlights and the music; scenes containing dialog being cut in the middle of it; unnecessary too short clips.

Since our proposed architecture makes the process of generating a tribute much more efficient by providing an intuitive UI where it is possible to tweak all the input parameters, returning live visual feedback of the changes. This facilitates the process of trial and error until the user is satisfied with the final result, by adjusting the input parameters.

Besides that, other features may improve the over quality of the generated tributes: **To improve Emotional Coherence scores computation**, we could modify the EmoMV model, creating a third branch dedicated to extracting and processing the text embeddings of the subtitles if the scene has it, in a similar way that we do with the visual and audio streams, so we could leverage more contextual information for computing the emotional coherence scores; **Experiment pre-training the EmoMV model with EmoMV-A and EmoMV-B datasets (separately and all together)**; **Fine-tune BERT with a dataset dedicated to the subtitle (movie dialog) format**; **Improve Post-production**.

6. References

References

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

Aditya, D., Manvitha, R., Samyak, M., and Shamitha, B. (2021). Emotion based video player. *Global Transitions Proceedings*, 2(2):368–374.

Aparício, A. M. S. (2015). Automated generation of movie tributes. Master's thesis.

Atencio, P., German, S.-T., Branch, J. W., and Delrieux, C. (2019). Video summarisation by deep visual and categorical diversity. *IET Computer Vision*, 13(6):569–577.

Aytar, Y., Vondrick, C., and Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, 29.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2014). One billion word benchmark for measuring progress in statistical language modeling.

Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2017). Lip reading sentences in the wild. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3444–3453. IEEE.

Cowen, A. S. and Keltner, D. (2017). Self-report captures 27 distinct categories of emotion bridged by continuous gradients. *Proceedings of the national academy of sciences*, 114(38):E7900–E7909.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Fajtl, J., Sokeh, H. S., Argyriou, V., Monekosso, D., and Remagnino, P. (2019). Summarizing videos with attention. In *Asian Conference on Computer Vision*, pages 39–54. Springer.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

Fernández-Aguilar, L., Navarro-Bravo, B., Ricarte, J., Ros, L., and Latorre, J. M. (2019). How effective are films in inducing positive and negative emotional states? a meta-analysis. *PloS one*, 14(11):e0225040.

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE.

Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25.

Gygli, M., Grabner, H., Riemenschneider, H., and Van Gool, L. (2014). Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer.

Gygli, M., Grabner, H., and Van Gool, L. (2015). Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3090–3098.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., et al. (2017). Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE.

James, W. (2007). *The principles of psychology*, volume 1. Cosimo, Inc.

Ji, Z., Xiong, K., Pang, Y., and Li, X. (2019). Video summarization with attention-based encoder–decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1709–1717.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kreibig, S. D., Samson, A. C., and Gross, J. J. (2013). The psychophysiology of mixed emotional states. *Psychophysiology*, 50(8):799–811.

Kuo, J. R., Neacsiu, A. D., Fitzpatrick, S., and MacDonald, D. E. (2014). A methodological examination of emotion inductions in borderline personality disorder: A comparison of standardized versus idiographic stimuli. *Journal of Psychopathology and Behavioral Assessment*, 36(1):155–164.

Li, B. and Kumar, A. (2019). Query by video: Cross-modal music retrieval. In *ISMIR*, pages 604–611.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297. University of California Los Angeles LA USA.

Mahasseni, B., Lam, M., and Todorovic, S. (2017). Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211.

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures.

Money, A. G. and Agius, H. (2008). Video summarization: A conceptual framework and survey of the state of the art. *Journal of visual communication and image representation*, 19(2):121–143.

Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S. A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., et al. (2019). Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*,

42(2):502–508.

Morency, L.-P., Mihalcea, R., and Doshi, P. (2011). Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 169–176.

Nagrani, A., Chung, J. S., and Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*.

Narasimhan, M., Rohrbach, A., and Darrell, T. (2021). Clip-it! language-guided video summarization. *Advances in Neural Information Processing Systems*, 34.

Olah, C. (2015). *Understanding LSTM Networks*. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/` [Accessed: Whenever].

Otani, M., Nakashima, Y., Rahtu, E., and Heikkilä, J. (2019). Rethinking the evaluation of video summaries. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7588–7596.

Pandeya, Y. R., Bhattarai, B., and Lee, J. (2021). Deep-learning-based multimodal emotion classification for music videos. *Sensors*, 21(14):4927.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. *British Machine Vision Association*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Ribeiro, R. and de Matos, D. M. (2011). Centrality-as-relevance: support sets and similarity as geometric proximity. *Journal of Artificial Intelligence Research*, 42:275–308.

Sharghi, A., Gong, B., and Shah, M. (2016). Query-focused extractive video summarization. In *European conference on computer vision*, pages 3–19. Springer.

Taylor, W. L. (1953). "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Thao, H. T. P., Roig, G., and Herremans, D. (2023). Emomv: Affective music-video correspondence learning datasets for classification and retrieval. *Information Fusion*, 91:64–79.

Tiwari, V. and Bhatnagar, C. (2021). A survey of recent work on video summarization: approaches and techniques. *Multimedia Tools and Applications*, pages 1–35.

Truong, B. T. and Venkatesh, S. (2007). Video abstraction: A systematic review and classification. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, 3(1):3–es.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Verma, G., Dhekane, E. G., and Guha, T. (2019). Learning affective correspondence between music and image. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3975–3979. IEEE.

Vianna, E. P., Weinstock, J., Elliott, D., Summers, R., and Tranel, D. (2006). Increased feelings with increased body signals. *Social cognitive and affective neuroscience*, 1(1):37–48.

Wang, J.-C., Yang, Y.-H., Jhuo, I.-H., Lin, Y.-Y., and Wang, H.-M. (2012). The acousticvisual emotion guassians model for automatic generation of music video. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1379–1380.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

Yadati, K., Katti, H., and Kankanhalli, M. (2013). Cavva: Computational affective video-in-video advertising. *IEEE Transactions on Multimedia*, 16(1):15–23.

Zhang, K., Chao, W.-L., Sha, F., and Grauman, K. (2016a). Summary transfer: Exemplar-based subset selection for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1059–1067.

Zhang, K., Chao, W.-L., Sha, F., and Grauman, K. (2016b). Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer.

Zhao, B., Li, X., and Lu, X. (2018). Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7405–7414.

Zhao, B. and Xing, E. P. (2014). Quasi real-time summarization for consumer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2513–2520.

Zhao, S., Li, Y., Yao, X., Nie, W., Xu, P., Yang, J., and Keutzer, K. (2020). Emotion-based end-to-end matching between image and music in valence-arousal space. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2945–2954.

Zhou, K., Qiao, Y., and Xiang, T. (2018). Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhu, X., Goldberg, A. B., Van Gael, J., and Andrzejewski, D. (2007). Improving diversity in ranking using absorbing random walks. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 97–104.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.