

**MUSICAL HYPERLAPSE:
A MULTIMODAL APPROACH TO
ACCELERATE FIRST-PERSON VIDEOS**

DIOGNEI DE MATOS

**MUSICAL HYPERLAPSE:
A MULTIMODAL APPROACH TO
ACCELERATE FIRST-PERSON VIDEOS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: ERICKSON RANGEL DO NASCIMENTO

Belo Horizonte
Outubro de 2021

DIOGNEI DE MATOS

**MUSICAL HYPERLAPSE:
A MULTIMODAL APPROACH TO
ACCELERATE FIRST-PERSON VIDEOS**

Thesis presented to the Graduate Program
in Computer Science of the Federal Univer-
sity of Minas Gerais in partial fulfillment of
the requirements for the degree of Master
in Computer Science.

ADVISOR: ERICKSON RANGEL DO NASCIMENTO

Belo Horizonte

October 2021

© 2021, Diognei de Matos.
Todos os direitos reservados.

Matos, Diognei de.

M433m Musical hyperlapse [manuscrito] a multimodal approach to
accelerate first-person videos / Diognei de Matos – 2021.
xiv, 75 f. il.

Orientador: Erickson Rangel do Nascimento.
Dissertação (mestrado) - Universidade Federal de Minas
Gerais, Instituto de Ciências Exatas, Departamento de Ciência da
Computação.

Referências: f.71-75.

1. Computação – Teses. 2. Visão computacional – Teses. 3.
Reconhecimento de emoções. I Nascimento, Erickson Rangel do
II. Universidade Federal de Minas Gerais, Instituto de Ciências
Exatas, Departamento de Ciência da Computação. III. Título.

CDU 519.6*82.10(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa CRB
6ª Região nº 1510



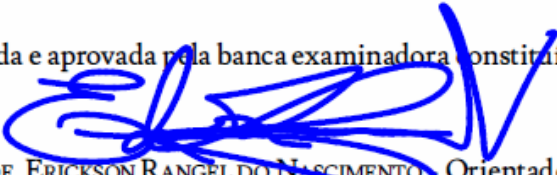
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

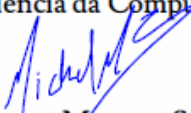
FOLHA DE APROVAÇÃO


Musical Hyperlapse: A Multimodal Approach to Accelerate First-Person
Videos

DIOGNEI DE MATOS

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. ERICKSON RANGEL DO NASCIMENTO, Orientador
Departamento de Ciência da Computação - UFMG


Prof. MICHEL MELO DA SILVA
DPI - UFV


PROFA. ANA PAULA COUTO DA SILVA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 4 de Junho de 2021.

Acknowledgments

I would like to thank everyone who contributed to the development of this project:

- Firstly, I thank God for accompanying me throughout my work journey in developing this project.
- I thank my family for the support they have given me throughout all this time.
- A lot of gratitude to my advisor Erickson Rangel do Nascimento, who accompanied me from beginning to end, always giving encouragement and motivation and helping to solve problems and doubts that arose during the project's development.
- Much gratitude also to the colleagues Washington Luis de Souza Ramos and Luiz Henrique Romanhol, who helped me immensely by contributing to the project's development.
- I also thank CNPq, CAPES, and FAPEMIG for funding this work.

Thanks also to the entire VerLab team, and to PPGCC-UFMG, which also contributed a lot.

“Aos bons, que após o último aceno, choram pela alvura, com a qual seus frágeis atos bailariam numa verde baía: Lutem, lutem contra a luz cujo esplendor já não fulgura.”

(Dylan Thomas)

Resumo

Com a facilidade de obtenção de dispositivos portáteis como câmeras e smartphones, a gravação de vídeos em primeira pessoa vem se tornando um hábito comum. Esses vídeos normalmente são muito longos e cansativos de assistir, sendo necessárias edições manuais. Com isso, surgiram métodos de aceleração que buscam reduzir o tamanho desses vídeos, maximizando a estabilidade visual sem perder as informações relevantes e produzindo um vídeo acelerado agradável de assistir. Apesar do progresso recente dos métodos de aceleração, esses métodos não consideram a inserção da música de fundo nos vídeos. A inclusão da música de fundo pode tornar os vídeos acelerados ainda mais agradáveis, pois o usuário poderá assistir o vídeo acelerado combinado com sua música de interesse. Esta dissertação apresenta uma nova metodologia que cria vídeos acelerados e insere automaticamente a música de fundo, combinando as emoções induzidas pelas modalidades visuais e acústicas. Nosso método reconhece as emoções induzidas pelo vídeo e pela música ao longo do tempo, usando redes neurais artificiais, criando curvas de emoção para o vídeo e para a música, representadas no modelo de Russell, um modelo de representação da emoção usado na área de psicologia. Nosso método possui também um algoritmo de otimização que calcula as similaridades entre os quadros do vídeo e segmentos da música, criando uma matriz custo dinâmico e computando o caminho ótimo que alinha a curva de emoção do vídeo com a da música, preservando também a estabilidade visual e continuidade temporal do vídeo acelerado. Avaliamos o nosso método em um conjunto de vídeos e músicas com conteúdos e estilos variados, comparando-o quantitativamente e qualitativamente com outros métodos de aceleração de vídeo presentes na literatura. Os resultados mostram que nosso método atinge o melhor desempenho em maximizar a similaridade das emoções, aumentando-a significativamente na maioria dos casos, enquanto também mantém a estabilidade visual dos vídeos acelerados em comparação com os outros métodos da literatura.

Palavras-chave: Visão computacional, Reconhecimento de Emoção em Músicas, Reconhecimento de Emoção em Imagens, Hyperlapse Semântico.

Abstract

With the ease of obtaining portable devices such as cameras and smartphones, the recording of first-person videos has become a common habit. These videos are usually very long and tiring to watch, requiring manual edition. Thereby, fast-forward methods emerged seeking to reduce the size of these videos, maximizing the visual quality without losing the relevant information and producing an accelerated video that is pleasant to watch. Despite the recent progress of fast-forward methods, these methods do not consider inserting background music in the videos. Inserting background music can make accelerated videos even more pleasant, as the user will be able to watch the accelerated video combined with their music of interest. This thesis presents a new methodology that creates accelerated videos and automatically inserts the background music, combining the emotions induced by the visual and acoustic modalities. Our method recognizes the emotions induced by video and music over time, using artificial neural networks, creating emotion curves for video and music, represented in Russell's model, an emotion representation model widely used in psychology. Our method also has an optimization algorithm that calculates the similarities between video frames and music segments, creating a dynamic cost matrix and computing the optimal path that aligns the video's emotion curve with the music's emotion curve, preserving also the visual quality and temporal continuity of the accelerated video. We evaluated our method in a set of videos and songs with varied content and styles, comparing it quantitatively and qualitatively with other fast-forward methods present in the literature. The results show that our method achieves the best performance in maximizing the similarity of emotions, increasing it significantly in most cases, while also maintaining the visual quality of the accelerated videos compared to other methods in the literature.

Keywords: Computer Vision, Music Emotion Recognition, Image Emotion Recognition, Semantic Hyperlapse.

List of Figures

1.1	Video acceleration illustration	2
1.2	Russel’s valence-arousal plane	3
1.3	An overview of our methodology	4
2.1	Main stages of Kopf <i>et al.</i> [2014] method	8
2.2	Main stages of Joshi <i>et al.</i> [2015] method	9
2.3	An output frame produced by the Halperin <i>et al.</i> [2018] method	9
2.4	Overview of the method proposed by Ramos <i>et al.</i> [2016]	10
2.5	The interface proposed by Higuchi <i>et al.</i> [2017]	11
2.6	The pipeline of the algorithm proposed by Lai <i>et al.</i> [2017]	12
2.7	Main steps of the Furlan <i>et al.</i> [2018] methodology	13
2.8	Emojigrid emotion representation	14
2.9	Best features according to Panda <i>et al.</i> [2018]	15
2.10	Architecture presented by Chowdhury <i>et al.</i> [2019]	16
2.11	Valence-arousal estimations by the Thammasan <i>et al.</i> [2016] method	17
2.12	Network structure proposed by Dong <i>et al.</i> [2019]	18
2.13	Examples of figures with aesthetics scores	19
2.14	Emotion prediction examples obtained by Zhao <i>et al.</i> [2014]	20
2.15	Overview of the framework created by Borth <i>et al.</i> [2013]	21
2.16	Examples of adjective-noun pairs used by Borth <i>et al.</i> [2013]	21
2.17	Classification examples obtained by Mittal <i>et al.</i> [2020]	22
2.18	Overview of the Sasaki <i>et al.</i> [2013] system	22
2.19	Framework proposed by Hong <i>et al.</i> [2018]	23
3.1	Our full methodology overview	26
3.2	Detailed block diagram of the video emotion curve creation	28
3.3	Video emotion curve example	29
3.4	Illustration of conversion from Plutchik’s Wheel to EmojiGrid	31
3.5	Illustration of the video curves smoothing	31

3.6	Detailed block diagram of the music emotion curve creation	32
3.7	Illustration of the audio classification step	34
3.8	Annotation interface of the DEAM dataset for arousal label	35
3.9	Illustration of the song curves smoothing	35
3.10	Illustration of the 3D dynamic cost matrices	39
4.1	Confusion matrix for the image classifier	47
4.2	Confusion matrix for the music valence classifier	48
4.3	Confusion matrix for the music arousal classifier	48
4.4	Selected frames distribution example	51
4.5	Emotion classification example 1	54
4.6	Emotion classification example 2	54
4.7	Emotion classification example 3	55
4.8	Emotion classification example 4	55
4.9	Qualitative comparison example 1	57
4.10	Qualitative comparison example 2	57
4.11	Qualitative comparison example 3	58
4.12	Qualitative comparison example 4	58
4.13	Qualitative comparison example 5	59
4.14	Qualitative comparison example 6	59
4.15	Qualitative comparison example 7	60
4.16	Qualitative comparison example 8	60
4.17	Song selection result example 1	63
4.18	Song selection result example 2	63
4.19	Song selection result example 3	64
4.20	Song selection result example 4	64
4.21	Song selection result example 5	65
4.22	Song selection result example 6	65
4.23	Song selection result example 7	66
4.24	Song selection result example 8	66

List of Tables

2.1	Works summary	24
4.1	Audio-visual dataset	45
4.2	Optimal path selection evaluation	50
4.3	Comparison with baselines	53
4.4	Results with song selection	61

Contents

Acknowledgments	ix
Resumo	xiii
Abstract	xv
List of Figures	xvii
List of Tables	xix
1 Introduction	1
1.1 Context and Motivation	1
1.2 Problem Definition	4
1.3 Objectives	5
1.4 Contributions	5
1.5 Organization	6
2 Related Work	7
2.1 Hyperlapse	7
2.2 Semantic Hyperlapse	10
2.3 Music Emotion Recognition	13
2.4 Image Emotion Recognition	18
2.5 Music Recommendation for Video	21
2.6 Summary	24
3 Methodology	25
3.1 Overview	25
3.2 Emotion Curves Creation	28
3.2.1 Video Emotion Curve	28
3.2.2 Music Emotion Curve	31

3.3	Optimal Path Selection	36
3.3.1	2D Cost Matrices Construction	36
3.3.2	3D Dynamic Cost Matrix Construction	38
3.3.3	Optimal Path Traceback	40
3.4	Song Selection	41
4	Experiments	43
4.1	Implementation Details	43
4.1.1	Emotion Curves Creation	43
4.1.2	Optimal Path Algorithm	44
4.2	Experimental Setup	44
4.2.1	Dataset	44
4.2.2	Evaluation Metrics	45
4.3	Quantitative Evaluation	47
4.3.1	Emotion Classification	47
4.3.2	Optimal Path Selection	49
4.3.3	Comparison with Hyperlapse Creation Methods	52
4.4	Qualitative Evaluation	53
4.4.1	Emotion Classification Examples	53
4.4.2	Comparison with Hyperlapse Creation Methods	56
4.4.3	Results Using Song Selection	61
5	Conclusion and Future Works	69
5.1	Conclusion	69
5.2	Future Works	71
	Bibliography	73

Chapter 1

Introduction

This thesis covers two research fields: automatic emotion recognition from visual and acoustic data and the fast-forwarding of egocentric videos. This chapter introduces these two fields, presenting a context and motivation for the work and then defining the problem, objectives, and contributions.

1.1 Context and Motivation

In recent years, we have witness an increasing volume of audio-visual data on the Internet due to the ease in people's access and usage of new digital technologies. The cost of multimedia mobile devices such as wearable cameras and smartphones is constantly decreasing while their storage capacity increases. As a result, many people start recording videos in an egocentric perspective of their daily activities, referred to as egocentric videos, resulting in long and untrimmed streams. Usually, egocentric videos are tiring to watch since they contain redundant segments, and post-edition is commonly disregarded. Consequently, there has been a great interest in the computer vision community in reducing the videos' total length to speed up browsing and creating a pleasant watching experience. In the video acceleration process, many factors must be considered, such as the smoothness of the transitions between frames, the camera's stability, and the relevance of the content preserved in the accelerated video. Therefore, just speeding up the video uniformly can result in unstable and unpleasant videos to watch.

Over the past several years, many works have been proposed to create a shorter accelerated version of egocentric videos using different strategies and under various restrictions to reduce the burden of watching the videos entirely, such as Kopf *et al.* [2014] and Joshi *et al.* [2015]. The accelerated video is commonly called hyperlapse, in

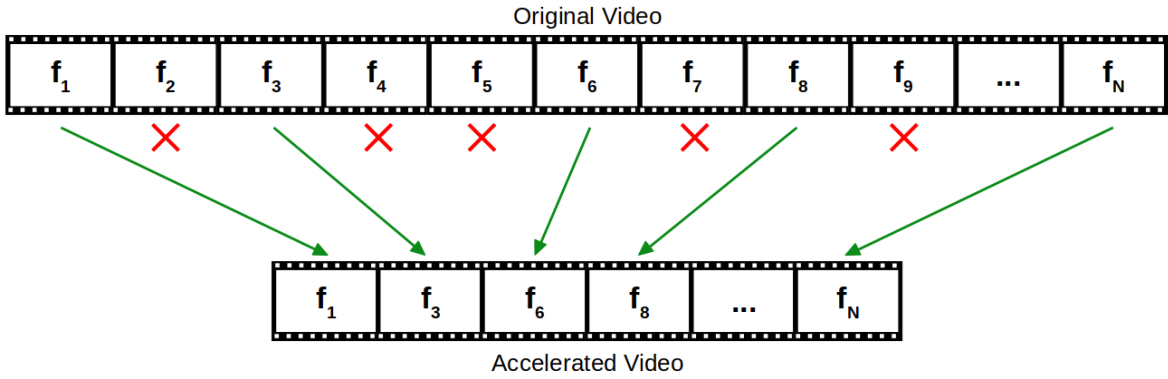


Figure 1.1. Video acceleration illustration. The video acceleration process consists of removing a subset of frames from the original video, creating a shortest video, seeking to maintain the visual video quality.

which the construction is a technique in time-lapse photography that allows creating motion shots, where the goal is to optimize the output number of frames and the visual smoothness [Silva *et al.*, 2018a]. The video acceleration process consists of removing a subset of frames from the original video, as illustrated in Figure 1.1, creating a shortest video, seeking to maintain the visual video quality.

An important extension of the traditional hyperlapse is the semantic hyperlapse, which includes the semantic relevance for each frame [Silva *et al.*, 2021], making the most important frames with lower acceleration rates than the other frames on the output video. As presented by Ramos *et al.* [2016], the goal is to accelerate the video based on the semantic extraction, generating a semantic curve that assigns scores to the frames, according to its information importance. Later works, such as Silva *et al.* [2018b], improved the results to solve other issues, such as the need to smooth out and avoid sharp cuts and transitions in the output video without losing relevant information. Furlan *et al.* [2018] gave importance also to the sound information in the video, calculating the psychoacoustic annoyance and accelerating the video based on psychoacoustic metrics.

Despite the advances, these works did not give attention to the background song that the user wants to include in the video. Both visual and sound streams play a significant role in the video watching experience, generating videos that include the current hyperlapse techniques usually overlook audio. Adding background music into an accelerated video based on its content is non-trivial.

Since we are interested in combining music with video, the contents of both must be associated in some way. However, to associate these contents, there are many challenges. It is necessary to compare and try to match video and music content over

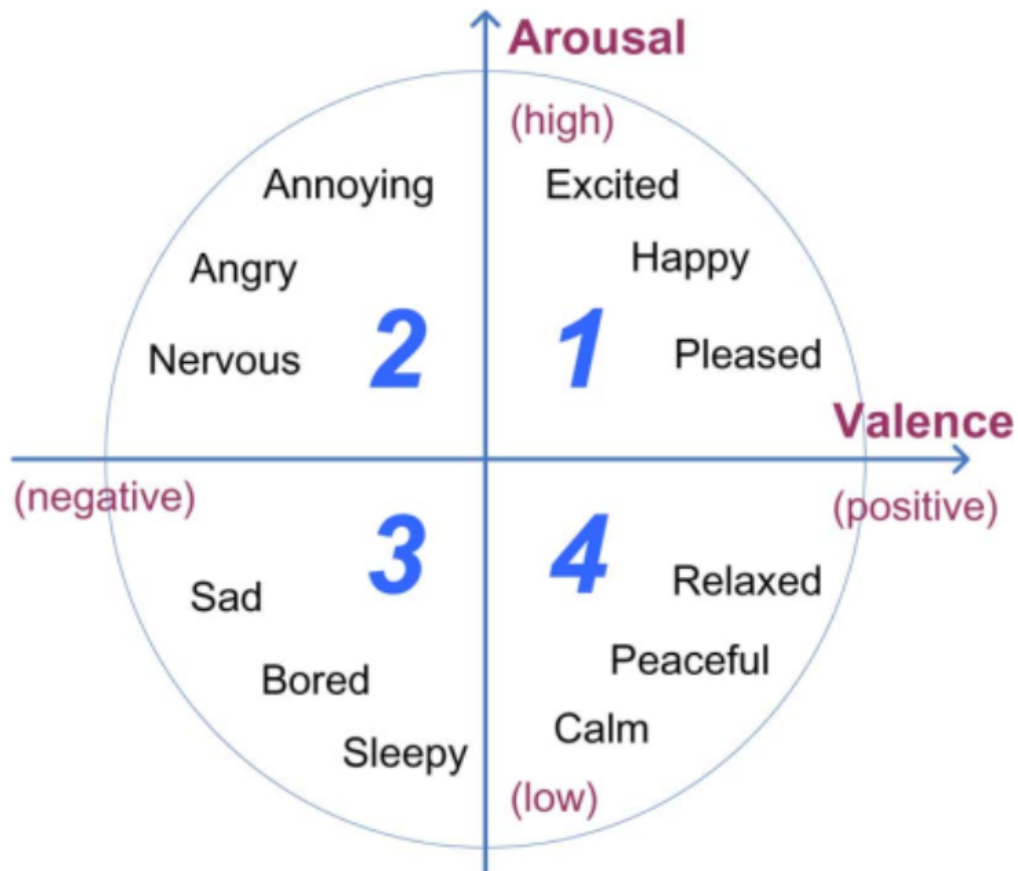


Figure 1.2. Russel’s valence-arousal plane. The x axis represents how good is the emotion (valence), and the y axis represents how exciting is the emotion (arousal). The center of the plane represents a neutral emotional state (extracted from Yang *et al.* [2008]).

time. The video acceleration must allow matching these contents without harming the visual quality of the video, concerning the transition smoothness and the camera stability. One way to combine music and video content is considering different emotions induced by both to produce a final video that maintains the video and music’s emotion similarity. The emotions estimated from images and from music can be used to align the scenes of the video with the excerpts of the music that arouse similar emotions in whoever is going to see the video and listen to the music simultaneously.

Music plays an essential role in society, especially in our digital age. Since many music files are scattered across storage media, a need has arisen to classify them by different emotions. There are many works in music emotion recognition, such as Chowdhury *et al.* [2019] and Dong *et al.* [2019], which consist of estimating the induced emotion by a specific piece of music. A classic representation of emotion is given by Russell’s model [Yang *et al.*, 2008], showed in Figure 1.2, where songs are classified

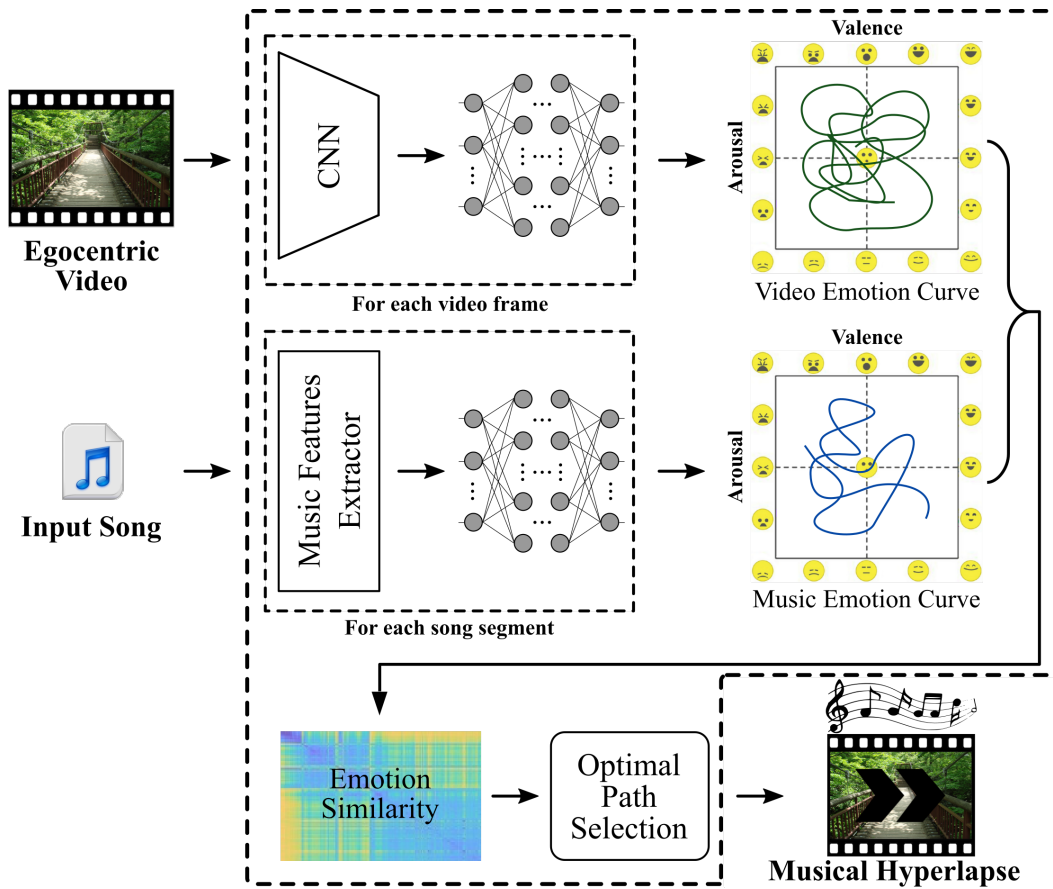


Figure 1.3. An overview of our methodology. Our method computes the emotion similarity between the video and the song after creating curves in the valence-arousal plane. It accelerates the input video by removing frames according to an optimization algorithm that seeks the best matches between the video and the song.

with different labels around two axes: the valence and the arousal. By using this representation, it is possible to represent emotions such as angry, happy, calm, bored, etc., numerically. Since images also affect our affective states, we can apply the same model to classify the emotions induced by images and use this audio-visual classification to synchronize an input video with background music when accelerating the video.

1.2 Problem Definition

In this thesis, we introduce a novel problem called *Musical Hyperlapse*, where the goal is to accelerate a video to the length of a song while matching the visual and audio signals to trigger, continuously, the same emotions during the output video exhibition. To tackle this problem, we propose a new multi-modal method to create hyperlapse

videos based on synchronizing the feelings in the video scenes and background song segments. Specifically, given the predicted continuous emotion curves for the video and audio streams, our approach seeks the best set of frames to be discarded in the video stream restricted to preserving the smoothness in the visual continuity and the matching between the emotion induced by segments of video and audio. Our approach assumes that the accelerated video will only have visual content from the original video and a background song that the user wants to hear along with the accelerated video. Figure 1.3 shows a simplified overview of our proposed method.

1.3 Objectives

The objective of this thesis is, given a first-person recorded video and a background song (selected by the user), accelerate the video to the length of the song seeking to maximize the similarity of the emotions induced by both over time, also maximizing the visual quality of the output video. Additionally, if there is a list of songs in the input (selected by the user), choose the song that best matches with the video. We can also define the specific objectives, listed below:

- Generate continuous emotion curves for the input video and input song;
- Accelerate the video by combining the song and video's emotion curves, also maximizing the video's visual quality;
- Evaluate the results by measuring the similarity of the produced curves, as temporal series, and also measuring the visual quality of the video.

1.4 Contributions

The contributions of this work can be summarized as follows:

- New models for automatic music and image emotion recognition;
- A novel optimization algorithm to create hyperlapse videos whose function is to reduce the video by matching its emotion curve to the music emotion curve;
- A new dataset comprising eight first-person videos and five songs with different genres, sizes, and styles.

Portions of this work have been published in the SIBGRAPI proceedings:

- Matos, D.; Ramos, W.; Romanhol, L.; Nascimento, E. R.. Musical Hyperlapse: A Multimodal Approach to Accelerate First-Person Videos. In: Conference on Graphics, Patterns and Images, 34. (SIBGRAPI), 2021, Gramado (Virtual), Brazil.

1.5 Organization

We have organized this thesis into five chapters. Regardless of this introduction, the remaining of this document is presented as follows:

- **Chapter 2 - Related Work:** discusses relevant work in the area of emotion recognition and video acceleration;
- **Chapter 3 - Methodology:** presents details about our proposed method to address the problem;
- **Chapter 4 - Experiments:** presents quantitative and qualitative experiments performed to validate our method and shows several results;
- **Chapter 5 - Conclusions:** closes this thesis with our conclusions and directions for future work.

Chapter 2

Related Work

In this chapter, we present the works most related with our approach. The chapter is divided into four parts. The first part comprises the Hyperlapse, where we present works in the context of accelerating videos. The second part comprises the Semantic Hyperlapse, where we present works with recent methods to accelerate videos based on semantic information. The third part comprises the Music Emotion Recognition, where we present works that classify music by its induced emotions in humans. And the fourth part comprises the Image Emotion Recognition, where we present works about automatic recognition of emotions in images. We also present works in the field of music recommendation for video.

2.1 Hyperlapse

Over the past decade, hyperlapse methods have been proposed to reduce the length of long egocentric videos. The evolution of these works is focused on improving the quality of the output video by keeping it as smooth and stable as possible, with the desired short length, and without losing relevant information, which is usually the content of interest defined by user.

Kopf *et al.* [2014] present a classical work in creating hyperlapse from first-person videos. The video is accelerated by using techniques based on image rendering, such as projecting, stitching, and blending after the optimal trajectory of the camera poses is computed. As a drawback, their method has a high computational cost and requires camera motion and parallax to compute the 3D model of the scene. Figure 2.1 show the main stages of Kopf *et al.* method for creating hyperlapses: (a) 3D camera and point cloud recovery, followed by smooth path planning; (b) 3D per camera proxy estimation; and (c) source frame selection and Poisson blending.

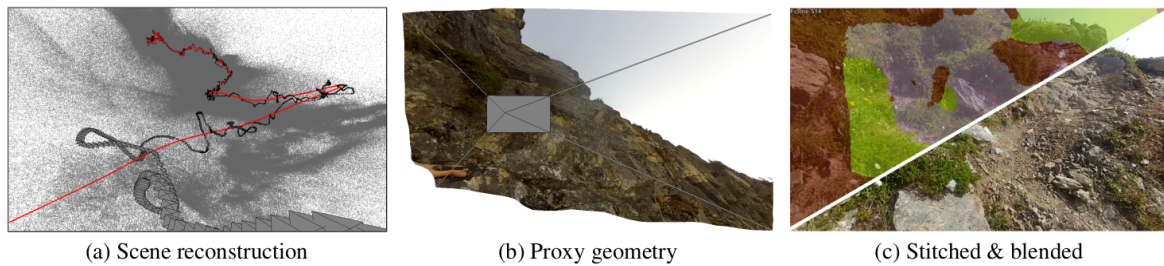


Figure 2.1. Main stages of Kopf *et al.* [2014] method. Their system convert first-person videos into hyperlapse summaries using a set of processing stages: (a) 3D camera and point cloud recovery, followed by smooth path planning; (b) 3D per camera proxy estimation; (c) source frame selection and Poisson blending (extracted from Kopf *et al.* [2014]).

The Instagram Hyperlapse App [Karpenko, 2014] is an approach that creates hyperlapse videos by combining video stabilization and the phone gyroscope. The method have the limitation of needing inertial data, making it unfeasible when recording videos using a standard camera. Poleg *et al.* [2015] present another method to create classical hyperlapse videos using a graph to model the frame selection. In the graph, the nodes represent the frames of the input video and the edge weights between pair of nodes represent the cost of including the pair of frames sequentially in the accelerated video. By this way, they create the accelerated video finding the shortest path in the graph.

Joshi *et al.* [2015] presented a real-time hyperlapse creation algorithm. The algorithm uses feature tracking to recover the camera motion and compute the optimal path with an algorithm inspired by dynamic programming and Dynamic Time Warping (DTW). Figure 2.2 shows the pipeline of their algorithm. The first step is the frame matching: using sparse feature-based techniques, they estimate how well each frame can be aligned to its temporal neighbors and store these costs as a sparse matrix. The second step is the frame selection: a dynamic programming algorithm finds an optimal path of frames that balances matching a target rate and minimizes frame-to-frame motion. The third step is the path smoothing and rendering: given the selected frames, smooth the camera path and render the final hyperlapse result.

Halperin *et al.* [2018] present an adaptive frame sampling to create stable fast-forwarded videos. They formulate the adaptive frame sampling as an energy minimization problem, which can find the optimal solution in polynomial time. They reduce the perception of instability by enlarging each of the input frames with neighboring frames. As shown in Figure 2.3, the authors' method looks into different directions and collect frames from the input video, creating mosaics around each frame. The mosaics are

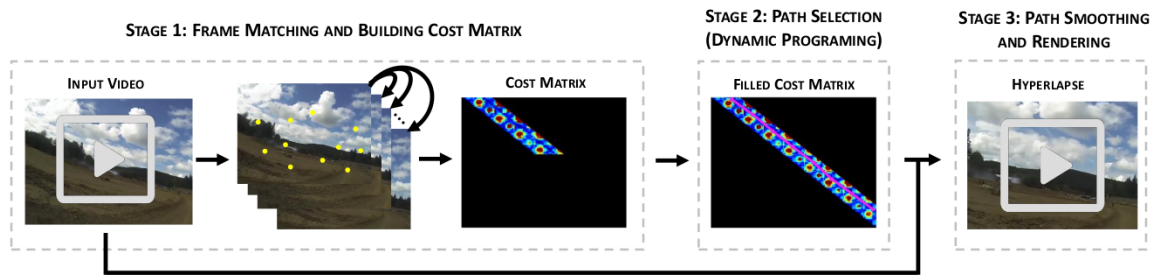


Figure 2.2. Main stages of Joshi *et al.* [2015] method. 1) Frame matching: using sparse feature-based techniques, they estimate how well each frame can be aligned to its temporal neighbors and store these costs as a sparse matrix. 2) Frame selection: a dynamic programming algorithm finds an optimal path of frames that balances matching a target rate and minimizes frame-to-frame motion. 3) Path smoothing and rendering: given the selected frames, smooth the camera path and render the final hyperlapse result (extracted from Joshi *et al.* [2015]).



Figure 2.3. An output frame produced by the Halperin *et al.* [2018] method. They look into different directions and collect frames from the input video, creating mosaics around each frame. The mosaics are sampled to meet playback speed and video stabilization requirements. The different original frames are marked with white lines (extracted from Halperin *et al.* [2018])

sampled to meet playback speed and video stabilization requirements. Their method can also generate a single hyperlapse video from multiple egocentric videos.

Our work shares similarities with the work of Joshi *et al.*, since our optimal path selection also draws inspiration from dynamic programming. However, different from Joshi *et al.*, which considers only the visual modality during their optimization process, we handle two modalities: the input visual stream and the output audio stream.

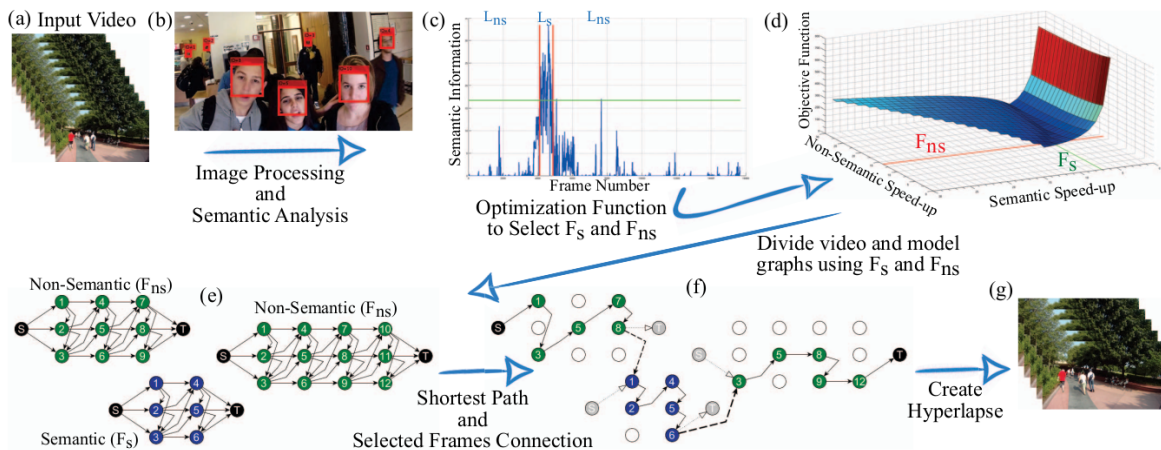


Figure 2.4. Overview of the method proposed by Ramos *et al.* [2016]. (a) Input is a first-person recorded video (egocentric video). (b) A semantic image content analysis is made to determine the relevance of each frame. (c) An optimization function is generated, separated into semantic and non-semantic types. (d) Different speed-up is assigned to semantic and non-semantic parts. (e) The frames are represented with graphs. (f) The shortest path algorithm is used to select the optimal video subset of frames. (g) The output is a semantic hyperlapse video (extracted from Ramos *et al.* [2016]).

2.2 Semantic Hyperlapse

Recent approaches in creation of fast-forwarded videos include visual semantics as part of the process. These methods, referred to as semantic hyperlapse, aim to accelerate the input video, optimizing camera stability, target speed-up rate, and semantics.

Okamoto and Yanai [2014] proposed a method to summarize egocentric moving videos, generating a walking route guidance video. They analyze the video by detecting pedestrian crosswalk and ego-motion classification, estimating importance scores for each video session, based on the detected contents. Their method control the video speed dynamically, instead of generating a summarized video file. They outperform a single summarization method in their experiments.

Ramos *et al.* [2016] introduced a new adaptive frame sampling process that considers the semantic information during the optimization. Their method is shown in Figure 2.4. Their approach assigns a semantic score for each video frame and split the video into temporal segments according to their relevance. The authors applied different playback rates such that more relevant segments are exhibited at a lower rate. Their optimization balances the semantics and traditional hyperlapse objectives using energy cost minimization in a graph representing the frames' transition. The work of

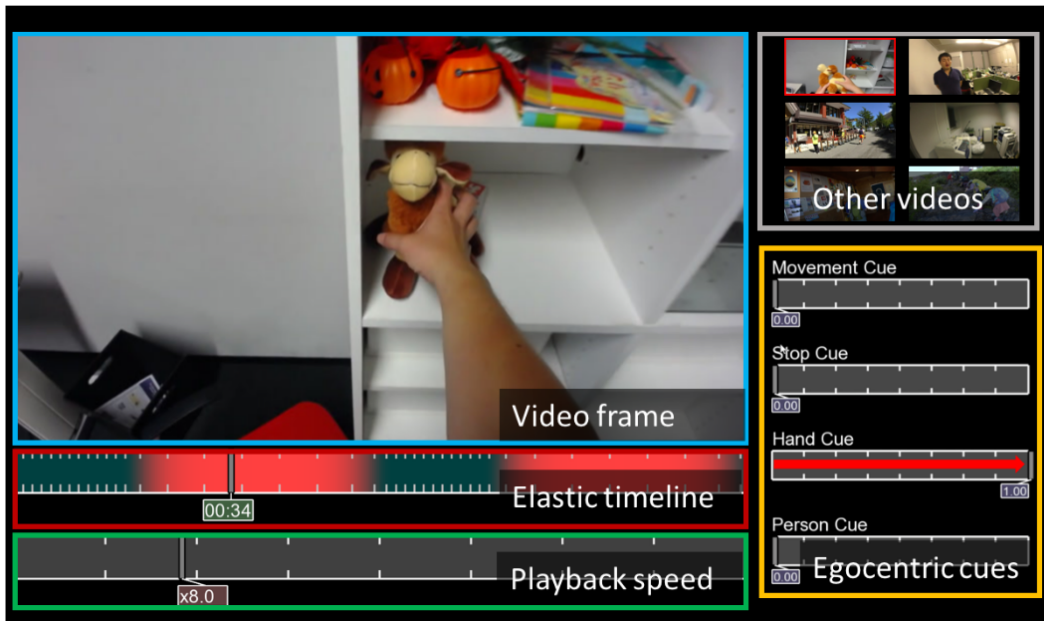


Figure 2.5. The interface proposed by Higuchi *et al.* [2017]. Salient parts of the video are emphasized by the elastic timeline. With this interface, the users can input which of such cues are relevant to their events of interest. The red arrow in the bottom right indicates a user’s input. In this example, the interface emphasizes hand-related events (extracted from Higuchi *et al.* [2017]).

Ramos *et al.* was later extended by Silva *et al.* [2016], where a homography-based stabilization was included in the process.

Yao *et al.* [2016] focused on the discovery of major or special user interest (highlights) in the input video. They used deep learning techniques to learn the relationship between highlight and non-highlight video parts. They associated low speed-ups for highlighted parts and high speed-ups for non-highlighted parts to perform the video summarization. Confirming the relevance of considering the semantic information, Higuchi *et al.* [2017] presented the EgoScanning, an interface to users find important events in long egocentric videos, shown in Figure 2.5. The interface allows users to input relevant cues to their events of interest. Using uniform sampling in the speed-up selected by the user, the remainder of the video is played faster.

Ogawa *et al.* [2017] proposed a fast-forwarding method for 360° videos (omni-directional videos). They used an adaptive subsampling scheme that selects optimal frames by minimizing a cost function based on 3D camera positions. Their approach successfully generate a subsampled and stable 360° video. Lai *et al.* [2017] presented a system capable of converting a 360° video into a normal field-of-view hyperlapse. Figure 2.6 shows the pipeline of their algorithm. After determining the per-frame viewing directions to the regions of interest, their approach produces a saliency-aware frame

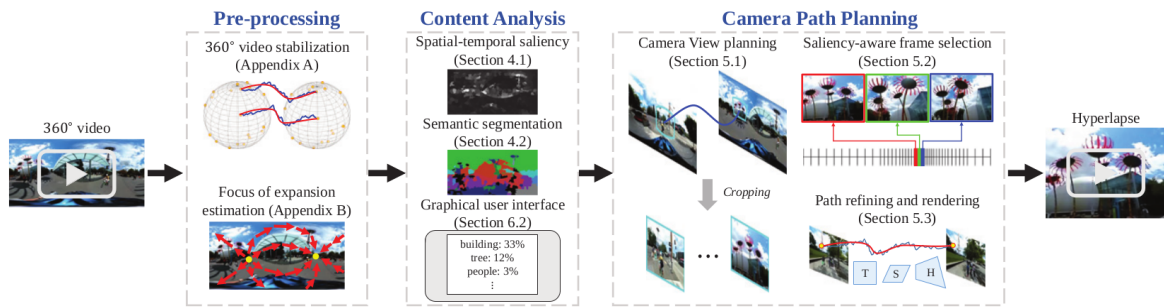


Figure 2.6. Pipeline of the algorithm proposed by Lai *et al.* [2017]. Given a 360° video, they first stabilize the video sequence. Then, estimate the focus of expansion as prior information for the camera path planning. To extract the regions of interest, they compute the spatial-temporal saliency and semantic segmentation. The detected regions of interest are used to guide the camera path planning. Finally, an adaptive 2D video stabilization is used to render a smooth hyperlapse (extracted from Lai *et al.* [2017]).

selection that considers denser sampling at attractive regions and attending the target speed-up rate.

Silva *et al.* [2018b] and Silva *et al.* [2021] modeled the adaptive frame sampling as a weighted minimum sparse reconstruction problem. Similar to the work of Ramos *et al.*, Silva *et al.* split the video temporally using frame-wise levels of relevance. Then, each segment is represented as a dictionary from which the output video frames are sparsely selected, aiming to reduce abrupt camera motions.

Unlike previous works, which are mainly focused on visual information, Furlan *et al.* [2018] proposed to use the input sound information. The main steps of their methods are shown in Figure 2.7. Their approach uses psychoacoustic metrics extracted from the video soundtrack to set the frames’ importance. The original video’s soundtrack is segmented, and for each segment, the Psychoacoustic Annoyance (PA) [Zwicker and Fastl, 2013] is computed. The PA values guide the semantic hyperlapse creation since they are used as semantic scores. Although using the source audio in the optimization process, Furlan *et al.* ignored the audio in the output video, making their problem fundamentally different from ours. Our main goal is to create a hyperlapse video with background music where both visual and acoustic signals induce similar emotions during the exhibition.

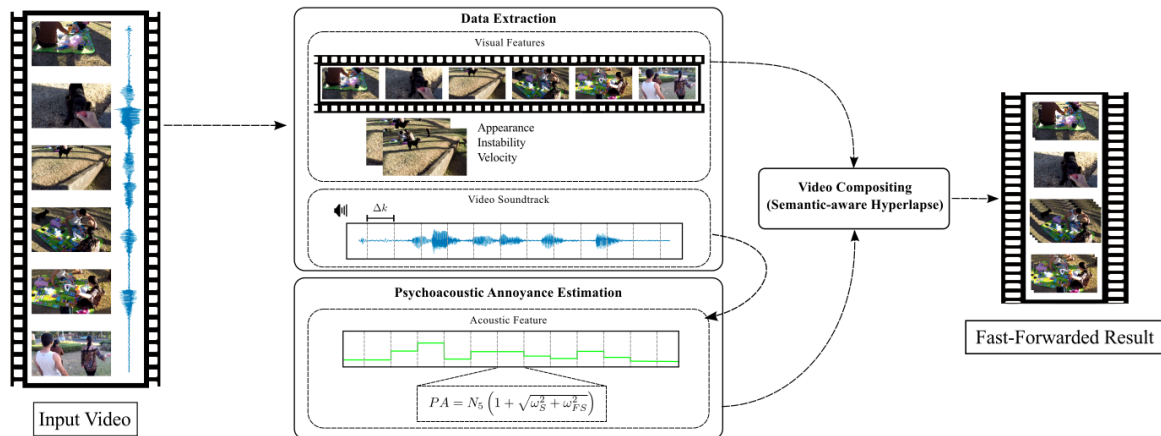


Figure 2.7. Main steps of the Furlan *et al.* [2018] methodology. After segmenting the video soundtrack into slices, they compute the PA metric (green curve), which is a semantic score assigned to each segment. The semantic score is used to create a relevant profile of the video used in the video compositing step to select the relevant frames (extracted from Furlan *et al.* [2018]).

2.3 Music Emotion Recognition

Significant progress has been made by researchers in the field of music emotion recognition. Some of these works aim to classify an entire song to a specific emotion, such as happy, sad, and angry, [Yang *et al.*, 2008; Panda *et al.*, 2018; Chowdhury *et al.*, 2019]. Others focus on the prediction of arousal and valence emotional values from segment-wise continuous features extracted from the song [Lu *et al.*, 2006; Thammasan *et al.*, 2016; Dong *et al.*, 2019].

Lu *et al.* [2006] present a research about different mood models and features. According to the authors, music features such as rhythm, melody, harmony, pitch, and timbre play an essential role in human physiological and psychological functions, altering their mood. With these features, the music mood can be divided into different types of moods. Some of these features, precisely the intensity, timbre, pitch, and rhythm, are acoustic features.

In music emotion recognition, a commonly used model is the Russell’s model [Alpher, 1980], that allows us to represent emotions in a 2D plane. The Russell’s valence-arousal emotion plane is showed in Figure 1.2, where the x axis is the valence, which represents how pleasant is the feeling, and the y axis is the arousal, which represents how exciting is the feeling.

An alternative emotion model is the EmojiGrid [Toet and van Erp, 2019], which uses emojis on the edges of the plane to indicate emotions. Figure 2.8 shows the space

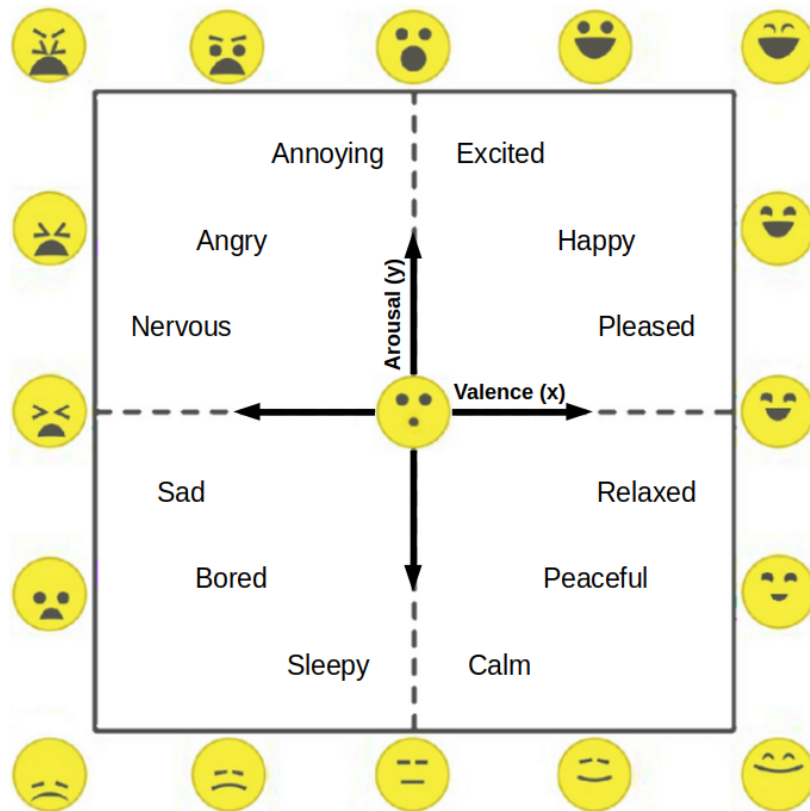


Figure 2.8. Emojigrad emotion representation. As in the Russell’s model, the x axis represents how good is the emotion (valence), and the y axis represents how exciting is the emotion (arousal). The center of the plane represents a neutral emotional state (extracted from Toet and van Erp [2019]).

representation considering valence and arousal as dimensions of emotions and extreme locations are represented with emojis. As in Russell’s model, the x axis represents how pleasant the emotion is, and the y axis represents how exciting is the emotion. The center of the plane represents a neutral emotional state.

Yang *et al.* [2008] formulated the musical emotion recognition as a regression problem to predict the valence and arousal values of the music samples. For each music sample, they extract features and use two regressors to predict the labels, one for valence and one for arousal. Thus, each music sample results on a point in the valence-arousal plane, and then the users can obtain the music sample by specifying a desired point in the plane. They apply principal component analysis to reduce the correlation between arousal and valence, reducing the processing time.

Panda *et al.* [2018] introduced another approach to generate audio features to improve the classification performance. They reviewed the existing audio features obtained in the literature and their relationships with the musical concepts, which are

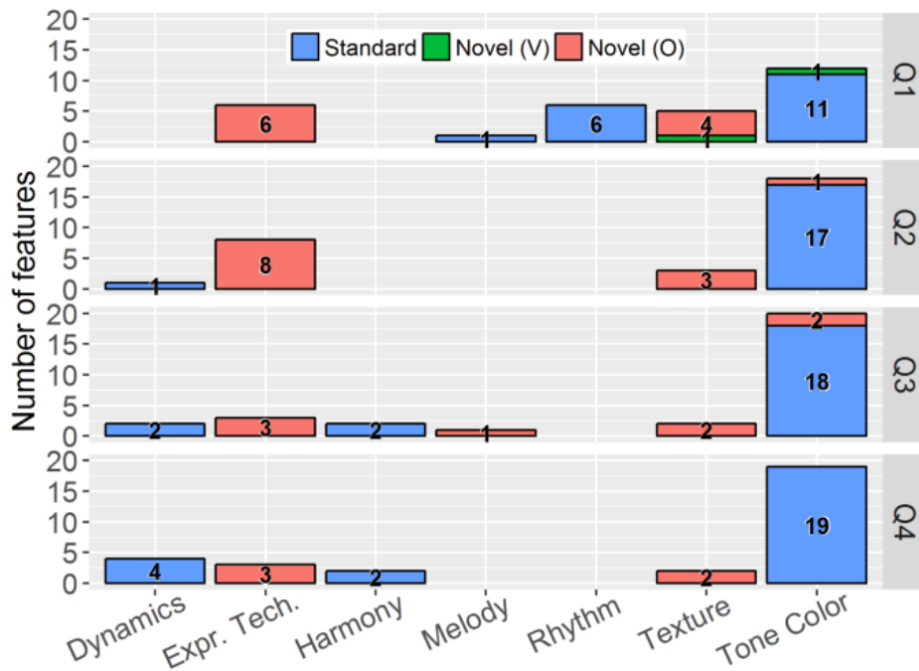


Figure 2.9. Best features according to Panda *et al.* [2018]. The best 30 music features to discriminate each quadrant, organized by musical concept. The Novel (O) group contains features that are extracted from the full original audio signal. The Novel (V) group contains features that are extracted from the voice-separated signal. Q1 to Q4 are the quadrants in the valence-arousal plane (Q1 - High valence, high arousal; Q2 - Low valence, high arousal; Q3 - Low valence, low arousal; Q4 - High valence, low arousal) (extracted from Panda *et al.* [2018]).

characteristics of a sound that defines it as a piece of music. New musical concepts were uncovered related to musical texture and expressive techniques. They rely on clues like melodic lines, notes, intervals, and scores to access higher-level musical concepts such as harmony, melody, articulation, or texture. They also gave importance to the determination of musical notes, frequency, and intensity contours mechanisms to capture the music information. Figure 2.9 shows the best features to discriminate each quadrant in valence-arousal plane, according to Panda *et al.* [2018]. The Novel (O) group contains features that are extracted from the full original audio signal. The Novel (V) group contains features that are extracted from the voice-separated signal. Q1 to Q4 are the quadrants in the valence-arousal plane (Q1 - High valence, high arousal; Q2 - Low valence, high arousal; Q3 - Low valence, low arousal; Q4 - High valence, low arousal).

Most recently, Chowdhury *et al.* [2019] aimed to create a model to give a musically meaningful and intuitive explanation for its predictions. They proposed a VGG-style deep neural network to obtain emotional features from a music piece through

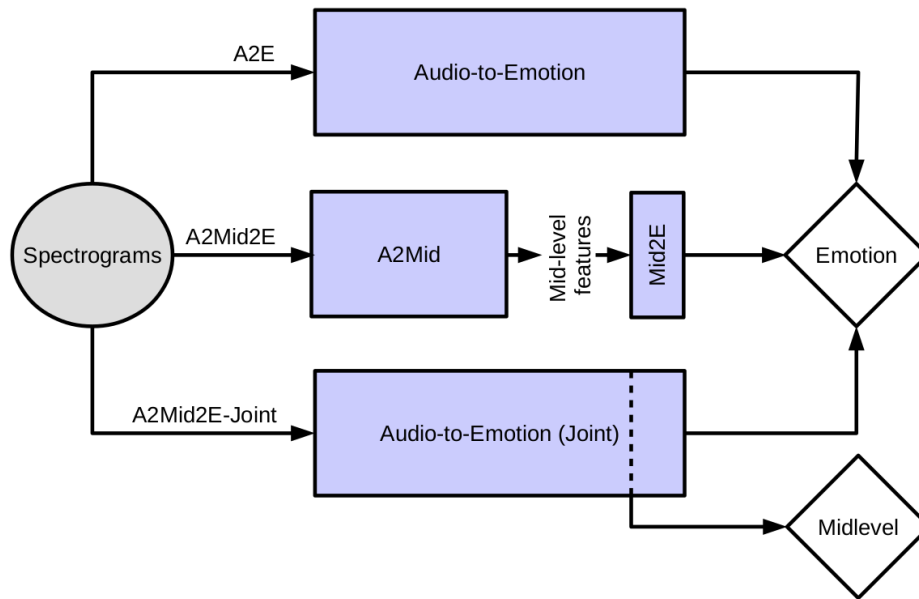


Figure 2.10. Architecture presented by Chowdhury *et al.* [2019]. The three different architectures proposed by Chowdhury *et al.* [2019] for predicting emotion from audio: A direct audio to emotion scheme (A2E), an audio to mid-level and mid-level to emotion scheme (A2Mid2E), and an audio to emotion and to mid-level jointly (A2Mid2E-Joint) scheme (extracted from Chowdhury *et al.* [2019]).

human interpretable mid-level perceptual features, using the audio spectrogram as input. These are features easily perceived and recognized by most listeners, without any music-theoretical training, such as melodiousness, rhythmic stability, and dissonance. Their approach is compared with another identical network but without considering the mid-level features, observing that the average loss with mid-level features is surprisingly low. Their system also allowed to visualize the effects of perceptual features on individual emotion predictions, concluding that the slight loss is given by the gain in the explainability of the projections. Figure 2.10 shows the three different architectures proposed by Chowdhury *et al.* [2019] for predicting emotion from audio: A direct audio to emotion (A2E), an audio to mid-level and mid-level to emotion scheme (A2Mid2E), and an audio to emotion and to mid-level jointly (A2Mid2E-Joint) scheme.

Several researchers also seek to predict arousal and valence emotional values from segment-wise continuous features extracted from the song. Thammasan *et al.* [2016] proposed a continuous music emotion recognition approach based on brainwave signals from the music listeners to obtain the audio features. Their experiment included self-reporting and continuous emotion annotation in the valence-arousal space. They used the Fractal Dimension [Li, 2002] and power spectral density [Rani, 2016] to extract

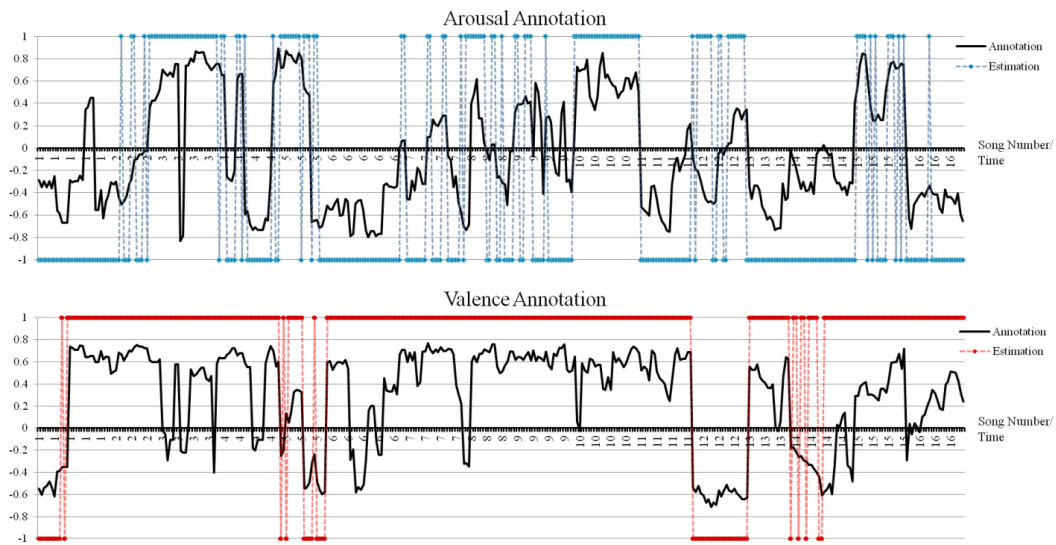


Figure 2.11. Valence-arousal estimations by the Thammasan *et al.* [2016] method. The valence and arousal annotations from a specific subject and their estimation by the model constructed with SVM and FD values from all instances. The horizontal axis represents the order of the selected songs by the subject. Data are plotted in time order (extracted from Thammasan *et al.* [2016]).

informative features from Electroencephalography (EEG) signals. They then applied it to classification algorithms to discriminate binary classes of emotion. However, their approach only classifies arousal and valence in two classes: high/low arousal and high/low valence, as shown in Figure 2.11.

A relevant work on literature is presented by Dong *et al.* [2019], where the songs are classified continuously on the arousal-valences plane with segments of 0.5 seconds. Figure 2.12 shows their network structure. They implemented a bidirectional convolutional recurrent sparse network (BCRSN) for music emotion recognition, based on convolutional neural networks (CNNs) applied to the audio spectrogram, and recurrent neural networks (RNNs). Their model adaptively learns the sequential information included affect salient features (SII-ASF) from the spectrogram of the song segments. Thus, their model can achieve continuous emotion predictions of audio files. Moreover, they propose a weighted hybrid binary representation (WHBR) method that converts the regression prediction process into a weighted combination of multiple binary classification problems, reducing the computational complexity. They also applied a PCA on the spectrograms to reduce the computational load without losing important information. Their model was evaluated with the DEAM dataset Solymani *et al.* [2018], outperforming the state of the art.

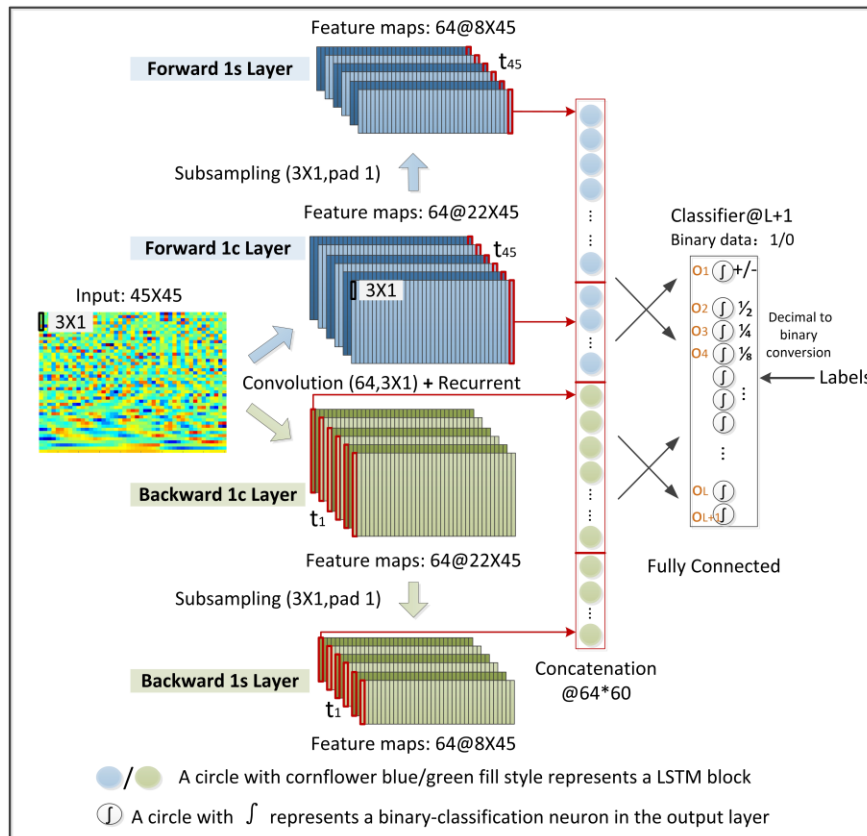


Figure 2.12. Network structure proposed by Dong *et al.* [2019]. The input is a 2D spectrogram. The local connectivity and parameter sharing architecture is used to replace the connection between the input and hidden (i.e., the forward and backward 1c layer) layers of the model at each frame t_i . The bidirectional recurrent is set at $t_i - t_{i+1}$ to transfer the sequential information. The subsampling operation obtains more advanced features. The output layer is fully connected with the neurons in the last frame t_{45} of the feature maps of the forward 1c/s layer and t_1 of the backward 1c/s layer. $L + 1$ binary classifiers with different weights are used for the final emotion prediction (extracted from Dong *et al.* [2019]).

2.4 Image Emotion Recognition

In the Affective Sciences, detection of emotion from scenes and from facial expressions are some of the essential tasks [Toet and Erp, 2019]. Datasets relating images to emotions such as GAPED [Dan-Glauser and Scherer, 2011] have been created for research purposes both for attention and emotion. In image emotion recognition, the problem consists of retrieving the emotional content automatically from a given image. In general, the categorization of such images is made upon human annotation or automatically by using learned representations that rely on both high and low-level features.

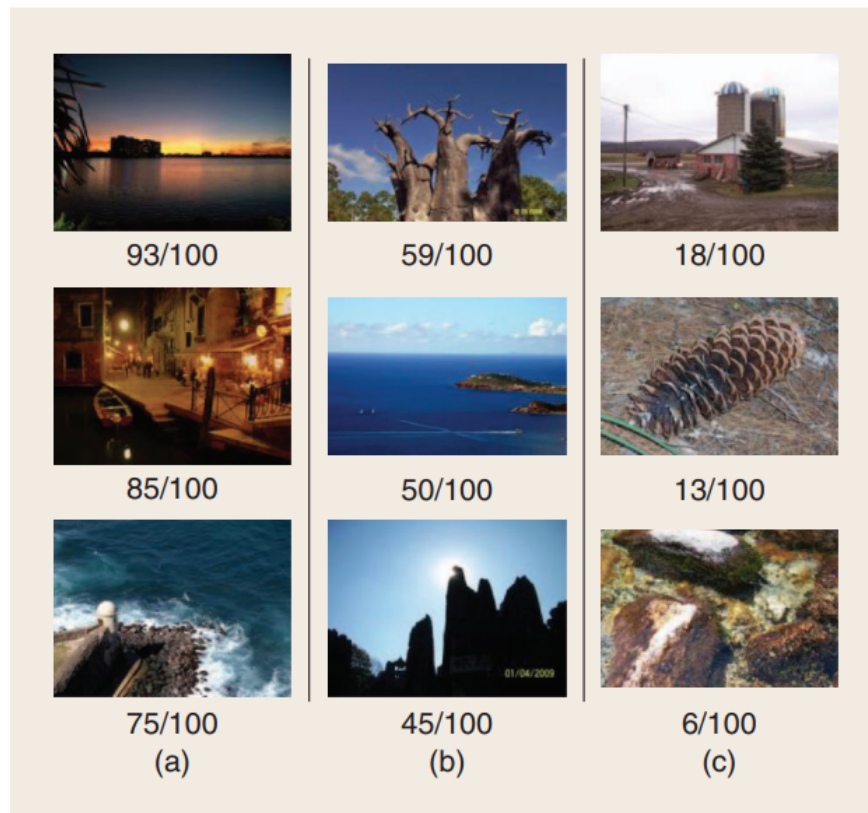


Figure 2.13. Examples of figures with aesthetics scores. Pictures with (a) high, (b) medium, and (c) low aesthetics scores from the Aesthetic Quality Inference Engine (extracted from Joshi *et al.* [2011]).

Joshi *et al.* [2011] and Zhao *et al.* [2014] explored the use of psychology and art-theory knowledge to determine which emotions may be evoked by a picture. However, as shown by Jia *et al.* [2012], the use of high-level features like social network data when analyzing images is much more effective than raw low-level features such as primary colors in the image. Figure 2.13 shows examples of aesthetics scores, which are scores assigned to images based on its artistic characteristics, used in the work of Joshi *et al.* [2011]. Figure 2.14 shows examples of emotion prediction obtained by Zhao *et al.* [2014], where the black plus signs and blue circles represent the ground truth and the predicted values of image emotions, respectively.

Descriptive data also play a role in several solutions to recognizing the emotion induced by the image. For instance, the work of Borth *et al.* [2013] uses pairs of adjectives and nouns to classify each picture. Figure 2.15 shows an overview of the framework proposed by Borth *et al.* [2013]. They use each of the 24 emotions defined in Plutchik's theory [Plutchik, 1980] to derive search keywords and retrieve images and videos from FLickr and Youtube. Tags associated with the retrieved images and videos

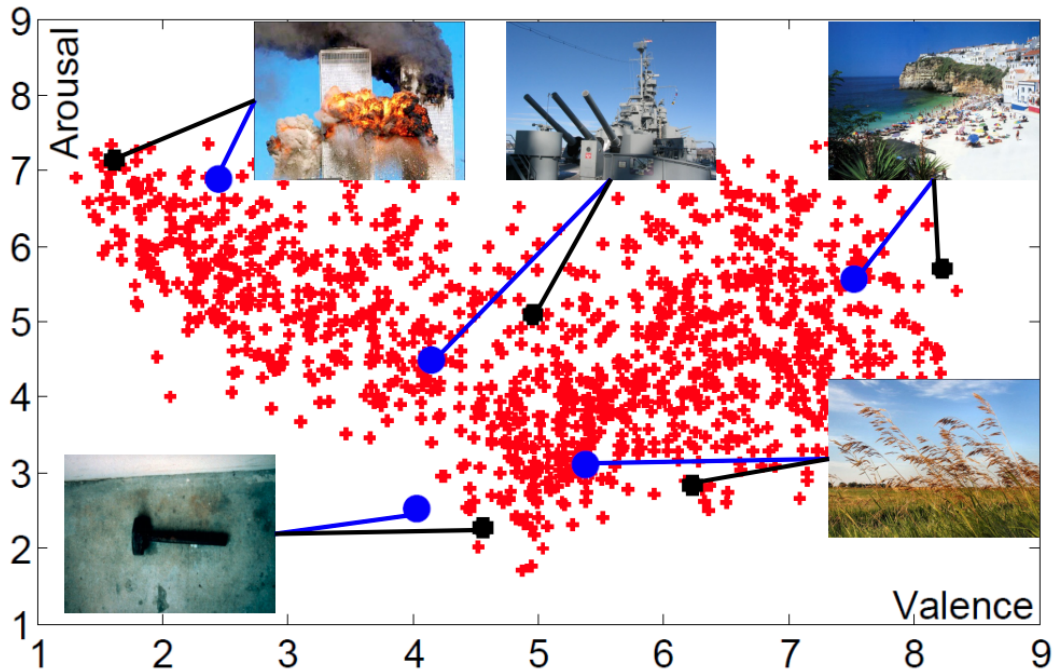


Figure 2.14. Emotion prediction examples obtained by Zhao *et al.* [2014]. Emotion prediction results of the Zhao *et al.* [2014] method. The black plus signs and blue circles represent the ground truth and the predicted values of image emotions, respectively (extracted from Zhao *et al.* [2014]).

are extracted, and then analyzed to assign sentiment values and to identify adjectives, verbs, and nouns. Then, the adjectives and nouns with strong sentiment values are used to form adjective-noun pairs (ANP). Individual detectors are trained using Flickr images to detect the ANP of each image. They apply SentiBank and train classifiers to predict sentiment values of the images, based on the image ANP. Some examples of adjective-noun pairs are shown in Figure 2.16.

Mittal *et al.* [2020] take a wider range of objects in the scene to later sort the most important ones regarding the induced emotion. Examples of classifications obtained by them are shown in Figure 2.17, each from the EMOTIC dataset (left) and GroupWalk Dataset (right), respectively.

Despite the progress of both emotion induced by music and images, it is worth noting that none of the works investigate the interplay between acoustic and visual signals regarding the induced feeling. Conversely, in this thesis, we propose to apply both visual and acoustic data to accelerate an input video by aligning video segments with the emotion induced by the frames and music.

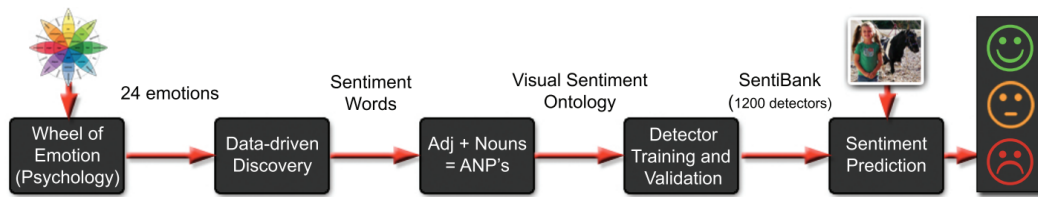


Figure 2.15. Overview of the framework created by Borth *et al.* [2013]. An overview of the proposed framework for constructing the visual sentiment ontology and SentiBank. They use each of the 24 emotions defined in Plutchik’s theory to derive search keywords and retrieve images and videos from Flickr and Youtube. Tags associated with the retrieved images and videos are extracted, and then analyzed to assign sentiment values and to identify adjectives, verbs, and nouns. Then, the adjectives and nouns with strong sentiment values are used to form adjective-noun pairs (ANP). Individual detectors are trained using Flickr images to detect the ANP of each image. They apply SentiBank and train classifiers to predict sentiment values of the images, based on the image ANP (extracted from Borth *et al.* [2013]).

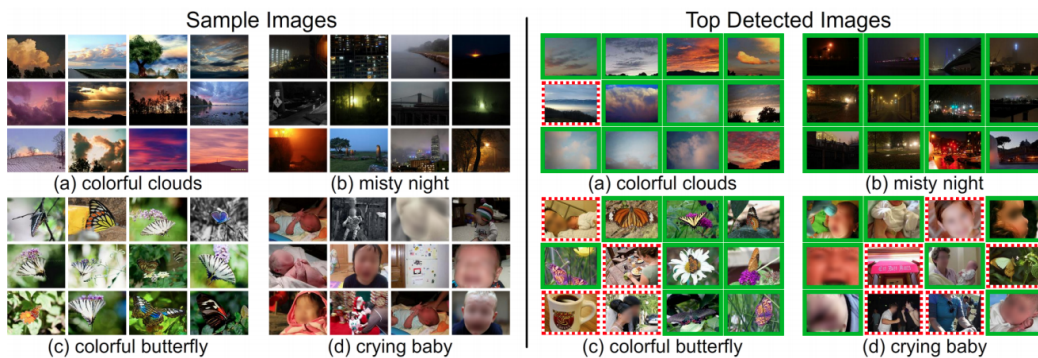


Figure 2.16. Examples of adjective-noun pairs used by Borth *et al.* [2013]. Left: Selected images for four sample adjective-noun pairs, (a),(c) reflecting a positive sentiment, and (b), (d), a negative one. Right: top detected images by SentiBank ANPs with high detection accuracy (top) and low accuracy (bottom). Correct detections are surrounded by green and thick frames and incorrect ones by red and dashed frames. Faces in the images are blurred (extracted from Borth *et al.* [2013]).

2.5 Music Recommendation for Video

There are many works to recommend background music for videos automatically. In general, these works aim to correlate the features of the songs with the features of the videos, with different approaches.

Kuo *et al.* [2013] proposed a framework to recommend background music for videos based on the multi-modal latent analysis between video and music. They related audiovisual features extracted from the videos and audio features extracted from

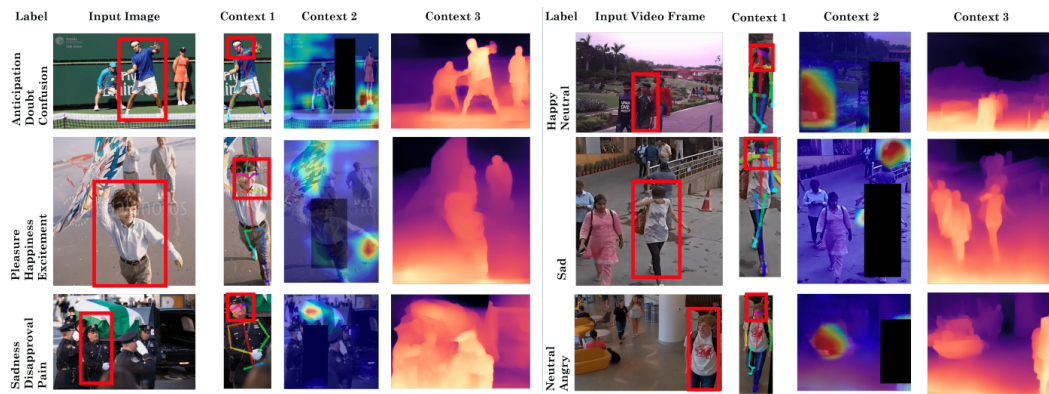


Figure 2.17. Classification examples obtained by Mittal *et al.* [2020]. Classification results on three examples, each from the EMOTIC dataset (left) and GroupWalk Dataset (right), respectively (extracted from Mittal *et al.* [2020]).

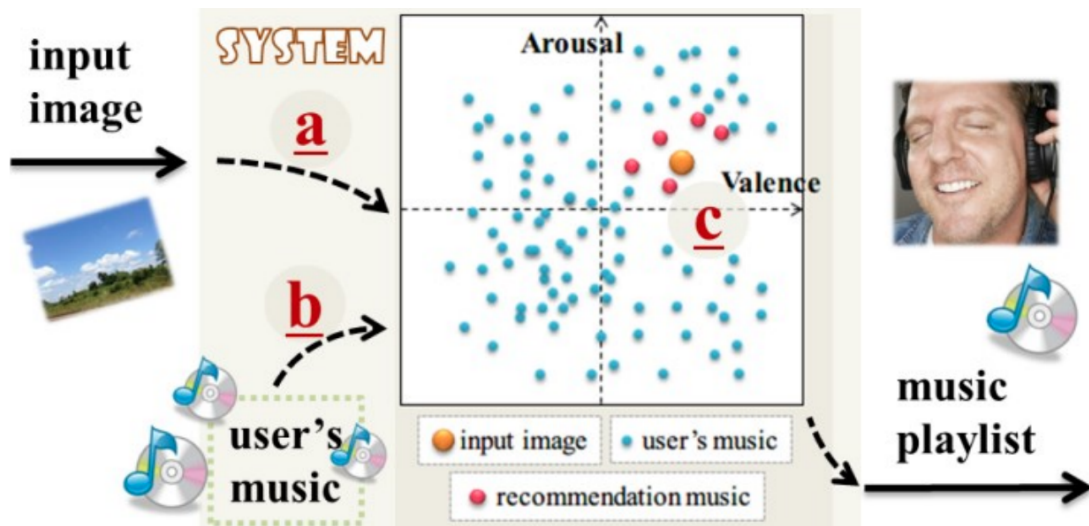


Figure 2.18. Overview of the Sasaki *et al.* [2013] system. Their system uses the valence-arousal plane to create a playlist of the recommended songs for a specified input image (extracted from Sasaki *et al.* [2013]).

the songs to select the background song. Based on the correlation of the features, a ranked music list is derived from the model for a specific video. Also, they proposed an algorithm to align the music beat with the video shot, generating the final recommendation list as the combined result of the correlation and alignability. They did not use music emotion recognition models.

Sasaki *et al.* [2013] proposed an one-directional video to music recommendation using the arousal-valence plane, shown in Figure 2.18. Their system uses input images without textual information and assumes a relationship between mood and images once visual information affects the human mood when listening to music.

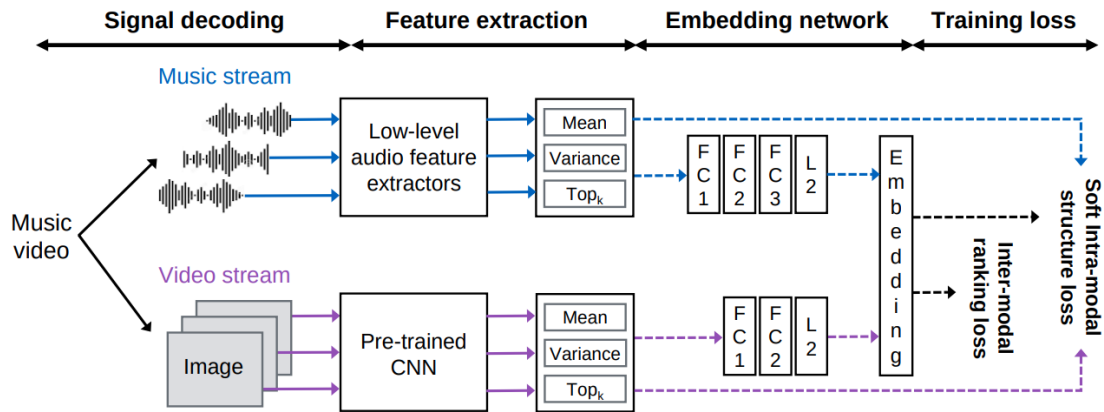


Figure 2.19. Framework proposed by Hong *et al.* [2018]. Given a video and its associated music as input, they extract video features with a CNN and extract music features with low-level audio feature extractors. The features are then aggregated and fed into a two-stream neural network followed by an embedding layer. The whole embedding network is trained by two losses with different purposes: ranking loss for inter-modal relationship and our newly proposed loss for soft intra-modal structure. Each dotted arrow indicates a flow that can be trainable, while the solid arrows indicate flows that do not (extracted from Hong *et al.* [2018]).

Most recently, Hong *et al.* [2018] proposed a cross-modal retrieval method for video and music using deep neural networks, trained via inter-modal ranking loss such that videos and songs with similar semantics get together in embedding space. They proposed a soft intra-modal structure loss that uses the distances between intra-modal samples before embedding. Figure 2.19 shows their proposed framework. Given a video and its associated music as input, they extract video features with a CNN and extract music features with low-level audio feature extractors. The features are then aggregated and fed into a two-stream neural network followed by an embedding layer. The whole embedding network is trained by two losses with different purposes: ranking loss for inter-modal relationship and our newly proposed loss for soft intra-modal structure. Each dotted arrow indicates a flow that can be trainable, while the solid arrows indicate flows that do not.

However, these works are focused on generic videos instead of accelerated videos. Our project has as input an egocentric video that the length is much longer than the average length of the background songs. Thus, we need to select a song and manipulate the video acceleration step to match the video’s semantic and song’s emotional curves. This task is more difficult because just choosing the best song based on features is not sufficient. We also need to create the hyperlapse, maintaining smoothness and temporal continuity on the video, continuously matching the curves.

Table 2.1. Works summary. Comparison of the areas covered by our work and by the most relevant related works.

Work	Video Fast-Forwarding	Music Emotion Recognition	Image Emotion Recognition	Music Recommendation for Video
Kopf <i>et al.</i> [2014]	✓	✗	✗	✗
Karpenko [2014]	✓	✗	✗	✗
Poleg <i>et al.</i> [2015]	✓	✗	✗	✗
Joshi <i>et al.</i> [2015]	✓	✗	✗	✗
Halperin <i>et al.</i> [2018]	✓	✗	✗	✗
Okamoto and Yanai [2014]	✓	✗	✗	✗
Ramos <i>et al.</i> [2016]	✓	✗	✗	✗
Silva <i>et al.</i> [2016]	✓	✗	✗	✗
Yao <i>et al.</i> [2016]	✓	✗	✗	✗
Higuchi <i>et al.</i> [2017]	✓	✗	✗	✗
Ogawa <i>et al.</i> [2017]	✓	✗	✗	✗
Lai <i>et al.</i> [2017]	✓	✗	✗	✗
Silva <i>et al.</i> [2018b]	✓	✗	✗	✗
Silva <i>et al.</i> [2021]	✓	✗	✗	✗
Furlan <i>et al.</i> [2018]	✓	✗	✗	✗
Lu <i>et al.</i> [2006]	✗	✓	✗	✗
Toet and van Erp [2019]	✗	✓	✗	✗
Yang <i>et al.</i> [2008]	✗	✓	✗	✗
Panda <i>et al.</i> [2018]	✗	✓	✗	✗
Chowdhury <i>et al.</i> [2019]	✗	✓	✗	✗
Thammasan <i>et al.</i> [2016]	✗	✓	✗	✗
Dong <i>et al.</i> [2019]	✗	✓	✗	✗
Solymani <i>et al.</i> [2018]	✗	✓	✗	✗
Dan-Glauser and Scherer [2011]	✗	✗	✓	✗
Joshi <i>et al.</i> [2011]	✗	✗	✓	✗
Jia <i>et al.</i> [2012]	✗	✗	✓	✗
Zhao <i>et al.</i> [2014]	✗	✗	✓	✗
Borth <i>et al.</i> [2013]	✗	✗	✓	✗
Mittal <i>et al.</i> [2020]	✗	✗	✓	✗
Kuo <i>et al.</i> [2013]	✗	✗	✗	✓
Sasaki <i>et al.</i> [2013]	✗	✓	✓	✓
Hong <i>et al.</i> [2018]	✗	✗	✗	✓
Ours	✓	✓	✓	✓

2.6 Summary

We presented, in this chapter, a lot of works related to our thesis. Table 2.1 shows the areas covered by our work and by the most relevant related works. In the first column we list the main related works, including our work in the last line. In the other columns we show the areas covered by each work. A ✓ indicates that the work covers that area and a ✗ indicates that the work does not cover that area. Our work covers all four areas presented in the table.

Chapter 3

Methodology

In this chapter, we present our methodology. First, we present an overview of the proposed methodology, and then we explain in more detail each part.

3.1 Overview

We model the problem of accelerating a video according to the emotions induced by visual and acoustic information as a time-series matching problem. Formally, given a long first-person video $V = [v_1, v_2, \dots, v_F]$ with F frames and a target song $M = [m_1, m_2, \dots, m_S]$ with S segments (pieces of an audio track), we aim at creating a shorter video $\hat{V} = [\hat{v}_1, \hat{v}_2, \dots, \hat{v}_S]$ by maximizing the similarity between valence-arousal emotion curves $X \in \mathbb{R}^{F \times 2}$ and $Y \in \mathbb{R}^{S \times 2}$ of the video and audio, respectively. The input can also be a M_L song set, from which the song M most similar to the video V will be selected using an uniform comparison. Figure 3.1 shows an overview of our methodology, which is divided into two main steps: *i)* Emotion Curves Creation and *ii)* Optimal Path Selection.

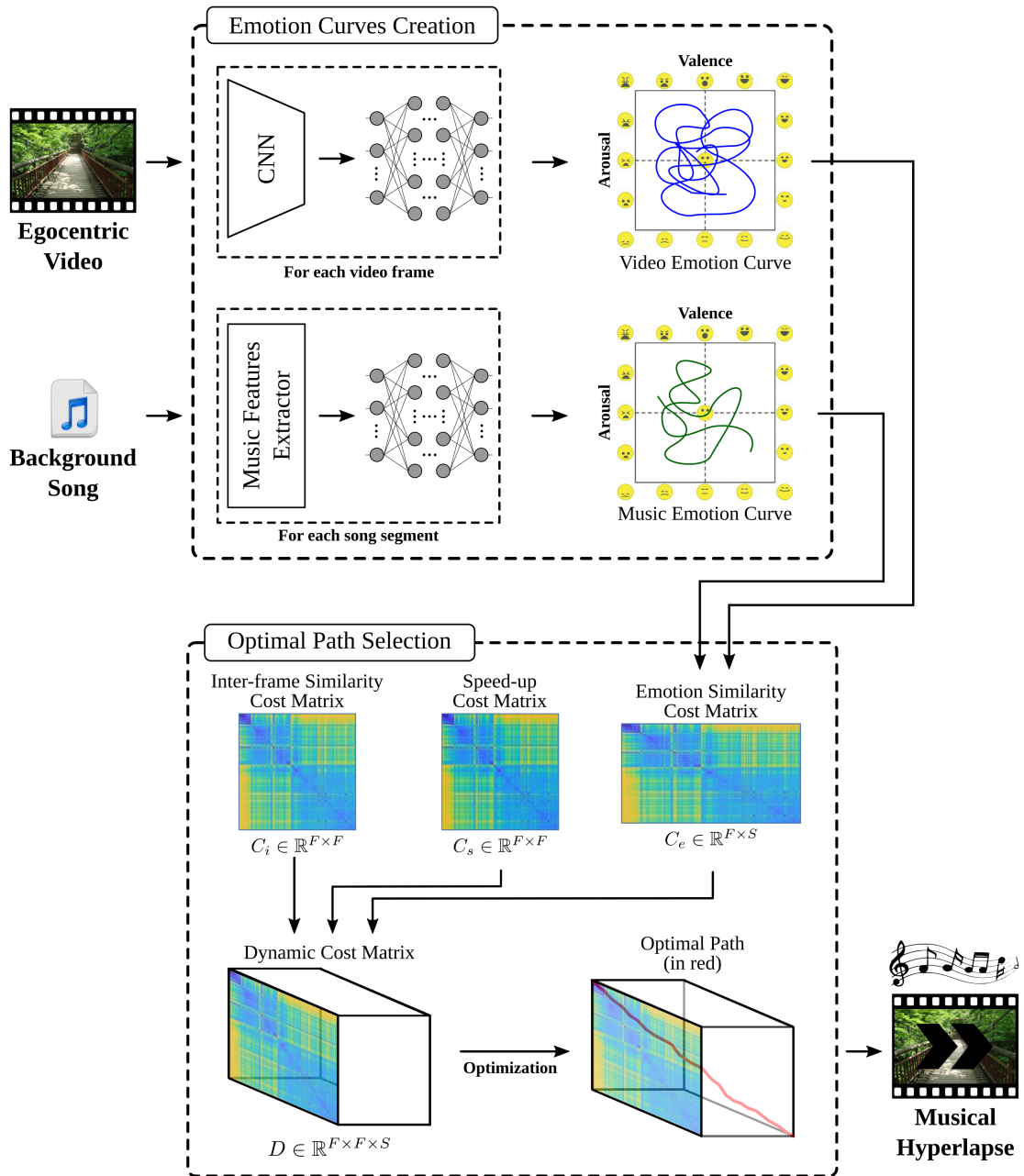


Figure 3.1. Our full methodology overview. In the first step, We extract features from each video frame and each song segment and classify them to obtain their induced emotion. With the classification results, we create continuous two-dimensional emotion curves in the valence-arousal plane. In the second step, we calculate inter-frame and cross-modal cost matrices to create a three-dimensional dynamic cost matrix to compute an optimal path that aligns the emotion induced by a song with the emotion induced by the frames while preserving the visual and temporal continuity.

We also show an overview of our methodology in Algorithm 1, where the inputs are the media streams V and M , and the output is the produced *musical hyperlapse* V_H . In the first loop, the video emotion curve X' is produced as a sequence of image valence and arousal pairs, predicted by the functions `predictVideoFrameValence(\cdot)` and `predictVideoFrameArousal(\cdot)`. Then, the curve X' is smoothed using the function `smoothTransitions(\cdot)`. In the second loop, the song emotion curve Y' is produced as a sequence of audio valence and arousal pairs, predicted by the functions `predictAudioSegmentValence(\cdot)` and `predictAudioSegmentArousal(\cdot)`. Then, the curve Y' is also smoothed using the function `smoothTransitions(\cdot)`. The optimal path \hat{X} is obtained by the function `findOptimalPath(\cdot, \cdot)` and the accelerated video V is then generated from \hat{X} by the function `copySelectedFrames(\cdot)`, that copies selected images from the original video. Finally, the video \hat{V} and song M are concatenated by the function `concatenateVideoAudio(\cdot, \cdot)`, generating the final hyperlapse video V_H .

Algorithm 1: Full Methodology Overview

Input: A video stream V , and an audio stream M

Result: The final hyperlapse V_H

$X' \leftarrow \emptyset$

for $v_i \in V$ **do**

$valence_i \leftarrow \text{predictVideoFrameValence}(v_i)$

$arousal_i \leftarrow \text{predictVideoFrameArousal}(v_i)$

$emotion_i \leftarrow [valence_i, arousal_i]$

$X'.append(emotion_i)$

end

$X \leftarrow \text{smoothTransitions}(X')$

$Y' \leftarrow \emptyset$

for $m_i \in M$ **do**

$valence_i \leftarrow \text{predictAudioSegmentValence}(m_i)$

$arousal_i \leftarrow \text{predictAudioSegmentArousal}(m_i)$

$emotion_i \leftarrow [valence_i, arousal_i]$

$Y'.append(emotion_i)$

end

$Y \leftarrow \text{smoothTransitions}(Y')$

$\hat{X} \leftarrow \text{findOptimalPath}(X, Y)$

$\hat{V} \leftarrow \text{copySelectedFrames}(X)$

$V_H \leftarrow \text{concatenateVideoAudio}(\hat{V}, M)$

return V_H

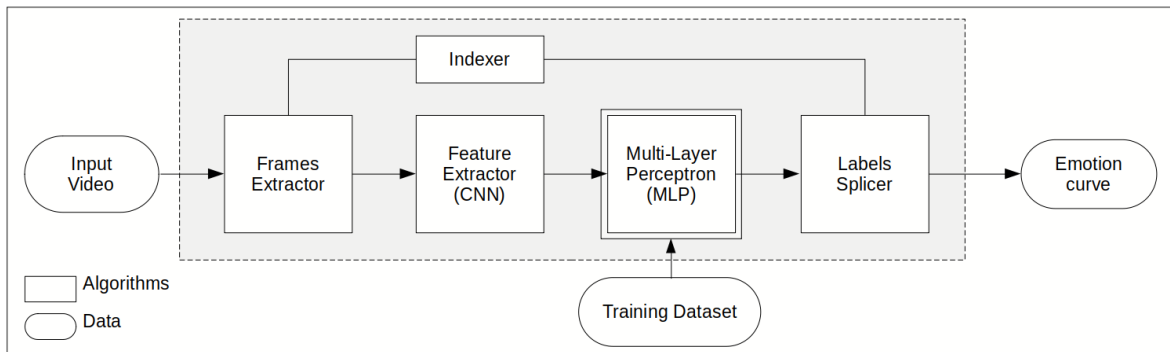


Figure 3.2. Detailed block diagram of the video emotion curve creation. The frame extractor extracts the images from the input video. Then, the features extractor, a Convolutional Neural Network (CNN), extracts the image features for each image. These features are inserted in a classifier (a multiple layer perceptron), which generates the label for the respective video image. The labels splicers join all generated labels in a vector, respecting the order of the original video sequence, which is maintained through the indexer.

3.2 Emotion Curves Creation

Our method creates two emotion curves in the first step, one for the video stream and another for the audio stream. The values in these curves reflect the induced emotion at each instant in time. Based on image and audio feature extraction, classifiers are used to estimate each emotion value, as illustrated in Figure 3.1. Next, we detail these classifiers and the estimation of these curves.

3.2.1 Video Emotion Curve

Figure 3.2 shows a detailed block diagram of the algorithm to create the music emotion curve. In this subsection, we show details about this algorithm. The frame extractor extracts the images from the input video. Then, the features extractor, a CNN, extracts the image features for each image. These features are inserted in a classifier (a multiple layer perceptron), that generates the label for the respective video image. The labels splicers join all generated labels in a vector, respecting the order of the original video sequence, which is maintained through the indexer.

3.2.1.1 Image Classifier

To create the video emotion curve, frames of the video stream V are used to feed an image emotion classifier as $X' = \phi(V)$. The classifier ϕ outputs the valence and

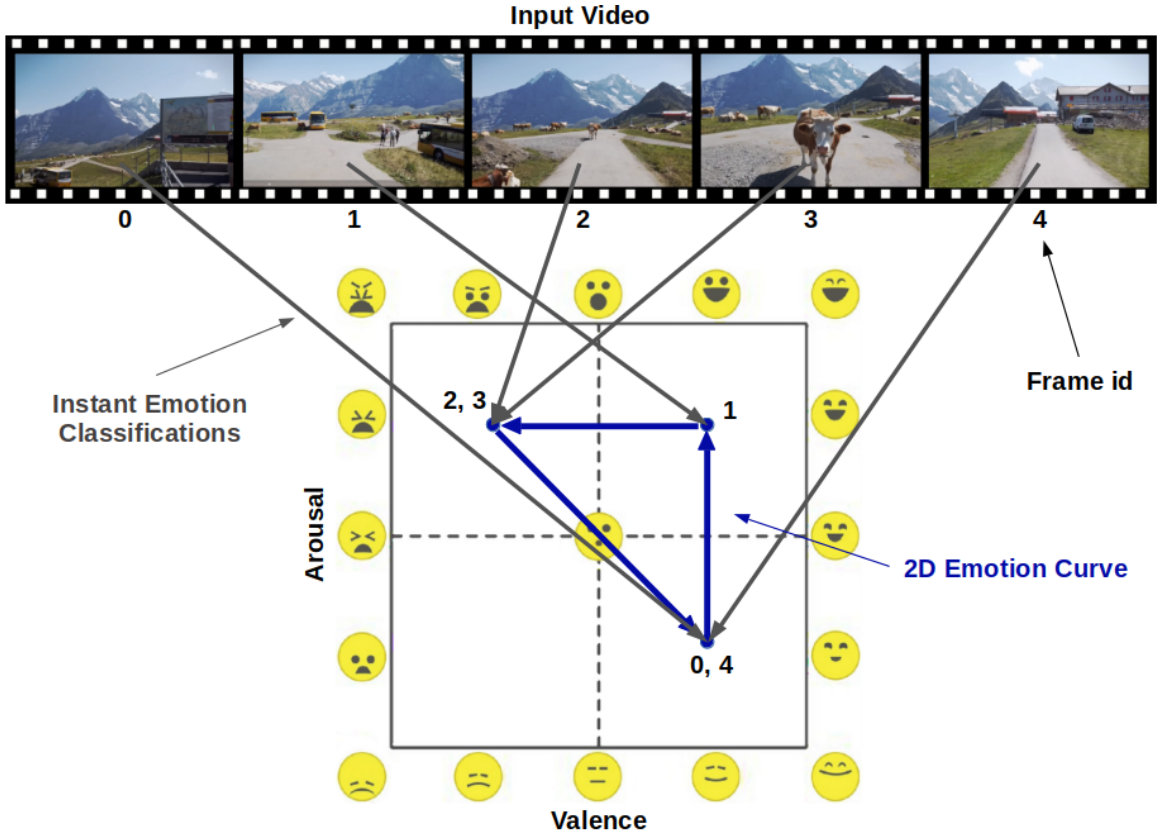


Figure 3.3. Video emotion curve example. Each video frame, an image, is classified as a 2D point in the valence-arousal plane. The sequence of classifications compose the video emotion curve (in blue), which is a time series.

arousal values for each frame composing a discrete two-dimensional emotion curve $X' = [x'_1, x'_2, \dots, x'_F]^T \in \{-1, +1\}^{F \times 2}$, as illustrated in Figure 3.3. We decomposed the curve into separated values of valence $X'_v = [x'_{v1}, x'_{v2}, \dots, x'_{vF}]^T \in \{-1, +1\}^F$ and arousal $X'_a = [x'_{a1}, x'_{a2}, \dots, x'_{aF}]^T \in \{-1, +1\}^F$. Thus, the video frame v_i has the coordinates x'_{vi} and x'_{ai} that represent it in the valence-arousal plane. We use a 2D-CNN, pre-trained in object detection task, as a backbone network topped with a fully-connected network to approximate the function ϕ .

3.2.1.2 Image Dataset

To train the network, we use a subset of the MVSO dataset [Dalmia *et al.*, 2016]. The entire MVSO dataset comprises about 7 million images and their respective contents defined in the form of adjective-nouns pairs such as *colorful-clouds*, *tiny-dog*, *old-books*, *crying-baby*, *happy-people* and others. Each of these adjective-noun pairs is associated with a distribution over the 24 emotion categories from Plutchik’s Wheel of Emotions

[Plutchik, 1980], representing the emotion induced by the image content. We converted these categories to the valence-arousal plane to create the final valence-arousal labels for the images in the MVSO dataset. For each image, we took the predominant emotion out of the 24 and use its quadrant in the Russel’s model as label, as illustrated in Figure 3.4. To find the correct quadrant for each one of the emotions from the Wheel of Emotions, we rely on more detailed versions of the valence-arousal plane, presented in the works of Ahn *et al.* [2010] and Scherer [2005], based on their similarity to the original Russel’s model and also on emotion synonyms.

A filtering was done in which images whose predominant emotion was not well defined were discarded. By this way, we selected only images in which the score for the predominant emotion was higher than the second highest score plus a threshold e_t , defined as $e_t = 0.40$. Images classified as pensiveness were discarded, since this emotion was not well defined in the valence-arousal plane. As result of this filtering, we are then using an MVSO subset with 4,736 images. Next, we list the quadrants and the emotions positioned in each one after the conversion:

- **Quadrant 0 (high valence, high arousal):** Ecstasy, joy, amazement and surprise;
- **Quadrant 1 (low valence, high arousal):** Terror, fear, loathing, rage, anger and annoyance;
- **Quadrant 2 (low valence, low arousal):** Apprehension, grief, sadness, disgust, boredom and vigilance;
- **Quadrant 3 (high valence, low arousal):** Serenity, admiration, trust, acceptance, distraction, anticipation and interest.

Finally, we randomly split the final set into training, validation, and test sets in the proportion 70:15:15 and perform the training using the cross-entropy loss. During the training, the feature extraction layers were kept frozen.

3.2.1.3 Video Curve Smoothing

In the inference stage, the discrete video emotion curve is converted to a continuous emotion curve as $X = f(X') \in \mathbb{R}^{F \times 2}$, where $f : \{-1, +1\} \rightarrow \mathbb{R}$ is a smoothing function that applies a quadratic interpolation to the sequential values, separated for each label (valence and arousal), as illustrated in Figure 3.5. Smoothing is intended to allow the optimization algorithm to work with real numbers instead of integers and avoid creating abrupt transitions in the curves.

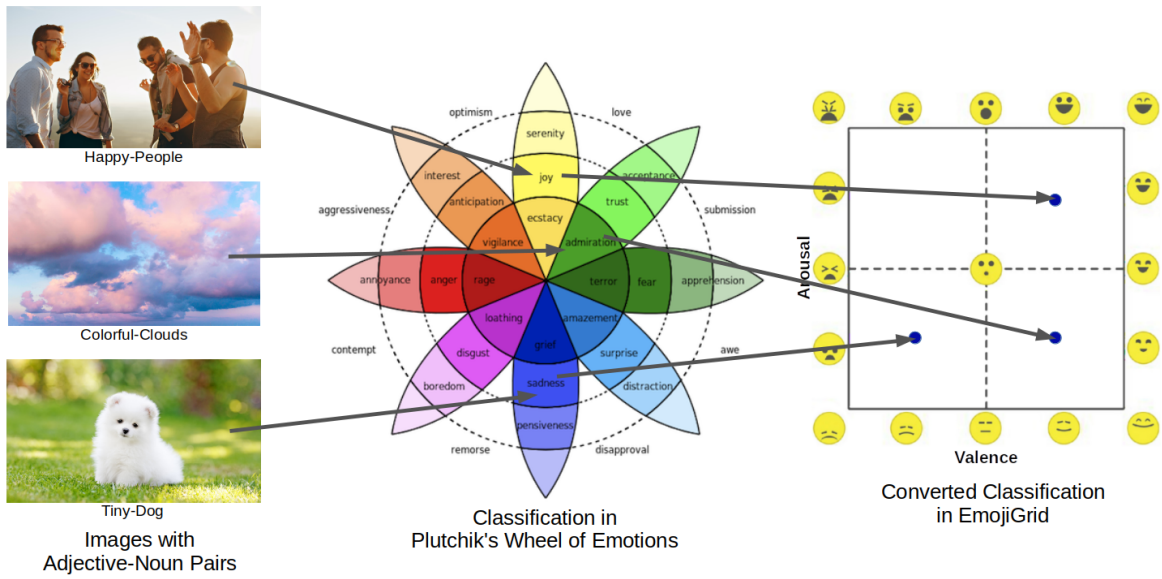


Figure 3.4. Illustration of conversion from Plutchik's Wheel to EmojiGrid. In the MVSO dataset, each image has scores associated with each emotion in Plutchik's Wheel of Emotions. We took the predominant emotion for each image and used its quadrant in the EmojiGrid as label.

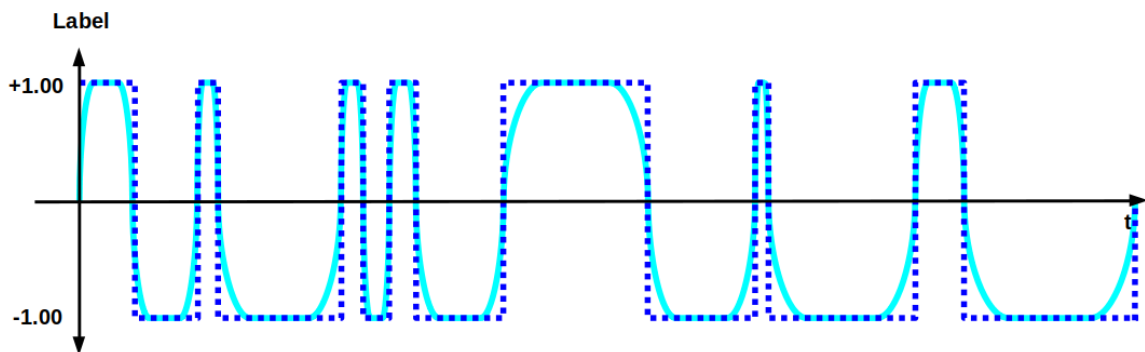


Figure 3.5. Illustration of the video curves smoothing. In dark blue, the discrete curve produced by the sequence of discrete classifications of the neural network for one of the labels (valence or arousal). In light blue, the continuous curve produced after smoothing the discrete curve. In this case, the label was discretized into $N = 2$ different levels.

3.2.2 Music Emotion Curve

Figure 3.6 shows a detailed block diagram of the algorithm to create the music emotion curve. The spectrogram extractor and splitter extracts and divides the input song spectrogram into 6-second samples at intervals of 0.5-second. Then, the features ex-

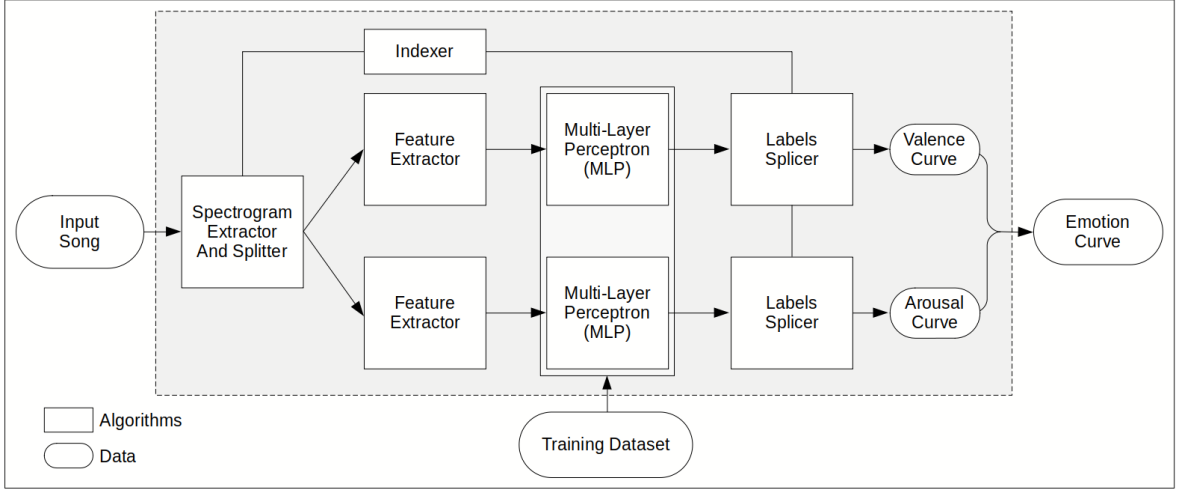


Figure 3.6. Detailed block diagram of the music emotion curve creation. The spectrogram extractor and splitter extracts and divides the input song into 6-second samples, at each 0.5-second of the song. Then, the features extractor extracts feature dedicated to the song’s melody, generating a vector with 48 features. We use two separate extractors because the valence features are different from the arousal features. These features are inserted in a classifier (a multiple layer perceptron), which generates the label for the respective song sample. Two separate classifiers are also used, one for valence and one for arousal. The labels splicers joins all generated labels in a vector, respecting the order of the segments of the original song, which is maintained through the indexer.

tractor extracts feature dedicated to the song’s melody, generating a vector with 48 features. We use two separate extractors, one for valence and other for arousal, treating these variables as being independent. These features are inserted in a classifier (a multiple layer perceptron), that generates the label for the respective song sample. Two separate classifiers are also used, one for valence and one for arousal. The labels splicers joins all generated labels in a vector, respecting the order of the segments of the original song, which is maintained through the indexer.

3.2.2.1 Music Classifier

To create the music emotion curve for an audio stream M , we use a pair of music emotion classifiers $Y' = \psi(M)$ that provides the valence and arousal one-dimensional discrete curves $Y'_v = [y'_{v1}, y'_{v2}, \dots, y'_{vS'}]^T \in \{c_1, c_2, \dots, c_N\}^{S'}$ and $Y'_a = [y'_{a1}, y'_{a2}, \dots, y'_{aS'}]^T \in \{c_1, c_2, \dots, c_N\}^{S'}$, where N is the number of discrete values, that is, categories, in which a song segment can be classified in the valence-arousal plane, and S' is the number of song segments at the sampling rate used in the classifier. Combining the valence and arousal values of these two curves, we ob-

tain a two-dimensional emotion curve $Y' = [y'_1, y'_2, \dots, y'_{S'}]^T \in \{c_1, c_2, \dots, c_N\}^{S' \times 2}$ in the valence-arousal plane. Thus, for a song segment $m_k, k \in \{1, \dots, S'\}$, (y'_{vk}, y'_{ak}) is represented as being one of the $N \times N$ points of a grid in the valence-arousal plane, where higher y'_{vk} values indicate a more positive valence and higher y'_{ak} values indicate a higher arousal.

Our music emotion classifier ψ comprises a feature extractor topped with two fully connected networks, one for each dimension (valence and arousal). We create a window of size $\alpha = 6$ seconds and slide it over the audio stream with a stride of $\delta = 0.5$ seconds to extract the features for each song segment from the audio spectrogram, as illustrated in Figure 3.7. Then, we extract from each spectrogram a d -dimensional feature vector $\hat{m}_k \in \mathbb{R}^d$ dedicated to the melody of the song. The selection of the audio features is based on the research of Panda *et al.* [2020]. Finally, we feed each feature vector \hat{m}_k to the classifiers to obtain the discrete curves Y'_v and Y'_a .

3.2.2.2 Music Dataset

We use the DEAM dataset [Solymani *et al.*, 2018] to train the music emotion classifier. The DEAM dataset comprises about 1,802 songs of various styles, such as rock, classic, country, and others, with durations between 45 seconds and 7 minutes. For each song, some raters (10 in most cases) annotated its valence and arousal values in a range of $[-1, +1]$ at each step of 0.5 seconds, starting from the 15th second of the song. There are approximately 126,000 annotated song segments in the entire dataset. Figure 3.8 shows the interface used to rotate the DEAM dataset [Aljanaki *et al.*, 2017].

To define the song segment label, we averaged the rater’s annotated valence and arousal values after filtering all values distant by 0.5 standard deviations from the mean. After this filtering, the dataset had 26,457 annotated segments for valence and 26,481 annotated segments for arousal. Then, to create the pairs of segments and labels used in our training procedure, we discretize the valence and arousal annotations provided in the DEAM dataset into N classes. Similar to our image emotion recognition classifier, we train the music emotion recognition classifiers using training, validation, and test splits in the same proportion.

3.2.2.3 Music Curve Smoothing

We also perform a smoothing in the music emotion curves, similarly to the video emotion curves, as illustrated in Figure 3.9. Note that, by using a stride of $\delta = 0.5$ seconds, during inference, we only obtain 2 samples per second, while the video stream operates at a higher rate, usually 30 frames per second. Therefore, to match the video’s

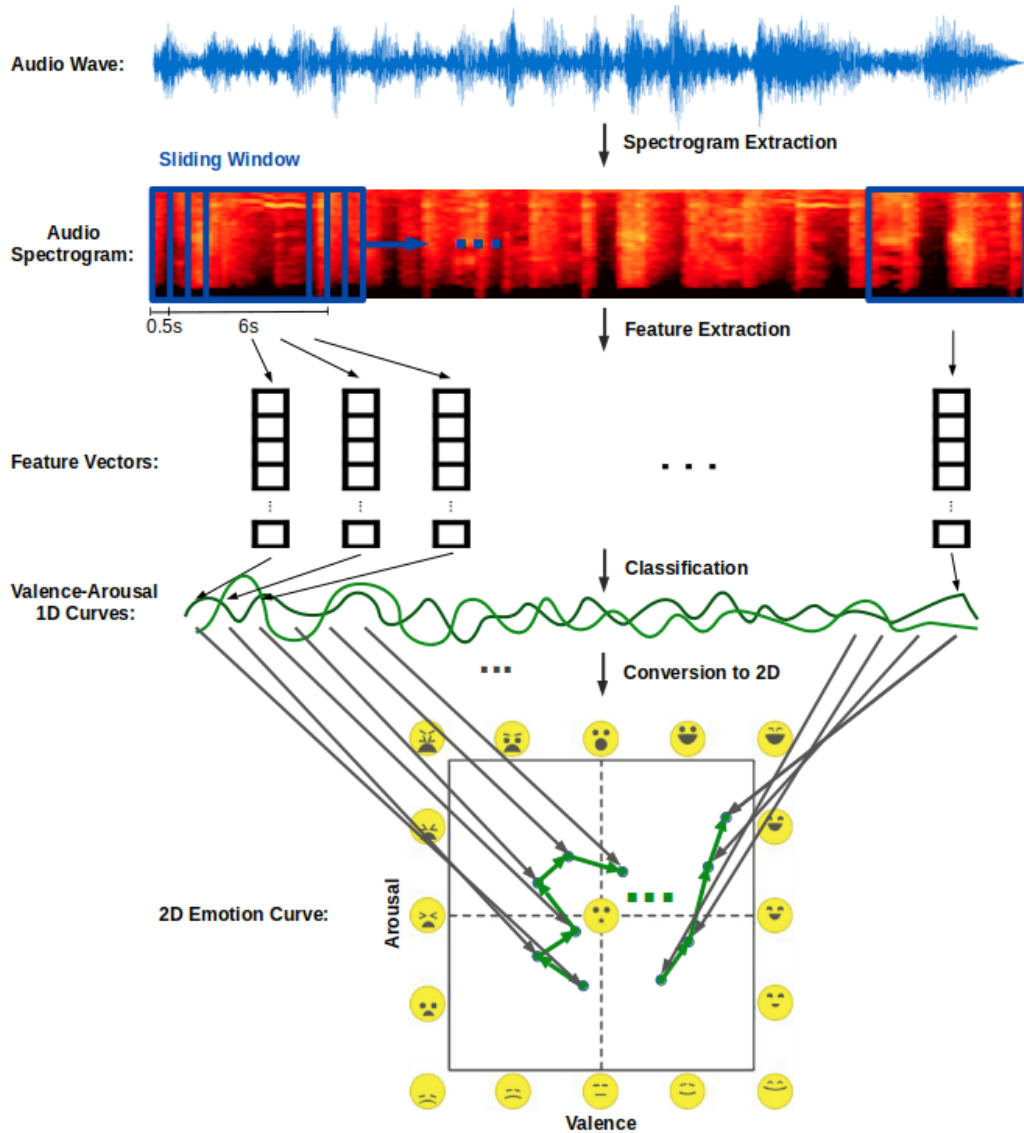


Figure 3.7. Illustration of the audio classification step. We create a window of size $\alpha = 6$ seconds and slide it over the audio stream with a stride of $\delta = 0.5$ seconds to extract the audio features for each song segment from the audio spectrogram. For both valence and arousal, each feature vector is classified as a value between -1 and $+1$, resulting in one curve for valence and one for arousal. These curves together form the emotion curve in the valence-arousal plane.

sampling rate, we perform a upsampling in the curves before applying a smoothing function that creates the final continuous curve $Y = g(Y') \in \mathbb{R}^{S \times 2}$. Thus, S is the number of song segments at the same sampling rate of the video. Again, smoothing is intended to allow working with real numbers and avoid creating abrupt transitions.

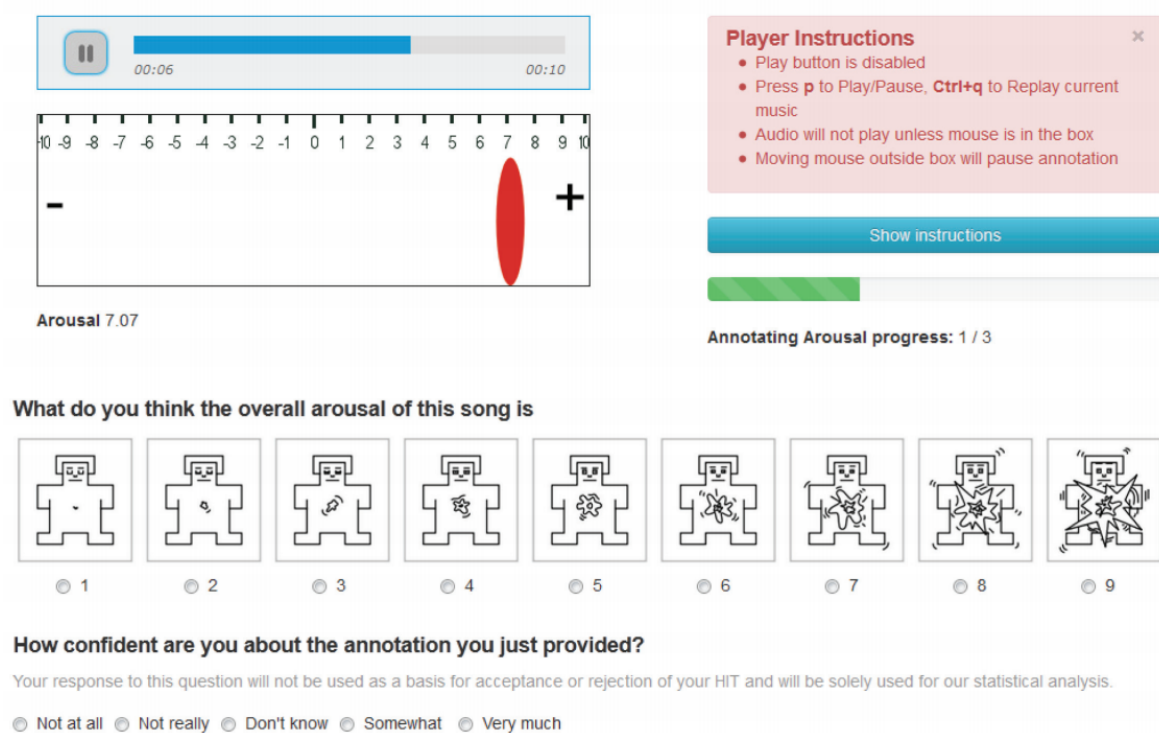


Figure 3.8. Annotation interface of the DEAM dataset for arousal label. For each segment of each song in the dataset, a set of users listen to the segment and sets the arousal value in a range of -10 to +10, through this labeling interface. The user can also assign an overall arousal for the entire song and a confidence level to his annotation. A similar interface is used to label the valence. (Extracted from Aljanaki *et al.* [2017]).

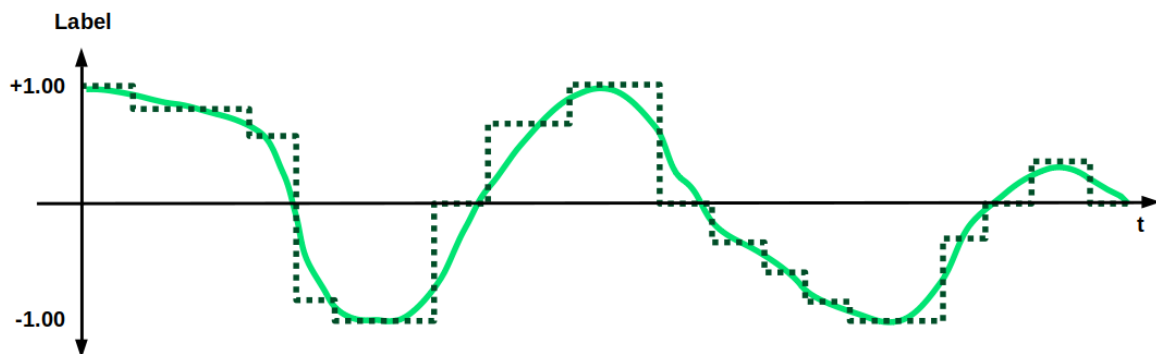


Figure 3.9. Illustration of the song curves smoothing. In dark green, the discrete curve produced by the sequence of discrete classifications of the neural network for one of the labels (valence or arousal). In light green, the continuous curve produced after smoothing the discrete curve. In this case, the label was discretized into $N = 8$ different levels.

3.3 Optimal Path Selection

After creating the emotion profile of the video and audio streams, we aim to find the optimal path that matches the emotion induced by the video and the song. The optimal path selection procedure is shown in Algorithm 2. The inputs of the algorithm are the video emotion curve X , and the song emotion curve Y . The output is the accelerated video \hat{V} , after dropping the frames to maximize the emotion similarity and visual video quality. The functions and variables presented in this algorithm are described in the next subsections.

Algorithm 2: Optimal Path Selection

Input: The video V , the song M , and the emotion curves X and Y

Result: The accelerated video \hat{V}

$F \leftarrow \text{length}(X)$

$S \leftarrow \text{length}(Y)$

$C_i, C_s, C_e \leftarrow \text{create2DCostMatrices}(V, M, F, S)$

$D, T \leftarrow \text{create3DCostMatrices}(F, S, C_i, C_s, C_e)$

$\hat{V} \leftarrow \text{createOptimalPath}(F, S, D, T)$

return \hat{V}

3.3.1 2D Cost Matrices Construction

When shrinking the video size, besides aligning the emotions in both modalities, we also need to produce a visually continuous video that presents a smooth motion during the exhibition. To attend to both objectives, we draw inspiration from the optimization process proposed in the work of Joshi *et al.* [2015], which creates a hyperlapse video with smooth transitions between frames using dynamic programming and DTW-based algorithm. However, unlike Joshi *et al.*, which optimizes the output video path regarding only inter-frame transitions and on visual modality, in our work, we must consider not only the inter-frame transitions but also the audio-visual relation regarding the induced emotion. Therefore, our algorithm creates cross-modal, inter-frame and speedup cost matrices to perform the optimization.

3.3.1.1 Inter-Frame Similarity Cost Matrix

To keep the video with a continuous visual motion, we create an Inter-frame Similarity Cost Matrix, $C_i \in \mathbb{R}^{F \times F}$, with each element computed as

$$C_i(i, j) = 1 - \text{SSIM}(v_i, v_j), \quad (3.1)$$

where $i, j \in \{1, 2, \dots, F\}$ are the frames indices in the input video and $\text{SSIM}(\cdot, \cdot)$ is the structural similarity index measure [Zhou Wang *et al.*, 2004]. Higher SSIM values indicate that the input frames are more similar to each other.

3.3.1.2 Speedup Cost Matrix

The algorithm also uses a cost matrix to avoid skips that are too distant from the target speed-up rate. Specifically, let $Sp^* = F/S$ be the target speed-up rate, where F is the number of video frames and S is the number of song segments, respectively. Each element in the Speed-up Cost Matrix, $C_s \in \mathbb{R}^{F \times F}$, is given by

$$C_s(i, j) = \min(((j - i) - \lfloor Sp^* \rfloor)^2, c_{min}), \quad (3.2)$$

where c_{min} is a threshold, empirically set to 200.

3.3.1.3 Emotion Similarity Cost Matrix

Finally, we create a cross-modal matrix to determine the cost of skipping relevant frames regarding the video and audio stream emotion similarity. The Emotion Similarity Cost Matrix, $C_e \in \mathbb{R}^{F \times S}$, is computed as

$$C_e(i, k) = \frac{\sqrt{(x_{vi} - y_{vk})^2 + (x_{ai} - y_{ak})^2}}{d_0}, \quad (3.3)$$

where $k \in \{1, 2, \dots, S\}$ is the song segment index, x_{vi} and x_{ai} are coordinates that represent the video frame in the valence-arousal plane, and y_{vk} and y_{ak} are coordinates representing the song segment. The constant d_0 , a normalization factor, is the distance between the points $(+1, +1)$ and $(-1, -1)$ in the valence-arousal plane ($d_0 = \sqrt{8}$).

The cost matrices C_i , C_s , and C_e are normalized into a range of $[0, 1]$. We show the detailed procedure to create the normalized cost matrices in Algorithm 3. First the matrices C_i and C_s are built and then normalized to $[0, 1]$. Finally, the matrix C_e is also built, already normalized.

Algorithm 3: 2D Cost Matrices Creation**Input:** The video V , the song M , and F and S numbers, respectively**Result:** The cost matrices C_i , C_s and C_e

```

for  $i=0$  to  $F$  do
  for  $j=0$  to  $F$  do
     $C_i[i, j] \leftarrow 1 - \text{SSIM}(v_i, v_j)$ 
     $C_s[i, j] \leftarrow \min(((j - i) - \lfloor F/S \rfloor)^2, c_{\min})$ 
  end
end
 $\max_{ci} \leftarrow \max(C_i)$ 
 $\max_{cs} \leftarrow \max(C_s)$ 
for  $i=0$  to  $F$  do
  for  $j=0$  to  $F$  do
     $C_i[i, j] \leftarrow C_i[i, j] / \max_{ci}$ 
     $C_s[i, j] \leftarrow C_s[i, j] / \max_{cs}$ 
  end
end
for  $i=0$  to  $F$  do
  for  $k=0$  to  $S$  do
     $C_e[i, k] \leftarrow \frac{\sqrt{(x_{vi} - y_{vk})^2 + (x_{ai} - y_{ak})^2}}{d_0}$ 
  end
end
return  $C_i, C_s, C_e$ 

```

3.3.2 3D Dynamic Cost Matrix Construction

The normalized cost matrices C_i , C_s , and C_e are further used to create a 3D Dynamic Cost Matrix $D \in \mathbb{R}^{F \times F \times S}$. Each entry $D(i, j, k)$ represents the minimal cost of the path that ends at the frame v_j and song segment k . We also create a traceback matrix $T \in \mathbb{R}^{F \times F \times S}$ that stores in $T(i, j, k)$ the index of the frame that precedes v_j in the path, given the song segment k . Next, we populate D and T by setting the first song segment slice as $D(i, j, 0) = C_s(i, j)$ and the following slices recursively as

$$\begin{aligned}
 D(i, j, k) = & \lambda_i C_i(i, j) + \lambda_s C_s(i, j) + \lambda_e C_e(j, k) \\
 & + \min_{h=1}^w (D(i-h, i, k-1)),
 \end{aligned} \tag{3.4}$$

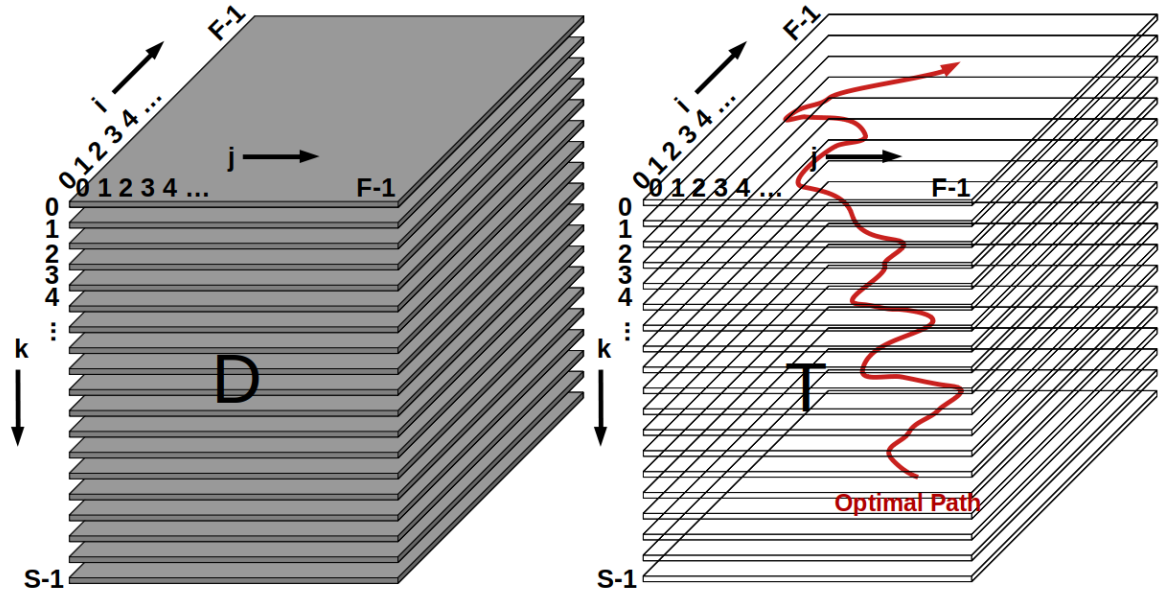


Figure 3.10. Illustration of the 3D dynamic cost matrices. Each position of the matrix D stores the cost of jumping from video frame i to j , associating video frame j with song frame k . Each position of the T array stores the index of the frame before the i frame whose cost is minimal. The path of the selected frames is shown in red, which is a sequence of frames, one of each slice of T .

where λ_i , λ_s and λ_e are the weights associated with each cost term and w is the maximum skip between adjacent frames in the path. More specifically, the cost weights were empirically set to $\lambda_i = 1.00$, $\lambda_s = 0.01$, $\lambda_e = 0.01$, and we set the maximum allowed skip to $w = 2Sp^*$ whose value is bounded to the interval $[4, 16]$. When populating D , we concurrently populate the traceback matrix by computing $T(i, j, k) = \operatorname{argmin}_{h=1}^w (D(i-h, i, k-1))$. Figure 3.10 illustrates the matrices D and T , and the optimal path.

We show the detailed procedure to construct matrices D and T in Algorithm 4. First, we initialize the cost weights and the maximum allowed skip. We also use a constant $g = 2$, which determines how many frames can be skipped at the beginning and at the end of the video. Then, we fill each slice of the matrices D and T according to the equations defined above.

Algorithm 4: 3D Dynamic Cost Matrix Creation

Input: The F and S numbers, and the cost matrices C_i , C_s and C_e **Result:** The 3D dynamic cost matrices D and T $[\lambda_e, \lambda_i, \lambda_s] \leftarrow [1.00, 0.01, 0.01]$ $w \leftarrow \min(16, \max(4, \text{int}(2 * F/S)))$ $g \leftarrow 2$ **for** $k=0$ to S **do** **for** $i=g$ to $\min(k * w + 1, F)$ **do** **for** $j=i+1$ to $\min(i + 1 + w, F)$ **do** $c \leftarrow \lambda_i C_i[i, j] + \lambda_s C_s[i, j] + \lambda_e C_e[j, k]$ $D[i, j, k] \leftarrow c + \min_{h=1}^{h=w} (D[i - h, i, k - 1])$ $T[i, j, k] \leftarrow \operatorname{argmin}_{h=1}^{h=w} (D[i - h, i, k - 1])$ **end** **end****end***return* D, T

3.3.3 Optimal Path Traceback

With matrices T and D filled, we traceback the optimal path, starting from the position $k = F$, and selecting, at each step, the index stored in $T(i, j, k - 1)$ while $k \geq 0$. The reversed order of the frames selected during this step is the final set that composes the hyperlapse video. Note that exact S frames are selected in this step. Therefore, the video length is reduced to the song length. We add the input audio stream to the composed hyperlapse video to generate the *musical hyperlapse* video.

We show the detailed procedure to create the optimal path in Algorithm 5. We use the variables w and g with the same values used in Algorithm 4. We select in the last slice of the matrix T , the position corresponding to the lowest cost of the last slice of the matrix D . Then, we initialize \hat{V} as an empty list, and go inserting the next frame stored in the selected position in T , until we reach the first slice of T . After doing this, we have the sequence of frames of the accelerated video \hat{V} .

Algorithm 5: Optimal Path Traceback**Input:** The F and S numbers, and the 3D matrices D and T **Result:** The accelerated video \hat{V} $w \leftarrow \min(16, \max(4, \text{int}(2 * F/S)))$ $g \leftarrow 2$ $k \leftarrow S$ $s \leftarrow \operatorname{argmin}_{i=F-g, j=i+1}^{i=F, j=i+w} (D[i, j, M])$ $d \leftarrow \min_{i=F-g, j=i+1}^{i=F, j=i+w} (D[i, j, M])$ $\hat{V} \leftarrow \emptyset$ $\hat{V}.append(d)$ **while** $s > g$ and $k > 1$ **do** $\hat{V}.prepend(s)$ $b \leftarrow T(s, d, k - 1)$ $d \leftarrow s$ $s \leftarrow b$ $k \leftarrow k - 1$ **end***return* \hat{V}

3.4 Song Selection

To improve the quality of the emotion curves matching, we also include an algorithm to choose the best M_i song to be inserted in the video V . If the user only defines a song M as input of the algorithm, the optimization is done only with it. However, the user can also define a directory with several songs as input $M_L = [M_1, M_2, \dots, M_{N_m}]$, allowing the algorithm to choose the best song for the video, among the N_m songs present in this directory.

The procedure to make the song selection is detailed in Algorithm 6. First, given an input video V and a list of input songs M_L , for each song $M_i \in M_L$, we reduce the size of the video V to the size of the song M_i , by uniformly removing frames from V , creating the reduced video V' . Then, we use the function `getEmotionCurve(\cdot)` to create the smoothed emotion curves X and Y , as previously described, for the video V' and the song M_i , respectively. Then, we calculate the similarity of the emotion curves X and Y for each song M_i , and select the song with the highest average emotion similarity as the best song M_b . We use the function `getEmotionSimilarity(\cdot, \cdot)` to get the similarity of a pair (x_i, y_i) , given by $p_{sim} = 1 - \frac{\sqrt{(x_{vk} - y_{vk})^2 + (x_{ak} - y_{ak})^2}}{d_0}$, where x_{vk} and x_{ak} are the image valence and arousal values, respectively; y_{vk} and y_{ak} are

the audio valence and arousal, respectively; and d_0 is the normalization factor. After running this algorithm, our optimization method can then be performed with the best song M_b and the input video V .

Algorithm 6: Song Selection

Input: A video stream V , and a list of audio streams M_L

Result: The reduced emotion curve \hat{X}

```

 $[M_b, S_b] \leftarrow [\emptyset, -inf]$ 
for  $M_i \in M_L$  do
   $M \leftarrow M_i$ 
   $[F, S] \leftarrow [\text{length}(V), \text{length}(M)]$ 
   $w \leftarrow F/S$ 
   $[V', M'] \leftarrow [\emptyset, \emptyset]$ 
   $[i_s, i_v] \leftarrow [0, 0]$ 
  while  $i_s < S$  and  $i_v < F$  do
     $M'.append(M[int(i_s)])$ 
     $V'.append(V[int(i_v)])$ 
     $i_s \leftarrow i_s + 1$ 
     $i_v \leftarrow i_v + w$ 
  end
   $X \leftarrow \text{getEmotionCurve}(M')$ 
   $Y \leftarrow \text{getEmotionCurve}(V')$ 
   $S_{list} \leftarrow \emptyset$ 
   $S_b \leftarrow 0$ 
  for  $x_k, y_k \in X, Y$  do
     $s_k \leftarrow \text{getEmotionSimilarity}(x_k, y_k)$ 
     $S_{list}.append(s_k)$ 
  end
   $S_k \leftarrow \text{mean}(S_{list})$ 
  if  $S_k > S_b$  then
     $S_b \leftarrow S_k$ 
     $M_b \leftarrow M_i$ 
  end
end

 $M' \leftarrow M_b$ 
return  $M', V'$ 

```

Chapter 4

Experiments

In this chapter, we present our experiments to provide both quantitative and qualitative evaluation.

4.1 Implementation Details

Our method was fully implemented in Python. In this section, we describe details about the implementation.

4.1.1 Emotion Curves Creation

For the image emotion recognition, we used the ResNet50 [He *et al.*, 2016] as the backbone network topped with a fully connected network with four layers of 1,000 neurons. We also tested other networks, such as AlexNet, VGG19, DenseNet, Inception, and SqueezeNet. ResNet50 had the best training and testing accuracies. We also tried other numbers of neurons for the fully connected layers, such as 200, 500, and 2,000, but these numbers had no significant influence on the results. The classification layer in the image emotion classifier comprises $N = 4$ neurons that represent each of the valence-arousal quadrants. We also tested values greater than four for the number of quadrants N , but the network could not get test accuracies above 50% for these values.

For the music emotion recognition, we also used a fully connected network with four layers of 1,000 neurons. We also tested other numbers of neurons for the fully connected layers, such as 200, 500, and 2,000, but these numbers had no significant influence on the results. The classification layer in the music emotion classifier comprises $N = 8$ neurons that represent each of the discretization levels separately for valence and arousal, totalizing 64 levels in the valence-arousal plane. We also tested greater

discretization levels, but the best results were obtained with 64 levels. We used the *essentia* Python library to extract the $d = 48$ music features used in the classification process.

To train the classifiers, we used a batch size of 200 for image classifier and 10000 for music classifiers, and the Adam optimizer with a learning rate of 1×10^{-5} and weight decay of 1×10^{-3} to train both classifiers. We used early stopping in both image and music classifiers, limiting the maximum number of epochs to 100 for the image classifier, and 10000 for the music classifiers. The image classifier training took approximately 24 hours, while for audio classifiers, the training took approximately 12 hours.

4.1.2 Optimal Path Algorithm

For the optimal path selection algorithm, the cost weights were empirically set to $\lambda_i = 1.00$, $\lambda_s = 0.01$, $\lambda_e = 0.01$. We used a large weight on the similarity of emotions, as this is the main objective of our method. For our method, we set the maximum allowed skip to $w = 2Sp^*$ whose value is bounded to the interval $[4, 16]$. For the D and T matrices, we used sparse representations to avoid high memory consumption issues. We run the optimal path selection algorithm using parallel processing to reduce the computational cost of processing time. First, we split the video and music into N_p parts, where N_p is the number of machine processors. We then run the optimization algorithm separately for each part, producing multiple accelerated chunks of the video. Finally, we concatenate all the accelerated parts forming the full accelerated video.

4.2 Experimental Setup

This section presents details about the experimental setup, such as the videos, songs, and metrics used in the evaluation.

4.2.1 Dataset

To compare our method with other existing methods, we organized a dataset composed of eight videos presenting different contents such as scenes of nature, cities, parks, buildings, cars, people, animals, and others; and five fixed songs with varied styles and emotion induction. Table 4.1 shows the list of videos and songs used in part of the experiments. We collected the videos from various sources, including the YouTube platform, other works in the literature, and self-acquisition. The specific source of

Table 4.1. Audio-visual dataset. List of videos and songs used for comparison with baselines.

Video Name	Duration
Berkeley1 (Self-acquisition)	17:41
Berkeley2 (Self-acquisition)	13:40
Bike3 [Kopf <i>et al.</i> , 2014]	13:10
CityWalk1 (YouTube)	10:00
MontOldCity1 (YouTube)	10:01
NatureWalk1 (YouTube)	09:50
StockHolm1 (YouTube)	24:59
Walking4 [Ramos <i>et al.</i> , 2016]	08:43
Song Name	Duration
Last To Know (Three Days Grace)	03:28
Onward to Freedom (Trailerhead)	02:58
My Immortal (Evanescence)	04:32
Little Talks (Of Monsters And Men)	04:23
In The End (Linkin Park)	03:38

the video and the song authors are indicated right after the video and song names, respectively. In order to standardize the size of images during data processing, we re-sampled all videos to the exact resolution of 640×480 pixels, but the algorithm can also be run with videos in other resolutions.

4.2.2 Evaluation Metrics

To assess the hyperlapse methods, we need to quantify the emotion induced by the video and audio streams, whether the target speed-up rate was achieved, and the visual continuity and stability of the final video.

4.2.2.1 Emotion Similarity

We quantify the emotion in the output video using the Emotion Similarity metric defined as

$$E_{sim} = \frac{1}{S} \sum_{k=1}^S \left(1 - \frac{\sqrt{(\hat{x}'_{vk} - y_{vk})^2 + (\hat{x}'_{ak} - y_{ak})^2}}{d_0} \right), \quad (4.1)$$

where \hat{x}'_{vk} and \hat{x}'_{ak} are the discrete valence and arousal values of the reduced video \hat{V} .

4.2.2.2 Speedup Ratio

Given a target speed-up rate Sp^* , to verify whether the target speed-up rate was achieved or not, we use the Speed-up Ratio metric, which is calculated as

$$Sp_r = \frac{\max(Sp^*, \hat{Sp})}{\min(Sp^*, \hat{Sp})}, \quad (4.2)$$

where $\hat{Sp} = \hat{F}/S$ is the speed-up rate achieved by the hyperlapse method. The purpose of this metric is to show the magnitude of the difference between the two speedups, in terms of percentage. The optimal value is 1, the greater the value, the greater the percentage of error between the desired and the obtained speedups.

4.2.2.3 FID-Score

We also measure if the output visual content is similar to the input and if it is stable. To calculate the similarity, we use the Fréchet Inception Distance (FID), proposed by Mathiasen and Hvilshøj [2020], which gives the similarity between two sets of images. This metric is commonly used to assess the quality of synthetic images created by generative adversarial networks. We apply this metric to determine the similarity between the original and accelerated videos regarding visual content.

4.2.2.4 Shaking Ratio

To compute the stability of the output frame transitions, we use the Shaking Ratio [Ramos *et al.*, 2020]. The Shaking Ratio uses homography transformations to calculate the average motion of the central pixel between pairs of frame transitions. We calculate it as the average motion of the midpoint between the frame transitions through the video, which is given by

$$Sk_r = \frac{1}{|\hat{V}| - 1} \sum_{n=1}^{|\hat{V}|-1} \frac{H(\hat{v}_n, \hat{v}_{n+1})}{d(v_n)}, \quad (4.3)$$

where \hat{v}_n is the n_{th} frame in the output video, H computes the transition of the central point of \hat{v}_n when applying the estimated homography between \hat{v}_n and \hat{v}_{n+1} , $d(\cdot)$ is the half of the frame diagonal and $|\hat{V}|$ is the size of set \hat{V} .

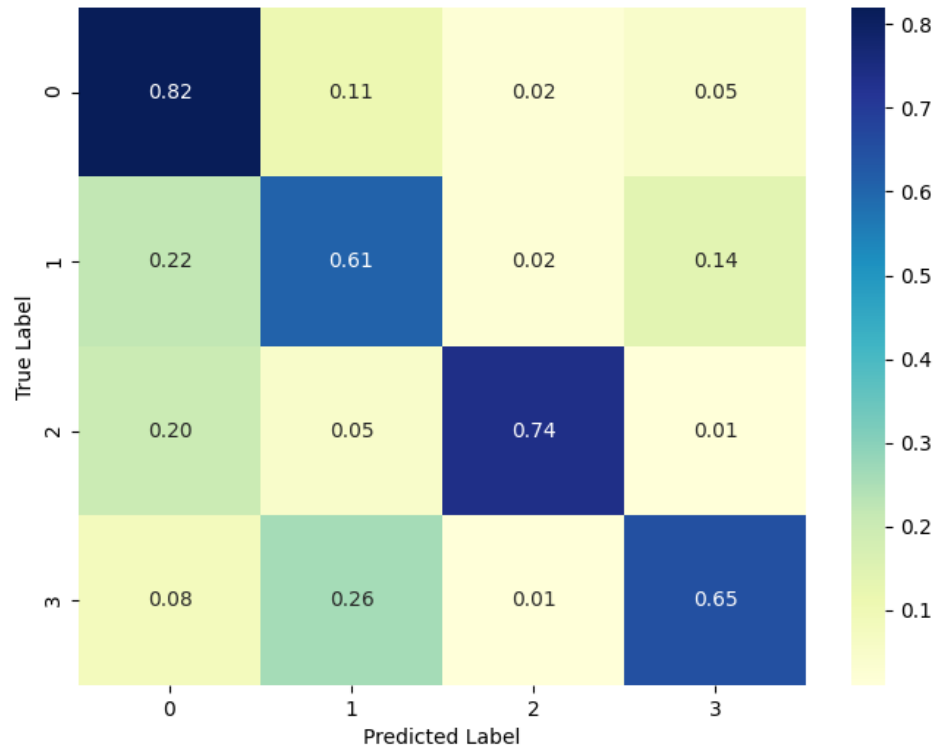


Figure 4.1. Confusion matrix for the image classifier. A normalized confusion matrix for the image classifier, showing the percentage of times that each label l_i (rows) was predicted as each label l_j (columns).

4.3 Quantitative Evaluation

This section presents the quantitative evaluation and results, first separated for the emotion classification and optimal path selection, and then for the entire methodology. We compare the optimal path method with other methods present in the literature. And we also compare our full methodology, to create the musical hyperlapse with other existing hyperlapse creation methods.

4.3.1 Emotion Classification

This section presents the obtained accuracy for the emotion classification, both for video and audio modalities

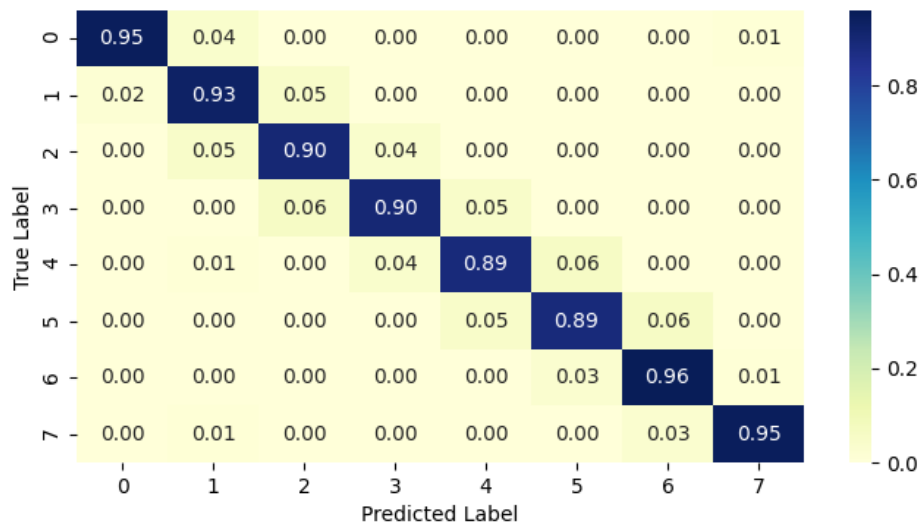


Figure 4.2. Confusion matrix for the music valence classifier. A normalized confusion matrix for the music valence classifier, showing the percentage of times that each label l_i (rows) was predicted as each label l_j (columns).

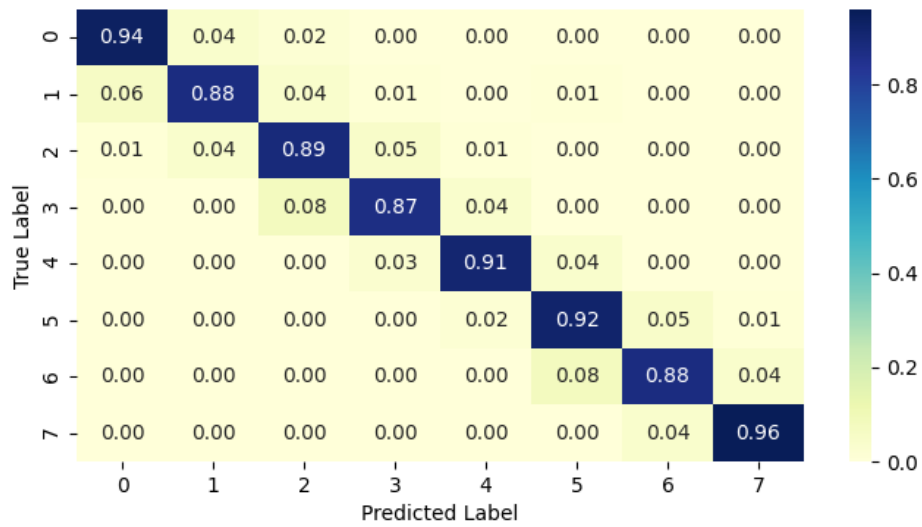


Figure 4.3. Confusion matrix for the music arousal classifier. A normalized confusion matrix for the music arousal classifier, showing the percentage of times that each label l_i (rows) was predicted as each label l_j (columns).

4.3.1.1 Image Classifier

For the classification of image emotions, using the presented MVSO subset, the accuracy obtained by the neural network in the test set was 71%, with $N = 4$ classes (quadrants) for the emotion curve. We show in Figure 4.1 the confusion matrix for the image classifier in the test set. The confusion matrix shows the percentage of times a label l_i (rows of the table) was predicted as a label l_j (columns of the table). Thus, higher values on the diagonal indicate a higher accuracy.

4.3.1.2 Music Classifier

For the classification of music emotions, using the DEAM dataset, the accuracy obtained by the neural network in the test set is 92% for valence and 91% for arousal, each label discretized at $N = 8$ levels, totaling $N^2 = 64$ classes for the emotion curve. We show in Figures 4.2 and 4.3 the confusion matrix for the image classifier in the test set, for valence and arousal labels, respectively.

4.3.2 Optimal Path Selection

We evaluate the use of our and other two simple optimal path selection approaches for performing the curve matching. The first is a greedy optimization method, and the second is the dynamic time warping (DTW) approach, both presented below.

4.3.2.1 Baselines

Greedy Optimization Method. This method greedily selects the next video frame with the maximum similarity for every song segment until it reaches the last segment. Specifically, given the emotion curves X and Y , for each $y_k, k \in \{1, 2, \dots, S\}$ the method seeks the next frame, v_l , to store in the path by computing $l = \operatorname{argmin}_{i=l}^{l+w} x_i$, where l stores the frame index of the last selected frame, initially set to $l = 1$, and w is the maximum frame skipping.

Dinamic Time Warping (DTW). This is an algorithm for measuring and aligning similarity between two temporal sequences, which may vary in speed [Müller, 2007]. To maximize the similarity of the input curves, the original DTW version may repeat video frames. It occurs because during the tracing of the optimal path, the algorithm had vertical transitions in similarity the matrix, which would mean repeating the same frame of the video in some transitions, implying that the video is stopping at certain times. Since this is not allowed in a hyperlapse, we adapted the method to our problem

Table 4.2. Optimal path selection evaluation. Comparison between the different optimization methods for frame sampling (best in bold).

Video	Emotion Score \uparrow			Speedup Ratio \downarrow			FID-Score \downarrow		
	<i>Greedy</i>	<i>DTW</i>	<i>Ours</i>	<i>Greedy</i>	<i>DTW</i>	<i>Ours</i>	<i>Greedy</i>	<i>DTW</i>	<i>Ours</i>
Berkeley1	0.74	0.74	0.77	1.23	1.03	1.00	22.06	22.14	3.30
Berkeley2	0.72	0.73	0.77	1.17	1.02	1.00	26.75	27.86	5.40
Bike3	0.72	0.72	0.76	1.16	1.02	1.00	16.75	18.10	5.04
CityWalk1	0.70	0.70	0.71	1.09	1.02	1.00	12.64	13.01	1.75
MontOldCity1	0.74	0.75	0.77	1.08	1.04	1.00	15.47	15.58	3.10
NatureWalk1	0.71	0.71	0.73	1.08	1.03	1.00	15.57	15.55	2.73
StockHolm1	0.71	0.71	0.73	1.36	1.01	1.00	37.58	36.07	4.21
Walking4	0.73	0.73	0.76	1.08	1.02	1.00	14.40	15.32	2.74
Mean	0.72	0.72	0.75	1.16	1.02	1.00	20.15	20.45	3.53

by adding a constraint that forces the algorithm to never repeat frames, by avoiding vertical transitions in the optimal path tracing. This adaptation slightly reduces the quality of the results of the DTW but ensures that the generated video is a valid hyperlapse. We feed the algorithm with the X and Y curves, and it returns two new \hat{X} and \hat{Y} curves with similar sizes and maximized similarity.

4.3.2.2 Results

Table 4.2 shows the quantitative results obtained by our method and the two previously listed baselines (Greedy and DTW). We show the scores for three metrics: Emotion Similarity, Speedup Ratio and FID-Score, by running each method for the eight videos, each one with the five songs, both presented in Table 4.1. In Table 4.2, each column shows the average score of the five songs obtained when running the respective method for the videos presented in each line. For each video, with each song, the emotion similarity is calculated by Equation 4.1. In the same way, the Speedup Ratio is calculated by Equation 4.2. For the FID-Score metric, we used a python implementation, which measures the similarity between the original video and the accelerated video. These results are not using the song selection algorithm.

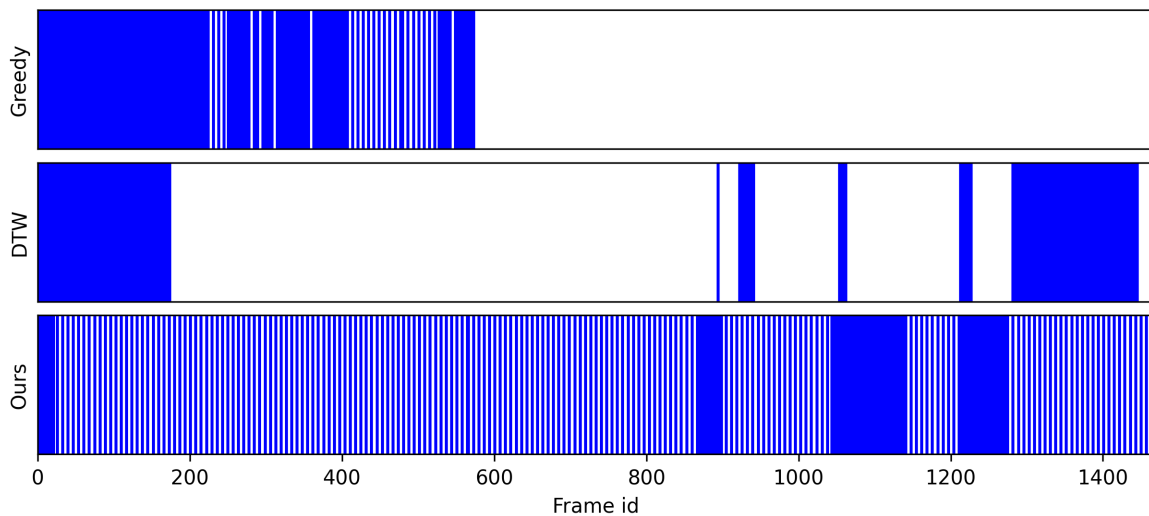


Figure 4.4. Selected frames distribution example. Selected frames for a part of the video "Bike3" with the song "In The End", in blue. The result of each method is shown on each line.

We can see that our method, when using the optimal path selection algorithm, achieved the best results across all metrics. The greedy approach maximizes the emotion similarities locally, leading to a significant error in the achieved speed-up, which might remove a big number of frames in the end of the video, also resulting in a high FID. The DTW, in its turn, seeks to find the best alignment globally, which creates many gaps between segments reducing the representability of the accelerated video regarding the original one, also resulting on a high FID. Although DTW tries to match the curves, the need to prevent it from repeating frames makes it obtain emotion similarities close to those obtained by the greedy approach. Our method manages to maximize the emotion similarities without repeating frames, reaching the optimal speed-up ratio by taking the exact amount of frames required by the song, also maintaining a balance between frame transitions by using the speed-up and inter-frame similarity cost matrices, guaranteeing a lower FID.

We show in Figure 4.4 an example of how the selected frames are distributed over time for the three optimal path selection methods. We show a part of the accelerated video "Bike3" for the song "In The End". The selected frames are in blue, and the result of each method is shown on each line. Note that the greedy method leaves a large gap at the end of the frameset, and the DTW method creates some gaps over time. Our method, in turn, has a more uniform selection of frames, avoiding gaps.

4.3.3 Comparison with Hyperlapse Creation Methods

We compare our methods against two hyperlapse baselines present in the literature: the Microsoft Hyperlapse (MSH) [Joshi *et al.*, 2015], and the extended version of the Sparse Adaptive Sampling (SASv2) [Silva *et al.*, 2021]. We used the desktop version of the MSH. For the SASv2, we set the hyperparameters as recommended by the authors. For our method and both baselines, the target speed-ups were defined as $Sp^* = \lceil F/S \rceil$, where F is the number of video frames and S is the number of song segments. For both our method and baselines, no additional stabilization algorithm was performed.

Table 4.3 presents the results for the comparison with the baselines, also without using the song selection algorithm. The columns show the Emotion Similarity, Speed-up Ratio, FID-Score, and Shaking Ratio values for each video in the dataset averaged over the five songs in Table 4.1. These metrics were calculated in the same way as was done for Table 4.2. Unlike the results of the previous section, in this one we are evaluating our entire hyperlapse creation methodology from start to finish, comparing it to other hyperlapse creation methods. In the previous section, we only assessed the optimal path selection algorithm to align emotion curves by comparing it to other curve alignment methods.

Our approach presents the best Emotion Similarity and Speed-up Ratio values while it is on par with the other methods in the Shaking Ratio. We accredit these results to our optimization algorithm that seeks to create a path that is visually stable, temporally continuous, and with high-quality emotion matching. Our approach samples exact S frames from the input video, it also presents the best Speed-up Ratio values in all cases. MSH, on the flip side, shows the worst values. The reason is that it favors optimizing the stability of frame transitions over achieving target speed-up.

We also observed that the SASv2 method obtained the best FID-Score in almost all cases. This is because this method is very focused on maintaining the temporal continuity between the story presented in the original video and in the accelerated video, which consequently makes the accelerated video more similar to the original, minimizing the FID-Score. However, our method also had low values for this metric.

Although, in general, the MSH presents the best Shaking Ratio values. Since the MSH algorithm neglects the video content and only optimizes the frame transition, their FID-Score values are worse than the other approaches by a significant margin. Moreover, the MSH algorithm includes image warping in its path smoothing and rendering step. This step may crop the image borders; therefore, increasing the FID-Score. In comparison to the MSH, our method presents FID-Score values closer to the SASv2 method, which is, by design, a content-based approach.

Table 4.3. Comparison with baselines. Comparison of our method and two other hyperlapse creation methods.

Video	Emotion Score \uparrow			Speedup Ratio \downarrow			FID-Score \downarrow			Shaking Ratio \downarrow		
	MSH	SASv2	Ours	MSH	SASv2	Ours	MSH	SASv2	Ours	MSH	SASv2	Ours
Berkeley1	0.73	0.72	0.79	1.19	1.01	1.00	28.90	4.30	6.82	0.02	0.02	0.02
Berkeley2	0.72	0.71	0.77	1.25	1.01	1.00	34.03	3.74	7.44	0.02	0.02	0.02
Bike3	0.71	0.71	0.77	1.02	1.01	1.00	28.31	3.02	6.21	0.03	0.05	0.05
CityWalk1	0.72	0.70	0.72	1.57	1.00	1.00	32.52	1.09	2.55	0.02	0.02	0.03
MontOldCity1	0.74	0.73	0.77	1.31	1.02	1.00	41.09	2.09	4.46	0.01	0.01	0.01
NatureWalk1	0.72	0.71	0.74	1.47	1.03	1.00	48.43	7.28	3.63	0.01	0.01	0.01
StockHolm1	0.71	0.70	0.74	1.13	1.16	1.00	23.99	7.66	5.13	0.02	0.01	0.02
Walking4	0.73	0.73	0.77	1.12	1.00	1.00	37.62	1.40	3.34	0.02	0.03	0.03
Mean	0.72	0.71	0.76	1.26	1.03	1.00	34.36	3.82	4.95	0.02	0.02	0.02

4.4 Qualitative Evaluation

In this section, we present several qualitative results obtained by our method. First, we show examples of emotion classifications obtained by the video classifier. Then, we show examples focused on the emotion curves matching, comparing our method with the other hyperlapse creation methods presented in Table 4.3, without the use of the song selection algorithm. Then, we show results using the song selection algorithm, which improves the emotion similarity in the final hyperlapses.

4.4.1 Emotion Classification Examples

Figures 4.5 to 4.8 show examples of the video emotion predictions for the videos used in the experiments. The selection of songs to be shown was random but trying to show all songs at least once. In each figure, we show the continuous video emotion curves with each axis scaled to a range of -1.0 to $+1.0$. We also show some video frames pointing to the predicted coordinate in the valence-arousal plane. In general, as can be seen in the figures, we observed that scenes with cars and smiling people are classified as high valence and high arousal. Nature scenes are classified with high valence and low arousal. Scenes of enclosed places and animals are classified as low valence and high arousal. The cases with low valence and low arousal are sporadic, but they are usually also scenes of closed places and walls.

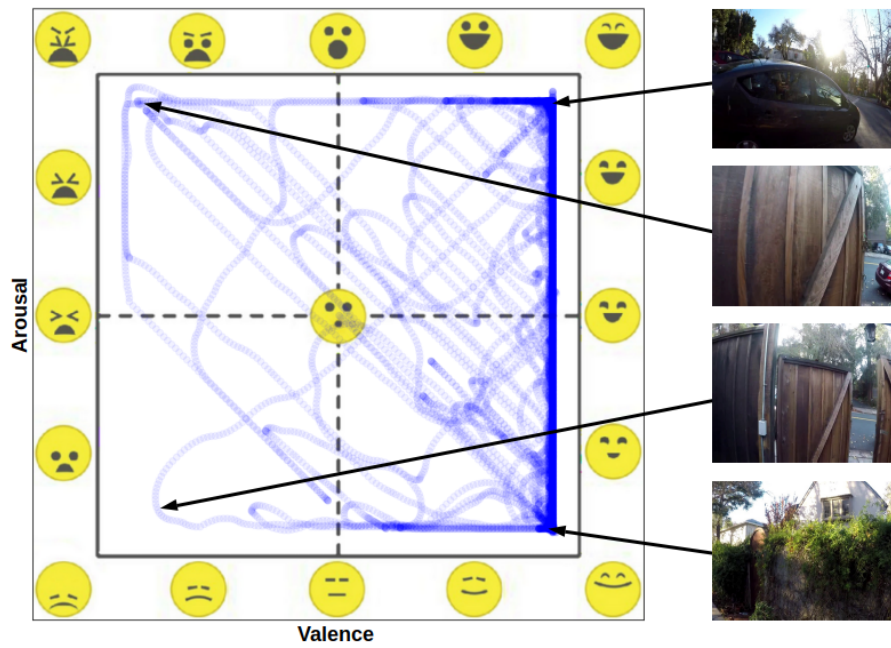


Figure 4.5. Emotion classification example 1. Emotion classification example for video Berkeley1.

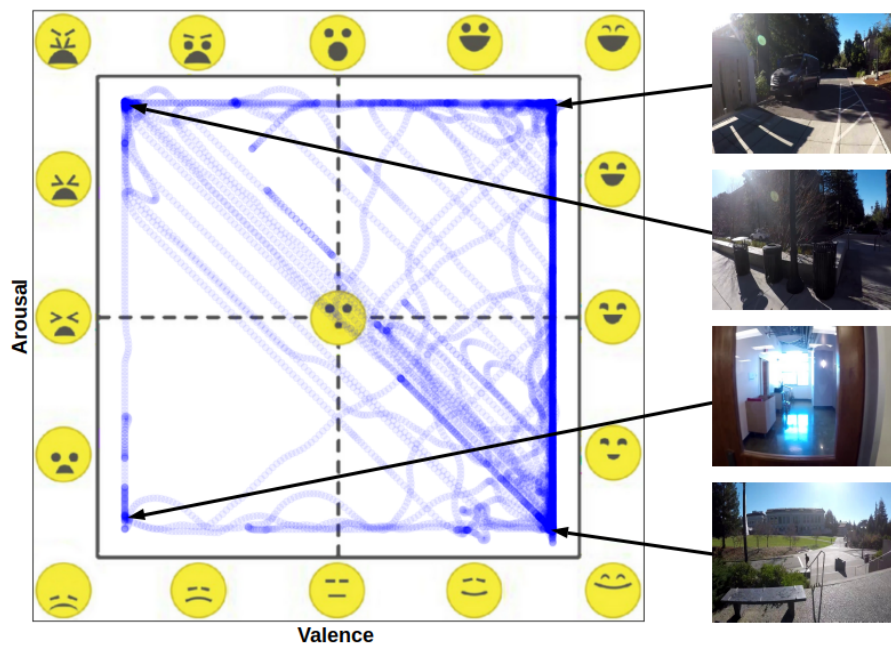


Figure 4.6. Emotion classification example 2. Emotion classification example for video Berkeley2.

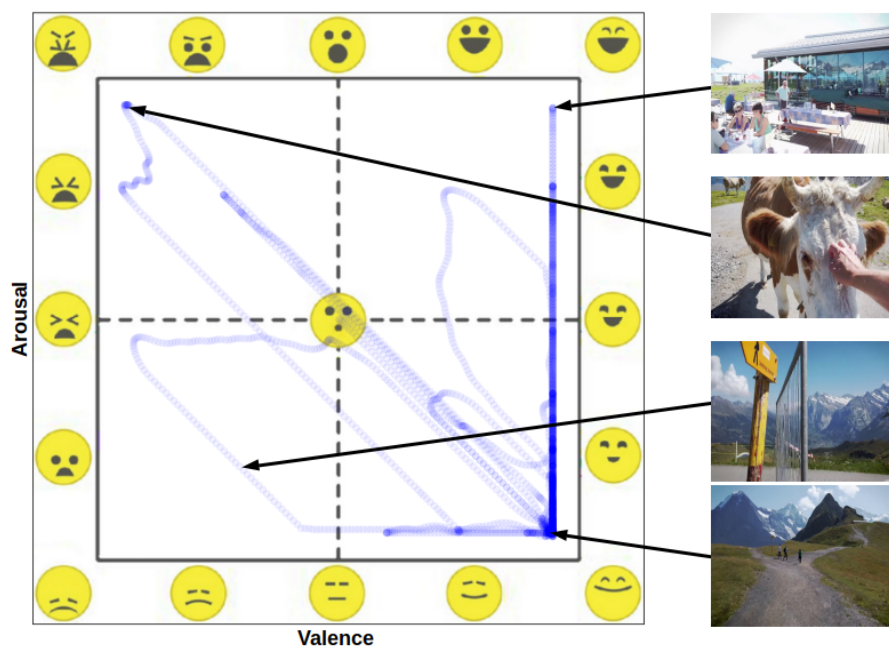


Figure 4.7. Emotion classification example 3. Emotion classification example for video CityWalk1.

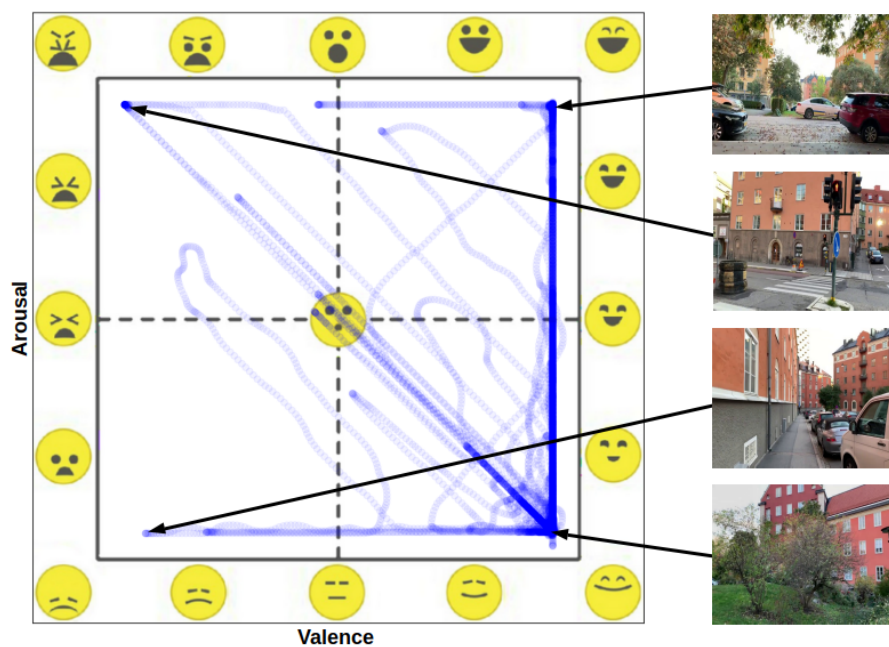


Figure 4.8. Emotion classification example 4. Emotion classification example for video StockHolm1.

4.4.2 Comparison with Hyperlapse Creation Methods

Figures 4.9 to 4.16 show examples of qualitative comparisons with the baselines presented in Table 4.3, one figure for each video. In each figure, we show the valence and arousal curves of the music and the accelerated video and the emotion similarity curves. At the top, we show the continuous music emotion curve in red, with greater intensity in the regions where there was a greater similarity with the continuous emotion curve of the video (the more red, the higher the similarity). At the bottom, we show the valence and arousal curves separately, the music curves in green, the video curves in blue, and the similarity curves in red. We also show the average emotion similarity of the entire curve. The results of each method are displayed in each column, respectively.

We can see that our method presents a distribution with higher intensities in the valence-arousal plane, indicating a higher matching in the induced emotions. MSH and SASv2, on the other hand, have a lower concentration of correct matching. Looking only at the valence and arousal curves, it is not possible to notice much difference since it is small, but we can see that the average similarity is more significant in our method in all cases. Also, note that on the valence-arousal plane, our approach, in most cases, resulted in more red dots, indicating that the emotion induced by the music was more similar to the video in our method than in the other methods.

Specifically, the results with more significant differences can be seen in Figures 4.9, 4.10, 4.11, 4.14, 4.15, and 4.16. Our method tends to widen or narrow some regions of the video curve in some cases or move some regions in other cases to make it more similar to the music curve, which doesn't happen in baselines, which have uniform frame sampling. This is achieved by just removing frames from the video without touching the music. Note that the music curves are always the same in all methods.

We can notice that our method does not increase the similarity of the curves so much in some cases because in these cases the music curve is very different from the video curve. This way, the algorithm can almost find a set of frames that maximize the similarity of the curves. We emphasize that as there are several restrictions during the video acceleration process, it is difficult to maximize the similarity of the emotion, which makes the average similarities very close to those of the baselines in some cases. We also emphasize that the purpose of the method is not only to match the curves, but also to guarantee the visual quality of the video.

Specifically, the results with lower differences can be seen in Figures 4.12, and 4.13. In these cases, the mean emotion similarity was very similar in both methods, and the intensities in the valence-arousal plane had no notable difference in our method. It is also complicated to notice a difference in the valence and arousal curves.

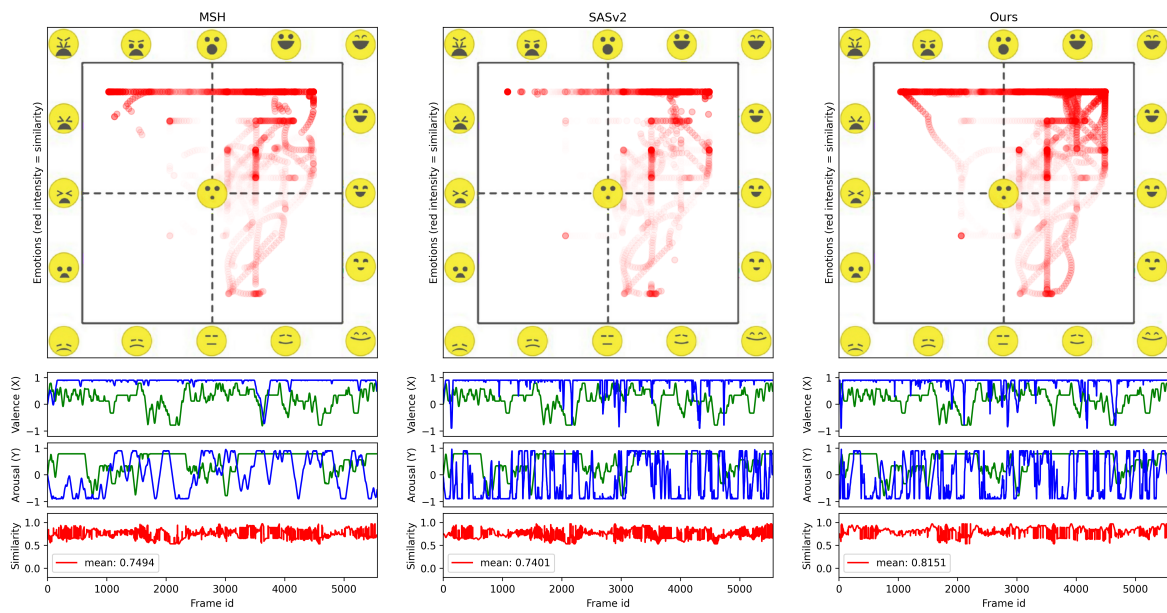


Figure 4.9. Qualitative comparison example 1. Emotion curves and similarities for video Berkeley1 with song Little Talks.

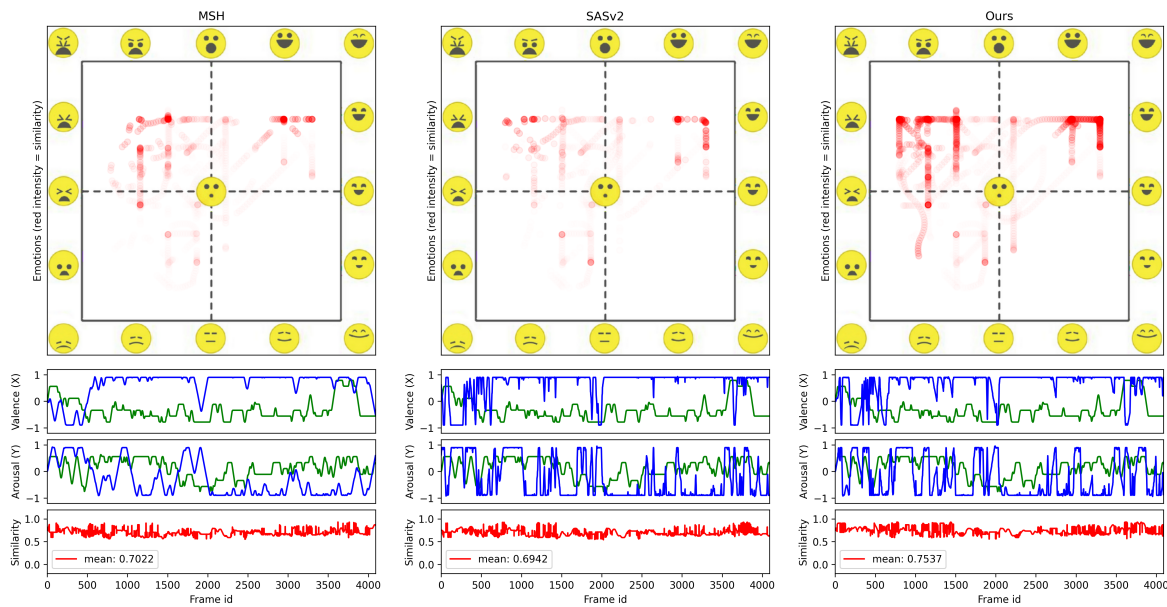


Figure 4.10. Qualitative comparison example 2. Emotion curves and similarities for video Berkeley2 with song Onward To Freedom.

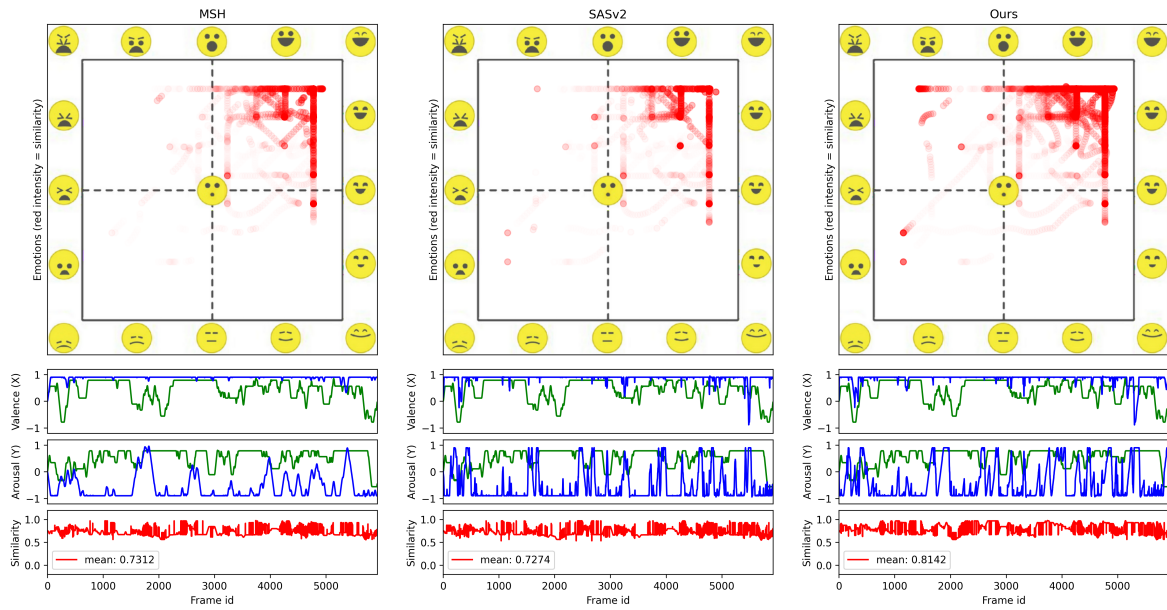


Figure 4.11. Qualitative comparison example 3. Emotion curves and similarities for video Bike3 with song In The End.

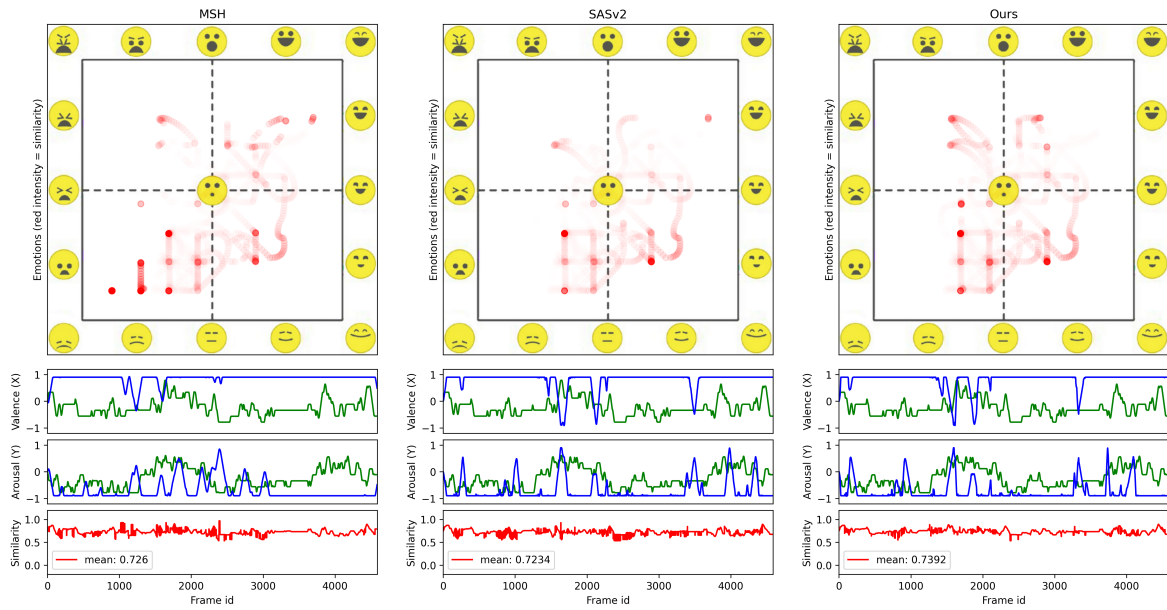


Figure 4.12. Qualitative comparison example 4. Emotion curves and similarities for video CityWalk1 with song My Immortal.

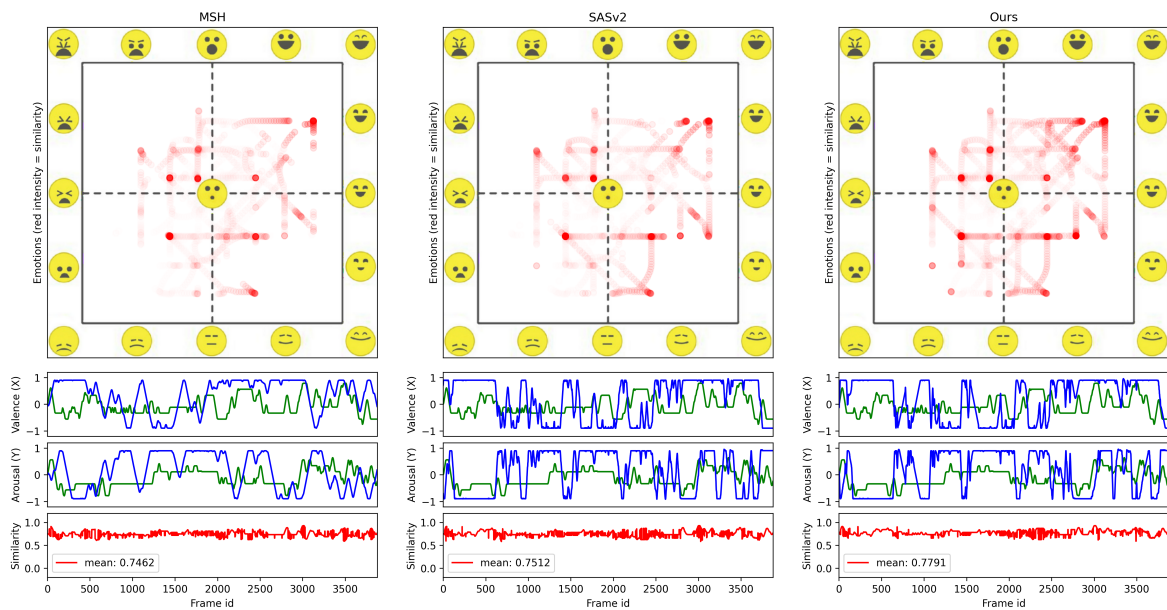


Figure 4.13. Qualitative comparison example 5. Emotion curves and similarities for video MontOldCity1 with song Last To Know.

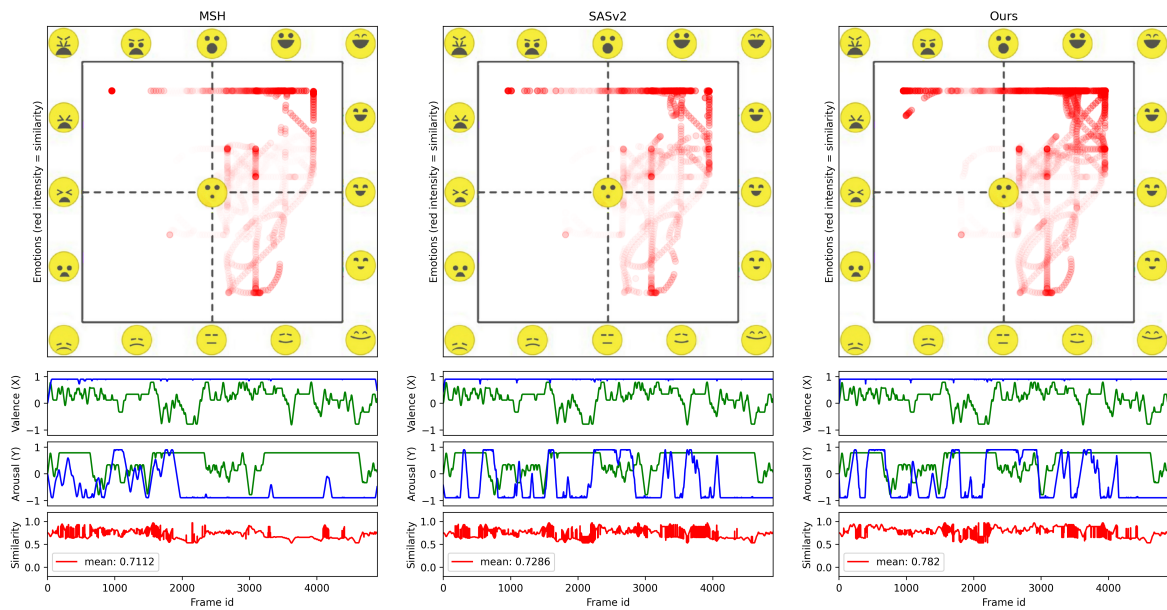


Figure 4.14. Qualitative comparison example 6. Emotion curves and similarities for video NatureWalk1 with song Little Talks.

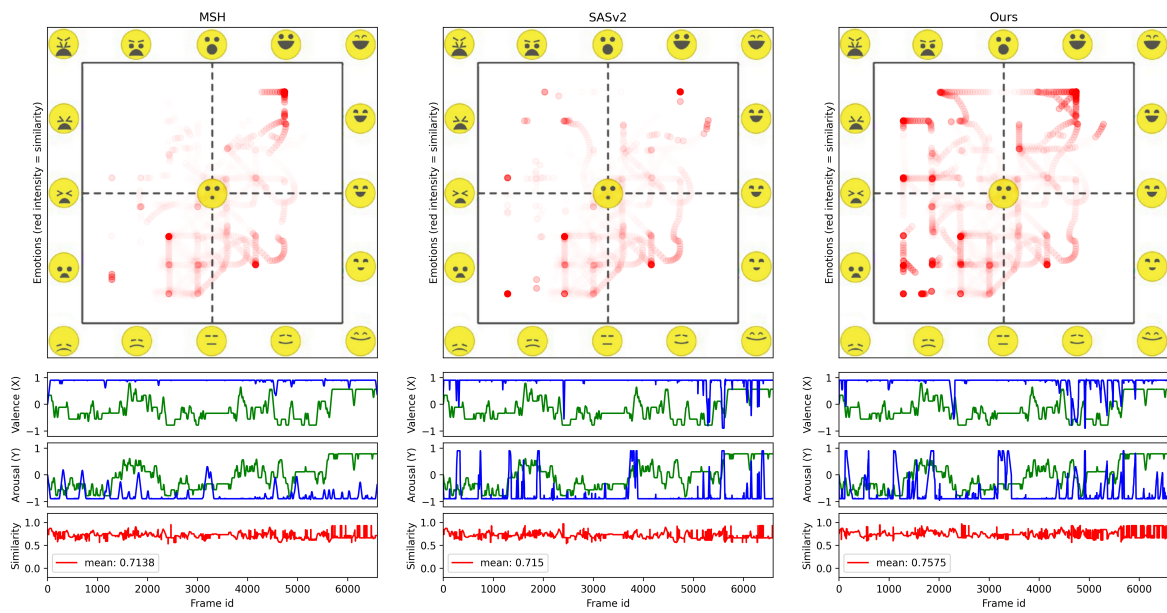


Figure 4.15. Qualitative comparison example 7. Emotion curves and similarities for video StockHolm1 with song My Immortal.

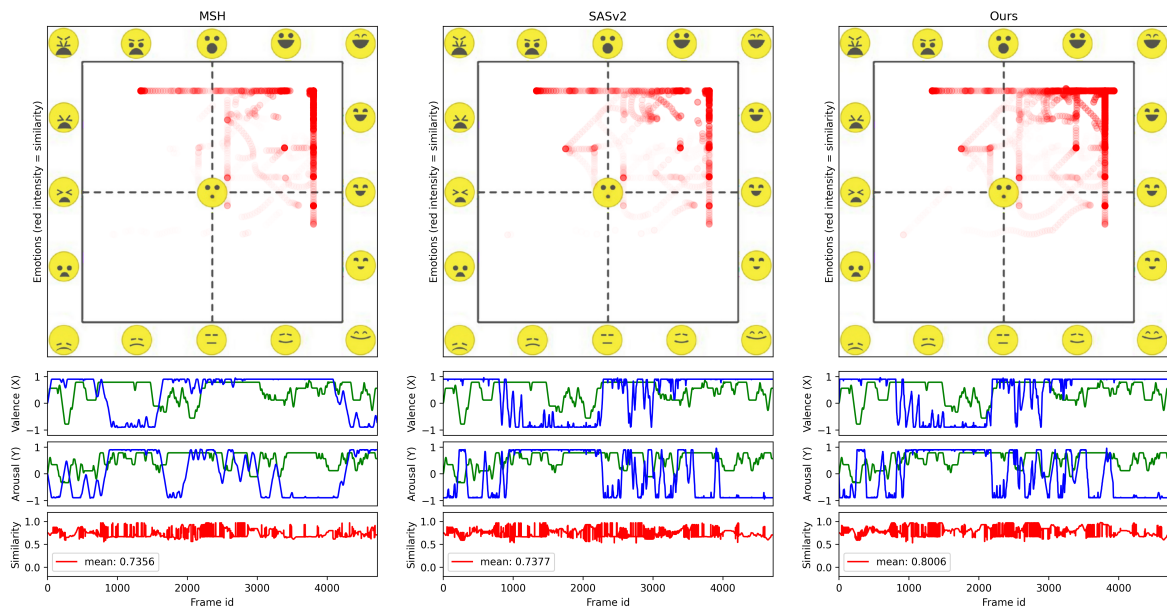


Figure 4.16. Qualitative comparison example 8. Emotion curves and similarities for video Walking4 with song In The End.

Table 4.4. Results with song selection. Comparison of emotion similarities using and not using the song selection algorithm.

Video	Emotion Similarity \uparrow		
	Mean of 5 Songs	Best of 5 Songs	Best of 1000 Songs
Berkeley1	0.79	0.82	0.90
Berkeley2	0.77	0.81	0.89
Bike3	0.77	0.81	0.91
CityWalk1	0.72	0.74	0.84
MontOldCity1	0.77	0.78	0.86
NatureWalk1	0.74	0.79	0.89
StockHolm1	0.74	0.76	0.89
Walking4	0.77	0.79	0.84
Mean	0.76	0.79	0.88

4.4.3 Results Using Song Selection

In this subsection, we show results using the song selection algorithm. To perform the selection, we run the algorithm using as input a set of 1000 songs of varied styles, such as Rock, Country, Folk, Classic, and others.

Table 4.4 shows the comparison of our method using and not using the song selection algorithm with 1,000 songs. In first column we show the mean emotion similarity using the five songs presented in Table 4.1. In second column we show the best emotion of these five songs. And in third column we show the emotion similarity using the song selection algorithm with 1,000 songs. We can observe that with the use of the algorithm, our method is able to achieve similarities above 0.9, proving that it is possible to find songs that match the video very well, allowing the algorithm to also be used for recommending music for video.

Figures 4.17 to 4.24 show qualitative results using the song selection algorithm. In each figure, we show the valence and arousal curves of the music and the accelerated video and the emotion similarity curves. At the left, we show the continuous music emotion curves in the valence-arousal plane, blue for video and green for audio. At the right, we show the valence and arousal curves separately, the music curves in green, the video curves in blue, and the similarity curves in red. We also show the average

emotion similarity of the entire curve.

In several cases, we get similarities higher or very close to 0.90, such as in Figures 4.17, 4.18, 4.19, 4.22, and 4.23. The high similarity can be noted just by looking at the curves. It is observed that in the valence-arousal plane, both curves tend to remain in the same quadrants. It is also noted that the separate valence and arousal curves are much more similar than in the results presented in the previous section, without the song selection.

In some cases, as in Figures 4.17, 4.19, and 4.22, the valence or arousal curve of the most similar song has few variations, remaining most of the time at one extreme, that is because the curve of the original video also tended to be more concentrated at that extreme. As a result, by speeding up the video for this song, the video curve becomes more constant since the algorithm is trying to make it look similar to the music curve, which has slight variation.

There are also cases in which parts of the valence and arousal curves seem to be complementary, one being the inverse of the other, as in Figure 4.20 and 4.21 from frame 1,200. However, this is just a coincidence that occurs when valence is the opposite of arousal, which is expected because there are only two discretization levels for each label.

The smallest similarities occur in Figures 4.20 and 4.24, where the average similarity is close to 0.84, but still higher than all results obtained without the song selection. In these cases, the curves are more visually different, but you can still notice that they tend to stay in the same quadrants in general.

In general, comparing these results with those of the previous section, we can observe that our method, when having a large set of songs as input, manages to improve even more the emotion curves matching, being able to recommend a good song for the video, besides making acceleration.

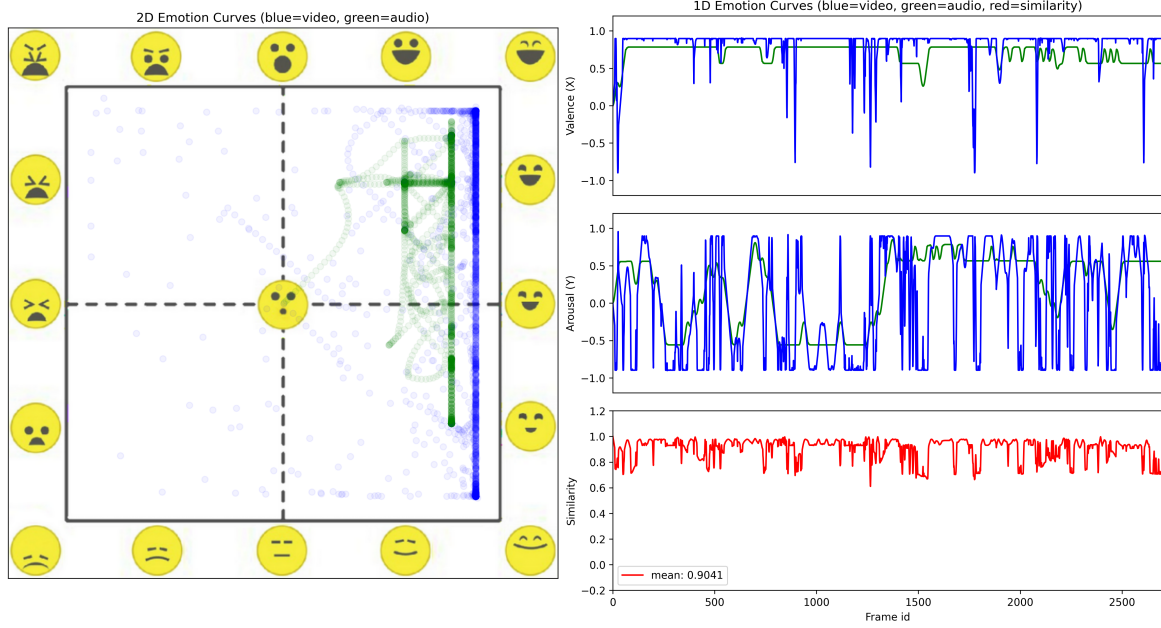


Figure 4.17. Song selection result example 1. Emotion curves for video Berkeley1 using song selection.

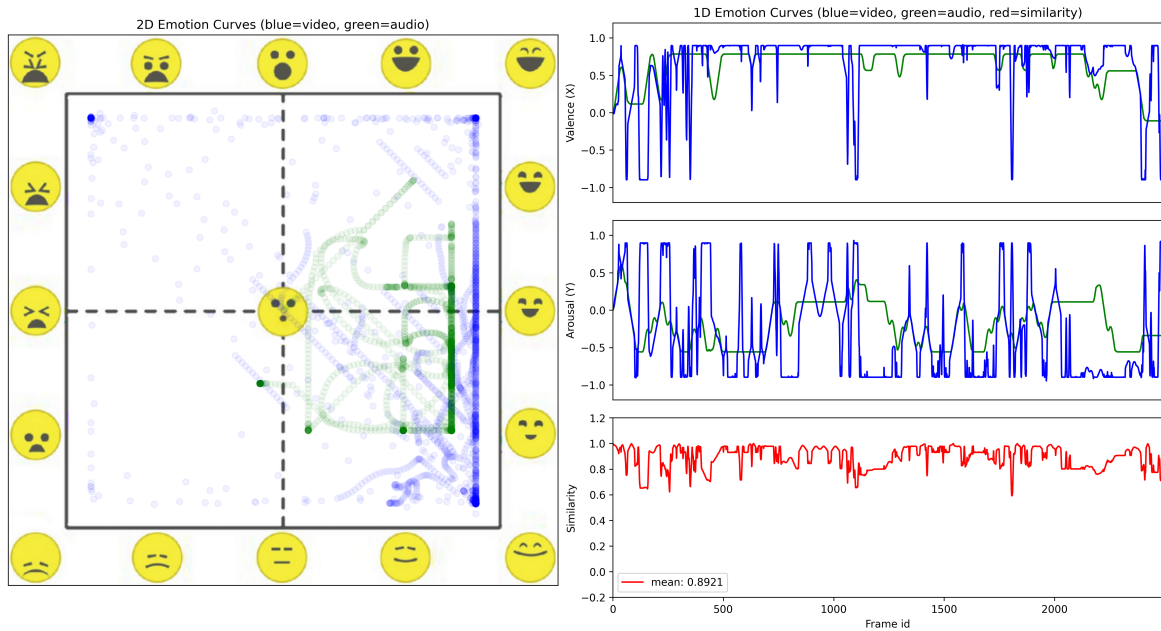


Figure 4.18. Song selection result example 2. Emotion curves for video Berkeley2 using song selection.

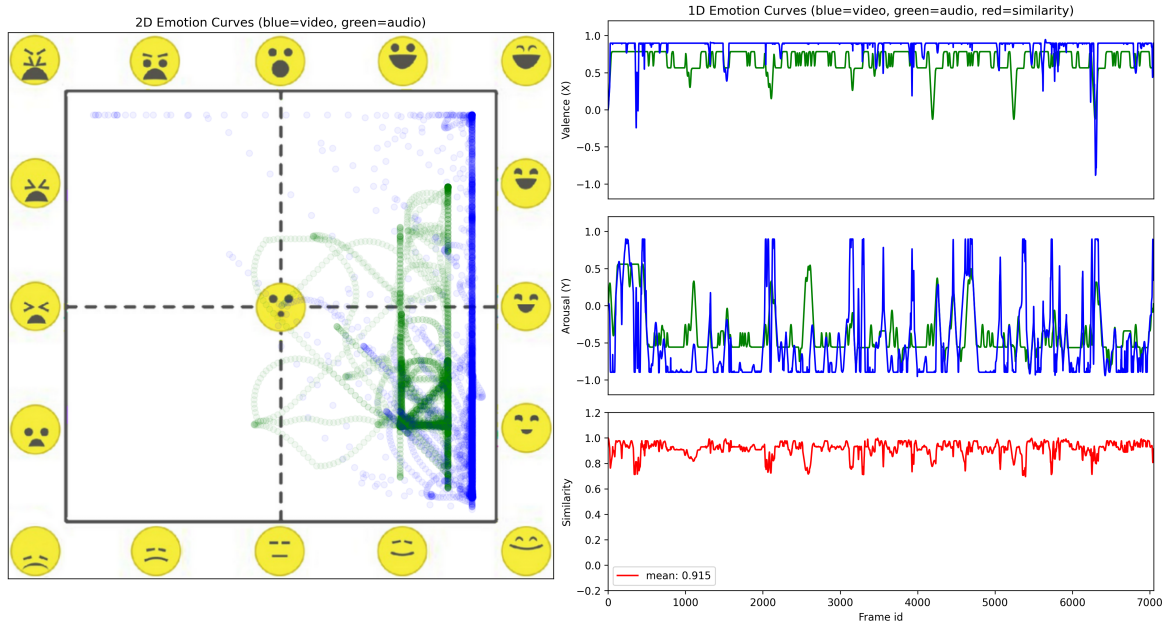


Figure 4.19. Song selection result example 3. Emotion curves for video Bike3 using song selection.

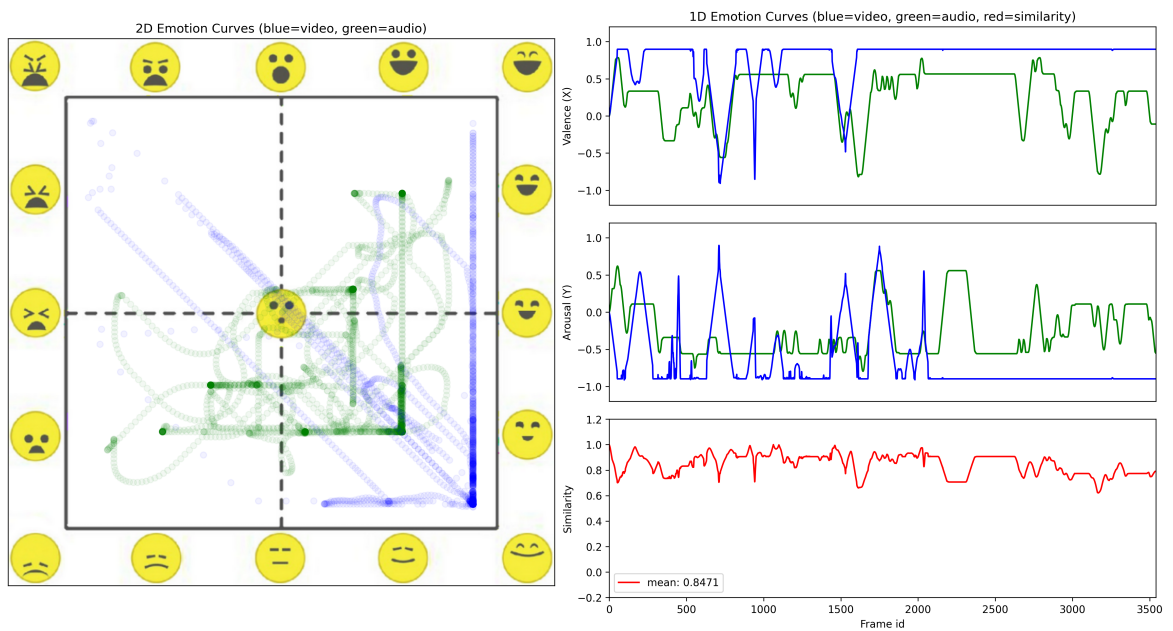


Figure 4.20. Song selection result example 4. Emotion curves for video CityWalk1 using song selection.

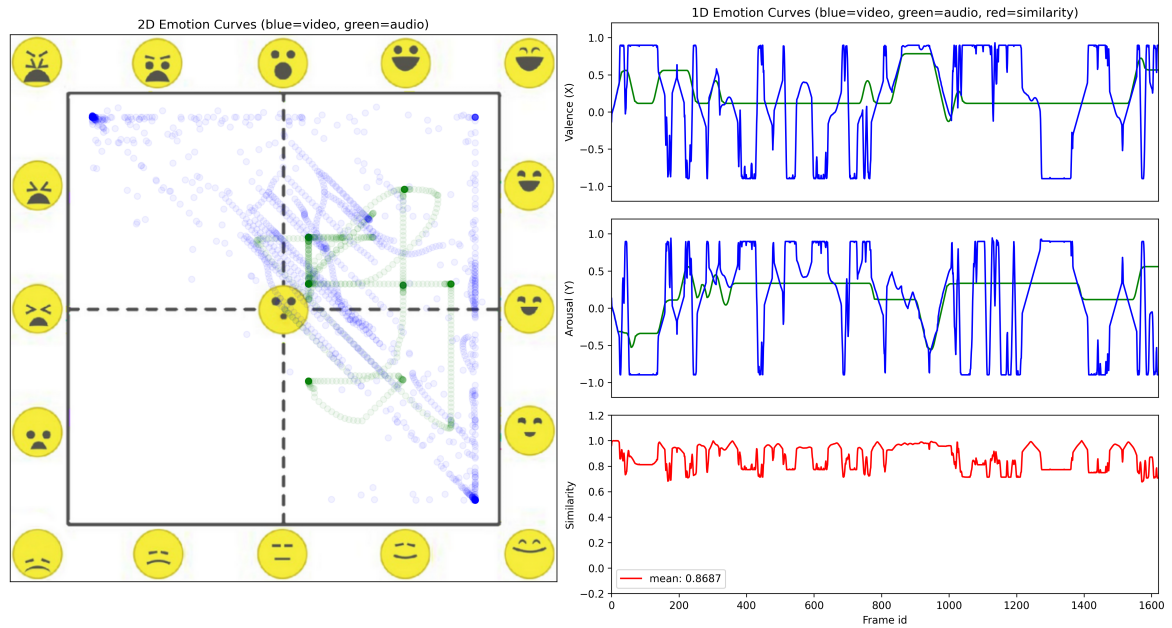


Figure 4.21. Song selection result example 5. Emotion curves for video MontOldCity1 using song selection.

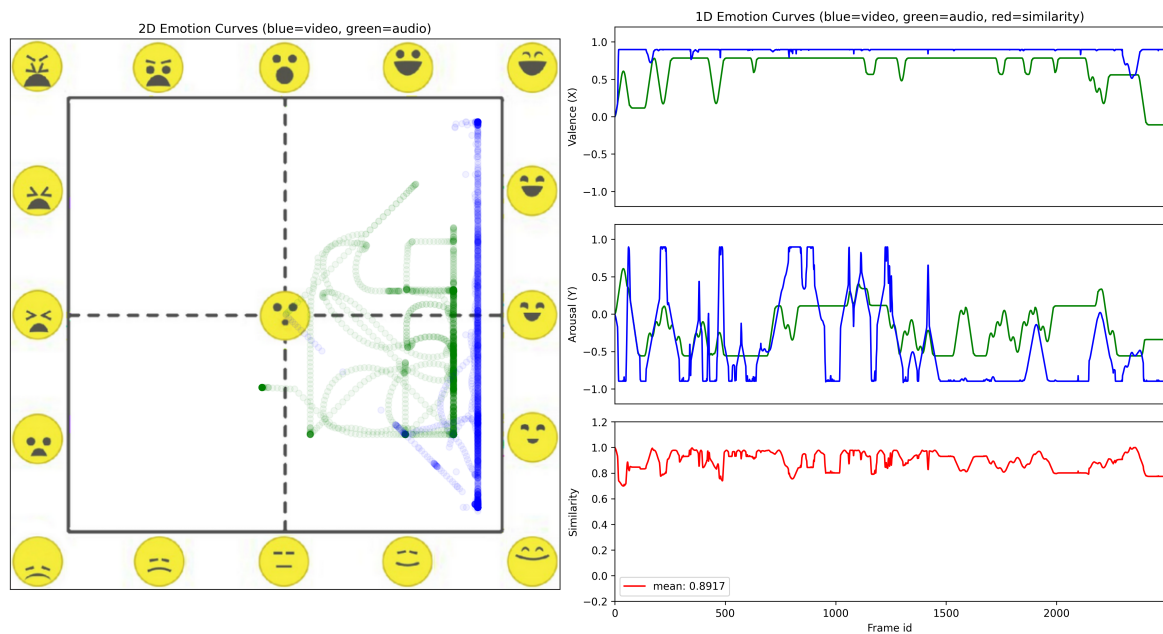


Figure 4.22. Song selection result example 6. Emotion curves for video NatureWalk1 using song selection.

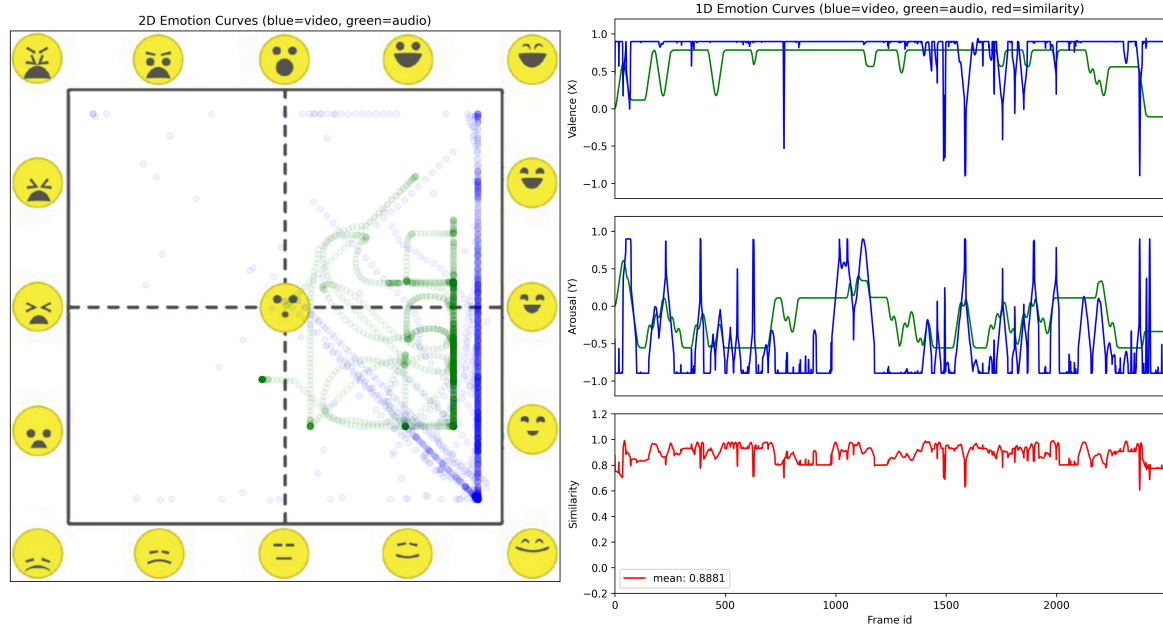


Figure 4.23. Song selection result example 7. Emotion curves for video Stockholm1 using song selection.

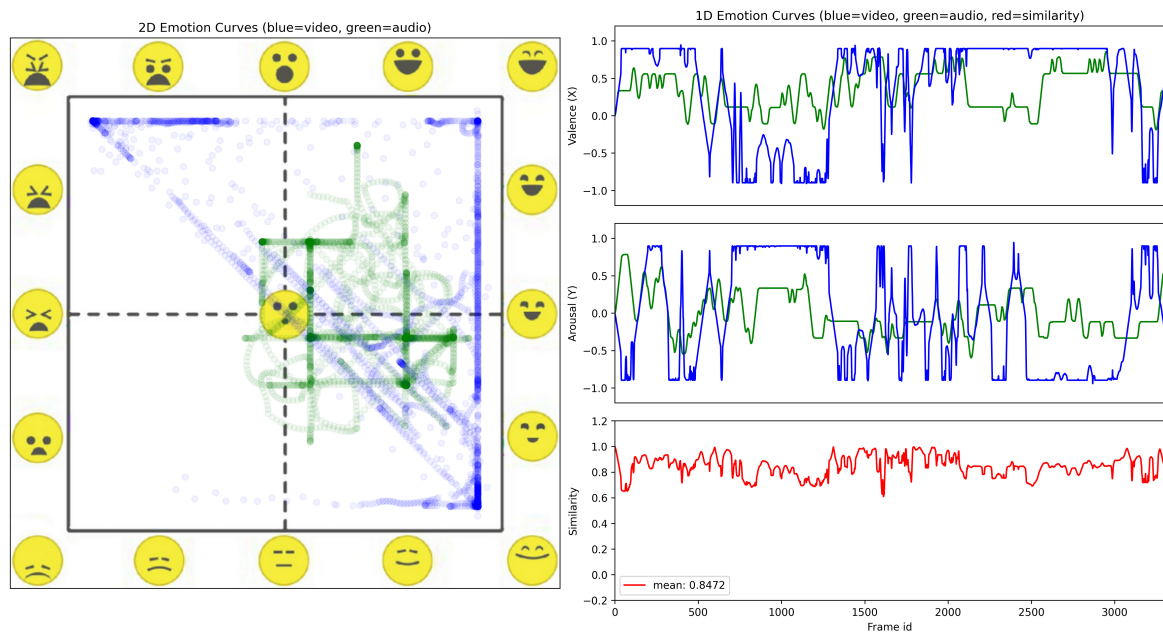


Figure 4.24. Song selection result example 8. Emotion curves for video Walking4 using song selection.

In this chapter, we presented several experiments performed to evaluate our method, comparing it with other methods present in the literature and the results obtained for each experiment. With the implemented models for the classification of emotions in images and music, we managed to create continuous emotion curves for the input video and the input song. And with the implemented optimal path selection algorithm, we were able to speed up the video by matching the emotion curves of the music with that of the video and maximizing the visual quality of the video.

The quantitative results showed that our method is able to maximize the similarity of emotion curves more than the baselines without losing visual quality. In addition, we quantitatively prove the effectiveness of the trained neural networks through the presented confusion matrices. We also confirmed that with the use of the song selection algorithm, the similarity of the curves can become even more significant in the accelerated video. Through the qualitative results, it was possible to visually verify that our method succeeds in making the emotion curves more similar for all videos and songs, allowing the video to be accelerated to combine well the emotions induced by the video and music throughout the time.

In summary, we argue that the proposed methodology achieved the objectives presented in the introduction once the effectiveness of our method was proven through the performed experiments.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

The development of this work was motivated by the need to speed up egocentric videos, which are usually long and tiring to watch, considering not only the visual information in the videos but also the acoustic information in the music that the user wants to insert into the accelerated video, making it more enjoyable to watch. Based on recent work on recognizing emotion in music and images, we decided to use emotion representation models to combine the video content with the music content. Thus, we introduced in this work the new task of accelerating first person videos including the alignment of emotions induced by visual and acoustic signals.

To solve the problem of speeding up the video, including background music, we developed a new method to create continuous emotion curves for music and video over time. We also developed an optimization algorithm to select the best subset of video frames, which maximizes the emotion similarity while also maintaining the visual quality of the accelerated video.

We extracted features from the music segments and video frames to create the emotion curves and use artificial neural networks to classify the emotions induced from these features. We represented emotions in a plane where the x-axis represents valence, and the y-axis represents arousal, both for music and video, which allowed us to calculate the similarity of data from different nature by measuring only the Euclidean distance between two points in the emotion plane. However, to classify the emotions, it was necessary to create discrete representations of the valence and arousal values labeled in the audio and image datasets, in order to transform the regression problem into a classification problem. The dataset used to train the audio classifier allowed a more detailed discretization, totaling 64 possible points in the valence-arousal

plane, for each music segment. The dataset used to train the image classifier, on the other hand, required a conversion to the valence-arousal plane, allowing the data to be distributed in only four classes, corresponding to the four quadrants of the plane. It reduced the quality of the continuous curve generated for the video, when compared to the continuous curve generated for music.

The proposed optimization algorithm selects the best subset of frames in the video based on the similarity of the generated emotion curves and similarities between video frames, seeking to guarantee the matching of the curves without losing visual quality. We observed that the proposed optimization method performed better than the greedy method and the DTW to accelerate the video and match the emotion curves. One difficulty in implementing the optimization method was that the dynamic cost matrix takes up many memory spaces, bringing the need to use sparse representations, in addition to limiting the size of videos in experiments. Another problem was that the algorithm took a long time to process the videos, which the solution was the reduction of the maximum frame skipping in selecting the frames to be removed. Increasing the size of the maximum frame skipping could improve the quality of the results, but it would significantly increase the processing time. However, we did not consider the processing time in the evaluation of the proposed method.

From the quantitative results, we observed that the developed method manages to increase the similarity of the video and music emotion curves for most videos and obtains adequate values for the temporal continuity and video stabilization while also reducing the video to the exact size of the song. From the qualitative results, we observed that despite maximizing the average similarity of the emotion curves, it still have a lot of difference. This is because there are many restrictions in the optimization process, such as the size of the maximum frame skipping and the target length of the accelerated video. With the inclusion of the best song selection algorithm, it is possible to produce results in which the emotion curves are more similar, achieving similarities close to 0.9 from a set with 1,000 songs.

In general, the experiments show that the proposed method achieved superior performance in terms of video representability, required speed-up and emotional alignment for different videos and songs without losing the visual quality of the hyperlapse, compared to previous methods. The results show that it is possible to create a hyperlapse combining media of distinct nature according to their respective affective semantic.

5.2 Future Works

For future works, once we presented a new problem of creating a musical hyperlapse, there are many improvements that can be done to improve the quality of the results.

An important improvement that we can make is increasing the number of discretization levels in the video emotion classifications and performing the training of the neural networks again, using for example, 64 classes for both audio and video emotion representations in the valence-arousal plane. This will certainly require further study in the dataset used to train the image classifier, as well as improvements in the training of the used convolutional neural network. In another hand, instead of discretizing and then smoothing the curves, regressions could be made, seeking to maximize the accuracy of the emotion representation. This task would be a little more complex, but it would be possible to classify the music segments and video frames to continuous values, directly generating the continuous emotion curves.

Another way to compare the similarities between music and video, instead of using Russell's emotion model, is to create an embedding space where each video frame and music segment would be positioned as a point in a n -dimensional space, and the distance from each pair of points would define the similarity between them, the closer the more similar. This approach would also eliminate the need to convert the discrete curves into continuous ones since the similarities would be obtained directly from the input video and input music.

The optimization algorithm implementation can also be improved, with the main objective of reducing the consumption of space and time. To do that, a more detailed study on the optimization algorithm would be necessary, as well as on the way the data is pre-processed and stored. New experiments must be performed evaluating the memory consumption and time spent by the algorithm with different videos and in other contexts. This improvement is significant when we want to use the algorithm in longer videos, for example, with over one hour.

We can also consider another approach for inserting a song into the accelerated video, that is the usage of machine learning methods capable of automatically generating the music for the video. There are already many works on automatic music generation in the literature. These works combined with results on fast-forwarding videos can also be a way to solve the problem of creating the musical hyperlapse.

Another idea is to also consider the acoustic content recorded in the original video, and calculate the emotion induced by it as well. That way we could include the music in the accelerated video without removing the audio from the original video.

Bibliography

- Ahn, J., Gobron, S., Silvestre, Q., and Thalmann, D. (2010). Asymmetrical facial expressions based on an advanced interpretation of two-dimensional russells emotional model.
- Aljanaki, A., Yang, Y.-H., and Soleymani, M. (2017). Developing a benchmark for emotional analysis of music. *PLOS ONE*, 12(3):1–22.
- Alpher, A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.
- Borth, D., Ji, R., Chen, T., Breuel, T., and Chang, S. (2013). Large-scale visual sentiment ontology and detectors using adjective noun pairs. pages 223–232.
- Chowdhury, S., Vall, A., Haunschmid, V., and Widmer, G. (2019). Towards explainable music emotion recognition: The route via mid-level features. *International Society for Music Information Retrieval Conference*.
- Dalmia, V., Liu, H., and Chang, S. (2016). Columbia mvso image sentiment dataset. *ArXiv*, abs/1611.04455.
- Dan-Glauser, S. E. and Scherer, R. K. (2011). The geneva affective picture database (gaped): a new 730-picture database focusing on valence and normative significance. *Behavior Research Methods*, pages 468–477.
- Dong, Y., Yang, X., Zhao, X., and Li, J. (2019). Bidirectional convolutional recurrent sparse network (bcrsn): An efficient model for music emotion recognition. *IEEE Transactions on Multimedia*, 21(12):3150–3163.
- Furlan, V. S., Bajcsy, R., and Nascimento, E. R. (2018). Fast forwarding egocentric videos by listening and watching. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Sight and Sound*, pages 2504–2507, University of California Berkeley, USA. IEEE Computer Society.

- Halperin, T., Poleg, Y., Arora, C., and Peleg, S. (2018). Egosampling: Wide view hyperlapse from egocentric videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(5):1248–1259.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, University of California Berkeley, USA.
- Higuchi, K., Yonetani, R., and Sato, Y. (2017). Egoscanning: Quickly scanning first-person videos with egocentric elastic timelines. In *SIGGRAPH Asia 2017 Emerging Technologies*, SA '17, New York, NY, USA. Association for Computing Machinery.
- Hong, S., Im, W., and Yang, H. S. (2018). Cbvmr: Content-based video-music retrieval using soft intra-modal structure constraint. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, ICMR '18*, New York, NY, USA. Association for Computing Machinery.
- Jia, J., Wu, S., Wang, X., Hu, P., Cai, L., and Tang, J. (2012). Can we understand van goghs moods learning to infer affects from images in social networks. *ACM International Conference on Multimedia*.
- Joshi, D., Datta, R., Fedorovskaya, E., Luong, Q., Wang, J. Z., Li, J., and Luo, J. (2011). Aesthetics and emotions in images. *IEEE Signal Processing Magazine*, 28(5):94–115.
- Joshi, N., Kienzle, W., Toelle, M., Uyttendaele, M., and Cohen, M. F. (2015). Real-time hyperlapse creation via optimal frame selection. *ACM Trans. Graph.*, 34(4). ISSN 0730-0301.
- Karpenko, A. (2014). The technology behind hyperlapse from instagram. <https://instagram-engineering.com/the-technology-behind-hyperlapse-from-instagram-4aae8b5c0d0a>.
- Kopf, J., Cohen, M., and Szeliski, R. (2014). First-person hyperlapse videos. In *ACM Transactions on Graphics (Proc. SIGGRAPH 2014)*, volume 33. ACM - Association for Computing Machinery.
- Kuo, F.-F., Shan, M.-K., and Lee, S.-Y. (2013). Background music recommendation for video based on multimodal latent semantic analysis. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

- Lai, W.-S., Huang, Y., Joshi, N., Buehler, C., Yang, M.-H., and Kang, S. B. (2017). Semantic-driven generation of hyperlapse from 360 video. *ArXiv*, abs/1703.10798.
- Li, B.-L. (2002). Fractal dimensions. *Encyclopedia of Environmetrics*.
- Lu, L., Liu, D., and Zhang, H.-J. (2006). Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):5–18.
- Mathiasen, A. and Hvilshøj, F. (2020). Fast fréchet inception distance. Aarhus University.
- Mittal, T., Guhan, P., Bhattacharya, U., Chandra, R., Bera, A., and Manocha, D. (2020). Emoticon: Context-aware multimodal emotion recognition using frege’s principle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Müller, M. (2007). Dynamic time warping. *Information Retrieval for Music and Motion*, 2:69–84.
- Ogawa, M., Yamasaki, T., and Aizawa, K. (2017). Hyperlapse generation of omnidirectional videos by adaptive sampling based on 3d camera positions. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2124–2128.
- Okamoto, M. and Yanai, K. (2014). Summarization of egocentric moving videos for generating walking route guidance. pages 431–442.
- Panda, R., Malheiro, R. M., and Paiva, R. P. (2018). Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, pages 1–1.
- Panda, R., Malheiro, R. M., and Paiva, R. P. (2020). Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, pages 1–1.
- Plutchik, R. (1980). *Emotion, a Psychoevolutionary Synthesis*. Harper & Row. ISBN 9780060452353.
- Poleg, Y., Halperin, T., Arora, C., and Peleg, S. (2015). Egosampling: Fast-forward and stereo for egocentric videos. pages 4768–4776.
- Ramos, W. L. S., Silva, M. M., Araujo, E. R., Neves, A. C., and Nascimento, E. R. (2020). Personalizing fast-forward videos based on visual and textual features from social network. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3260–3269.

- Ramos, W. L. S., Silva, M. M., Campos, M. F. M., and Nascimento, E. R. (2016). Fast-forward video based on semantic extraction. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3334–3338.
- Rani, S. S. (2016). Power spectral density in communication systems.
- Sasaki, S., Hirai, T., Ohya, H., and Morishima, S. (2013). Affective music recommendation system reflecting the mood of input image. In *2013 International Conference on Culture and Computing*, pages 153–154.
- Scherer, K. R. (2005). What are emotions? and how can they be measured? *Social Science Information*, 44(4):695–729.
- Silva, M., Ramos, W., Campos, M., and Nascimento, E. R. (2021). A sparse sampling-based framework for semantic fast-forward of first-person videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):1438–1444.
- Silva, M. M., Ramos, W. L. S., Chamone, F. C., Ferreira, J. P. K., Campos, M. F. M., and Nascimento, E. R. (2018a). Making a long story short: A multi-importance fast-forwarding egocentric videos with the emphasis on relevant objects. *Journal of Visual Communication and Image Representation*, 53:55–64. ISSN 1047-3203.
- Silva, M. M., Ramos, W. L. S., Ferreira, J. P. K., Campos, M. F. M., and Nascimento, E. R. (2016). Towards semantic fast-forward and stabilized egocentric videos. In *International Workshop on Egocentric Perception, Interaction and Computing (EPIC) at European Conference on Computer Vision (ECCV)*, pages 557–571, Amsterdam, NL.
- Silva, M. M., Ramos, W. L. S., Ferreira, J. P. K., Chamone, F. C., Campos, M. F. M., and Nascimento, E. R. (2018b). A weighted sparse sampling and smoothing frame transition approach for semantic fast-forward first-person videos. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2383–2392, Salt Lake City, USA.
- Solymani, M., Aljanaki, A., and Yang, Y.-H. (2018). DEAM: Mediaeval database for emotional analysis in music.
- Thammasan, N., Moriyama, K., Fukui, K.-i., and Numao, M. (2016). Continuous music-emotion recognition based on electroencephalogram. *IEICE Transactions on Information and Systems*, E99.D:1234–1241.

- Toet, A. and Erp, v. (2019). Emomadrid: An emotional pictures database for affect research.
- Toet, A. and van Erp, J. B. (2019). The emoji-grid as a tool to assess experienced and perceived emotions. *Psych*, 1(1):469--481. ISSN 2624-8611.
- Yang, Y., Lin, Y., Su, Y., and Chen, H. H. (2008). A regression approach to music emotion recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):448–457.
- Yao, T., Mei, T., and Rui, Y. (2016). Highlight detection with pairwise deep ranking for first-person video summarization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 982–990.
- Zhao, S., Gao, Y., Jiang, X., Yao, H., Chua, T.-S., and Sun, X. (2014). Exploring principles-of-art features for image emotion recognition. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, pages 47–56.
- Zhou Wang, Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.
- Zwicker, E. and Fastl, H. (2013). *Psychoacoustics: Facts and models*, volume 22. Springer Science & Business Media.