Dissertation on

"Video Trailer Generation using Multimodal Data Analysis"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE21CS461A – Capstone Project Phase - 2

Submitted by:

Nikhil Giridhar	**PES1UG21CS384**
Prajna R	**PES1UG21CS417**
Pranathi Praveen	**PES1UG21CS428**
Shreeja Rajesh	**PES1UG21CS564**

Under the guidance of

Dr. Surabhi Narayan
Professor
PES University

August – Dec, 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Video Trailer Generation using Multimodal Data Analysis'

is a bonafide work carried out by

Nikhil Giridhar	**PES1UG21CS384**
Prajna R	**PES1UG21CS417**
Pranathi Praveen	**PES1UG21CS428**
Shreeja Rajesh	**PES1UG21CS564**

in partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE21CS461A) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period August – Dec, 2024. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature	Signature	Signature
Dr. Surabhi Narayan	Dr. Mamatha H R	Dr. B K Keshavan
Professor	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **"Video Trailer Generation using Multimodal Data Analysis"** has been carried out by us under the guidance of Dr. Surabhi Narayan, Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August – Dec, 2024. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG21CS384	**Nikhil Giridhar**
PES1UG21CS417	**Prajna R**
PES1UG21CS428	**Pranathi Praveen**
PES1UG21CS564	**Shreeja Rajesh**

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Surabhi Narayan, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE21CS461A - Capstone Project Phase – 2.

I am grateful to the project coordinator, Dr. Priyanka H, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Mamatha H R, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

Manually crafting emotion-aware and attractive trailers is often cumbersome and challenging. There is a need to automatically generate intelligent and attractive trailers. Our project aims to address this problem by using multi-modal analysis integrating audio, video and other metadata. The purpose of our project is to explore and create novel approaches for audio-video processing to capture the “trailer-worthy” key moments from short movies and tailor them into visually attractive and emotionally aware trailers.

This project aims to develop a unique audio-guided video feature extraction technique which helps automatically generate an appropriate trailer when a short movie is given as input. This strategy ensures equal prominence is given to both auditory and visual features when getting the timestamps. The project also introduces “feature-fusion” which takes in the timestamps obtained from audio and video processing modules and gives an output of plausible “trailer-worthy” scenes. These scenes are ordered to craft an attractive and intelligent trailer capturing the essence of the short movie. We also aim to show that there is no need for synthetic content to generate good quality trailers.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Within the multimedia context, trailers are among the most effective ways to capture the audience's interest in wanting to know more and promoting content. An efficiently created trailer is a wrap-up of a movie, game, or series in which only the best moments are aired to entice future viewers. The combination of visual and audio in the trailers is actually what communicates the essence of that content; thus, it becomes the main determinant of viewership and box-office outcomes. A meaningful trailer, which really captures the attention of the audience, is not an easy process to create.

Influencing the audience's perception, raising anticipatory anxiety, prompting engagement, shaping the audience's thoughts and driving suspense – all these things and more can only be achieved through the effective generation of trailers. A well-done video clip can stir up curiosity which in turn makes people feel something leading to their viewing a movie of their choice or interest through a particular channel. Thus, it is an indispensable tool in the hands of movie-makers, producers as well as vendors. In addition, trailers profoundly influence the narrative discourse around movies and in turn critical reception, the expectations of the audience and cultural dialogues. This in turn means that generating great trailers is not only important in influencing how financially successful a certain film may become but also its standing as a work of art while majoring towards which it will be part of the culture.

The point of our project is to basically change how trailers are made by creatively combining analysis techniques of information from many modes; usage of sophisticated artificial intelligence tools, and keeping user interfaces simple and user-friendly. It does this by using sound, speech and text mining which makes it possible to make movies before they're even shot within a short space of time. Our project focuses on a comprehensive exploration of trailer generation methodologies, encompassing a variety of functionalities such as feature extraction, content analysis, stylistic modeling, user customization etc.

Our video preview generator system is a sophisticated layer with a variety of features that include audio processing, visual analysis, text mining, and trailer synthesis. Use cases for the same can extend to a variety of domains ranging from film marketing to content promotions to academic research. We base our approach to trailer creation on the interdisciplinary merging of computer vision, audio processing, natural language processing, and machine learning methods. Our goal is to continuously modify and enhance our trailer generation framework using algorithmic development, data-informed experiments, and user feedback inclusion in order to improve its performance and user satisfaction.

It is the purpose of our project to expand the limits of creating trailers by employing advanced technologies and pioneer methods to provide an innovative tool for producing exciting movie trailers to artists, film directors and various authors of advertising content. In the course of our activities, we set forth towards a new creativity-oriented way of making trailers – one that would be much more accessible as far as users are concerned and would have more influence on promoting digital media than it has ever been before.

CHAPTER-2

PROBLEM STATEMENT

Movie trailers have been around for a century, yet they always pique viewers' curiosity and excitement. However, creating engaging short movie trailers with conventional techniques such as selecting key moments is difficult and time-consuming. Due to their reliance on manual labour, traditional methods are not very effective for creating engaging trailers.

The proposed solution involves an automated, multimodal trailer generation approach that integrates features from distinct modalities and includes audio-guided video key moment extraction. The challenge is creating an automated approach for choosing keyframes that conveys story details without producing spoilers. The success of this solution will lead to an increase in efficiency and time saving.

CHAPTER - 3

LITERATURE SURVEY

Novel approaches have recently been suggested that automatically and semi-automatically identify relevant trailer moments directly from video analysis by automatic scene segmentation into shots, turning point detection, and identification of trailer-suitable moments. This is further enhanced through highlight detection based on visually appealing and contextually relevant frames. Multimodal analysis combines visual, audio, and textual cues to pinpoint scenes that best depict the movie. Embedded learning refines this process while factors such as editing biases, emphases on scenes, and focuses on characters add up to the dimensions of trailer component identifications in order to understand the evaluation factors of a trailer. The following literature review represents some of the different ways and advancements in the field of automatic trailer generation, putting a focus on multimodal analysis. It discusses crucial works that shaped not only the understanding but also developed the methods to find impactful video moments using audio, visual, and text data for optimal trailer creation. These have created foundations for making automation in trailer generation effective and efficient.

P. Papalampidi et al. [1] describes new techniques to automatically detect key moments using video analysis, focusing on the narrative structure of the movie. Wang et al. [2] presented the model CCANet, which performs the selection of trailer highlights using both highlight detection and multimodal approaches to capture relevant frames in movies both visually and contextually. Gan, Bei et al. [3] propose learning movie highlights from noisy labels by training a model on trailers to segment movies into scenes. Wei, Fanyue et al. [4] introduced a method of "Pixel-Level Distinction Video Highlight Detection" which can model movies at the pixel level, learning both temporal and spatial relations and subtle context to detect viewer-appealing segments of unedited videos. Bretti, Carlo, et al. [5] shows that multimodal analysis aids in obtaining scenes which are most trailer-worthy. Work on embedded learning for identifying common key components for trailers in Sheng, Jiachuan, et al. [6]. Further, other methods such as MEGA by Sadoughi et al. [7] offer very promising results. Automatic summarization using textual information and

multimodal data in P. Mishra et al. [8] provide more insights. K. Porwal et al. [9] and Dong et al. [10] deal with deciphering video, audio and text files and deriving meaningful insights to classify and synthesize frames using language-vision models. Experiments on the impact of editing biases on viewer impressions in Kakimoto et. al. [11] and efforts towards developing frameworks for genre classification based on textual features extracted from movie subtitles in M. Hesham et. al. [12] make up a complete platform for advancing video trailer generation techniques.

P. Papalampidi et al. [1] proposes that movie narratives involve intricate relationships between the character and the events; hence they require us to understand their long-term cause-and-effect. The sheer size (e.g., movie length) and multimodal nature (video, audio, text) of movies make it difficult for standard AI/ML models to process. Moreover, creating large datasets with the video and corresponding textual information (metadata) for movie narratives, is a very challenging task, thus limiting the training data available.

This paper proposes the following Hypothesis - Summarizing a movie is possible using the information about its narrative structure. They introduce a new dataset called **TRIPOD**. Their proposed workflow is as follows:

- Identifying the crucial moments or turning points based on the plot using information from the video which is broken down into segments or shots
- Summarizing screenplays from their underlying narrative structure.
- Summarizing movies using both audio and visual information and building a graph structure to represent the relationships between key moments.
- Creating the trailers by relying on human input for certain prompts.

Based on the results of this paper, it can be concluded that:

- Screenplays are used during training the model, but not needed while using the model.
- The model can identify the turning points in the story and trailer-worthy moments.

Wang, L. et al. [2] proposes that in order to eliminate the exhaustive process of annotating the highlight moments in movies, the authors propose a novel method "Co-contrastive Attention network (CCANet)" to automatically detect trailer-worthy moments in movies by learning these key moments directly from the movie

The authors have constructed a novel dataset "Trailer Moment Detection Dataset (TMDD)" to test their model. The contrastive attention network was also tested using YouTube Highlights, TV Sum and other on the VHD benchmark datasets

A two-step approach is has been proposed as a part of this research :

- Co-Attention Network : "Co-Attention" between movies and trailers help determine the training pairs where the moments are scored higher if they are more visually correlated to the scenes from the movie.
- Contrastive Attention Network : This module enhances the feature representation of the visual content, where the contrast between the moments in trailer and the movie are highlighted by maximising it.

The authors have used MAP, Rank@N and Rank@Global to test their findings; they have also compared CANet to other benchmark methods.

Co-Attention scores are maximum when trailer shots are from the movie itself

The augmenting makes the highlight features more distinguishable from the non-highlight features

The proposed Contrastive Attention module explicitly models the relations between the key and non-key moments so that "trailer" features stand out from the "non-trailer" moments helping build a more robust model.

The following strengths and weaknesses were observed from the findings of the paper

There is an area of opportunity to utilize multi-modal features to boost the trailer moment detection. This method shows performance gain on benchmark datasets and hence CCANet can be used for detecting key highlights in videos. It is also targeted toward large movies, so scaling would be eased.

Gan, Bei et al. [3] proposes a method to learn highlights in movies through noisy labels. This model is trained on trailers which helps segment the movie into scenes. The shots in these scenes are noisy labels as not all trailer scenes incorporate highlight moments and most scenes are tailored to appeal to the audience. A novel approach known as CLC or collaborative noisy label cleaner is proposed to handle noisy labels in highlight detection.
The research uses a novel dataset “MovieLights” on which the approach has been tested. The research also uses “YouTube highlights” for validation purposes.

ACP - Augmented Cross Propagation exploits the related audio and visual signals and fuses them to learn single multi-modal representation.
MMC - Multi modal cleaning helps obtain cleaner labels by observing multi-modal losses

A three-step workflow has been proposed as a part of this research

- Feature extraction
- ACP : This module helps to capture multi-modal feature interactions. It also comprises of a “consistency loss” to show how much these modalities agree with each other in a feature space
- MMC : to tackle noisy labels

The research work shows that CLC outperforms the baseline Video Highlight Detection models by almost 20 - 23% . This model is also robust to varying label noise levels. CLC accounts for temporal and multi modalities helping in better understanding which scenes must be included in the highlights. ACP and MMC can be used in video highlight detection and further enhanced with other techniques to generate trailers. The authors, however have not implemented scene understanding

Wei, Fanyue et al. [4] proposes a novel method “Pixel-Level Distinction Video Highlight Detection” (PLD-VHD) that models movies to obtain pixel level demarcations. This accounts for temporal and spatial relations between the content and also explores fine-level context to suggest

what segments are appealing to the viewer from an unedited video. The authors have exploited the property of movie highlights being “context-dependent”
The following datasets have been used in this research : “YouTube Highlights”, “TvSum”, “CoSum” – existing benchmark datasets for testing

A two-step workflow has been followed

- Modelling Temporal Dependency : The video is segmented, each segment is associated with a label y, which determines if the segment is a highlight or not. Instead of adding individual frames as input to the estimation function, they add clips (L frames to I) and find the loss with ground truth
- Visual Saliency: In order to find out pixel level distinctions, visual attraction of the frame is considered. The saliency mask is used as pseudo-labels to obtain pixel level distinctions.

The following results have been observed

- PLD-VHD improves the benchmark methods on “TvSum” and “CoSum” by 3.1% and 9.9%
- TASED-Net with temporal and spatial works best on YouTube highlights dataset
- The PLD-VHD with TASEDNet accounts for spatial and temporal context. This may help in finding proper clips or shots to include in parts of the trailer

In addition to achieving best performance, the proposed approach has also shown expandability. However, their method mainly fails in “first-person” videos with a lot of cluttered background

Bretti, Carlo, et al. [5] proposes a multimodal approach based on utilizing pre-existing trailers to analyze which segments exhibit high trailerness, utilizing visual and subtitle data over various durations. This methodology aims to forecast the trailerness of specific sections within a TV episode or film, aiding editors in identifying optimal moments for trailer inclusion from extended video content.

This paper utilised the GTST dataset. It consists of sixty three episodes from the ”Goede Tijden, Slechte Tijden” which is a long-running Dutch soap opera

A novel approach , Trailerness Transformer based on three main stages which are, encoding, multi - modal and multi - scale transformers, and prediction aggregation.

- When input is given as a video representing a movie or television series, creating encodings at both the clip and shot levels for the visual and textual modalities of the video.
- The data is further encoded and processed by individual transformers, with the transformer output then undergoing a sigmoid function for normalization.
- Subsequently, to predict trailerness, the four prediction sequences are combined.
- The late fusion of predictions involves combining the frame-level predictions from the four multi - modal multi - scale streams using various possible combinations, such as averaging the likelihood of predictions is performed

Looking at the results discussed, quantitatively, the proposed model outperforms all three baselines (random, MLP based ,Vasnet) by incorporating sequential order and temporal positioning in terms of clips being assigned for trailerness Qualitatively,

Best Performing model - To yield higher trailerness, scenes with brighter visuals and emphatic dialogue delivery are required.

A few points can be noted regarding the methodology of the paper as given below,

Strengths

- The trailerness for each clip or shot aids editors in creating trailers. This inturn helps the editors in boosting their creativity due to selection and recommendation moments they would otherwise not have picked initially.
- This approach emphasizes the benefits from contextual information.
- This proposed method is targeted towards narrative-based videos (such as soap operas)

Weaknesses

- Although it does estimate the trailerness for the clips and shots it is noteworthy to mention

that segmentation of these shots in a trailer is crucial

- The clips and shots selected are based on parameters such as high emotion visuals and high pitched dialogue delivery. Incase of incorporating this to our project , subjectiveness plays a crucial role , soap operas are region specific and hence can't be scaled to movie , where trailerness is found for all genres.

Sheng, Jiachuan, et al. [6] proposes an embedded learning algorithm for the automatic generation of movie trailers, eliminating the need for human involvement. The development of a novel embedded classification algorithm for the purpose of identifying common key components within movie trailers.

The paper utilised ImageNet dataset for training. For testing dataset with 10 Movies namely The Dark Knight Rises , Inceptions, Transformers 3, Iron Man3 , Thor 2 , Prometheus and so on .

The workflow includes,

- Utilizing the VGG-F model for feature extraction from candidate frames.
- Applying SURF matching results to compare frames from a movie with frames from its trailer,
- Implementing semi-supervised learning based on "Friends being close and enemies being apart" principle, utilizing S4VM for the final classification of frames.
- Generating the trailer by stitching clips around the key frames identified during classification.

The results are as mentioned below ,

- Movies containing (a) a lot of attractive and exciting contents, such as actions and explosions (b) scenes with composition on protagonists' and dialogue delivery; have achieved high accuracy. An example for these are movies such as "Edge of Tomorrow" and "Resident Evil: Retribution". It can be seen here that "Prometheus" gets the least accuracy since the trailer does not have such characteristics.

- The proposed model outperforms the other three models based on similarity between the scenes picked by the model and trailer scenes

A few points can be noted regarding the methodology of the paper as given below,

Strengths

- Incorporates the creation of suspense through avoidance of story ending, thereby avoiding spoilers in the trailer generated.
- The features considered for the system aids in generating trailers for multigenre movies.

Weakness

- To enhance the presentation of movie content, the importance of the casts and sounds could be included, which may not satisfy conditions for creating a trailer.
- Trailers generated with just high-impact factors are not helpful.

Sadoughi, Najmeh, et al. [7] proposes a novel method for combining and aligning many modalities to interpret long-form videos (more than 60 minutes) in an effective and efficient manner. MEGA 1. MEGA addresses the issue of cinematic long-video segmentation by leveraging multiple media modalities .

The method is trained on Movienet-318 , IMDB , Places and testing on Movienet-318 , TRIPOD

The workflow includes,
The proposed method pipeline consists of the following steps .

- Initially, the video is preprocessed by dividing it into shots.
- Then, from each shot, features are retrieved and then pooling and normalisation are performed. The second step is the alignment and fusing of cross-modalities.
- Bottleneck fusion tokens and alignment positional encoding are used to accomplish this. Finally, scene and act segmentation is a part of the pipeline.

- Act segmentation is achieved by using knowledge transfer loss in this method, whereas scene segmentation uses CE loss.

The results are as mentioned below,

- In case of scene segmentation, MEGA Outperforms when trained on the three datasets M+P+I as compared to MEGA trained only on MovieNet 318, Mega outperforms all previous SoTA models in this aspect .
- In case of act segmentation, on the TRIPOD dataset, MEGA sets a new SoTA performance.
- Based on visual modality alone MEGA performs better than the prior SoTA, it also outperforms GRAPHTP, illustrates the alignment and fusion components and the suggested methodology. With +5.51% TA, +9.15% PA, and -%0.81 D, MEGA nearly doubles the performance of earlier studies.

A few points can be noted regarding the methodology of the paper as given below,

Strengths

- The suggested approach is versatile for use in real-world scenarios since it is scalable and generalizes to a variety of numbers of modalities at different scales.
- The method focuses on knowledge transfer between modalities using Knowledge Distillation

Weakness

- The appearance, location, activity, acoustic and textual features are the only areas explored in this work.
- Providing of actor name and identification can aid in scene/ act segmentation for long movie segmentation however it needs to be implemented to MEGA

P. Mishra et al. [8] opposed the problem of manually developing trailers for online academic courses that consumes a lot of time and needs significant human labor and proficiency. Their

proposed AI-based platform intends to computerize the creation of trailer content with regard to the text and visual components by employing machine learning technology plus natural language processing mechanisms.

Datasets for this included chapters from a textbook for ML and speech-to-text transcriptions of video lectures from an academic course on NLP..

The proposed workflow is as follows,

- The system is an approach for producing video trailers by using templates that contain components like splash images, titles of trailers, information on authors, synopses, metadata, endorsements, calls-to-action.
- To create a trailer, the template constraints are used which specify how it should look – for example which fonts should be used, what should be the pacing, where should cut scenes be, audio editing etc.
- The system comprises a video combiner module stitching together all elements likes frame data, voice-over text, and text-to-speech to make the final trailer video.

The results obtained were as follows,

- A user evaluation with sixty-three human evaluators is part of the research and it offers positive feedback.
- Human evaluation results were positive suggesting that the developed approach can be effective in creating online educational courses trailers.
- The authors also plan to enhance the present system through incorporating user evaluation feedbacks and introducing more fascinating topics.

These were the key takeaways,

- Improving trailer content by refining the pacing, readability, and general effectiveness of the trailers according to user feedback suggestions it is possible to enhance the auto-generated content in the system.

- Interactive Dashboard: More customization options and flexibility in trailer creation could be incorporated into the proposed interactive dashboard for content creators wishing to make edits to the auto-generated content.
- Advanced Themes: The system's capabilities would be advanced by more advanced themes and features like automatically detecting learning outcomes given resources, thereby providing a more comprehensive overview of course content in the trailers.

K. Porwal et al. [9] addresses the challenges caused due to reduced time, tight schedules, and a growing number of online events and activities, using audio and video content online is hard. The research aims to provide a solution through using Natural Language Processing (NLP) methods for creating transcripts and summarizing video clips. With this approach, audio-visual content can be converted into textual format and summaries that are qualitative are produced while retaining the original essence of the content.

Datasets included generic video files (not explicitly mentioned about specifics).

The proposed workflow is as follows,

- The media file gets divided into audio chunks consisting of frames further cut into tokens; which on their own are fed into Hugging Face Model that extracts them into text.
- The main ones are Statistical Information Retrieval (SIR) based analyses and Natural Language Processing (NLP) based information extraction approaches.
- The document contains tables showing metrics such as duration of videos, memory use in gigabytes and processing time as an evaluation parameter for measuring the efficiency of the proposed algorithm for video summarization.
- An unsupervised graph-based text summarization technique, called the Text-Rank algorithm is utilized for generating the video summaries.

These were the key takeaways,

- The lack of specific datasets mentioned in the document are for video transcription and summarizer. The absence of comparative analysis in the paper does not compare the proposed video summarization algorithm with existing methods or tools within this area.
- If the author had included a comparative analysis with other video summarization techniques, it would have shown the advantages and disadvantages of their proposed method.
- The document does not clearly define potential future directions for the research or perceive possible improvements or applications for the project.

Dong et al. [10] considers the issue of text-to-speech conversion which can be quite problematic especially because it is not easy to obtain well-structured texts necessary for training audio recordings. The main objective of our work is to handle the lack of actual texts related to sound signals due to their unavailability through an exploration into raw instructional videos combined with visual linguistic pre-possessed systems

The paper utilizes the following datasets

1) VGGSound: Consisted of 171,899 YouTube videos which had around 10 sec duration, covering 310 classes of sounds.

2) MUSIC: Consisted of 1,055 full-length YouTube videos of people playing various musical instruments, with around 21 instrument types.

The authors propose the following workflow

- Teaching a condition diffusion model for producing video audio tracks from video frame images, by using pretrained contrastive language-image pretraining (CLIP) models. Researching zero-shot modality translation through conditioning the diffusion model with a CLIP encoded text query at test time.
- Utilization of a pretrained diffusion prior model has been applied to solve the modality gap problem between textual and visual zed queries which lead to an equal parity regarding text-to-audio transformation as well as image-to-audio synthesis.

A detailed analysis of the research showed the following results

- The proposed CLIPSonic model effectively learns text-to-audio synthesis without text-audio pairs, leveraging unlabeled videos and pretrained language-vision models.
- The model demonstrates competitive performance in both text-to-audio and image-to-audio synthesis, offering a promising research direction for leveraging images as rich conditioning signals for audio synthesis.
- The study identifies a noticeable performance drop in text-to-audio synthesis when using text queries in a zero-shot setting, indicating a modality gap between the CLIP's image and text embedding spaces.
- The CLIPSonic-ZS model shows a performance drop in fidelity and relevance when using text queries, suggesting a challenge in effectively synthesizing audio from text queries.

H. Kakimoto et. al. [11] proposes that movie trailers are tailored to a specific target audience depending on the movie's genre. Very few scenes are taken from the movie and hence the duration of the trailer is very less. As a result, it is very hard to edit a trailer. If the trailer isn't captivating, viewers will lose interest in the movie.

When a movie is summarized and a trailer is edited, there could be some biases like background music, characters, dialogues, scenes, sentiments and so on. In this paper, seven of these biases are analyzed to check if they may be used in curating a trailer which will cater to the preferences of the majority of the viewers.

The movie plots and summaries are taken from Wikipedia and IMDb Websites. Users' impressions are also evaluated for the movie trailers, through questionnaires, for analysis against the editing biases.

The proposed workflow is as follows:

- Defining two categories of video editing bias - Audio-Visual and Contents. 7 of these biases are extracted - scenes' length, reordering, background music, characters, lines, topics and sentiment.

- Their effects in the trailers are analysed through a preliminary experiment using existing movie trailers.

Based on the results, scene reordering, length of scenes, emphasis of characters and number of topics have the greatest influence on how the impressions of the viewers differ from the movie to the trailer.

M. Hesham et. al. [12] considers video summarization as a promising approach for efficacious realization of video content through Identifying and picking out descriptive frames of the video.
In this paper, an adaptive framework called the Smart-Trailer is proposed to use only the subtitles of any English movie to automate the trailer creation process. The dataset used here is the Kaggle movie dataset which contains around 5000 movies belonging to different genres.
The following workflow is proposed in the paper:

- The framework analyzes the movie subtitles and classifies the movies into their corresponding genres.
- The system uses many deep learning methodologies to capture the opinions and the behaviors of users to recommend relevant scenes based on their preferences.

The results are as follows:

- Initial experimentation generated a corpus for genre based classification, which is tested on the dataset of real movies. The accuracy rate was 0.89.
- The system also gave automated trailers which have an average accuracy of 47% for selecting scenes which are there in the original trailer.

Video trailer generation has been in the research field for over 15 years, and there have been numerous attempts to solve the challenge of generating efficient, concise and contextual trailers. This field has been marked with continuous advancements and refinements over the years. The aforementioned papers represent a few of them. From this we can conclude the following :

Incase of the method proposed by Bretti, Carlo, et al. [5] , the novel method proposed i.e the multimodal approach at shot level outperforms the unimodal baselines which are text and visual. In movie highlight detection, Collaborative Noisy Label Cleaner [2] outperformed baselines by 20-23%, proving to be robust across label noise levels by utilizing temporal and multi-modal cues. “Smart Trailer” generates trailers only using movie subtitles, and automated trailers accurately selected scenes present in the original trailer with an average accuracy of 47%. The CLIPSonic model [10] exhibits good performance in text-to-audio and image-to-audio synthesis, but experiences a drop in relevance when using text queries in a zero-shot setting

The extensive literature survey performed gave us a good amount of insights into the existing approaches to tackle the challenge of trailer generation. We could conclude that processing modalities separately is lighter on computation and more effective in retrieving key scenes to be added in the trailer. We could also conclude that there are various baseline models and algorithms to infer from incase of possible challenges encountered in the future.

CHAPTER - 4

DATASET

The data repository consists of links compiled from various sources like Youtube, Director's Note, Short of The Week, vimeo and other online resources. The data was mainly compiled from publicly available and reputable data sources. These platforms cater to short movie enthusiasts as well as serve as a stage for the directors to launch their upcoming movies and trailers to film festivals and garner new audiences' attention. The total number of movies (with corresponding trailers) is 311. The average duration of the movies is around 12 minutes. The average duration of the trailers is around 50 seconds. The dataset has 5 columns. The movie name, movie video link, movie duration, trailer video link and trailer duration. A majority of these movies can be categorized as horror-thriller genre.

The data in the repository are readily available, high auditory and visual quality, and age appropriate feature content. All the short movies and their corresponding trailers have been taken from the following online sources - YouTube, Vimeo, FilmShortage, ShortOfTheWeek, Reddit, FilmsShort, LetterBoxd.

Provided below is a snapshot of the curated dataset to provide a concise overview:

Sl No	Short Movie Name	Movie Link	Trailer Link
1	2:00 am	https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.youtube.c	https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=ht
2	97%	https://vimeo.com/143233550	https://vimeo.com/75476170
3	623	https://vimeo.com/groups/shortfilms/videos/100066515	https://vimeo.com/104512427
4	(Otto)	https://vimeo.com/jobjorisenmarieke/otto	https://vimeo.com/jobjorisenmarieke/otto-trailer
5	@Social #Connection	https://vimeo.com/87743718	https://vimeo.com/102975960
6	11 Minutes	https://vimeo.com/96462953	https://vimeo.com/96464712
7	14 in February	https://youtu.be/YnpmZk3oWzg?si=dunmkSDA9e6Wz6IH	https://youtu.be/UcvTJWhQra0?si=-rfBtU2Ro95kqPM8
8	26000 Days	https://youtu.be/mKhHaWnlY1c?si=0iMCbwZyGj2h677J	https://youtu.be/nFZ55wtD5xs?si=t1Ewo4lUK8fuEmj4
9	6 Days	https://vimeo.com/9957697	https://vimeo.com/10275533
10	7,83 Hz	https://vimeo.com/255016268	https://vimeo.com/groups/shortfilms/videos/255032452

Fig 4.1. Dataset Snapshot

CHAPTER - 5

SYSTEM REQUIREMENT SPECIFICATIONS

5.1 Introduction

5.1.1 Purpose

Designing emotionally persuasive trailers for short films through ordinary means is a tiresome and scary job because it is hard to capture people's interest by combining audio, video, and other metadata in an intelligent way through multimodal data analysis techniques. For this to be done there are some system requirements that you will need. They must have a strong ability when it comes to dealing with sound by enabling the removal of sounds from the radar as well as its analysis. Efficient video manipulation software should also be availed so as to facilitate proper scrutiny on the visual elements in any multimedia data. At the same time, there should be mechanisms of incorporating metadata into such systems so as to make sure these embedded files have necessary information for comprehension. In addition, there is also a need for computational capacities which can work many things on large multimedia files effectively.

5.1.2 Intended Audience and Reading Suggestions:

Intended Audience includes Production houses, Professional Video Editors and Film Enthusiasts who want to use this software to explore automatically created trailers for short movies; and Researchers who intend to work with Audio/Video Processing to improve the existing approaches for trailer generation.

5.2 Overall Description

5.2.1 Product Perspective:

The software is developed for crafting intelligent trailers using Multi-Modal Analysis and Deep Learning Methodologies. It can be used by Professionals to generate trailers for advertising their short movies. This can also be used by Enthusiasts for experimentation and entertainment, for movies that don't have trailers.

5.2.1.1 System Interface

The system would contain multiple modules for Audio Processing, Video Processing, Frame Selection, Shot Selection, Trailer Scene Rearrangement, Metadata Analysis and Trailer Generation.

5.2.1.2 Software Interface

The software includes Python Libraries, Deep Learning Models and other Audio-Video Processing tools. The Software will be compatible on Linux, Mac and Windows Operating Systems.

5.2.2 Product Functions

- Given the video of a short movie, the software separates the audio and video, processes the audio and video files separately to find certain key moments.
- Based on the anomalies detected in the audio files, or attractiveness of the video frames, shot selection is done.
- The Scenes are combined in such a way that the end trailer is visually attractive and also has key moments based on the audio (music or dialogues).

5.2.3 User Classes and Characteristics

The user characteristics in this research project revolve around educational or experimental roles. Those who contribute are usually academic researchers, short-film producers or social media

content creators who supply video datasets and set guidelines for creating trailers, which drives the project forward.

5.2.4 Design and Implementation Constraints:

Climax or Story Reveal: The system must be designed to correctly identify key moments like surprise reveals or tension-building scenes based on the genre of the movie. Also, ensuring that these trailer moments do not spoil the movie by revealing the story or climax is very essential.

Length of the movie and trailer: Creating a captivating trailer with the limited duration of 30-90 seconds from a 15-30 minutes short film is a huge task requiring careful selection and editing. The algorithms used have to be optimised to work effectively with this constraint.

Multimodal Analysis: There is a lack of publicly available benchmarks for processing multimodal data, especially for movie trailer generation. To evaluate the proposed software, a novel evaluation metric must be developed.

Data Acquisition: Finding a publicly available dataset of short films with clear audio and video, obtained legally and with proper permissions, is extremely challenging. Alternate data collection methods have to be explored, such as collaborating with short film festivals or communities.

5.2.5 Assumptions and Dependencies:

The main assumptions of our project would be:

1. Genre Consistency: It is assumed that the short films provided mostly belong to the horror, thriller, or mystery genres. This assumption makes it possible to create analytical algorithms that are especially suited to these narrative modalities. By concentrating on these genres, the study can go more deeply into the particular components and methods employed in these genres, improving the analysis's accuracy.

2. Content Adherence: The process of creating trailers will closely follow the short films' original content.The story won't be changed, and no outside components (like generated content) will be included.

In addition to these assumptions, there are dependencies that must be considered:

1. Deep Learning Frameworks: To create and train models for audio and video analysis, the project mostly uses deep learning frameworks like TensorFlow or PyTorch. These frameworks enable the project to extract significant information from the audio and visual components of the short films by offering strong tools and algorithms for processing and evaluating complex data.

2. Audio/Video Processing Libraries: The extraction and manipulation of features from audio and video data will be greatly aided by libraries such as Librosa or OpenCV. With the help of these libraries, the processing and manipulation of audio and video data in a multitude of ways can occur , extracting essential elements that aid in analysis and trailer creation.

3. Computational Resources: Training of Deep learning models can incur significant computational costs. Hence, in order to effectively train and improve the models, access to computational resources like GPUs or cloud computing services may be required. These resources ensure the fast and accurate analysis of the short films required for trailer curation.

A thorough study plan that takes into account these presumptions and dependencies can be created to meet the obstacles and take advantage of the opportunities this project presents. The implementation and accomplishment of the research objectives will be facilitated by the careful consideration of these elements.

5.3 Functional Requirements

The core functionalities that our Automatic Trailer Generation software will offer are as follows:

5.3.1 Separation of Audio and Video

The user must give a short movie as input to the software. The system will separate the short movie into its audio and video components.

5.3.2 Audio Processing:

The audio is processed first, to retrieve specific segments of audio that highlight the key moments of the short movie. For example, “screaming” in the case of a horror movie. The audio segment determines to what extent the dialogues, music etc., matter while a trailer is generated.

5.3.3 Video Processing:

The video processing component does the following

1. Picks the video segments corresponding to the audio segment and analyses the attractiveness
2. Selects any other frames / shots that may be visually and in terms of content, a highlight and therefore a necessity to include in the trailer

5.3.4 Meta-Data and Result analysis:

This component is concerned with analysing some important metadata like trailer duration, movie duration, trailer worthy segments, non trailer worthy segments etc.

5.3.5 Shot Arrangement:

The segments are either compiled into a trailer or presented to the director as a set of clips to choose from. Additional effects will not be added in the generation process.

CHAPTER - 6

HIGH LEVEL DESIGN

6.1 Introduction

This project addresses the need for effective and automatic trailer generation by using multimodal data analysis, integrating audio, video, and other metadata to intelligently craft meaningful and concise trailers that capture the attention of the viewers/audience. Our project aims to explore various Audio and Video Processing techniques for analyzing and capturing the key moments of a short movie.

6.2 Current System

Crafting emotionally resonant short movie trailers using traditional methods is laborious and challenging. Current softwares and approaches for automatic trailer generation include having a professional content creator or experienced director to provide inputs and suggestions for choosing key moments for the trailer.

6.3 Design Details

6.3.1 Novelty

The Novelty of this project is that a multi-modal analysis is performed for short movies. Audio is processed first to extract emotional moments and then the video processing is performed to extract attractive frames. Then a novel algorithm is applied for rearranging the scenes to create a captivating trailer. The metadata, like genre, is also taken into consideration.

6.3.2 Innovativeness

The project showcases a lot of innovation in the field of audio-video processing. The work expresses innovation by creating a comprehensive trailer using a multimodal data analysis approach which has been under-explored in prior work. The project is also aimed at creating a novel audio guided video processing technique and a sophisticated algorithm for ordering the obtained scenes.

This creative method helps produce a trailer with visual and auditory attractive features that captures the essence of the given short movie. The project guarantees that by using independent audio and video analysis, equal weightage will be given to both visual and audio features thereby ensuring a thorough comprehension of the short film's content.

The trailer generation process gives emphasis on not giving out the plot or disclosing the climax. Key "trailer-specific" moments are chosen through the audio driven video analysis which ensures that the highlights are selected from the movie based on a certain threshold. Through this the project aims to manipulate the trailer scenes to not completely give out the plot.

This project represents a significant advancement in the trailer creation industry due to its innovative use of multimodal data analysis techniques.

6.3.3 Interoperability

The project follows a modular approach. The significance of this approach is that it becomes open to extension. Any further features like adding additional template content can easily be integrated into the existing code modules. The existing plan of action can be split into the following modules

1. Audio Analysis Module : This feature is specifically to return key timestamps corresponding to emotionally significant audio segments which have some auditory impact (ex. A Scream)

2. Video Analysis Module : This module is to utilize the segments returned by the audio module and select highlights from the video and return the corresponding timestamps
3. Feature Fusion : This module aims to combine the features obtained from the previous modules
4. Scene Formation: The video and audio segments selected will be combined and an algorithm will arrange them to form a visually attractive trailer.

6.3.4 Performance

Performance of an application is a measure of effectiveness and user satisfaction. The users of the trailer generation software must be happy with the trailer contents they get as output

Certain aspects like memory utilization, processing speed, correctness of output etc. show the performance of an application

Utilizing cloud services like AWS to store large short movie data and generated trailers helps reduce dependence on local storage which inturn prevents system failure.

The processing speed can be increased by the use of good quality GPUs. The trailer generation software must give an appropriate trailer, else the user will not be satisfied.

6.3.5 Reliability

The goal in terms of reliability is to establish a robust system which establishes smooth, uninterrupted operations. Comprehensive error handling and fault tolerance mechanisms to deal with potential failures and overall stability of the system shall be implemented.

We will also strive to implement data integrity and backup procedure checks with the goal to prohibit any form of data loss or corruption. Rigorous types of testing would be conducted to address as many potential vulnerabilities as possible.

6.3.6 Maintainability

A modular design approach is applied to maintain the project: Audio and visual analysis components are kept separated from each other to update them independently. Documentations describing data models, methods, and system architecture are updated on a regular basis. It helps

keep the code name and relevant comments to maintain the code. The code is tracked with tools such as Git and other version control systems that encourage teamwork and perform system backups. High-level testing such as unit and integration tests helps ensure that the system is still reliable and accurate even after modifications. The system should be made as easy to maintain and flexible regarding changing needs as possible.

6.3.7 Portability

To ensure a smooth operation across various systems and platforms, platform-independent Python libraries and portable deep learning frameworks for audio and visual analysis are used. Cross-platform development tools and libraries will assist in the code being compiled and run on different operating systems. The use of Docker allows for containerization in which an application and its dependencies are bundled in standardized components. A flexible and scalable way to deploy is done with the use of cloud platforms such as AWS, Azure, or Google Cloud. A steady compatibility testing method will ensure a proper working scope of the application in various settings, screen resolutions, aspect ratios, and hardware combinations. This will make the application more accessible and portable in environments.

6.3.8 Reusability

The modular architecture makes it easy to reuse it, hence efficiency in pre-processing and frame extraction. The pre-processing module splits short movies into audio and video parts. The audio module simplifies audio processing across various movies while extracting significant anomalies or sentiments. It enhances processing coherence. The video module selects visually appealing frames and can adapt to various video inputs. The scene arrangement module uses inputs from both audio and video modules to arrange segments in the correct order. It can be reused for all trailer generation use cases.

6.3.9 Application compatibility

The system can run seamlessly on any operating system like Windows, Linux or Mac OS, Also, it supports all the commonly used audio formats like WAV and MP3, and video formats like MP4,

MOV, and MVA. The software also offers a variety of formats for exporting or downloading the trailers, including MP4 and MOV.

6.3.10 Resource utilization

The resource utilization depends on many factors like the length and complexity of the movie. Short movies with basic cuts and transitions will require comparatively less processing power than those of longer duration having elaborate effects and color grading.

1. CPU: Primarily for basic audio and video loading and processing
2. GPU: A hardware accelerator to boost the speed of training, processing and feature extraction time
3. RAM: The software needs RAM to store project data, including video clips, audio files, and cached information.
4. Storage: The final trailer file size will depend on the chosen format and resolution.

CHAPTER - 7

LOW LEVEL DESIGN

7.1. Introduction

7.1.1. Overview

This outlines a modular design incorporating separate processing of audio and visual data to predict the most trailer worthy segments from a given short movie.

The data is analysed for relevant features with the intent of giving equal weightage to the audio and video data, ensuring a balanced contribution from both the modalities. Audio features extracted are MEL spectral features, MEL Frequency cepstral coefficients, chroma features and spectral contrast features. Given the selected genre is horror, these features were selected for their ability to best capture the suspenseful atmosphere and intense auditory moments. A Siamese model with an LSTM layer is designed to generate embeddings and predict the similarity of audio segments to a centroid, indicating trailer-worthiness.

The trailer-worthy timestamps are tabulated and stored to guide the visual processing. Features from each frame of the movies and trailers are extracted using the CLIP model. This model is utilized to exploit the cross modal embeddings and semantic features of the frames. A one class Support Vector Machine is trained on this data to learn a hyperplane that distinguishes only the trailer worthy segments from all the segments obtained during audio processing.

Key segments from both the processes are identified using clustering and thresholding distance metrics. Each component is designed to be reusable, with feature extraction and prediction pipelines modularized for scalability and maintainability

7.1.2. Purpose

This section provides a detailed and comprehensive description of the various modules used in the process of generating a trailer for a short movie. This modules are -

1. Exploratory data analysis and data preprocessing
2. Audio feature extraction, modelling and prediction of trailer worthy timestamps
3. Video feature extraction, modelling and prediction of trailer worthy timestamps
4. Analysis of predicted time segments

The project employs a combination of machine learning algorithms, feature extraction techniques, clustering and thresholding methods to analyze and predict trailer-worthy segments from movies. This document describes in detail the following algorithms and techniques

- Preprocessing data into the necessary format using youtube_dl
- Feature extraction using librosa
- Siamese Model with LSTM layers to predict trailer worthy audio timestamps
- K-Means clustering to group similar audio features, aiding in the identification and prediction of trailer-worthy segments.
- Scene Segmentation, shot detection and frame extraction using scenedetect
- Feature extraction using CLIP ViT-B-32
- One Class SVM combined with distance thresholding to predict trailer worthy video timestamps

This document details the complete internal working of the trailer generation process. This serves as a comprehensive guide for future enhancements, ongoing documentation, maintenance tasks, and result analysis.

7.1.3. Scope

The scope of this document encompasses a detailed overview of the architecture, algorithms, and techniques employed in identifying key moments for trailer generation from horror themed short films. The algorithms, techniques and architecture are chosen specifically to give the most significant and impactful moments from the short films. The

audio features extracted are tailored specifically for the selected genre, whereas the Siamese network is employed due to its semi-supervised nature, making it well-suited for the task, especially since the dataset includes pairs of trailers and movies.

The One Class Support vector machine helps to identify the trailer worthy segments by treating the trailer data as a distinct class thereby distinguishing it from the non trailer class. The CLIP model is used for visual feature extraction owing to its zero shot capabilities and ability to uniquely identify genre specific cues.

Overall, this low-level design document serves as a comprehensive guide for implementing the software intended to create trailers for short movies, introducing a novel method and the corresponding architecture and pseudocode.

7.2. Proposed Approach

7.2.1. Constraints

The following section outlines the limitations and considerations that influence the design and implementation of the modules included in this project. The constraints included are -

1. Data Availability : The data collection process was carefully curated, selecting only short films that are publicly available on platforms like YouTube and Vimeo. Furthermore, each movie was required to have corresponding trailer and content suitable for the general audience
2. Computational resources : Processing audio and video simultaneously requires significant compute resources. Hence, we adopt a novel approach where audio is processed first to reduce computational load, serving as a lightweight guide for the subsequent video analysis.
3. Audio and Video Requirements : The audio and video used in the processing must adhere to the horror genre. The short films must not exceed twenty five minutes of duration. The expected key moments when compiled should not exceed 2 minutes.

7.2.2. Assumptions

The assumptions outlined in this document establish the foundational understanding required for the trailer generation process. These include assumptions on the genre, input data and output content.

1. Genre Consistency: The project is designed to adhere with extracting key moments from the short films of “Horror” genre. This assumption arises from the fact that the audio features extracted during analysis are specifically chosen to effectively represent the characteristics most relevant to the horror genre, thereby ensuring the most impactful frames. The visual features in horror films, such as stark dark-light transitions and impactful facial expressions, are particularly prominent and significant
2. Input Data : An assumption is made that the input data is obtained from legitimate sources and the content is age-appropriate
3. Content Adherence: The trailer worthy segments are selected only from the content of the corresponding short movie. No additional content will be synthesised. No transitions and videographic effects will be added.

7.2.3. Dependencies:

1. Preprocessing : youtube_dl and pytube for processing the data into usable form and splitting the data into audio and video files
2. Audio and Video Feature Extraction : Librosa for extracting MFCC, MEL, chroma and spectral contrast features during audio processing and CLIP ViT-B-32 for extracting visual features from frames
3. Machine Learning Algorithms : Siamese network for modelling audio features, K means for clustering related audio features and One Class SVM with decision
4. Computational resources and Environment : The project utilises Google Colab and kaggle environments for computational tasks, occasionally leveraging their GPU accelerators to efficiently process large datasets and train models.

7.3. Proposed Methodology

7.3.1. Algorithm

1. Pre-Processing - The short film and the corresponding is taken as input and is pre-processed to convert the files into the standard format like MP4. This is then split into audio and video data
2. Feature Extraction and data preparation -
    a. Audio Feature Extraction : The audio trailer and movie files are split into 5 second segments each. Each segment is passed through a feature extraction function where the MEL, MFCC, chroma and spectral contrast features are extracted for each segment. For each movie segment which is treated as an anchor point a corresponding label is associated. If there is a trailer segment with similar features as the movie segment, the label will be 1 and the trailer segment is called a positive. This together forms the triplet data
    b. Video Feature Extraction : The video trailer and movie files are segmented using a python library named scenedetect, that detects scene transitions. Frames are extracted from these scenes and stored. These frames are subjected to the CLIP model. This model extracts the feature embeddings from the them and stores them in a numpy array
3. Audio Processing : The triplet data is fed to a Siamese model with LSTM layers. The model is trained on pairs of movie - trailer triplet data. The LSTM layers retain the information over the training process. Euclidean distance is computed between the embedding output and similar feature segments are grouped together. Trailer worthiness predictions are made on a new movie by processing it in a similar manner
4. Video Processing : The numpy arrays are used to train a one class SVM model. This model learns a hyperplane that distinguishes the trailer worthy segments from non trailer worthy segments. The timestamps obtained from the audio processing are subjected to the SVM model. The final timestamps obtained are considered to be potentially the most trailer worthy

The trailer created using this approach will only have original content taken from the short movie and will not use any other external source.

7.3.2. Pseudo-code

FUNCTION generateTrailer(movie_file, trailer_file, test_audio):

// This function generates a movie trailer by processing audio and video contributions.

// It takes three inputs: the movie file, an initial trailer file, and a test audio,

// returning the final edited trailer.

// initialise global lists to store the training data

movie_audio_segments = []

movie_video_segments = []

trailer_audio_segments = []

trailer_video_segments = []

// Step 1: Split movie and trailer into audio and video

movie_audio, movie_video = splitAudioVideo(movie_file)

APPEND movie_audio_segments WITH movie_audio_segment

APPEND movie_video_segments WITH movie_video_segment

trailer_audio, trailer_video = splitAudioVideo(trailer_file)

APPEND trailer_audio_segments WITH trailer_audio_segment

APPEND trailer_video_segments WITH trailer_video_segment

// Step 2: Audio Feature Extraction

triplet_data = splitAndExtractAudioFeatures(movie_audio_seg, trailer_audio_seg)

// Step 3: Siamese Network for Filtering Audio

```
    filtered_audio = runSiamese(triplet_data, test_audio)
    // Step 4: Video segmentation and Feature Extraction using CLIP model
    scene_features = splitAndExtractVideoFeatures(movie_video_seg, trailer_video_seg)
    // Step 5: Build an SVM
    build_Model = trainSVM(scene_features)
    // Step 6 : Filter Segments
    trailer_filtered_segements = filterScenes(movie_video, audio_csv)
    // Step 6: Stitch Selected Scenes for Final Trailer
    final_trailer = stitchScenes(trailer_scenes)
    RETURN final_trailer
END

// Function Description
FUNCTION splitAudioVideo(file):
    // This function takes as input a media file and splits it into audio and video parts.
    // Input: media file (movie or trailer)
    // Output: audio part, video part
FUNCTION splitAndExtractAudioFeatures (select_movie_audio, select_movie_video):
    // This function splits the audio data into segments of 5 seconds
    // and extracts the audio features from each segment
    // This function also prepares the triplet data
    // Input: selected movie audio, selected trailer audio
    // Output: triplet data containing features and labels
FUNCTION runSiamese(triplet_data, test_audio):
    // This function filters audio segments using a Siamese Network to retain only relevant segments for the test audio.
    // Input: triplet data, test audio
```

// Output: filtered audio segments with start and end times

FUNCTION splitAndExtractVideoFeatures(movie_video_list, trailer_video_list):

// This function first splits the mp4 files into frames using scene segmentation

// This function extracts frames from video data using scene detection and

// applies CLIP model to extract features

// Input: movie videos, trailer videos

// Output: numpy array of scores representing trailer-worthy scenes

FUNCTION trainSVM(scene_features):

// This function trains an SVM model to recognise the trailer worthy moments

// Input: scene features

// Output: SVM model and scaler

FUNCTION filterScenes(movie_file_mp4, audio_pred_csv):

// This function filters the extracted scene features based on worthiness scores

// using the SVM model

// Input: Movie mp4 file and audio CSV file containing predictions for the same movie

// Output: selected trailer-worthy scenes with start and end times

FUNCTION stitchScenes(trailer_scenes):

// This function stitches together the selected scenes into a cohesive final trailer.

// Input: selected trailer-worthy scenes, // Output: final edited trailer

7.3.3. Implementation and Results

Various machine learning algorithms and feature extraction methods have been utilized, built and implemented as a part of the trailer generation process. For the audio processing, we have exploited features of the Librosa library to extract important features like MFCC, MEL, spectral contrast and chroma that highlight the impact audio segments in a short film of the horror genre. We have also built and modified a siamese network to include LSTM layers to capture temporal dependencies enabling it to efficiently learn and retain

audio features over time which significantly improves its ability to recognize patterns and relationships in the audio data. We have also utilized the K-means algorithm to cluster similar embeddings (as predicted by the siamese model) together. When a new movie audio is given to predict trailer worthy segments, those segments which are nearer to the mentioned “k” clusters are given out as output.

For video processing, the state of the art “PySceneDetect” library which works on scene detection and allows the automatic splitting of videos into individual scenes or clips based on shot changes is used to segment the video into scenes and then into frames. Another state of the art “Zero shot” based model - CLIP ViT-B-32 is utilized to extract visual features from the frames and store them as numpy embeddings. These embeddings are used to train a custom modified One class SVM. The model learns a hyperplane that determines the euclidean distance of the trailer worthy segments / anomalies from it. When the audio prediction segments of a new movie are fed to the video processing model, it predicts the trailer worthy segments according to a decision threshold. This operates on the principle that a higher degree of negativity results in increased leniency, whereas a higher degree of positivity leads to greater strictness.

The results show that the proposed novel technique of audio processing followed by video processing tasks has helped to significantly reduce the computational resources required. Upon comparing the final time segments returned after video processing with the ground truth data, it was observed that more than 50% of the segments were similar. This significant overlap indicates a high degree of accuracy in segment identification, validating the robustness of our methodology. This high percentage of matching segments demonstrates the effectiveness of our approach in capturing critical moments within the movie.

Furthermore, most of the time segments identified were key moments from the movie, both in terms of audio and video content. These key moments included scenes with intense dialogue, pivotal plot points, and high action sequences, which are crucial for

creating an engaging trailer. The success in identifying these moments underscores the importance of integrating both visual and audio data in the segmentation process.

These findings highlight the potential of our proposed methodology in automating the trailer generation process, ensuring the production of high-quality trailers that effectively capture the essence of the original content while significantly reducing manual effort and production time.

7.3.4. Further Exploration Plans

Further Exploration Plans include enhancing our trailer generation system by integrating more advanced multimodal analysis, potentially incorporating subtitles and captions to capture narrative depth and applying sentiment analysis to better align scenes with the intended emotional tone. Additionally, expanding our training dataset beyond horror movies to include diverse genres will improve the system's adaptability and genre-specific sensitivity. We plan to explore real-time adjustments, enabling the model to dynamically adapt to different pacing and styles, and investigate more efficient models for scalability with high-resolution content.

CHAPTER - 8

SYSTEM DESIGN

This system is designed to predict the most trailer worthy segments from a given movie by separately processing its audio and video. The split tasks not only ensure computational efficiency but also give equal importance to audio and visual features, a key characteristic of horror films. The detailed system design is described below -

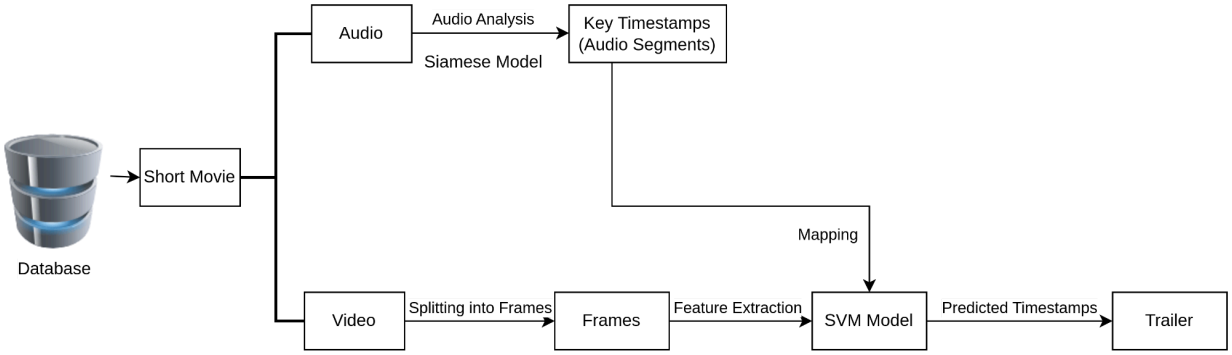

Fig 8.1 System Design

The Fig 8.1 shows the high level system design. As mentioned before, the audio and video processing tasks will be carried out separately. The timestamps generated from the predictions made by the audio model will guide the video processing task, allowing us to extract key trailer moments effectively.

The data which is in the form of links extracted from various resources is preprocessed to extract the audio and video files in the required format. The data is subjected to initial data analysis where a range of audio and video features are examined, including MEL spectrograms, Mel-frequency cepstral coefficients, color spectrograms, Weibull distributions, and color histograms. On initial data analysis, we observed that results for the genre of horror were

exceptional and since our novel proposed method is audio guided video processing , these results were found to be insightful. Hence the genre of horror was chosen for our novel approach.

As mentioned above regarding the audio processing, the audio files are first split into segments with a duration of 10 seconds for both movie and trailer. MEL, MFCC, chroma and spectral contrast features are extracted for all the segments. This is followed by the preparation of triplet data. For each movie segment (anchor segment) a corresponding trailer segment with similar features is paired. Positive pairs, drawn from trailers using the same feature types, are labeled "1" to signify a matching movie-trailer segment. Negative pairs are generated by reversing this setup: trailer segments act as anchors, paired with non-matching movie segments, and are labeled "0" to indicate dissimilarity.

A Siamese network consisting of two identical LSTM subnetworks is trained on this triplet data. Each LSTM processes the input sequence and generates a fixed size embedding. This model is compiled with Adam optimizer and contrastive loss function. Euclidean distance is computed to get the similarity between input pairs.

Once the model is compiled and trained , in regards to the prediction, initially the features such as MFCC, Mel-spectrogram, Chroma, and Spectral Contrast are extracted followed by averaging to form a single feature vector. Audio segments are padded to a uniform length with zero-padding to ensure the consistent input shapes for machine learning models.Further K Means clustering is employed to find unique clustering for the features. Hence the key segments are identified based on their proximity to the cluster centroids. Fig 8.2 illustrates the architecture of the audio model.

Further audio features are standardized using Standard Scaler , which is given as an input to the Siamese LSTM model. Finally the predictions are made, and significant segments are identified using the K Means clustering. The identified key moments are then saved to a CSV file which is given as the output.

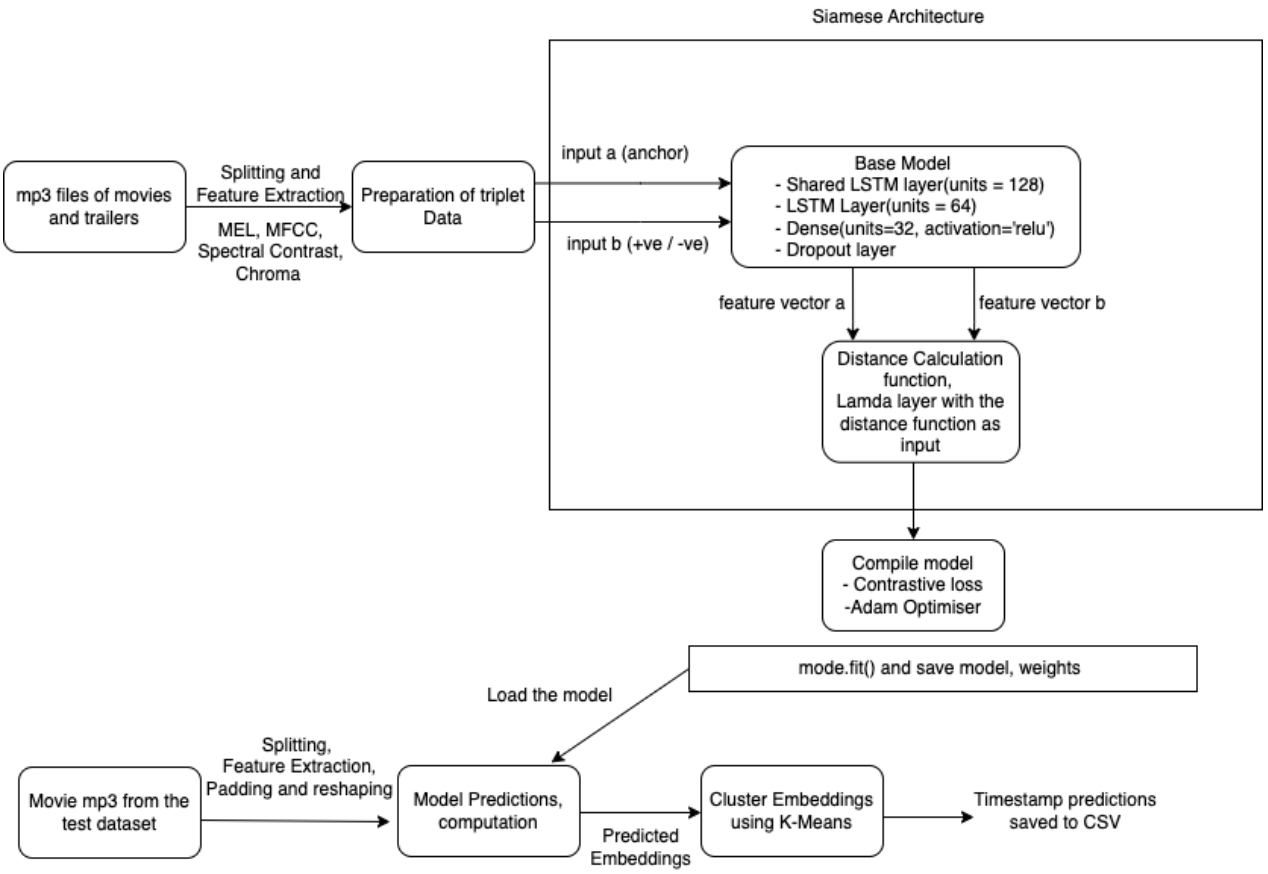

Fig 8.2 Architecture of the audio model

The audio timestamps extracted during audio processing are used to guide the video processing. During video processing the video file is processed using the State of the Art "PySceneDetect" algorithm, that segments the video into scenes and individual frames are extracted from and stored.

CLIP is a neural network trained on image-text pairs, allowing it to predict relevant text for any image through natural language prompts. The image and its associated text are mapped to the same vector space. The CLIP model (ViT-B/32) is loaded and features from the movie and trailer frames are extracted, averaging them to obtain a single feature vector per scene.

Movie features (normal data) and trailer features (anomalies) are stacked and scaled using Standard Scaler. A one class Support vector machine (OCSVM) is trained to distinguish the

anomalies from normal movie features. The model operates on the principle of novelty detection, where it learns a decision boundary around the normal data points (movie features) and identifies any new data points that fall outside this boundary as anomalies. Fig 8.3 illustrates the architecture of the video model.

The trained model and scaler is saved. When timestamps from audio analysis for a new movie are fed to the model, the video is split into five second segments and frames are extracted from them. For each frame, the frames are extracted using the CLIP model, preprocessed, and averaged to represent each segment. Extracted features were scaled and fed into the One-Class SVM model to predict trailer-worthiness, with a decision threshold score determining the distance from the boundary. The timestamps closer to the boundary are considered as the key moments for the prediction and hence returned as the output in the form of a CSV file. Hence the selected timestamps are then matched to the corresponding movie and the returned segments are pieced together to give a trailer.

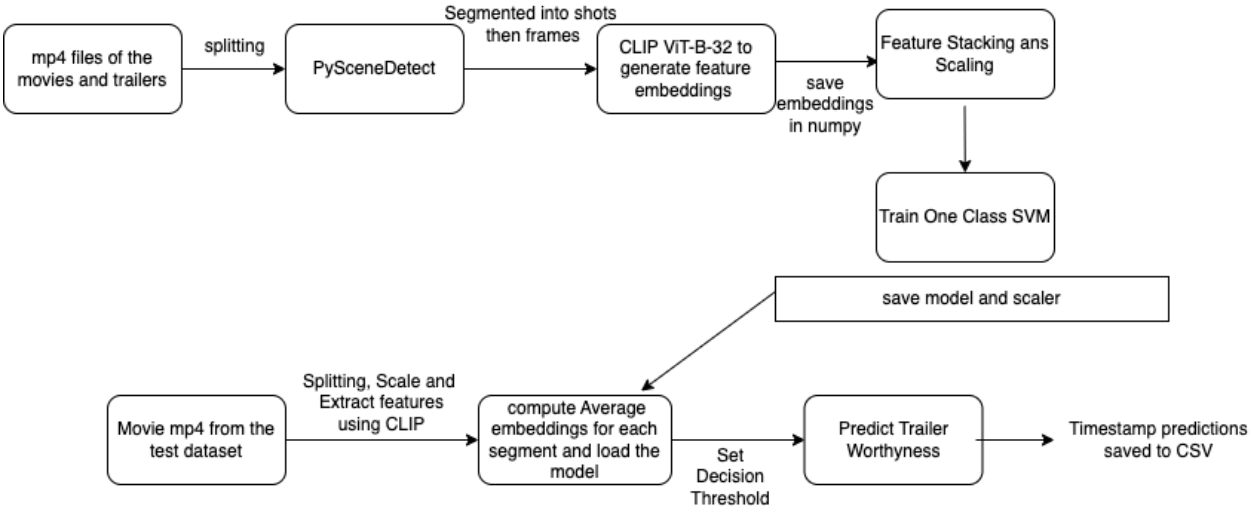

Fig 8.3 Architecture of the video model

CHAPTER - 9

IMPLEMENTATION AND PSEUDO-CODE

The following section details the implementation that has been performed in the entire project as mentioned in the System Design (section 8). The implementation details the preliminary data analysis, Audio Processing and Video Processing.

9.1. Exploratory Data Analysis

This section details how the audio is structured and its pattern using MEL Spectrogram and MFCC scores. These features give an understanding of what timestamps can be possibly extracted in the audio analysis phase that later guides the selection of key timestamps from the video.

The following are the details of the implementation:

A short movie named “Ignore It” [13] has been used to demonstrate the exploratory data analysis for the audio processing phase. This short film is of the “horror” genre. The movie is around 7 minutes As seen in the system design (section 7), the video is supposed to be in mp4 or mov format. The short film is extracted to required format using “pytube” [16], a python library. Further , the audio from this short film is extracted using “moviepy” [17] for the first step of the pseudo code i.e. audio processing. Code for audio extraction - the extracted audio is saved as audio.mp3.

Following this the extracted audio is subsequently split into vocal and accompaniment (background) audio files using “spleeter”, a python library [14]. Code for splitting audio - the “vocals and background” files are stored in /output directory.

A Fourier analysis on the vocals and background audio is performed, however fourier analysis only gives a compressed visual of the audio in a fixed interval of time, and there was a need to dynamically analyse the audio to get highlight information.

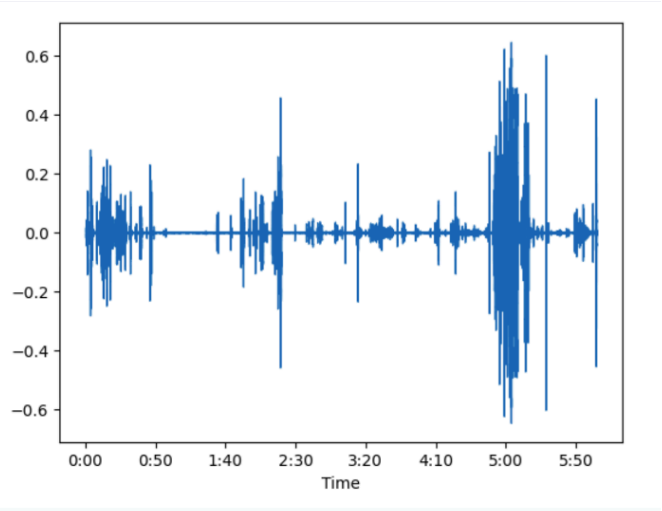

Fig 9.1 Fourier Analysis of the vocals audio

The MEL spectrogram is a spectrogram in which the y axis is in the scale of “MELs”. The MEL scale normalises the pitch difference. The intensity of audio is measured in terms of Decibels instead of amplitude. Decibels make the audio easier for humans to analyse. MEL spectrograms are plotted using the “librosa” library in python [15]. The MEL spectrograms are plotted in a 10 seconds interval.

Figure 9.2 shows the MEL spectrogram for the 32nd Interval (5 min 33 secs - 5 mins 43 secs). This shows intervals where there is loud periodic audio i,e. 1.5 s - 3 s depicted by close bands and darker orange indicating that the person is speaking with more intensity. The intervals 4.5s - 6s and 8.5s - 9s correspond to the man screaming “No Leave her alone !” and “Jessica, Pass”, both of which are told with tremendous intensity and fear (in the context of the movie, Jessica is about to be possessed by an evil spirit and the man, her husband is screaming). There are also continuous bands which indicate there is continuous vocals like talking.

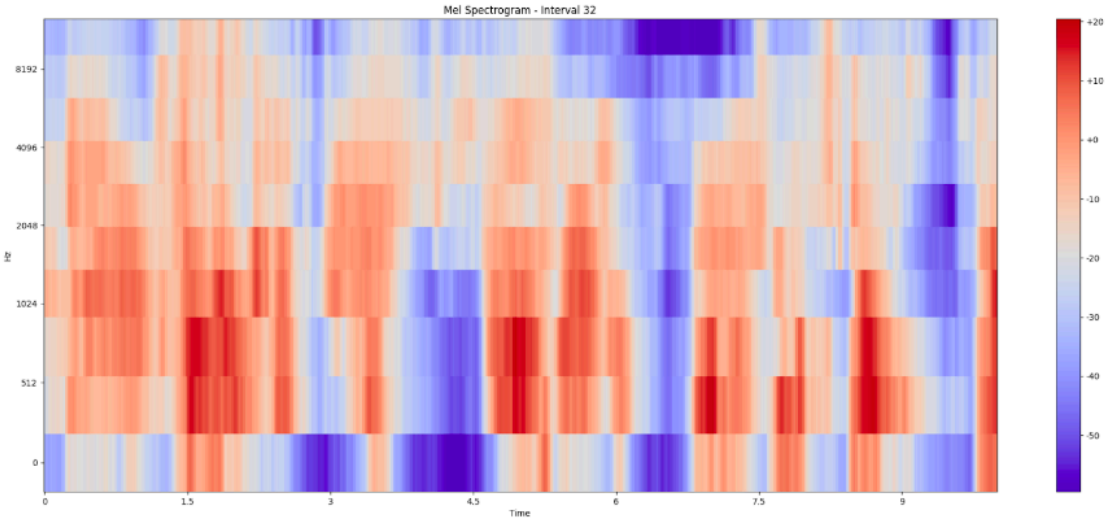

Fig 9.2 MEL Spectrogram for the vocals file

Figure 9.3 shows a comparison between MEL spectrograms for the vocals and background file. The MEL spectrogram for vocals between 4.5s and 9s shows high intensity which indicates a shout or very loud, intense talking. However the interval 1.5s - 4s is blank indicating there was no talking. On analysing the background we can infer that there was a significant audio event, like heavy breathing when the vocals were static.

These details are important in selecting key timestamps during audio analysis. A horror short film trailer usually has one scene in which there is a loud sound like a scream, a significant tagline dialogue and either eerie silence or breathing in the background.

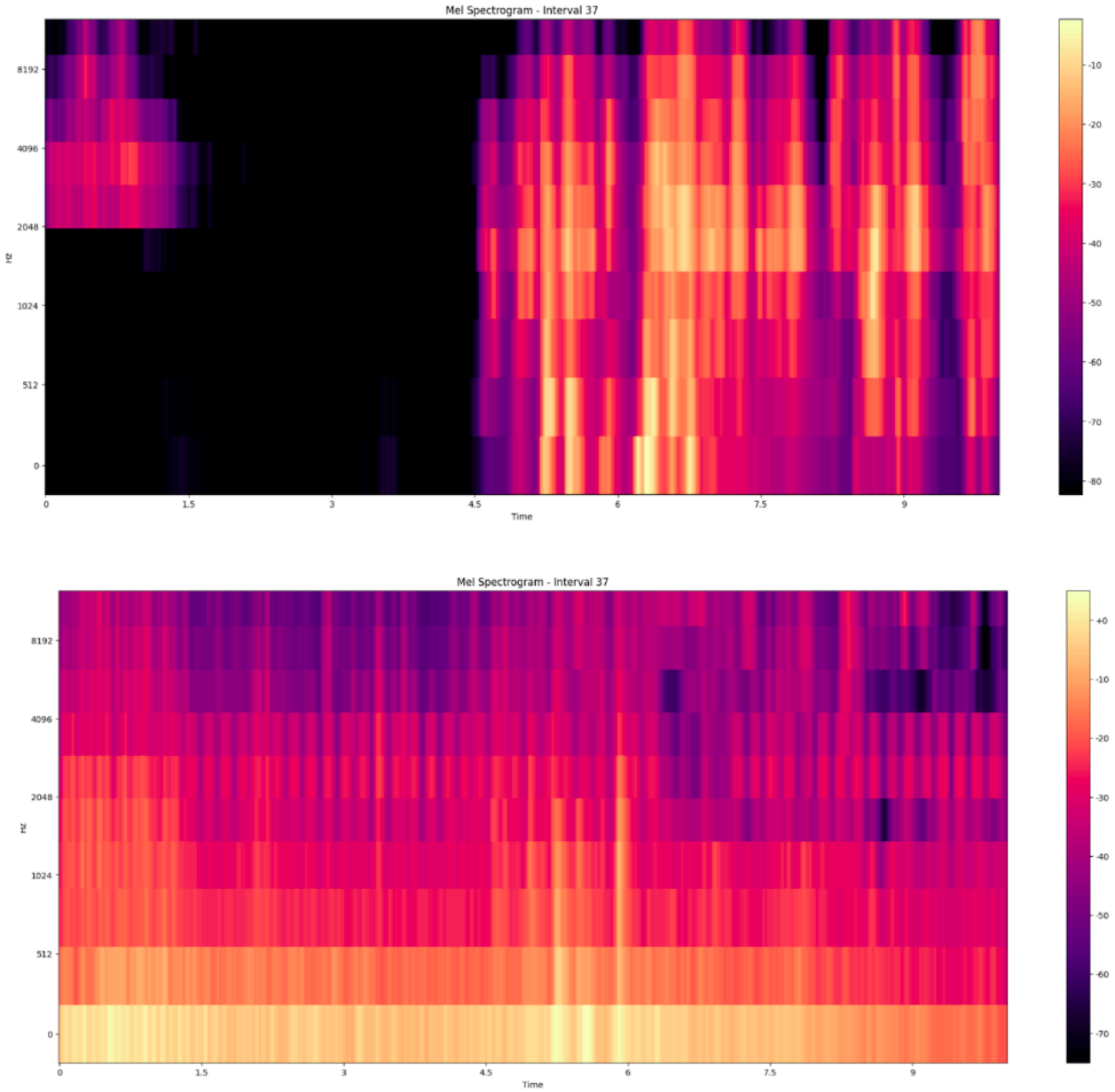

Fig 9.3 Comparison of MEL Spectrograms for Vocals and Background

Following the analysis of MEL Spectrograms , an analysis of MFCC coefficients has been made. MFCCs are a concise representation of audio features. Hence, MFCC coefficients contain information about the rate changes in the different spectrum bands. So they basically describe the shape of the spectral envelope.

MFCC are formed by applying DCT (Direct Cosine Transform) on a list of mel log powers. The MFCC are less sensitive to noise and the variation in pitch of the audio , hence making them most suitable for audio signal processing. As seen here MFCC are constricted to about 13 parameters to capture complex audio signals efficiently.

Given below is the MFCC plot of the entire short film spanning for about six minutes and 33 seconds.

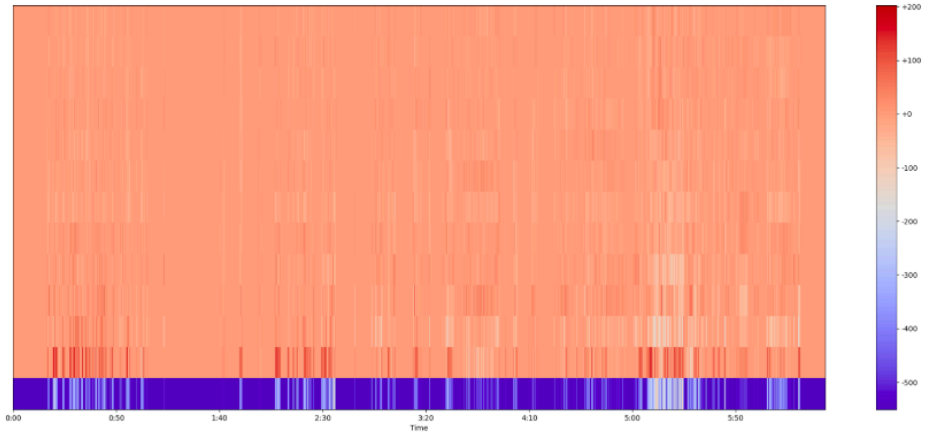

Fig 9.4 MFCC Power Spectrogram

MFCCs for the audio sample:

Time Frame	MFCC 1	MFCC 2	MFCC 3	MFCC 4	MFCC 5	MFCC 6	MFCC 7	MFCC 8	MFCC 9	MFCC 10	MFCC 11	MFCC 12	MFCC 13
0	-550.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	-550.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	-550.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1312	-346.01	-5.48	103.03	35.03	-14.29	58.43	-17.30	-4.83	-10.68	-4.78	-3.02	-7.48	-11.46
1313	-395.02	71.62	71.62	32.87	-8.39	56.58	-0.48	1.05	-1.79	-16.52	-6.46	-0.81	-13.10
1314	-422.79	96.37	55.53	17.84	2.55	46.51	11.16	19.68	-0.92	-21.21	-7.25	-15.75	-9.70
1315	-431.86	103.06	54.12	12.82	1.46	34.31	15.85	25.66	1.49	-18.79	-10.42	-25.08	-13.72
1316	-423.77	115.09	56.93	11.60	-2.50	21.39	9.10	25.24	0.86	-14.47	-5.51	-20.15	-11.09
1317	-385.79	145.99	53.73	-0.97	-18.24	-1.44	0.69	21.52	-9.94	-17.32	-0.58	-7.90	-1.94
1318	-337.11	154.53	30.89	-20.50	-23.35	-2.08	-1.69	28.38	-22.64	-11.18	8.02	0.29	-1.77
1319	-321.24	144.10	16.32	-27.69	-23.54	1.25	-3.94	33.58	-26.01	-3.26	19.48	5.43	-5.91
1320	-303.11	121.17	12.60	-14.37	-4.13	1.49	-7.79	17.85	-29.59	2.80	23.13	4.10	-11.96
1321	-305.14	101.70	11.31	15.70	13.80	-1.13	-6.85	3.64	-29.59	7.49	18.62	2.67	-8.92
1322	-371.70	95.20	13.94	33.44	18.58	-1.86	-3.95	3.41	-22.53	7.82	15.86	2.53	-3.82

Fig 9.5 MFCC Coefficient Values

The values above represent the MFCC features against time frames . It can be noted that the 1st feature describes the overall signal energy while the next twelve features represent spectral features such as shape, slope, roughness, flatness etc. Here the negative values under MFCC1 show that in the time frames as seen above, indicates a high signal energy that seems to be

consistent in these frames. There also occurs frames where the 1st coefficient contains positive values which indicate lower signal energy. MFCC plot and coefficient values can be obtained using “librosa”.

9.2 Audio Analysis

Initially the audio is split into segments of duration 5 seconds.

```
split_movie_audio_seg = splitAudio(movie_audio_file)
split_trailer_audio_seg = splitAudio(trailer_audio_file)
FUNCTION splitAudio(audio_file_mp3):
    // iterate through the audio file and split into segments of 5 seconds and store in a list
    // Input : Movie and trailer Audio File
    //  Output : Lists containing the audio segments
```

Further the audio features such as mfcc, mel, spectral contrast and chroma are extracted from the audio segments. This aids in further audio processing

```
FUNCTION extract_features(file, sr):
    // iterate through the list of segments to extract the audio features and store each feature in a list
    // Input : Lists containing the audio segments
    // Output: Lists containing the audio features computed by the pertaining function
    load audio from file with sampling rate sr
    compute mfcc features from audio
    compute mel spectrogram from audio
    compute chroma features from audio
    compute spectral contrast features from audio
    return  mfcc, mel, chroma, spectral_contrast
```

The features extracted from movie and trailer audio files are further reshaped to fit the audio processing model (here a Siamese Network) and returned at the end of the function. In the next steps, the function generates triplet data for the Siamese LSTM model. It creates pairs of audio feature vectors from movies and trailers and labels them as positive or negative based on their source. Also, different audio features (MFCC, Mel spectrogram, Chroma, and spectral contrast) are concatenated into single vectors. Ensure that every anchor is matched with its corresponding positive/negative sample. Finally, this function scales the feature vectors by StandardScaler and returns anchors and positives along with the labels as NumPy arrays.

Further the Compute Distances Function,calculates the Euclidean distance between two input tensors. This layer calculates the squared differences, sums these differences across the feature dimension, and finally returns the square root of summed squared differences, giving out the distance measure for use in the Siamese LSTM model.

Further the Siamese LSTM model is created using the Siamese LSTM architecture as mentioned in Section8. It takes two input sequences of audio features and processes these through shared LSTM layers. Each input goes through a 128-unit LSTM layer, followed by a 64-unit LSTM layer and finally a dense layer with units 32 and ReLU activation for each of the inputs. It then calculates a distance between these dense outputs using a custom Lambda layer, returning the Keras model for this input and the output distance.

Then the Contrastive Loss function, defines the custom loss that calculates the contrastive loss given a predicted distance between pairs of samples, using a predefined margin to tell apart the similar and dissimilar pairs. Therefore, with precautions against division by zero and handling possible NaN values by setting them to zero, computation of stable loss is guaranteed during training.

After the definition of these functions, the training data which is the movie and trailer audio files are first individually converted to segments . Further their audio features such as mel, mfcc, spectral_contrast and chroma are extracted and saved, Utilizing the feature data , triplet data for the same is generated.This is followed by the reshaping of anchors and positives for them to be used as inputs for the Siamese LSTM model. Subsequently , the Siamese LSTM model is created and a Dropout Layer is added to prevent the overfitting of the model. Accordingly, the model is

compiled with the Adam optimizer, where its learning rate is set to 0.0001 and the clipping value is 0.5, utilizing a custom contrastive loss function with a margin of 1. The model defines various callbacks for efficient training, including ReduceLROnPlateau and EarlyStopping, to adapt the learning rate according to validation loss and stop training in case improvements have stalled. Then, the model trains for 30 epochs with a batch size of 64 and utilizes 20% of the data for validation. The model summary is illustrated in Fig.9.6.

Model: "functional_5"

Layer (type)	Output Shape	Param #	Connected to
input_layer_4 (InputLayer)	(None, 1, 4)	0	-
input_layer_5 (InputLayer)	(None, 1, 4)	0	-
lstm_6 (LSTM)	(None, 1, 128)	68,096	input_layer_4[0][0], input_layer_5[0][0]
lstm_7 (LSTM)	(None, 64)	49,408	lstm_6[0][0]
lstm_8 (LSTM)	(None, 64)	49,408	lstm_6[1][0]
dense_4 (Dense)	(None, 32)	2,080	lstm_7[0][0]
dense_5 (Dense)	(None, 32)	2,080	lstm_8[0][0]
lambda_2 (Lambda)	(None, 1)	0	dense_4[0][0], dense_5[0][0]
dropout_2 (Dropout)	(None, 1)	0	lambda_2[0][0]

Total params: 171,072 (668.25 KB)
Trainable params: 171,072 (668.25 KB)
Non-trainable params: 0 (0.00 B)

Fig. 9.6. Siamese LSTM model Summary

Further the prediction involves the following ,

The extract_features method extracts, for a given audio signal in this case test data, various features like MFCC, Mel spectrogram, Chroma, and Spectral Contrast using the Librosa library. Then it makes all these features have an equal number of frames either by padding or truncating each to the maximum number of frames across those features. Finally, it concatenates these features into one array and computes the mean across time for every feature, thus the resulting feature set is returned.

This is followed by the pad_sequences function that normalizes the length of the input sequences through padding with zeros until it reaches some given maximum length. In cases where a

sequence is shorter than this maximum length, it's padded to match, and if it is longer than the maximum length, it's truncated to fit. It returns an array of the padded sequences.

In the next steps, the function get_key_audio_segments will return key audio segments from clustered features by proximity of cluster centroid. It will calculate the centroid for each unique cluster, loop through segments that belong to a particular cluster, determine their start and end times based on the given segment duration. This function ensures that segments are not larger than the overall audio duration of the movie and calculates the distance of each segment's features to the centroid. If a segment is at a distance less than or equal to a threshold value from the centroid, it is considered as a key moment and added to the list of key segments returned at the end.

```
FUNCTION get_key_audio_segments(labels, features, segment_duration=5, movie_audio_len=0, sr=22050, threshold_distance=0.1):
  // retrieves the key audio segments from the movie audio segment lists
  //Input: labels, features and threshold distance
  //Output: lists containing the start and end time of the keysegments
   initialize key_segments as an empty list
   compute total_audio_duration = movie_audio_len / sr

   get unique_labels from labels
   for each cluster in unique_labels
      find indices of segments belonging to the cluster
      extract features of the cluster
      compute centroid as the mean of cluster features

      for each index in cluster_indices
         calculate start_time = index * segment_duration
         calculate end_time = start_time + segment_duration
         if end_time > total_audio_duration, SKIP this segment
```

```
        compute distance_to_centroid for the segment
        if distance_to_centroid < threshold_distance
            add (start_time, end_time) to key_segments

    return key_segments
```

Further, the predict_timestamps function forecasts the timestamps of key audio moments in a movie by first loading a movie and its trailers, then extracting features using the extract_features function. After that, this function pads the extracted features for the upkeep of shape uniformity, then standardizes them using StandardScaler. Next, reshape the features to match the model expected input format and prepare the inputs as tensors to make a prediction. It obtains the prediction from the model, flattens the predicted features for clustering, and then applies K-means clustering to determine which segments best represent trailer-worthy moments. Finally, key audio segments are extracted based on their proximity to cluster centroids and returned as timestamps.

```
FUNCTION predict_timestamps(model, movie_file, trailer_files, sr=22050, threshold_distance=0.1):
    // return the trailer worthy segments with a prediction label of 1 from the key audio segments lists
    // Input:model, movie files, trailer files, threshold_distance
    //Output: list containing the prediction labels and audio segments
    load movie audio with sampling rate sr
    extract features from movie audio

    for each trailer_file in trailer_files
        load trailer audio with sampling rate sr
        extract features from trailer audio
    end for
```

calculate max feature length across movie and trailer features
pad movie and trailer features to max length

scale features using StandardScaler for normalization

reshape movie and trailer features into 3D input format for model

predict embeddings from model using movie and trailer features as input
reshape predicted embeddings for clustering

apply KMeans clustering to embeddings with num_clusters = 4
get cluster labels for embeddings

calculate key audio segments using labels, embeddings, segment duration, movie audio length, and threshold_distance

return key audio segments

Finally , Siamese LSTM model is loaded and instantiated with loss function and summary of the model architecture is displayed. The sample rate is set to 22,050 Hertz. The function predict_timestamps is invoked in order to identify key audio moments in the movie based on the loaded model and trailer files. The resulting key segments are then saved into a Pandas DataFrame and exported to a CSV file named 'key_audio_moments.csv'.

9.3 Video Analysis

The training movies and trailers that are preprocessed into mp4 format are loaded. A State of The art python library PySceneDetect detects scenes in the video and extracts frames from the start of each scene . The frames are saved for further use.

split_movie_videoSegments = splitVideo(movie_video_file)
split_trailer_videoSegments = splitVideo(trailer_video_file)
FUNCTION splitVideo(video_file)
// This function takes video file and segments it into scenes using scenedetect
// The frames are extracted from scenes and saved
// Input : movie and trailer video files
// Output : frames saved to directory for all the movies and trailers

In the next step, the code initialises the CLIP model selecting the ViT-B-32 architecture. The model is loaded with its corresponding preprocessing pipeline that resizes, normalises and then converts the images into tensors before feeding them into the model. The function “extract_clip_features_for_all_frames” processes all the frames stored in the directory by iterating through it and passing them through the CLIP preprocessing pipeline. The model outputs feature embeddings (vector representations) for each frame. These are averaged to store the embeddings for each shot.
The extracted feature vectors (embeddings) represent the images in a higher dimension where semantically similar images are nearer. The averaged feature vectors for entire shots provide a compact representation of each shot's visual content.

FUNCTION extract_clip_features_for_all_frames(frame_dir):
// This function loads the clip model and sends them to the CLIP preprocessing
// which extracts the visual embeddings
// Input : Frames extracted from all movies and trailers
// Output : Visual Features extracted in the form of embedding

The extracted average vectors are stored in .npy files. The code iterates through a list of movie and trailer files that were loaded before and extracts the features for the movies and trailers and saves them in separate directories.
The next step is to load all the features. The code is designed to iterate over all the files in the features directory that ends with .npy. It appends the features to a list and then stacks them into a single 2D NumPy array using np.vstack.

The next step is to combine the features of the movies and trailers into a unified array for training combined_features is a stacked array containing both the movie and trailer features. movie_features and trailer_features are vertically stacked (np.vstack) to form the final combined_features array. The labels array is created to represent whether the features come from a movie (1) or a trailer (-1). 1 represents inliers (movies) and -1 represents anomalies (trailers). The features are then scaled using a standard scaler to ensure that the model can work effectively and the features are on a similar scale.The Standard Scaler computes the mean and standard deviation. A One class SVM is used to classify the trailer points as anomalies. It is modelled to recognise the outliers from the normal distribution of movie features. The model is fit and saved along with the scaler.

OneClassSVM Model Summary	
Attribute	Value
Kernel	rbf
Nu (Anomaly Ratio)	0.5
Gamma	auto
Number of Support Vectors	8
Support Vector Indices	[0 1 2 3 4 5 6 7]
Dual Coefficients Shape	(1, 8)
Intercept (bias)	[-1.09188865]

Fig 9.7 : One Class SVM Model Summary

The next step is to predict whether these segments obtained from the audio analysis are "trailer-worthy" or not based on a One-Class SVM model trained in the previous step. The model

and scaler are loaded. For each specified segment in the audio timestamps, frames are extracted using the CV library.

The purpose of this is to extract CLIP features for each frame on the fly instead of storing. cv2.VideoCapture(video_path) opens the video file. The frames per second of the video are retrieved using cap.get(cv2.CAP_PROP_FPS) and start_frame and end_frame are calculated based on the timestamps in the audio file. cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) converts the frame from BGR to RGB. The frame is then converted to a PIL image and preprocessed using the CLIP preprocess pipeline. The image is passed through the ViT model to extract its feature embeddings. The feature embeddings for all frames in the shot are averaged to provide a single feature vector for the timestamp.

svm_model.decision_function computes the decision score, which is proportional to the distance of the sample from the separating hyperplane. The trailer features are labeled -1 for the model to identify anomalies The decision scores indicate how anomalous the movie segment is. The nearer the scores are to the hyperplane, the more trailer worthy they are. If the score is above the defined threshold, the segment is predicted as "trailer-worthy" (1), otherwise it is predicted as "not trailer-worthy" (0).

```
FUNCTION predict_trailer_worthy(extracted_features):
        //  Predicts trailer-worthiness using the decision function scores and a threshold.
        // Input : Movie features extracted using CLIP for the audio segments obtained
        //  during audio analysis.
        // Output : Trailer worthy timestamps and distance from hyperplane
```

Based on the decision threshold the trailer worthy segments are predicted, and saved in a CSV file. These segments are those which have the key audio and video moments and are the highlights of the movie

CHAPTER - 10

EXPERIMENTAL RESULTS AND DISCUSSION

The split tasks of audio video processing proved not only to be computationally effective but also showed results that gave equal importance to the most impactful auditory and visual features that can be expected in a horror film. The final segments extracted highlighted key acoustic elements such as screams and unsettling atmospheres and key visual elements like darkness and fear filled imagery.

As mentioned in section 8, the evaluated metrics aids in the effectiveness of the Machine learning models. As previously stated , three parameters Hamming Score(HS), Intersection Over Union(IOU) and Task Accuracy(TA) have been considered.

Hamming Score as a metric here represents the ratio of the number of segments that are highly trailer worthy(closer to the decision threshold) to the total number of extracted segments. When tested over 20 percent of the dataset the average Hamming Score was 0.6930. This showed that over 60 percent of predicted segments were found to be highly trailer worthy. Fig 10.1. illustrates the top 10 Hamming Scores for the movies analyzed.

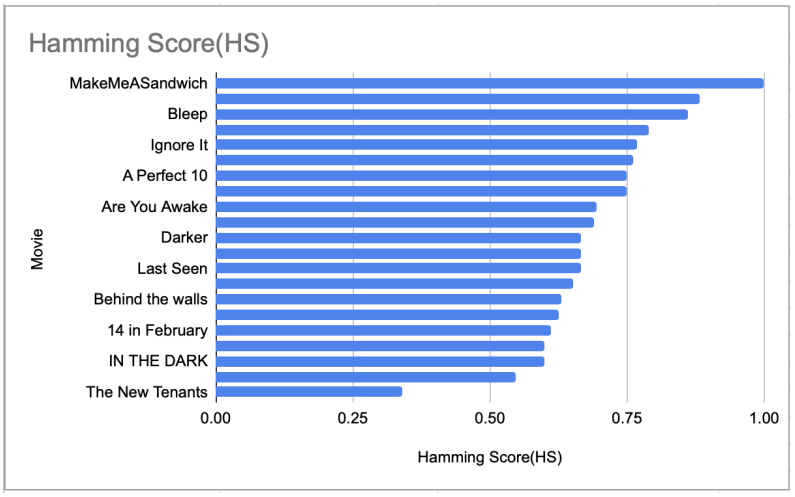

Fig. 10.1. Top ten Hamming Scores

Intersection Over Union is the ratio of the number of segments that lie in both extracted trailer segments and the actual trailer segments(ground truth) to the union of trailer worthy segments between the original trailer segments and the extracted trailer segments. The average IOU was computed as 0.3455. This indicates that over 30 percent of the extracted segments are present in the ground truth while there is a scope for other trailer worthy segments that could be used for the trailer. The top 10 IOU Scores are displayed in Fig 10.2.

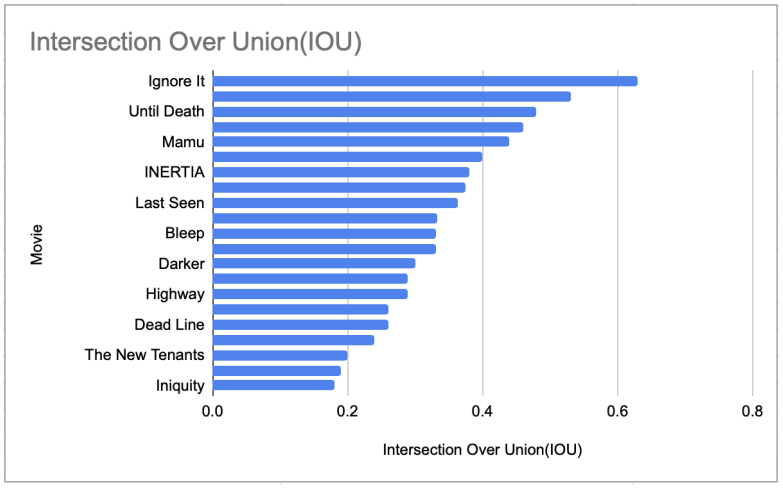

Fig. 10.2. Top ten IOU Scores

Task Accuracy here is the ratio of the number of predicted segments that are present in ground truth to the total number of segments. When computed over the testing data ,the average Task Accuracy was 0.5625.This likely indicates that over 50 percent of the extracted segments were predicted correctly. Fig 10.3. represents the top 10 Task Accuracy scores among the movies tested.

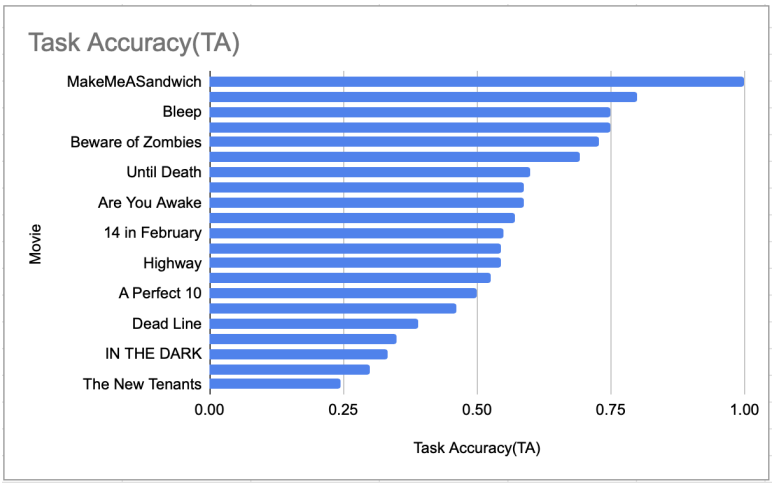

Fig. 10.3. Top ten Task Accuracy Scores

CHAPTER - 11

CONCLUSION AND FUTURE WORK

Trailers play a vital role to give a glimpse into an upcoming movie and pique audience attention. However manually curating a trailer for movies is labor intensive and time consuming. Hence prompting to automating the trailer generation process presents a valuable opportunity for the production team to work on other creative aspects of filmmaking.

The repository utilized consists of 311 short films and their corresponding trailers which mostly are of the horror genre. This aforementioned data is split into separate modalities and processed. As mentioned in section IV, our novel methodology proved to be computationally effective.

The results obtained as seen in section 10 shows that a significant proportion of the predicted segments are highly trailer worthy and are present in the ground truth. The evaluation metrics demonstrate the effectiveness of our novel method in predicting trailer worthy segments. The hamming score indicates that more than 60% of the predicted segments are highly trailer worthy. The Intersection over Union score suggests that more 30% of the predicted segments overlap with the ground truth and the Task Accuracy signifies highlights the model's capacity to predict segments accurately

The Hamming Score and Task Accuracy Score of 1 highlights the model's effectiveness in prediction of the most high trailer worthy moments that match the ground truth. The IoU scores could be relatively lower because there are a significant number of key trailer segments in this film, and not all of them may make it into the final trailer.

While the results are promising, the future research could explore the inclusion of more sophisticated audio and visual features that capture the nuances of a horror movie. To enhance the model's robustness, a wider variety of horror films across different subgenres can be included in the dataset.

REFERENCES

[1] P. Papalampidi, “Structure-aware narrative summarization from multiple views,” era.ed.ac.uk, Jan. 2023, doi: https://doi.org/10.7488/era/2946.

[2] Wang, L., Liu, D., Puri, R., Metaxas, D.N. (2020). Learning Trailer Moments in Full-Length Movies with Co-Contrastive Attention. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science(), vol 12363. Springer, Cham. https://doi.org/10.1007/978-3-030-58523-5_18

[3] Gan, Bei & Shu, Xiujun & Qiao, Ruizhi & Wu, Haoqian & Chen, Keyu & Li, Hanjun & Ren, Bo. (2023). Collaborative Noisy Label Cleaner: Learning Scene-aware Trailers for Multi-modal Highlight Detection in Movies.

[4] Wei, Fanyue et al. “Learning Pixel-Level Distinctions for Video Highlight Detection.” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022): 3063-3072.

[5] C. Bretti, P. Mettes, Hendrik Vincent Koops, Daan Odijk, and Nanne van Noord, “Find the Cliffhanger: Multi-modal Trailerness in Soap Operas,” Lecture Notes in Computer Science, pp. 199–212, Jan. 2024, doi: https://doi.org/10.1007/978-3-031-53308-2_15.

[6] J. Sheng, Y. Chen, Y. Li, and L. Li, “Embedded learning for computerized production of movie trailers,” Multimedia Tools and Applications, vol. 77, no. 22, pp. 29347–29365, Apr. 2018, doi: https://doi.org/10.1007/s11042-018-5943-3.

[7] N. Sadoughi et al., “MEGA: Multimodal Alignment Aggregation and Distillation For Cinematic Video Segmentation Supplementary Material.” Accessed:Mar.14,2024.[Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/supplemental/Sadoughi_MEGA_Multimodal_Alignment_ICCV_2023_supplemental.pdf

[8] P. Mishra, C. Diwan, S. Srinivasa, and G. Srinivasaraghavan, “AI based approach to trailer generation for online educational courses,” CSI Transactions on ICT, vol. 11, no. 4, pp. 193–201, Nov. 2023, doi: https://doi.org/10.1007/s40012-023-00390-1.

[9] K. Porwal, H. Srivastava, R. Gupta, S. Pratap Mall, and N. Gupta, “Video Transcription and Summarization using NLP,” SSRN Electronic Journal, 2022, doi: https://doi.org/10.2139/ssrn.4157647.

[10] Dong, Hao-Wen et al. “CLIPSonic: Text-to-Audio Synthesis with Unlabeled Videos and Pretrained Language-Vision Models.” 2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) (2023): 1-5.

[11] H. Kakimoto, Y. Wang, Y. Kawai and K. Sumiya, "Extraction of Movie Trailer Biases Based on Editing Features for Trailer Generation," 2018 IEEE International Symposium on Multimedia (ISM), Taichung, Taiwan, 2018, pp. 204-208, doi: 10.1109/ISM.2018.000-6.

[12] M. Hesham, B. Hani, N. Fouad and E. Amer, "Smart trailer: Automatic generation of movie trailer using only subtitles," 2018 First International Workshop on Deep and Representation Learning (IWDRL), Cairo, Egypt, 2018, pp. 26-30, doi: 10.1109/IWDRL.2018.8358211

[13] Ignore It: https://www.youtube.com/watch?v=hs3paMLb9Qg

[14] Spleeter library: https://pypi.org/project/spleeter/

[15] Librosa MEL Spectrogram : https://librosa.org/doc/latest/generated/librosa.feature.melspectrogram.html

[16] MoviePy : https://pypi.org/project/moviepy/

[17] Pytube : https://pytube.io/en/latest/

[18]CLIP: https://github.com/openai/CLIP

Prajna R Capstone_SBN08_Pg8.docx (1).pdf

ORIGINALITY REPORT

8%	4%	5%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to PES University Student Paper	3%
2	link.springer.com Internet Source	1%
3	arxiv.org Internet Source	<1%
4	Jiachuan Sheng, Yaqi Chen, Yuzhi Li, Liang Li. "Embedded learning for computerized production of movie trailers", Multimedia Tools and Applications, 2018 Publication	<1%
5	Dong, Hao-Wen. "Generative AI for Music and Audio", University of California, San Diego, 2024 Publication	<1%
6	journal.linguaculture.ro Internet Source	<1%
7	Yibo Sun, Zhe Sun, Weitong Chen. "The evolution of object detection methods",	<1%

	Engineering Applications of Artificial Intelligence, 2024 Publication	
8	www.coursehero.com Internet Source	<1 %
9	"Computer Vision – ECCV 2024", Springer Science and Business Media LLC, 2025 Publication	<1 %
10	pt.scribd.com Internet Source	<1 %
11	Submitted to Edge Hill University Student Paper	<1 %
12	"Advances in Visual Computing", Springer Science and Business Media LLC, 2020 Publication	<1 %
13	dblp.dagstuhl.de Internet Source	<1 %
14	Fanyue Wei, Biao Wang, Tiezheng Ge, Yuning Jiang, Wen Li, Lixin Duan. "Learning Pixel-Level Distinctions for Video Highlight Detection", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022 Publication	<1 %
15	deepai.org Internet Source	<1 %

16	"MultiMedia Modeling", Springer Science and Business Media LLC, 2024 Publication	<1 %
17	Honoka Kakimoto, Yuanyuan Wang, Yukiko Kawai, Kazutoshi Sumiya. "Extraction of Movie Trailer Biases Based on Editing Features for Trailer Generation", 2018 IEEE International Symposium on Multimedia (ISM), 2018 Publication	<1 %
18	"Pattern Recognition and Computer Vision", Springer Science and Business Media LLC, 2025 Publication	<1 %
19	ojs.aaai.org Internet Source	<1 %
20	Submitted to La Trobe University Student Paper	<1 %
21	"Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020 Publication	<1 %
22	wsl.iiitb.ac.in Internet Source	<1 %
23	medium.com Internet Source	<1 %
24	Submitted to Asian Institute of Technology	

	Student Paper	<1%
25	Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, Juan Carlos Niebles. "Home Action Genome: Cooperative Compositional Action Understanding", 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021 Publication	<1%
26	Submitted to University of Tennessee Knoxville Student Paper	<1%
27	assets.researchsquare.com Internet Source	<1%
28	Eslam Amer, Ayman Nabil. "A Framework to Automate the generation of movies' trailers using only subtitles", Proceedings of the 7th International Conference on Software and Information Engineering - ICSIE '18, 2018 Publication	<1%
29	Submitted to University of Canberra Student Paper	<1%
30	assets.amazon.science Internet Source	<1%
31	nitandhra.ac.in Internet Source	<1%

32	openaccess.thecvf.com Internet Source	<1 %
33	portal.sinteza.singidunum.ac.rs Internet Source	<1 %
34	Cem Ünsalan, Berkan Höke, Eren Atmaca. "Embedded Machine Learning with Microcontrollers", Springer Science and Business Media LLC, 2025 Publication	<1 %
35	Yuchen Fan, Dario Antonelli, Alessandro Simeone. "Chapter 5 Continual Learning for Human-Machine Collaboration in VUCA Environments", Springer Science and Business Media LLC, 2024 Publication	<1 %
36	export.arxiv.org Internet Source	<1 %
37	repository.kaust.edu.sa Internet Source	<1 %
38	www.research.ed.ac.uk Internet Source	<1 %
39	Lecture Notes in Computer Science, 2006. Publication	<1 %
40	Najmeh Sadoughi, Xinyu Li, Avijit Vajpayee, David Fan, Bing Shuai, Hector Santos-	<1 %

Villalobos, Vimal Bhat, Rohith Mv. "MEGA: Multimodal Alignment Aggregation and Distillation For Cinematic Video Segmentation", 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023
Publication

41 Prakhar Mishra, Chaitali Diwan, Srinath Srinivasa, G. Srinivasaraghavan. "AI based approach to trailer generation for online educational courses", CSI Transactions on ICT, 2023 <1%
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		