Cinematic Video Editing: Integrating Audio-Visual Perception and Dialogue Interpretation

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Engineering
by Research

by

Rohit Girmaji
2021900013
rohit.girmaji@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)

Hyderabad 500 032, INDIA
JUNE 2025

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Cinematic Video Editing: Integrating Audio-Visual Perception and Dialogue Interpretation" by Rohit Girmaji, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Vineet Gandhi

To
My Family and Friends

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my adviser, Professor Vineet Gandhi, for his expert guidance, patience, and unwavering support throughout my research. His insightful feedback and constant encouragement have been instrumental in shaping both this thesis and my academic development. I am especially grateful for his availability and willingness to assist at every stage of the work.

I would also like to thank Professor Ramanathan Subramanian for his valuable contributions and suggestions during our editing project, which significantly improved the quality of our work.

I am also deeply grateful to the CVIT Lab and Professor Vineet Gandhi for providing an enriching academic environment and the necessary resources to complete my research. I extend my sincere thanks to my research colleagues at the CVIT Lab for cultivating a collaborative and inspiring atmosphere throughout the research journey.

I am extremely thankful to my friends and colleagues Bhav and Siddharth, with whom I achieved the most research output. I am also grateful to Adhiraj, Sudheer, Sarthak, Ritvik, Shreyank, and Sarath Sivaprasad for their constant support and collaboration. I believe there is always something to learn from each of them, and I truly enjoyed the time spent working together. I really enjoyed my tea breaks with my friend and colleague Neil, chatting about things beyond the lab and just life in general.

Finally, I would like to thank my family for their love, encouragement, and understanding throughout this journey. Their unwavering belief in me gave me the strength to persevere, especially during challenging times. I am deeply grateful for their emotional support and constant motivation. This acknowledgment would be incomplete without a special mention of my wife, Sanju Lahari, to whom I am forever indebted.

Abstract

This thesis focuses on advancing automated video editing by analyzing raw, unedited footage to extract essential information such as speaker detection, video saliency, and dialogue interpretation. At the core of this work is **EditIQ**, an automated video editing pipeline that leverages speaker cues, saliency predictions, and large language model (LLM)-based dialogue understanding to optimize shot selection—the critical step in the editing process.

The study begins with a comprehensive assessment of active speaker detection techniques tailored for automated editing. Using the BBC Old School Dataset, annotated with active speaker information, we propose a robust audio-based nearest-neighbor algorithm that integrates facial and audio features. This approach reliably identifies speakers even under challenging conditions such as occlusions and noise, outperforming existing methods and closely aligning with manual annotations.

In the domain of video saliency prediction, we present ***ViNet-S*** and ***ViNet-A***, compact yet effective models designed to predict saliency maps and identify salient regions in video frames. These models are computationally efficient, balancing high accuracy with reduced model complexity.

Starting with a static, wide-angle camera feed, EditIQ generates multiple virtual camera feeds, mimicking a team of cinematographers. Speaker detection, saliency-based scene understanding, and LLMs-driven dialogue analysis guide shot selection, which is formulated as an energy minimization problem. This optimization ensures cinematic coherence, smooth transitions, and narrative clarity in the final output.

The efficacy of EditIQ is validated through a psychophysical study involving twenty participants using the BBC Old School dataset. Results demonstrate EditIQ's ability to produce aesthetically compelling and narratively coherent edits, surpassing competing baselines and showcasing its potential to transform raw footage into polished cinematic narratives.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The widespread availability of affordable, compact cameras with advanced features like 4K recording has made high-quality video capture more accessible. However, while recording is now easier, video editing remains a complex, skill-intensive process. This complexity is evident across industries such as live broadcasting, sports, news, and social media, where there is a high demand for swift, dynamic video production.

Producing professional live recordings involves skilled camera operators capturing performances from multiple angles. These feeds are then meticulously edited to create a polished final product. The challenges of live filming, where retakes are impossible and equipment options are limited, further complicate the process. The need for precision and the lack of room for error add to the difficulty.

Manual video editing, which is both slow and requires significant expertise, exacerbates the complexity of producing high-quality recordings. As a result, achieving a professional standard necessitates a well-coordinated team of camera operators, multiple cameras, and experienced editors, all contributing to the intricate and costly nature of the production.

In the world of video production, there are three key stages: pre-production, production, and post-production. Pre-production is all about preparation, where everything from scriptwriting to scheduling is planned before filming begins. Production is the stage where the actual filming happens, capturing the video content. Once the footage is collected, it moves into post-production, where it is carefully edited to tell a compelling story or convey a specific message. This stage is detailed and time-consuming, requiring a skilled editor to piece together the best moments from the footage.

Many video production companies often use fixed wide-angle cameras placed far from the action to capture the entire scene. While useful for archiving and providing a broad overview, these static shots cannot often engage viewers by missing out on close-ups of faces, emotions, and interactions that are key to storytelling. However, with high-resolution cameras, the need for multiple camera crews can be minimized. A virtual pan/tilt/zoom (PTZ) camera can be simulated using vertical editing by choosing a cropping window at each time inside the original recording. The pan/tilt movements are simulated by moving the cropping window vertically and horizontally inside the image boundaries of the original recording and the zoom is simulated by changing the size of the cropping window keeping a fixed output resolution. By using Virtual PTZ (Pan-Tilt-Zoom) simulation, editors can create the illusion of multiple

camera angles from a single static shot, adding variety and depth to the footage. This approach results in a more dynamic and visually engaging video, giving the impression that it was filmed from different perspectives.

The video editing pipeline consists of the following major steps:

- **Content Analysis:** Content analysis refers to extracting key information from a scene, which is essential for editing. For example, the location of the actors on stage, the events and actions happening on stage, identifying the speaker, salient regions (from saliency prediction), subtitle generation, etc.

- **Shot/Rush Generation:** The shot generation step involves using a virtual camera simulation to create multiple virtual PTZ shots from a static wide-angle recording. This is achieved by moving cropping windows within the master shot, each following a specific actor or group of actors. The process is framed as a convex optimization problem to ensure the shots are well-composed and cinematic.

- **Shot Selection:** The shot selection step involves choosing the most effective shot at each moment to best convey the storyline. This is done using an optimization algorithm that evaluates the importance of each generated shot while following cinematic principles like avoiding jump cuts, minimizing rapid transitions, and maintaining a consistent rhythm. Finding the importance of each of the multiple shots at any given time is obtained by using different cues like speaker information, salient regions, and dialogue interpretation obtained through content analysis.

This thesis explores content analysis in video editing, focusing on extracting key information such as active speakers, actions using saliency prediction, and dialogue interpretation with Large Language Models. It examines how these factors influence shot selection in the editing pipeline and their impact on the final edited output. We quantitatively evaluated our findings using the BBC Old School dataset, as discussed in Chapter 3 & Chapter 5.

We briefly discuss the significance of the speaker, human gaze, and dialogue interpretation through LLMs in video editing before outlining the key contributions of this thesis.

Speaker Driven Editing: Awareness of the person speaking at a given moment (active speaker) is crucial in video editing, especially in dialogue-driven scenes. Simpler editing styles like Dragnet (described by Murch in his book [1]) rely entirely on the active speaker information. In Dragnet style, both the video and audio of each character's entire line are included within each edit, and the editing process is akin to a tennis match, with rapid back-and-forth cuts that leave no space for reaction. The other more sophisticated cuts like L-cuts or J-cuts also hinge upon the knowledge of the active speaker, where the voice of the person seen in the outgoing shot continues, or the sound of the speaker who is about to be shown is heard before the cut happens. Just like human editors, accurate knowledge of the active speaker is essential for the realm of automated editing.

Naturally, majority of prior research efforts towards automated editing heavily rely upon active speaker information [2–4]. However, identifying and detecting the active speaker remains challenging for automated editing systems, unlike human editors who can do so effortlessly. Most existing methods either assume that the active speaker information is available [4] or utilize handcrafted features and methods to detect it [2]. To this end, evaluating the applicability of current state-of-the-art active speaker detection algorithms in the context of automated video editing is essential. In chapter 3, we address this by proposing a simple audio-based method to predict an active speaker at any given time in a video and evaluate the existing speaker detection algorithms and its impact on the video editing task.

Gaze Driven Editing: Previous efforts in automated video editing have aimed to enhance static wide-angle footage into more dynamic content through machine learning and optimization techniques. The objective is to lower production costs and complexity while maintaining high-quality output. To reduce dependence on multiple camera operators, Gandhi *et al.* [5, 6] proposed a method for automatically creating multiple clips suitable for editing by simulating pan-tilt-zoom movements within a single static camera's frame. Moorthy *et al.* [3] showed that effective camera selection could be achieved by utilizing eye-gaze data from viewers. Their approach assumes that people naturally focus on the most salient parts of a scene, and gaze can act as a proxy to identify important elements. While their technique yields strong results, its applicability is limited by the need for gaze data, which might not always be accessible. To address this limitation, we use a computational video saliency model to simulate human gaze. In Chapter 4, we present an efficient Video Saliency Prediction model designed to estimate salient cues such as actions within a scene.

LLMs for Dialogue Interpretation: To remove the reliance on external user data for video editing, we introduce *EditIQ*, an entirely automated multi-camera video production system for staged events, utilizing footage from one or more wide-angle, fixed-position cameras. *EditIQ* aims to harness the potential of large language models (LLMs) for the task of automated video editing. Recent research has demonstrated the powerful capabilities of LLMs in understanding essential scene components, such as emotions [7], entailment [8], and co-reference resolution [9]. Our approach is the first to show that these models can be effectively employed to guide camera selection based on the narrative and dialogue, and to determine which elements of a scene should be visually highlighted. In Chapter 5, we describe the *EditIQ* pipeline, explain how LLMs and video saliency contribute to the shot selection process, and evaluate the pipeline through a psychophysical experiment with twenty participants using the BBC Old School dataset.

1.1 Contributions

The following are the key contributions of this thesis:

- We annotate the BBC Old School Dataset with active speaker labels, background noise, buzzer presses, and silence, making it suitable for automated editing research. Additionally, we evaluate three classes of Active Speaker Detection (ASD) algorithms within this dataset, including an

off-the-shelf audio-visual algorithm and adapted audio-based algorithms that integrate face verification and visual tracking. Finally, we present comprehensive results and discussions on the effectiveness of these algorithms when used with a dynamic programming-based editing framework.

- We propose ViNet-S, a lightweight model with just 9 million parameters that outperforms the original ViNet, and ViNet-A, which uses an STAL backbone to improve feature representation in multi-subject videos, particularly human-centric ones. Our experiments demonstrate that an ensemble of ViNet-S and ViNet-A achieves state-of-the-art results across nearly all studied datasets. We present extensive qualitative and quantitative results using nine different datasets.

- **Semantic shot potentials:** We leverage innovative LLM and visual saliency-based predictions as potentials to assess the significance of various rushes derived from the original footage. LLMs offer dialogue-driven visual cues, whereas saliency enhances the understanding of scene actions.

- **Fully automated editing pipeline:** The potentials obtained from active speaker detection, LLM predictions, and visual saliency results are integrated with cinematic constraints, framing the selection of camera shots as a discrete optimization task. This entire process is automated, needing only the wide-angle video and scene data to produce the edit. *EditIQ* can edit a 2-minute video with five performers in just 2 minutes on a PC with an Nvidia 4090Ti GPU. In comparison, manual editing is far more labor-intensive and time-consuming.

- **Comprehensive Evaluation:** Our method is tested on the professionally curated BBC-OSD dataset [10], which is tailored for evaluating automated editing of wide-angle footage. Findings from an extensive user study show that *EditIQ* outputs are favored by users in terms of narrative impact, retention of emotions and actions, and the overall viewing experience.

Chapter 2

Background

2.1 Video Editing

2.1.1 Grammar of Cinematography

Cinematography is the art and science of capturing motion and live scenes on film or digital media. The term comes from the Greek words ”kinema” (movement) and “graphein” (to record). It encompasses visual elements like composition, lighting, and camera movement. The cinematographer, who once operated the camera, now primarily supervises the camera operator and decides on settings such as colour, depth-of-field, zoom, and framing. Just as grammar structures verbal communication, cinematography has its conventions and rules, forming its unique “grammar.” This section covers the accepted vocabulary and guidelines for constructing and presenting visual elements, based on standard cinematography and editing literature.

2.1.1.1 Aspect Ratio

The aspect ratio of an image refers to the proportional relationship between its width and height, specifically the ratio of width to height. It is typically presented in the format W:H, where W stands for width and H represents height. For instance, a 16:9 aspect ratio indicates that for every 16 units of width, there are 9 units of height. In cinema, common aspect ratios are 1.85:1 and 2.39:1, while television and online videos frequently use 4:3 and 16:9 aspect ratios. Figure 2.1 represents an image in different aspect ratios.

2.1.1.2 Shot Sizes

What sets movies apart from live theater is how filmmakers control the audience’s view. In theater, the audience sees the entire stage in a wide shot and can only focus on different elements as they choose. In contrast, movies use various shot sizes to guide what viewers see and how they see it. By changing shot sizes, filmmakers keep the visual experience dynamic and prevent monotony, while also highlighting key elements of a scene at different moments.

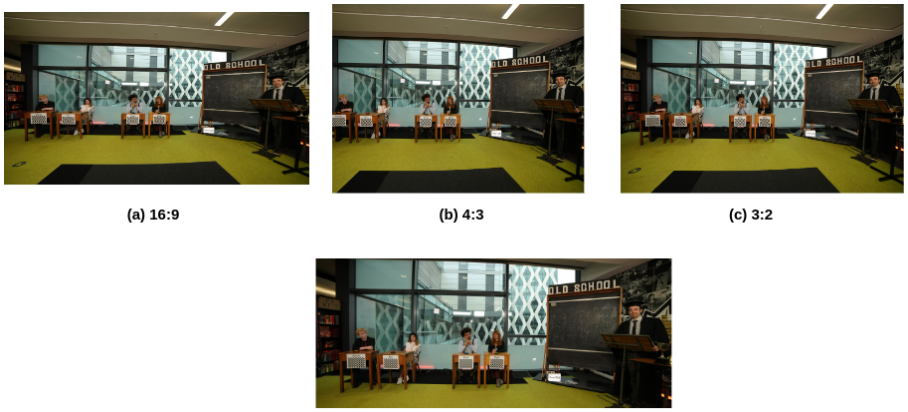


Figure 2.1: Different aspect ratios. Images taken from BBC-OSD

Figure 2.2: Example of 7 different Shot Sizes. Image reproduced from web link

Various shot sizes can be employed for specific purposes or to evoke different emotional responses. In his book *Grammar of the Shot*, Thomson classifies shot sizes into three primary groups: the Long Shot (LS) or Full Shot (FS), the Medium Shot (MS), and the Close Up (CU). These fundamental categories can be expanded into additional shot types. Figure 2.2 illustrates 7 different shot sizes. For the edited output videos, we utilize only ELS, MS, MLS, and MCU for any visual representation. Below is a brief description of each shot size:

- **Extreme Long Shot (ELS):** Captures a wide field of view, often used as an establishing shot to show the entire scene and its surroundings. It helps depict the relationship between actors and their environment.

- **Long Shot (LS) / Full Shot (FS):** Frames the entire body of the subject, with both head and feet visible. It emphasizes the actor while still showing some of the surrounding environment, making it easier to observe details like clothing and movement.

- **Medium Shot (MS):** Frames the subject from around the waist up, closely mirroring how we naturally observe others in conversation. It offers a balance between showing the subject and the surrounding environment, creating a comfortable and familiar view for the audience.

- **Close-Up (CU):** Focuses tightly on the subject, usually from just above the hairline to just above the shoulders. This shot emphasizes facial expressions and emotions, minimizing the visibility of the environment.

- **Medium Long Shot (MLS):** Frames the subject from around knee height, providing more context of the environment while still focusing on the subject.

- **Medium Close-Up (MCU):** Frames the subject from the chest up, offering a closer view of the subject's expressions while still providing some environmental context.

- **Extreme Close-Up (XCU):** This shot type occupies the entire frame with the subject's details, such as the eyes, lips, or fingers, bringing the viewer very close. When capturing individuals this intimately, we focus on their subtle gestures and expressions, almost as if through a magnifying lens. Extreme Close-Ups create an intense sense of closeness to the subject, providing a level of intimacy seldom experienced in daily life.

2.1.1.3 Composition

In addition to visual grammar elements like shot sizes and aspect ratios, components such as headroom, look room (Rule of Thirds), and camera angles (high and low) contribute to the overall composition. In cinematography, composition refers to the framing and arrangement of elements within the image. While following composition guidelines helps create visually pleasing shots, these rules are flexible and can be intentionally broken to achieve unique ideas and perspectives.

2.1.1.4 Camera Movements

The above described compositions come under static compositions where both the subject and the camera remain stationary. However, compositions can shift with camera subject movement or both. This dynamic aspect is what sets motion pictures apart from still photographs. While a photograph might imply movement, it only captures spatial relationships within a single, unchanging frame. In contrast,

motion pictures are composed in both space and time, allowing for a more dynamic and evolving visual experience.

Thomson [11] classifies the shot types based on the camera movement into simple, complex and developing shots:

- **Simple Shots:** The camera is completely static, with no movement at all (lens, camera, or mount).
- **Complex Shots:** Involve lens and camera movement, but the mount remains stationary.Includes pans, tilts, and zooms.
- **Developing Shots:** The camera's mount moves, optionally combined with lens and camera movement. Includes dolly and crane shots.

The following is a brief description of different ***Complex Shots***:

- **Pan:** Horizontal rotation of the camera from a fixed point, like turning your head side to side. Used to follow a subject or shift focus between subjects.
- **Tilt:** Vertical movement of the camera lens, similar to nodding your head up or down. Often reveals vertical elements like buildings or people.
- **Zoom:** Adjusting the lens to change the field of view without moving the camera. "Zoom-in" narrows the field, while "zoom-out" broadens it.

In this thesis, we work with the BBC Old School Dataset, detailed in section 3.2.1, which consists of footage captured with static cameras—meaning neither the lens, camera, nor mount moves. However, addressing complex shots is crucial for video editing. Therefore, we apply algorithms and methods from [5] to generate complex shots from these static recordings. Specifically, we simulate Pan, Tilt, and Zoom effects using videos originally recorded with stationary cameras.

2.1.2 Multi Camera Setup vs Single Camera Setup

Single Camera setup: A single-camera setup, as the name suggests, involves using only one camera to capture a scene. The simplest form is a static camera that records the entire action, commonly used for archival purposes like theatre recordings. This method is cost-effective and straightforward, requiring no camera operator and allowing for automatic recording. However, it's less engaging for professional productions due to the lack of varied angles. In traditional filmmaking, varying viewpoints with a single camera require multiple takes, with actors repeating their performances as the camera is repositioned for different shots. This approach gives directors greater control over each shot and is ideal for capturing specialized shots with complex camera movements. However, it is time-consuming and demands experienced actors who can consistently replicate their performances. Additionally, maintaining continuity during editing can be challenging, and the setup is unsuitable for live performances where multiple takes aren't possible.

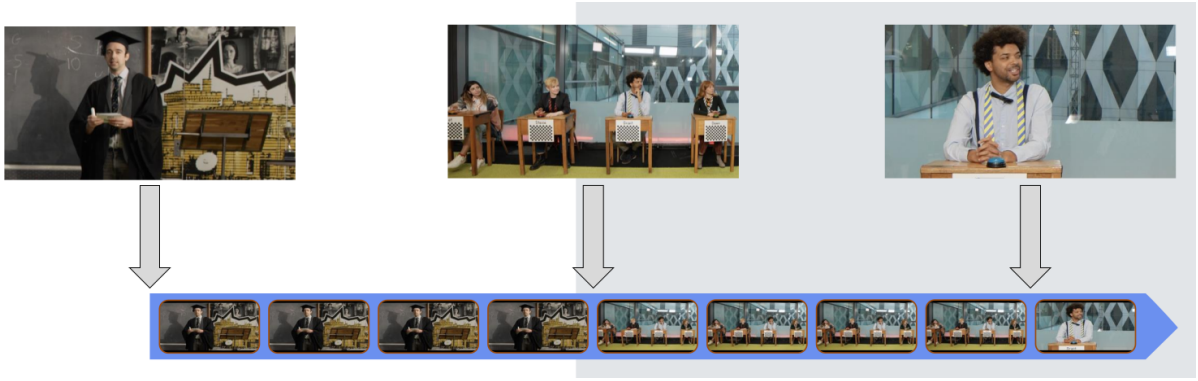


Figure 2.3: A sample edited sequence. Image reproduced from web link

Multi-Camera Setup: A multi-camera setup involves using multiple cameras to capture a scene from different angles and shot sizes simultaneously, allowing various shots to be obtained in a single take without interrupting the action. For example, one camera might focus on close-ups of actors while another captures a wide shot of the entire scene, known as the "master shot," which ensures continuity and serves as a fallback during editing. The primary advantage of a multi-camera setup is its efficiency in recording time and ease of editing, as it maintains continuity and allows for a variety of angles. However, this setup gives the director less control over individual shots, requires a larger crew and more equipment, and can limit complex camera movements due to the need to accommodate other cameras' fields of view. Despite these drawbacks, multi-camera setups are essential for live broadcasts, such as sports events, and are favoured in recording staged performances, offering viewers more dynamic and detailed perspectives than a single-camera setup.

2.1.3 Vertical Video Editing

Video Editing: Video editing, or montage, involves selecting, rearranging, and cutting shots from raw footage (known as "rushes") to create a coherent and effective narrative. While editing can also include applying visual enhancements, our focus is on the manipulation and sequencing of shots over time to ensure the story is told in the most compelling way. The editor's role is crucial in choosing the best shots and arranging them to present the action and narrative clearly and engagingly. Figure 2.3 shows a sample edited sequence.

Vertical Editing: Traditional editing with single or multi-camera setups involves capturing scenes with either one camera requiring multiple takes or multiple cameras each focusing on different angles. This method limits the number of available shots. Vertical editing, introduced by Walter Murch, offers an alternative by allowing editors to crop different shots from a single master shot. This technique, unlike traditional methods, enables more flexible shot selection without the need for multiple cameras or repeated takes.

Virtual PTZ Camera: Vertical editing also facilitates the creation of a virtual pan/tilt/zoom (PTZ) camera by adjusting the cropping window within the original master shot. By moving this window horizontally or vertically, editors can simulate pan and tilt movements, while changing the window

(a)

(b)

Figure 2.4: (a) Master Shot (b) Some shots generated from the Master Shot. Here only 5 shots are shown.

size mimics zooming. This method allows for dynamic shot variations without physically moving the camera and can be applied to both real and virtual environments, though in this context, it refers to editing real-world footage.

2.1.4 Rush Generation

Using vertical editing, multiple virtual cameras can be simulated from a single master shot, focusing on different elements within a scene. This process, known as "rush generation," mimics the work of a virtual camera crew, producing a variety of compositions that can be edited together like traditional multi-camera footage. This computational approach streamlines the editing process, providing flexibility and efficiency in shot selection and assembly. Gandhi et.al [5] presents the most comprehensive work to computationally generate rushes. We use this method in the ***Rush Generation*** (section 3.3.3.1 & section 5.2.2) phase of our experiments.

A significant advantage of using a virtual camera setup is that it requires only a single, non-operated camera to cover an entire scene (Figure 2.5(b)), allowing the editor to create multiple smaller shots from the master shot, as illustrated in Figure 2.4. For instance, Figure 2.4 demonstrates that, with a traditional multi-camera setup, generating five shots would require five different cameras. In contrast, using a single camera setup would necessitate five repetitions of the same scene. However, with a virtual camera setup, five shots can be produced from a single master shot. Furthermore, if the scene depicted in Figure 2.4(a) is captured from four different viewpoints using four cameras, it would yield 20 shots from all angles. In a traditional single or multi-camera setup, this would require 20 repetitions of the scene and 20 cameras. Therefore, a virtual camera setup is both cost-effective and time-saving compared to traditional methods, which would demand multiple cameras or repeated takes. Virtual camera simulation also provides flexibility, allowing the editor to explore a wide range of shot options, particularly when

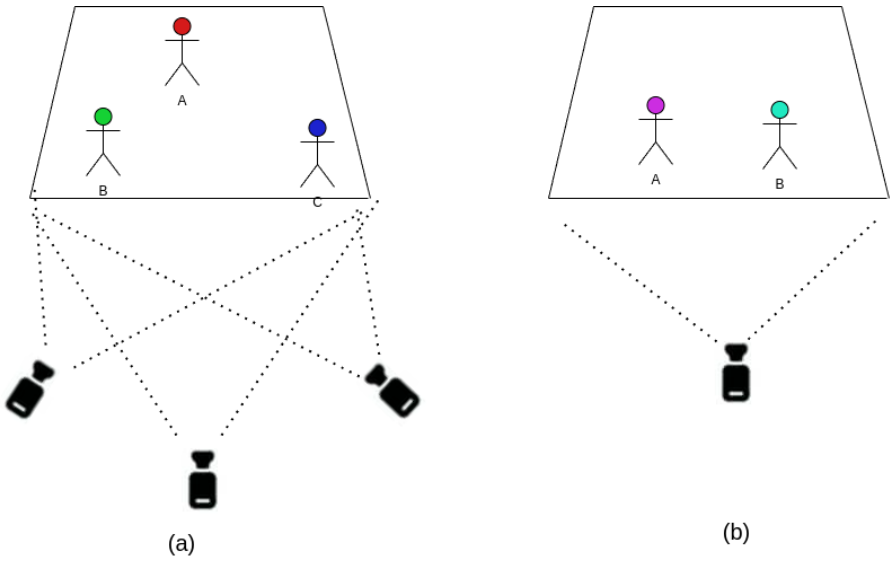


Figure 2.5: (a) Multi Camera Setup where each camera covers the entire scene in different view points, unlike the traditional multi-camera setup where each of the camera captures different shots/framings of different actors.(b) Single Camera Setup where a single camera covers the entire scene unlike the traditional single camera setup

more actors are involved, and can enhance traditional multi-camera setups by generating additional shots from the master shot without incurring extra costs.

However, a significant disadvantage of virtual camera simulation is the loss of resolution when creating tighter shots, as only a portion of the original image is used. This makes it suitable mainly for contexts where reduced resolution is acceptable, such as web streaming. Additionally, virtual camera shots cannot change the camera angle, meaning all generated shots retain the original perspective of the master shot. Despite these limitations, advancements like 4K cameras could expand the applicability of virtual camera simulation in the future.

2.1.5 Shot Selection

Given the multiple rushes, the shot selection step selects the most impactful shot to narrate the story at each moment. This process is framed as a discrete optimization problem, evaluating the significance of each shot while following cinematic rules—avoiding jump cuts, rapid transitions, and maintaining rhythm. Shot importance is determined as shot potential using scene elements like speakers, saliency, and dialogue, with cinematic principles modeled as penalties. The final edit is found by optimizing a path through an editing graph, where each frame contains $2^n - 1$ nodes representing shots, and edges indicate cuts or continuations.

2.1.6 Cinematic Constraints

Following cinematic principles is crucial for producing a refined, well-edited output. These principles are implemented through penalty terms. We identify three distinct types of penalties, which are

explained further below. The overall penalty for a specific shot is the sum of these individual penalties at each timestamp.

Overlap Penalty: When transitioning between two consecutive shots, too much overlap can result in a jump cut, disturbing the flow and creating a jarring visual effect. To prevent such abrupt transitions, we introduce an overlap penalty that aims to reduce the overlap between consecutive shots. This penalty is enforced only when two separate shots, s_t^i and s_{t+1}^j, are involved, and a cut is made between them.

$$O(s_t^i, s_{t+1}^j, \gamma) = \begin{cases} 0 & \text{if } \gamma \leq \alpha \\ \frac{\mu\gamma}{\alpha} & \text{if } \alpha < \gamma \leq \beta \\ \nu & \text{if } \gamma > \beta \end{cases}$$

In this case, γ denotes the overlap ratio, which is determined by the intersection-over-union (IoU) of two consecutive shots, s_t^i and s_{t+1}^j. The penalty is applied in a piecewise manner: no penalty is imposed if the IoU is below the threshold α, and a linear penalty is applied for IoU values between α and β, and a large penalty ν is assigned when the overlap ratio surpasses β, indicating a substantial overlap that breaches cinematic guidelines.

Rhythm Penalty: The timing of cuts is crucial in shaping the overall atmosphere of a scene during video editing. The length of a shot plays a key role in how the audience perceives the mood and intensity of a sequence. For instance, extended shots foster a slower pace, evoking calmness or emotional depth, often found in romantic or reflective scenes. Conversely, brief shots generate a quicker pace, amplifying tension or excitement, a common technique in action sequences. To control the rhythm of cuts, we introduce the rhythm penalty, which adjusts shot duration to preserve the cinematic flow. This penalty is applied depending on the duration of the current shot (τ), calculated as follows:

$$R(s_t^i, s_{t-1}^j, \tau) = \begin{cases} \gamma_1 \left(1 - \frac{1}{1+\exp(l-\tau)}\right) & \text{if } i \neq j \\ \gamma_2 \left(1 - \frac{1}{1+\exp(-m+\tau)}\right) & \text{if } i = j \end{cases}$$

In this equation, τ represents the duration of the current shot, while l and m are parameters that regulate the rhythm of cuts. The constants γ_1 and γ_2 act as scaling factors for the rhythm penalty. When transitioning to a new shot ($i \neq j$), the penalty becomes greater if the shot is cut too soon, i.e., before $\tau = l$ seconds, to discourage overly quick transitions. Conversely, if a shot is held for an extended period ($i = j$), the penalty increases as τ surpasses m seconds, prompting a cut to bring in fresh visual content. These two conditions work together to manage the rhythm of cuts, ensuring the scene maintains a balanced pace—neither too fast nor too stagnant.

Transition Penalty: Rapid cuts in video editing can disorient the viewer and diminish the emotional impact of a scene. Such quick transitions may obscure crucial elements, hinder narrative coherence, and reduce the overall visual appeal. To avoid this, we introduce a transition penalty that enforces a minimum duration for each shot. For two consecutive shots, s_t^i and s_{t+1}^j, the penalty is expressed as:

Figure 2.6: Sample Eye Tracker

$$T(s_t^i, s_{t+1}^j) = \begin{cases} 0 & \text{if } i = j \\ \lambda_{trans} & \text{if } i \neq j \end{cases}$$

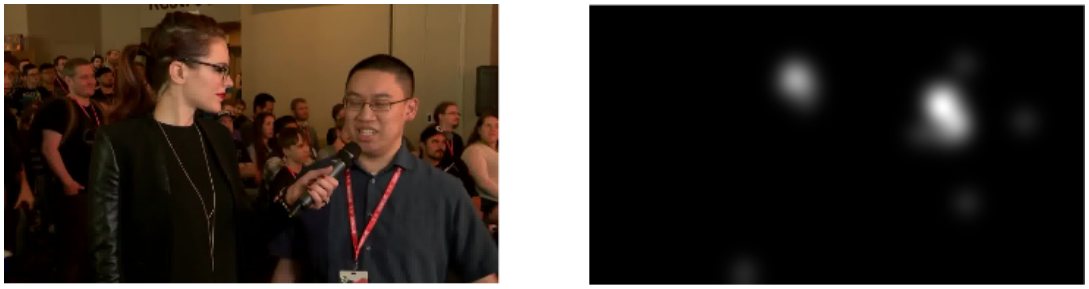
Here, λ_{trans} is the transition penalty parameter.

2.2 Video Saliency Prediction

2.2.1 Eye Tracker

Our objective is to explore how visual information is processed by the human brain, how the eyes respond to this information through movement, and how these reactions influence the subsequent visual data captured. To achieve this, we require a device that can accurately capture the eye's response to different types of visual stimuli and project this data onto a 2D plane. In the past, participants were asked to use a mouse to identify regions of interest within an image, but this approach resulted in inaccurate fixation maps and, consequently, flawed saliency maps. As a result, models trained on these datasets produced unreliable predictions.

However, with recent technological advancements, eye trackers have become available, enabling precise, real-time tracking of eye movements. Figure 2.6 illustrates a typical eye-tracker, which comprises cameras, projectors, and algorithms. The projectors generate a near-infrared light pattern on the eyes, and the camera captures high-resolution images of the eyes and the pattern. To determine eye position and gaze point, machine learning, image processing, and mathematical algorithms are employed. The device is mounted on the monitor displaying the image, and the participant views the image briefly while the eye movements are recorded at a predefined frequency. This data is used to create a binary fixation map, and applying a Gaussian filter yields a probability distribution known as a saliency map. These



(a) (b)

Figure 2.7: (a) Sample frame in a video from MVVA dataset. (b) Corresponding Saliency Map.

saliency maps are significantly more reliable as they are based on actual eye-tracking data, thereby aiding in the development of more robust saliency prediction models. Figure 2.7 shows a sample frame and its corresponding saliency map.

2.2.2 Video Saliency Datasets

In this section, we will discuss the creation of saliency datasets, using the DHF1K dataset as an example [12]. DHF1K is one of the largest and most comprehensive video saliency datasets, designed to capture gaze behaviour during free viewing of videos. It comprises 1,000 videos with diverse content and lengths, annotated with eye-tracking data from 17 observers.

The creators of DHF1K began by examining existing datasets to understand how dynamic human fixations were influenced by both static image features and temporal video information, as well as to identify the gaps in available data. For example, the Hollywood-2 dataset, which was the largest at the time, lacked generality because it contained only videos from Hollywood movies. To address this, the DHF1K team collected videos from YouTube using around 200 key terms (such as dog, walking, car, etc.) and carefully selected 1,000 videos. This large-scale, high-quality dataset was essential to ensure content diversity, which is crucial for creating a long-lasting benchmark. All videos were standardized to a 30fps frame rate and resized to a 640×360 spatial resolution.

Ensuring diversity was a key aspect of the dataset creation, so each video was manually annotated with one of 150 category labels and further classified into seven main categories (such as daily activities, sports, and art). The selection process also accounted for various objects within the videos, as previous research indicated that object information significantly influences human fixations.

After collecting the videos, the next step was to annotate them with real eye-tracking data from humans. This was done using the Senso Motoric Instruments (SMI) RED 250 system, which recorded eye movements at a sampling rate of 250 Hz. Eye-tracking data was collected from 17 participants (10 males and 7 females, aged between 20 and 28) who completed eye-tracker calibration and had less than a 10% fixation dropping rate. In total, 51,038,600 fixations were recorded across 1,000 video sequences, encompassing 582,605 frames and a total duration of 19,420 seconds.

Chapter 3

Assessing active speaker detection algorithms through the lens of automated editing

3.1 Related Works

3.1.1 Active Speaker Detection Algorithms

Active speaker detection (ASD) algorithms aim to identify active speakers in a scene [13]. ASD requires input from both visual and audio modalities. If we have beforehand knowledge of the actor's faces and their corresponding voices in a scene, then ASD can be performed by pure audio-based diarization [14]. In the wild setting, where such information is unknown, recent methods apply deep neural networks on face tracks to detect if the voice is synchronized with the lip and face movement [15]. Our work investigates a controlled setup where the numbers of actors and their identities are known upfront; hence, both approaches are applicable. We briefly review the advancements in speaker diarization and ASD below. We then discuss automated editing algorithms that utilize active speaker information.

3.1.1.1 Speaker Diarization

Speaker diarization is the process of separating an audio recording that contains multiple speakers into distinct segments based on the speaker's identity. Most speaker diarization systems [16–18] consist of multiple relatively independent components (a) a speech segmentation module, which removes non-speech segments, (b) a module to extract speaker-discriminative embeddings [19, 20], (c) a clustering module and (d) a refinement module which enforces additional constraints to further refine the diarization results [16]. More recent attempts have aimed to consolidate these modules and train diarization in an end-to-end manner. The end-to-end Neural Diarization (EEND) family of approaches [21–24] model diarization as a multi-label classification problem using permutation-invariant training.

3.1.1.2 Audio Visual ASD

In the early days of Audio Visual Active Speaker Detection (ASD), basic visual features such as upper body [25] and facial [26] movements were utilized to predict the active speaker. However, the ef-

fectiveness of this method was limited due to the weak correlation between body movements and speech activity. Later, the combination of audio and visual information proved to be much more beneficial in performing ASD [27]. Audio Visual Fusion techniques approached the task by assigning speech to one of the speakers in a video [28]. Some methods view ASD as a classification model that evaluates each speaker in the video and outputs an active speaker label for each of them [29]. Lately, various deep learning architectures have been proposed like TalkNet [30] with attention mechanism, and Graph Neural Networks [31] have been developed and have provided significant performance improvements in ASD. We briefly describe TalkNet which is used in our experiments below.

3.1.1.2.1 TalkNet is a complete system developed to process cropped face videos along with their corresponding audio to identify whether the individual is speaking in each frame. As shown in Figure 3.1, this system consists of a feature representation frontend and a speaker detection backend classifier. The frontend contains both an audio temporal encoder and a video temporal encoder, which transform the frame-based audio and video inputs into time-sequenced embeddings that encapsulate the temporal context. The backend classifier utilizes an inter-modality cross-attention mechanism to align audio and visual content in real time, as well as a self-attention mechanism to detect speaking activity within the temporal context at the level of the utterance.

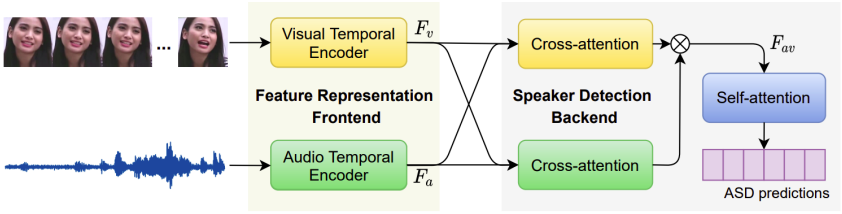


Figure 3.1: TalkNet Architecture. Image reproduced from [30]

3.1.2 Video Editing

Several previous automatic editing methods employ speaker information for editing. Classical computational editing systems [32–35], for example, use speaker detection algorithms to determine who is talking and select a camera known to have a shot of that person. More recently, [4] proposed an optimization-based approach for automatically creating well-edited movies from a 3D animation. They employ speaking as one of the main action categories. Work by [2] proposes an editing framework based on a user-specified set of film-editing idioms. They employ idioms like speaker visibility, which ensures that the speaker of each line of dialogue is visible. The work by Moorthy et al. [3] suggests that speech-based editing scores highly with respect to conveying actor emotions. We take a similar approach to [3] and replace gaze potential with speaker potential. Our work is also related to previous works that pose video editing as a discrete optimization problem, solved using dynamic programming [36], [4], [37], [38].

Figure 3.2: Screenshots of a scene from 4 different camera views in the Old School BBC Dataset. Screenshots from Camera 1 to Camera 4 can be seen from left to right.

3.2 Datasets

3.2.1 BBC Old School Dataset

The study utilizes the BBC Old School Dataset (BBC-OSD) [10]. BBC-OSD, curated by BBC R&D, is a comprehensive resource for advancing research into AI-driven automated video editing. It features wide-angle, zoomed-out camera views that capture the activities of multiple actors in each scene, offering a varied scenario for assessing ASD algorithms. Unlike typical medium-close-up shots (e.g., newsroom settings), these wide-angle views support both virtual camera simulations [5] and camera selection in terms of view and virtual shot [3]. The dataset includes raw footage from a multi-camera shoot of a game show called Old School, as well as processed videos representing multiple takes of different scenes from the show. Screenshots from different camera views of the dataset videos can be visualized in Figure 3.2. It is important to note that the multi-camera setup used for this dataset differs from traditional multi-camera setups. Instead of capturing various shots or framings of the actors, this setup covers the entire scene from four distinct viewpoints. Figure 2.5(a) illustrates a camera setup similar to that used in this dataset.

We use the Edit Decision List (EDL) corresponding to the human-edited programme to extract videos of a total duration of 30 minutes from the processed shoot videos as shown in Figure 3.3. They also provide a human-edited programme as a benchmark for automated editing systems.

3.2.2 Dataset Statistics

The dataset consists of 18 video clips with a total duration of about 30 minutes. There are 5 unique identities with voice tracks and faces. There is only one off-screen speaker in the data. *Bell ring* and *Buzzer* sounds are the non-speech sounds.The total duration of overlapped speech segments and speech activity in the video amounts to 1.3 minutes and 25 minutes respectively.

3.3 Methodology

3.3.1 Annotation Process

We provide ground truth speaker annotations for these videos taken from one camera view, which are the same for all the camera views.We use the open-source project VIA tool for annotations. Its GUI and annotation process can be visualized in [Figure-3.4]. We follow the below steps for annotation:

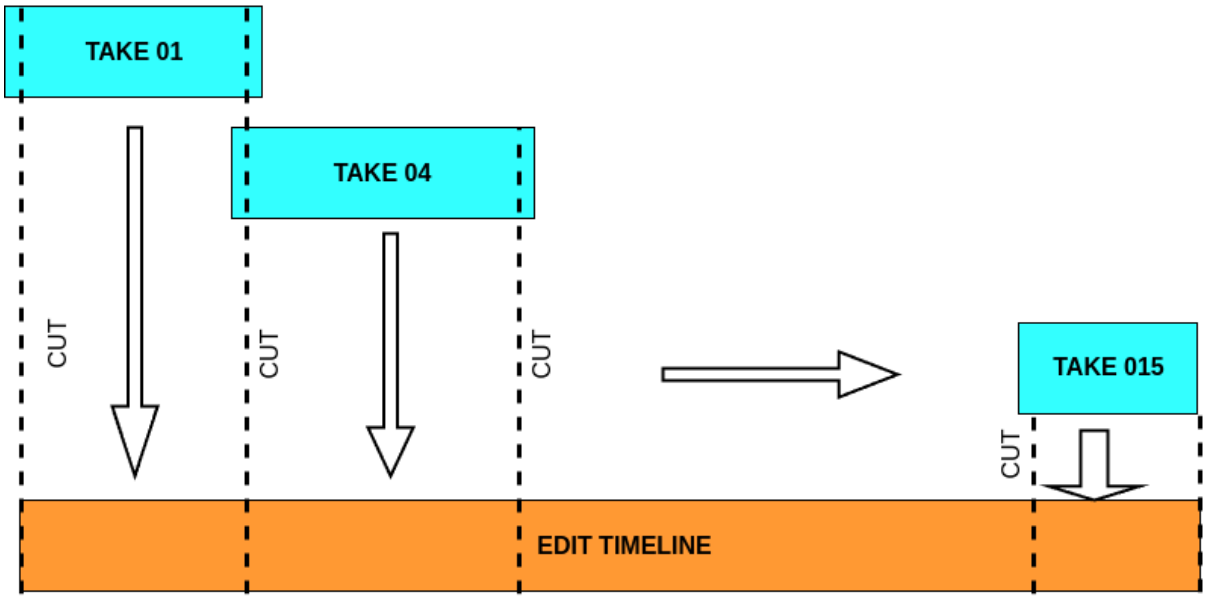


Figure 3.3: Edit Timeline/Edit Decision List: A sequence of chosen moments from different takes by professionals, woven together to tell the story.

3.3.1.1 Generating Face and Voice Tracks

For face crop tracks generation, we follow the steps described in the face track annotator of VIA tool. We first automatically generate face tracks using VGGFaceTracker [39, 40]. Then we manually filter, select and update the annotations. For all the videos, we manually generate voice tracks by watching the video and listening to the audio stream concurrently. Human annotators refined the start and end of speech segments to get accurate labels for the segments. We merge neighbouring segments of the same speaker if the gap between the segments is less than 1 second. We also consider the speech segments even if it is less than 1 second and generate voice tracks and labels for such segments. In addition to the speaker's voice tracks annotation we also provide labels for off-screen speakers, and non-speech sounds like bell ring, buzzer. As a final step, the annotations are manually verified by 3 annotators to get quality labels.

3.3.1.2 Labelling the face and voice tracks

As all the videos in the dataset have the same set of speakers, video level identity labels of speakers are the same across all the videos.

3.3.2 Speaker detection methods

We describe different types of active speaker detection methods used and the ways we rely on speaker information in the task of video editing with different approaches.We have used Audio Visual Active Speaker Detection (ASD), Audio-based Speaker Diarization and Audio-based KNN Classifier methods for speaker detection

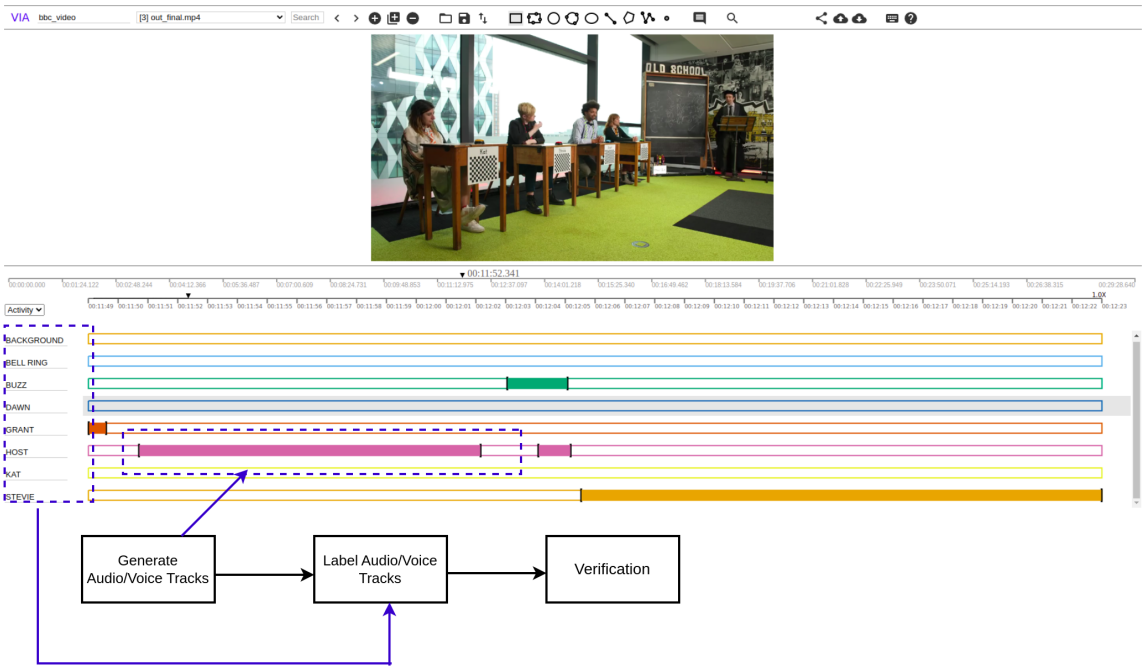


Figure 3.4: Annotation tool and process for speaker annotations. Background is the off-screen speaker, Bell ring is the non-speech sound.

3.3.2.1 Audio Visual Active Speaker Detection(ASD)

Active Speaker Detection (ASD) models predict the active speaker at the frame level. An Active Speaker in a video is a speaker whose face is visible and audible simultaneously. We consider TalkNet [30] an ASD model for our experiments. TalkNet is a classification model that takes cropped faces in the video and corresponding audio as input and outputs active speaker labels for each face. The model consists of a feature representation frontend and a speaker detection backend. The frontend generates visual and audio spatio-temporal features and the classifier backend consists of an inter-modal cross-attention and self-attention mechanism followed by a classifier head to generate the active speaker label.

3.3.2.2 Audio-based Speaker Diarization

In our work, we employ the recent state-of-the-art EEND method proposed by Bredin et al. [23, 24].The model trained on the *AMI*, *VoxConverse*, and *DIHARD* datasets [41–43] is used in our experiments.

3.3.2.3 Audio-based KNN Classifier

We use K-Nearest Neighbour (KNN) Classifier for our approach. We take three 10 second audio samples of each speaker to form the search space and perform nearest neighbour search of a test sample to get the class label prediction. We take the majority class label of the top-K nearest neighbours as the test sample label.

We generate X-vector speaker embeddings [44] with a pre-trained TDNN model using SpeechBrain [45]. These embeddings will now become our search space for the KNN algorithm.

For a given video, we perform a sliding window approach and predict labels for each segment through K-nearest neighbour search. For segments with more than one predicted label, we take the label with the highest similarity measure. We use cosine similarity as the similarity measure.

3.3.3 Video Editing

Here we describe the ways we leveraged active speaker information (*Speaker Potential*)in the task of video editing with two different approaches, i.e. Speaker Guided Greedy Editing(SGE) and Speaker Guided Optimization Editing(SOE). We also describe the pre-processing steps like Shot Generation which is required for both the approaches

3.3.3.1 Rush Generation

In the BBC old school data set, we are given multiple camera views for a video and a human-edited video. We employ the algorithm [5] for shot generation.

For a master shot with n actors, we create $2^n - 1$ virtual shots in total. This includes $\binom{n}{1}$ single shots (featuring individual actors), $\binom{n}{2}$ two-shots (featuring pairs of actors), $\binom{n}{3}$ three-shots, and so forth, all presented in a 16:9 aspect ratio. These virtual shots, together with the master shot, are collectively known as "rushes," which are later used for selection and editing.

For shot generation, we need tracks of the actors in a frame (bounding boxes). We use ByteTrack [46], which is a Multi-Object Tracking method that estimates bounding boxes and identities of objects in videos. They adopt an object detector YOLOX [47] to detect the bounding boxes with a confidence score. In the first step of the algorithm, the bounding boxes are divided into high-confidence and low-confidence score bounding boxes. In the second step, it tries to associate the high-confidence score bounding boxes with the tracklets and then associates the low-confidence score bounding boxes with the unmatched tracklets.

We use ByteTrack for its efficiency and its ability in handling special scenarios like person occlusions, crossing past each other etc, which are comprised in OSD.

3.3.3.2 Speaker Potential

Equation 3.2 quantitatively measures the importance of each shot at every time instant. Previous works [2] and [4] estimate the actions/emotions in a given shot by either relying on additional meta-data or bottom-up computational features. [48] and [49] have shown that the gaze data recorded from users enables effective localization of focal scene events. We extend this idea to calculate speaker potential similar to that of gaze potential in [3] of each shot using active speaker information at each time frame (s_t).

Assuming we have active speaker information (the actor who spoke at that instant of time) per each frame, we determine the speaker potential for each shot. We adopt a bottom-up approach, we first calculate the speaker potential of (lower-order shots) single shots, which capture individual actors using 5.6.

$$S(s_t^x) = \begin{cases} \lambda & x\ is\ speaker \\ 0 & otherwise \end{cases} \tag{3.1}$$

where s_t^x refers to a shot s at time t that contains a single actor x

Speaker potential for shots with multiple actors (higher-order shots) is then computed from the speaker potentials of constituent lower-order shots using equation 3.2.

$$S(s_t^{ab}) = S(s_t^a) + S(s_t^b) - |S(s_t^a) - S(s_t^b)| \tag{3.2}$$

where s_t^{ab} refers to a shot s at time t that contains set of actors $\{a, b\}$

For instance, speaker potentials of two 1-shots $S(s^a)$ and $S(s^b)$ can be used to compute the speaker potential of a 2-shot $S(s^{ab})$. Similarly, speaker potentials of two 2-shots $S(s^{ab})$ and $S(s^{bc})$ can be used to compute speaker potential of a 3-shot $S(s^{abc})$

It can be seen in Equation 3.2 if there is only one speaker 1-shot with the active speaker gets high speaker potential. If a and b are the active speakers, the 2-shot($S(s^{ab})$) that contains both the active speakers gets the maximum speaker potential.

3.3.3.3 Shot Selection

Speaker Guided Greedy Editing: For speaker-guided greedy editing(SGE) we select the shot that best captures the speaker from the generated shots. For speaker information, we use manual annotations and active speaker detection networks. When more than one person speaks simultaneously, their combined shot is selected. The algorithm continues with its current selection until a change of speaker occurs. A minimum shot duration (l) is enforced to avoid rapid shot transitions. If a silence lasting L seconds is detected, we display a wide shot during that particular segment of silence.

Speaker Guided Optimization Editing: GAZED [3] is an end-to-end system to automate the video editing process for staged performances. It outputs an edited video that adheres to common cinematic principles and is aesthetically pleasing to watch. In the regular Gazed approach, we use gaze potentials, which are then combined with other terms that model cinematic principles like avoiding jump cuts, rhythm (pace of shot transitioning), avoiding transient shots etc. But with speaker-guided optimization editing(SOE), we rely on active speaker information to build speaker potential (Equation 3.2) instead of the human gaze. Speaker potentials assign a higher cost to shots with an active speaker, which will enforce the *dynamic programming* optimization framework to select the shot with greater speaker potential by constraining other cinematic principles. The underlying algorithm poses shot selection as a discrete optimization problem, which examines the importance of each of the multiple shots generated

for every video frame while adhering to cinematic principles like avoiding cuts between overlapping shots (termed jump cuts), avoiding rapid shot transitions, maintaining a cutting rhythm, etc. Cinematic principles are modelled as penalty in the term $E_e(s_{t-1}, s_t)$ where s_t represents shot s at time frame t. This penalty term is the sum of three different costs, namely shot transition cost, shot overlap cost, and cutting rhythm cost, as described in Section 2.1.6. The final solution is obtained via a search for the optimal path through an editing graph.

In SOE setting, the shot selection will ensure to follow cinematic principles and provide an aesthetically pleasing experience to watch, which isn't constrained in the SGE approach. As there could be incorrect active speaker predictions from active speaker detection networks, cinematic-motivated penalty terms do also work as an error recovery mechanism (to not make bad mistakes) in some scenarios. Unlike the greedy approach, the optimization-based method cannot run in a real-time setting as it needs to construct a complete cost matrix to get a minimal cost path. This approach has a higher memory footprint compared to the greedy approach as it needs to build an editing graph for optimization framework and backtrack for the optimal cost path. For a video with f frames and n actors, it will have a space complexity of $O(2^n - 1 * f)$. We also propose that our SOE framework with cinematic principle penalizing terms is open to extension for other useful information such as action etc, but not limited to human gaze or active speaker.

Formally, given a sequence of frames $t = [1..T]$, the set of generated shots (rushes) $S_t = \{s_i^t\}_{i=1}^{2^n-1}$ and the active speaker information a_t corresponding to active speaker a at time frame t, our algorithm selects a sequence of shots $\epsilon = \{r_t\},\ r_t \in S_t$ for each frame t minimising the objective function 3.3

$$E(\epsilon) = \sum_{t=1}^{T} -\ln S(r_t) + \sum_{t=2}^{T} E_e(r_{t-1}, r_t) \tag{3.3}$$

where $S(r_t)$ that represents speaker potential for each shot and $E_e(r_{t-1}, r_t)$ represents cost for transitioning from one shot to another.

We solve equation 3.3 using dynamic programming. The algorithm outputs a sequence of shots r_t (where r is the selected shot at time frame t) from the set of shots generated over time $\{S_t\}$. We build a cost matrix $C(r_t, t)$ where $r_t \in s_t^i$ and $t = [1..T]$, each cell is computed with recurrence relation 3.4.

$$C(r_t, t) = \begin{cases} -\ln S(r_t) & t = 1 \\ min_k[C(r_k, t-1) - \ln S(r_t) + E_e(r_k, r_t)] & otherwise \end{cases} \tag{3.4}$$

For each cell in the matrix, we compute and store the minimum cost to reach it. Once the matrix is built, we then perform backtracking to retrieve the sequence of optimal shots.

3.4 Evaluation metrics

Frame-level speaker accuracy: for a video is calculated by dividing the number of frames with correct speaker prediction by the total number of frames with voice activity.

Frame-level Editing accuracy: is calculated by dividing the number of frames that have atleast one correct match with ground truth in terms of the subjects shown in the video edit by the total number of frames in the video.

3.5 Experiments

We perform all our experiments on the timeline edit videos from OSD. Our experiments are of two parts. We first predict the speakers at frame level using TalkNet, pyannote and KNN algorithm on the dataset videos and evaluate using frame level speaker accuracy. In the second part, we perform video editing using the speaker predictions from all three models from the first part and ground truth speaker data. We evaluate the three ASD approaches in the context of video editing using frame level editing accuracy.

3.5.1 Speaker Detection Experiments

We experiment with an Audio Visual ASD model TalkNet. TalkNet is pretrained on TalkSet data pretrained on AVA-Active Speaker dataset. We predict active speakers at frame level using this model. These models essentially predict the bounding box of the active speaker faces. To get the person-id for the predicted bounding box, we use faces of the actors and tracking (ByteTrack) information to associate the predicted bounding box with person-id.

For experiments related to Audio-based speaker diarization(pyannote) we rely on the open-source python library for speaker diarization called pyannote-audio. We used the official pre-trained (as described in section 4.1.2) Speaker Diarization pipeline from *pyannote* available on *Hugging-Face*.

We take three 10 second audio samples of each of the 5 speakers in the OSD for KNN algorithm. For a given video, we take sliding window approach with window length of 0.8 second and a stride of 0.4 second. For each segment, we predict the speaker by KNN search. We take top-3 nearest neighbours based on the cosine similarity score and use majority voting strategy to get the speaker for the segment. We experimented with different k values and found optimal performance for $k = 3$.

3.5.2 Video Editing Experiments

We evaluate our audio-based speaker detection approach in the task of video editing. For this, we take two existing editing algorithms namely speaker-based greedy editing(SGE) and speaker-guided optimization editing(SOE).

In SGE, we use minimum shot duration(l) of 1.5 second and silence duration(L) of 2 second.

In SOE, we use speaker potential function $S(s_t^x)$ [3.2] in optimization framework [3.3]. We constrain our optimization framework with a minimum shot duration (l) of 1.5 second. In the scenarios where there is silence or no clear speaker, similar to SGE approach speaker potential ranks master shot a higher cost and a lower cost for the other shots.

	Duration(sec)	Frame Level Accuracy					
		TalkNet(ASD)				Pyannote	KNN
		cam1	cam2	cam3	cam4	All cameras	
Full video	1768.64	54.51%	52.10%	52.76%	50%	73.71%	83.5%
Clip #1	192.76	58.29%	57.38%	51.97%	53.13%	84.98%	89.41%
Clip # 2	80.52	56.83%	56.20%	54.44%	54.32%	89.87%	91.15%
Clip #3	153.92	55.38%	53.12%	54.41%	53.12%	33.10%	88.73%
Clip #4	106.84	53.28%	49.47%	53.80%	46.63%	90.12%	86.00%
Clip #5	140.72	49.41%	47.46%	52.62%	51.51%	13.95%	86.17%
Clip #6	23.64	49.67%	47.66%	52.39%	51.02%	52.70%	86.28%
Clip #7	85.2	47.76%	46.09%	50.63%	52.91%	17.31%	85.54%
Clip #8	6.8	47.70%	46.06%	50.47%	52.65%	40.35%	85.63%
Clip #9	103.04	47.48%	46.07%	49.44%	51.75%	49.82	83.97%

Table 3.1: Frame Level Accuracy for the Full Video and Specific Segments.

		Editing Accuracy with TalkNet predictions		Editing Accuracy With GT Speaker Annotations		Editing Accuracy with KNN Speaker Predictions	
	Duration(sec)	Speaker Greedy	Speaker Optimization	Speaker Greedy	Speaker Optimization	Speaker Greedy	Speaker Optimization
		Cam 4	Cam 4	All Cameras		All Cameras	
Clip #1	192.76	46.99%	50.33%	71.63%	79.85%	66.33%	77.94%
Clip # 2	80.52	50.05%	58.06%	66.54%	76.49%	61.22%	76.39%
Clip #3	153.92	61.23%	66.76%	82.77%	88.00%	71.88%	78.79%
Clip #4	106.84	47.06%	61.40%	80.50%	85.37%	65.64%	71.93%
Clip #5	140.72	45.92%	50.45%	67.89%	74.68%	55.38%	72.40%
Clip #6	23.64	60.67%	66.88%	88.40%	92.74%	75.96%	80.57%
Clip #7	85.2	47.77%	52.75%	71.45%	74.21%	58.76%	66.71%
Clip #8	6.8	51.71%	58.91%	66.00%	68.42%	64.88%	68.42%
Clip #9	103.04	42.17%	52.56%	64.33%	65.75%	58.95%	60.48%

Table 3.2: Editing Frame level accuracies with ground truth and predicted speaker annotations using various methods.

3.6 Results and Discussions

We conducted experiments using three different methods to detect speakers and report metrics for each of these methods in Table 3.1. We assessed the accuracy of TalkNet, Pyannote, and KNN approaches for the full video and 9 segments of the video. We chose these 9 segments based on varying visual and acoustic conditions. Specifically, we selected segments that contained non-speech sounds such as buzzers, bell rings, and clapping, as well as instances of overlapping speech, and instances where all actors' speech was covered.

Based on the data presented in Table 3.1, it can be concluded that the KNN method outperforms both Pyannote and TalkNet. One reason for this could be that KNN uses audio samples of the speakers as extra input for predictions, while Pyannote doesn't use any extra data. TalkNet had the lowest performance, likely due to the complex visual conditions present in the videos such as face occlusions and low resolution. The performance of TalkNet varied depending on the camera views, indicating that its performance is influenced by visual features that change with different camera angles.

We conducted video editing experiments on the OSD dataset using speaker information obtained from ground truth annotations as well as speaker predictions obtained from the KNN and TalkNet methods. We use videos from one camera view for these experiments. Table 3.2 presents the frame-level

editing accuracy results for the greedy and optimization approaches across 9 segments. The data in Table 3.2 suggests that the input speaker information plays a significant role in video editing and that the speaker is the most important factor in the video editing decision-making process. According to the metrics, approximately 70-80% of the frames in an edited video contain the speaker. Our results indicate that the speaker optimization approach outperforms the greedy approach. Furthermore, there is a visible correlation between speaker accuracy and editing accuracy.

3.7 Summary

This study evaluates various ASD methods and presents a simple audio-based approach that can outperform existing methods in video settings, such as those found in the OSD dataset. The KNN approach outperforms both diarization and audio-visual ASD methods. This study also emphasizes the importance of speakers in the video editing process and assesses the effectiveness of different ASD models in this task. Our experiments demonstrate that speaker information significantly influences video editing output. Our proposed approach requires audio samples and actors' face images as additional inputs for predicting and tracking the active speaker. We believe that audio samples and face images of actors are often available before the editing process. A potential future direction would be to investigate other factors that affect the video editing task with the goal of achieving fully automated video editing and developing robust ASD models that can perform well in various video settings, thereby improving the task of video editing.

Chapter 4

Minimalistic Video Saliency Prediction via Efficient Decoder & Spatio Temporal Action Cues

4.1 Related Works

4.1.1 Visual Saliency Prediction

Video saliency prediction has historically been significant to the computer vision community [50]. Driscoll *et al.* [51] were among the first to discuss using neural networks for saliency map prediction in humanoid robots. SP models thrived with the surge of deep learning over the last decade. Bak *et al.* [52] proposed two-stream networks that utilized two convolutional backbones for leveraging RGB images and optical flow maps as input, respectively, to extract spatio-temporal features and fuse their outputs for saliency prediction. Wu *et al.* [53], and Zhang and Chen [54] additionally examined the two-stream structure and analyzed fusion methods to improve performance.

Since optical flow networks simply consider adjacent frames for modelling the temporal relations in the video, LSTM-based methods were employed to model long-term temporal dependencies efficiently. Gorji and Clark [55] make use of a multi-stream Convolutional Long Short-Term Memory network (ConvLSTM) that incorporates static saliency and learns long-term temporal dependencies to estimate SP. Wang *et al.* [12] propose ACLNet to extend CNN-LSTM architecture using a supervised attention mechanism. Lai *et al.* [56] presents STRA-Net, a spatio-temporal residual network based on ConvGRU, to model the attention transition across frames. SalEMA [57] utilizes a simple exponential moving average for feature fusion in the temporal domain. SalSAC [58] first extracts multi-level features and then uses a random shuffling mechanism for the multi-level attentions calculated from the multi-level features followed by a correlation-based ConvLSTM.

3D Convolution-based architectures gained a lot of traction due to their ability to simultaneously model both spatial and temporal information. These methods typically employ action classification networks as their backbone. TASED-Net [59] is a 3D convolutional encoder-decoder network that makes use of S3D [60] pre-trained on the Kinetics dataset [61] as the video encoder. Bellitto *et al.* [62] propose HD2S, which generates multiple intermediate saliency maps by using features extracted at different abstraction levels and fuses them to predict the saliency map. Jain *et al.* [63] present ViNet, a

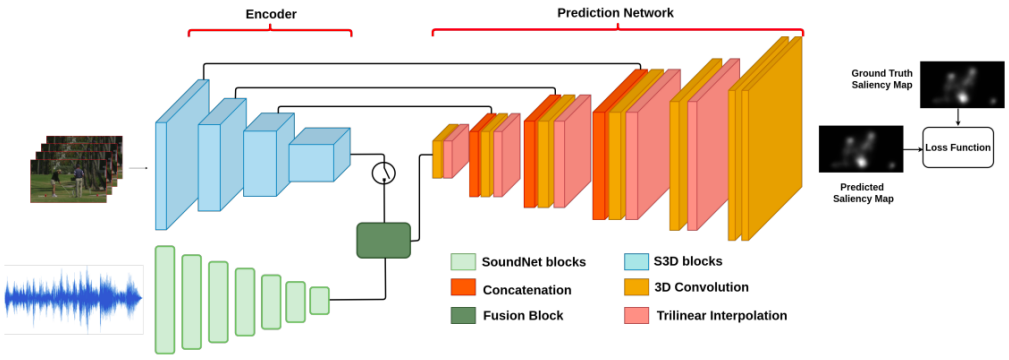


Figure 4.1: AViNet Architecture overview. Removing the audio branch, the resulting architecture is ViNet. Image reproduced from [63]

fully convolutional encoder-decoder architecture. It utilizes hierarchical features from the encoder and passes them as skip connections to the decoder in a UNet [64] like fashion to output the saliency map. We build upon this model due to its simplistic architecture, ease of training and real-time inference, augmenting the design with an efficient and lightweight decoder. Figure 4.1 illustrates the ViNet architecture. STSANet [65] utilizes multiple spatio-temporal self-attention modules at different levels of 3D convolutional backbone for capturing long-range relations between spatio-temporal features; however, its large model size renders it impractical for real-world applications.

Recent efforts utilize transformers for SP. Wang *et al.* [66] propose a spatio-temporal self-attention module for modelling the global correlation between pixels and human visual attention in both time and space. Ma *et al.* [67] extend video SP to video saliency forecasting (VSF), predicting attention regions of consecutive future frames. TMFI-Net [68] employs Video Swin Transformer [69] as the video encoder and a hierarchical decoder to predict saliency maps. THTD-Net [70] makes use of Video Swin Transformer [69] as well, gradually reducing the temporal dimension in the decoder layers to model long-range temporal dependencies.

The field of action recognition has evolved significantly, shifting from action classification [60] to STAL [71, 72]. While existing saliency models leverage backbones pre-trained on action classification tasks [60, 69], they have yet to exploit recent STAL research advancements fully. Our work addresses this shortcoming and bridges the gap between the two fields.

4.1.2 Audio-Visual Saliency Prediction

Recent works have also explored the amalgamation of audio and visual modalities for saliency prediction. Tsiami *et al.*introduces STAViS [73], which combines spatio-temporal visual and auditory features using linear weighting. They utilize 3D ResNet-50 [74] as the visual backbone and SoundNet [75] as the audio backbone. Chang and Zhu propose TSFP-Net [76], which constructs a temporal-spatial feature pyramid similar to FPN [77] and uses a hierarchical 3D convolutional decoder to predict the saliency maps. Audio features and visual features are fused and integrated into the original network in

the form of attention. CASP-Net [78] uses a two-stream encoder to associate video frames with their sound sources and a saliency decoder to combine multi-scale audio-visual features from previous decoder blocks. However, concerns persist regarding the practical utilization of audio in these models, as they remain agnostic to input audio during inference [63]. In this work, we omit audio and concentrate on maximizing the potential of the visual modality.

4.2 Datasets

4.2.1 Visual Datasets

We conduct experiments on three major visual-only saliency datasets, namely DHF1K, Hollywood-2 and UCF-Sports.

4.2.1.1 DHF1K

DHF1K [12] comprises 1,000 videos, with 600 designated for training and 100 for validation. Additionally, a test set of 300 videos is provided, though the ground truth for this set is not publicly available. The ground truth was generated with eye-tracking data collected from 17 observers. We use the DHF1K dataset for training and ablation analysis since it is the most diverse dataset.

4.2.1.2 Hollywood-2

Hollywood-2 [79] is the largest video SP dataset in terms of the number of videos, comprising 1707 videos. This dataset features short video sequences from 69 Hollywood movies, spanning 12 distinct human action categories. Following the standard protocol, we utilize the split of 823 training videos and 884 test videos for evaluation.

4.2.1.3 UCF-Sports

UCF-Sports [79] contains 150 videos of various sports-related actions. The videos cover 9 common sports action classes, such as diving, swinging and walking. This is also a task-specific dataset similar to Hollywood-2. The viewers have been biased towards task-aware observation by being instructed to "identify the actions occurring in the video sequence". We employ the standard split of 103 videos for training and 47 videos for testing.

4.2.2 Audio-Visual Datasets

We perform extensive experiments on six audio-visual saliency prediction datasets: AVAD, Coutrot1, Coutrot2, DIEM, ETMD and MVVA. Notably, our model utilizes only the visual components of these datasets.

4.2.2.1 AVAD

AVAD dataset [80] comprises 45 brief video clips, each lasting between 5-10 seconds, featuring a variety of audio-visual scenes such as dancing, guitar playing, birds singing, and more. Eye-tracking data from 16 participants have been recorded.

4.2.2.2 Coutrot1

Coutrot1 [81, 82] contains 60 clips depicting dynamic natural scenes, mainly consisting of four visual categories: one/several moving objects, landscapes, and faces.

4.2.2.3 Coutrot2

Coutrot2 [83] comprises 15 video clips featuring four individuals engaged in a meeting, along with eye-tracking data collected from 40 participants.

4.2.2.4 DIEM

DIEM [84] consists of 84 video clips based on advertisements, game trailers, movie trailers, documentaries, music videos, news clips and time-lapse footage. The eye gaze of 42 observers was collected via an Eyelink eye-tracker while watching the videos in random order and with the audio on. It contains 64 training videos and 20 testing videos.

4.2.2.5 ETMD

ETMD dataset [85] consists of 12 videos extracted from six diverse Hollywood movies.The eye gaze from 10 observers have been collected.

4.2.2.6 MVVA

MVVA [86] is a large-scale, extensive eye-tracking database of multi-face videos. It contains eye movement data of 34 subjects on 300 videos captured under audio-visual conditions. The dataset consists of videos with indoor or outdoor scenes classified into six categories: TV play/movie, interview, video conference, variety show, music, and group discussion.

4.3 Proposed Model Architectures

We propose an end-to-end trainable visual-only model called ViNet-A (fig. 4.2). It is a fully 3D-convolutional encoder-decoder architecture consisting of a SlowFast network [87] as the video encoder, a convolutional neck to reduce computational costs and an efficient, lightweight decoder for predicting the saliency map. We also propose a variation of the ViNet architecture [63], ViNet-S, which utilizes our efficient decoder, resulting in a small model while retaining the original ViNet's performance. Lastly,

we suggest an ensemble of the two proposed models, ViNet-E, which predicts the final saliency map by simply taking the pixel-wise average of the outputs from the proposed models. We elaborate on the suggested models in the following sections.

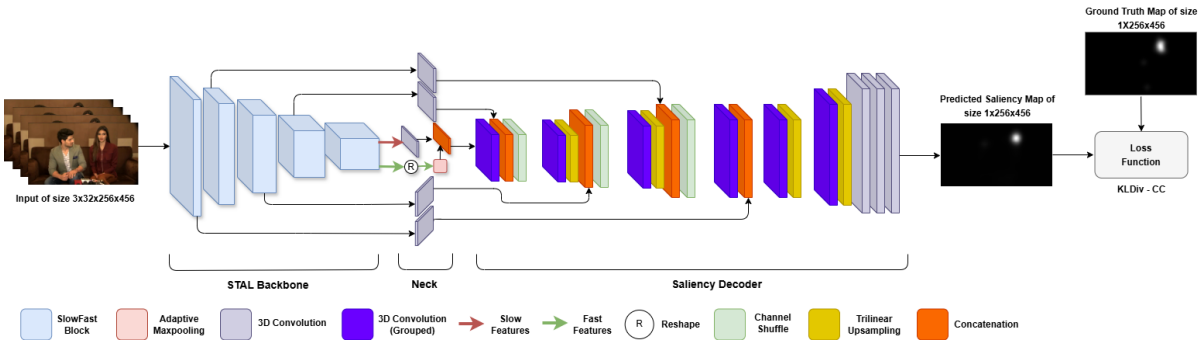


Figure 4.2: Our Model (ViNet-A) Architecture for SP (Best viewed in colour)

4.3.1 ViNet-A

4.3.1.1 Backbone

Our model architecture utilizes the SlowFast network [87] as the video encoder, which is pre-trained on the AVA actions dataset [88]. This dataset provides dense annotations of human atomic visual actions, including actions related to poses, such as sitting and walking, as well as interactions between people and objects or other people. The choice of this specific backbone is motivated by its ability to effectively map each spatial location to its corresponding action, enabling the model to capture the localized actions of each entity present in the scene

The SlowFast network consists of two parallel pathways: the Slow and Fast pathways. The Slow pathway operates at a low frame rate and captures spatial semantics, while the Fast pathway operates at a high frame rate and captures motion at fine temporal resolution. Both pathways are 3D convolutional networks with varying input sizes. Each pathway produces features at different stages of the network. To combine information from both pathways, lateral connections are made from the Fast pathway to the Slow pathway at these stages. These lateral connections are then passed to different layers of the saliency decoder as skip connections.

We input a video clip $x_{clip} \in \mathbb{R}^{3\times32\times256\times456}$ to the encoder which outputs slow features, $X_{slow} \in \mathbb{R}^{2048\times8\times16\times29}$, fast features, $X_{fast} \in \mathbb{R}^{256\times32\times16\times29}$ and hierarchical features X_1, X_2, X_3 and X_4.

4.3.1.2 Neck

The neck comprises of 1×1 convolutional blocks to reduce the number of channels, consequently decreasing the computational overhead. We reduce the number of channels in X_{slow} by half. X_{fast} is reshaped, doubling its channels while halving its temporal dimension, and then passed through an adaptive max pool to match the temporal dimension of X_{slow}. The two are concatenated channel-wise,

resulting in fused SlowFast features, $X_{slowfast} \in \mathbb{R}^{1536\times8\times16\times29}$. Hierarchical features X_1, X_2, X_3 and X_4 are also processed with 1×1 convolutional blocks to reduce their channels by half and improve computational efficiency. In short, the process can be described as follows:

$$X^{'}_{slow} = ReLU(Conv^{1\times1}(X_{slow})) \tag{4.1}$$

$$X^{'}_{fast} = AdaptiveMaxPool(Reshape(X_{fast})) \tag{4.2}$$

$$X_{slowfast} = [X^{'}_{slow}, X^{'}_{fast}] \tag{4.3}$$

$$X_i = ReLU(Conv^{1\times1}(X_i)), i \in 1, 2, 3, 4 \tag{4.4}$$

Here, ReLU represents the ReLU activation function $Conv^{1\times1}$ represents 1×1 convolutions for halving the channels and $[,]$ represents concatenation.

4.3.1.3 Saliency Decoder

The Saliency Decoder comprises six decoding blocks consisting of 3D convolutions, trilinear upsampling and channel shuffle [89] layers. We use 3D convolutions with filter groups [90] and channel shuffle to greatly reduce computation costs while maintaining accuracy. SlowFast features, $X_{slowfast}$, are input to the decoder and hierarchical features X_1, X_2, X_3, X_4 are passed as skip connections sequentially. All 3D convolutions except the last block utilize filter groups with 32, 16, 8, 8, 4 and 2 groups, respectively. Furthermore, channel shuffle layers are inserted after the first three grouped convolution layers. We experimented with different filter groups and channel shuffle layer configurations and found this optimal. We use the ReLU activation function after every convolutional layer except the last one, which employs the Sigmoid activation function to output the predicted saliency map.

4.3.2 ViNet-S & ViNet-E

ViNet-S employs the S3D [60] backbone as its video encoder and the efficient, lightweight decoder utilizing grouped convolutions and channel shuffle layers, similar to the ViNet-A saliency decoder described above.

ViNet-E is an ensemble of the proposed models ViNet-S and ViNet-A that generates a saliency map by performing a simple pixel-wise mean of the two predicted saliency maps. Since both models predict saliency maps of different sizes, we upsample the ViNet-S prediction to match that of ViNet-A before computing the pixel-wise mean.

$$Sal_E = Mean(Sal_A, Upsample(Sal_S)) \tag{4.5}$$

Here, Sal_S, Sal_A and Sal_E represent the output saliency maps of ViNet-S, ViNet-A and ViNet-E, respectively.

4.4 Evaluation Metrics

There are 5 evaluation metrics [91] broadly classified into 2 categories: Distribution-based and Location-based. These metrics are required to train and evaluate the saliency models.

4.4.1 Distribution Based Metrics

There are mainly three distribution based metrics KLDiv, CC, SIM as described below.

4.4.1.1 Kullback-Leibler divergence(KLDiv)

Kullback-Leibler divergence is an information-theoretic measure of the difference between two probability distributions.

$$KLDiv(P, Q) = \sum_i P_i \log(\epsilon + \frac{Q_i}{P_i + \epsilon}) \tag{4.6}$$

where P & Q are the predicted saliency map and ground truth respectively, ϵ is a regularization term.

Here, we interpret saliency maps as probability distribution and per- pixel value as probability. It is an asymmetric dissimilarity metric, where lower the value indicated better predictions. Since it uses log, it is highly sensitive to zero values, thus a sparse set of predictions is penalized very harshly, significantly worse than chance. Its value can vary from $-\infty$ to $+\infty$.

4.4.1.2 Correlation Coefficient(CC)

The Pearson's Correlation Coefficient (CC), also known as linear correlation coefficient, is a statistical method used generally in the sciences for measuring how correlated or dependent two variables are.

$$CC(P, Q) = \frac{\sigma(P, Q)}{\sigma(P, P) \times \sigma(Q, Q)} \tag{4.7}$$

where $\sigma(P, Q)$ represents covariance between P and Q

It is a symmetric metric which penalizes false positives and negatives equally and is invariant to linear transformations. The pixels where both predicted and ground truth saliency maps have similar values gives high CC value. CC value can vary from zero to one.

4.4.1.3 Similarity(SIM)

The Similarity metric (SIM), also referred to as histogram intersection, measures the similarity between two distributions, viewed as histograms. After normalising the input maps, SIM is computed as the sum of the minimum values at each pixel.

$$SIM(P,Q) = \sum_i min(p_i, q_i) \tag{4.8}$$

It was introduced as a metric for color-based and content-based image matching. Now it is used in saliency tasks for a simple comparison between saliency maps. A similarity score of one indicates that the maps are exactly same and zero indicates completely dissimilar.

4.4.2 Location Based Metrics

There are mainly two location-based metrics AUC-J and NSS considered for this study, as described below.

4.4.2.1 Area Under the ROC Curve(AUC-J)

The Area Under the ROC Curve (AUC-J) is the most widely used metric for evaluating saliency maps. The saliency map is treated as a binary classifier of fixations at various threshold values (level sets), and a ROC curve is swept out by measuring the true and false positive rates under each binary classifier. Its value can vary from zero to one and the higher the value better the predictions.

4.4.2.2 Normalized Scanpath Saliency (NSS)

Normalized Scanpath Saliency (NSS) aims to quantify the saliency map values at the fixated locations and to normalize it with the predicted map variance.

$$NSS(P,Q^B) = \frac{1}{N} \sum_i \overline{P}_i \times {Q_i}^B \tag{4.9}$$

where $N = \sum_i {Q_i}^B$ and $\overline{P} = \frac{P - \mu(P)}{\sigma(P)}$

It was introduced as a simple correspondence measure between saliency maps and ground-truth fixations. It is sensitive to false positives, relative differences in saliency across the image, and general monotonic transformations but since the maps are normalized, it is invariant to linear transformations. High valued predicted value at fixated locations gives higher NSS score. A zero value score indicates chance, positive score indicates correspondence between maps above chance, and negative score indicates anti-correspondence.

4.5 Experiments

4.5.1 Training

Following [63], we input a clip of 32 consecutive frames to the ViNet-S model and use the ground truth saliency map of the 32nd frame for supervision and prediction. For the ViNet-A model, the input

Table 4.1: Results on DHF1K validation set, UCF-Sports, and Hollywood2 test sets. Best results highlighted in red and second best in blue.

METHOD	DHF1K (Validation Set)				UCF-Sports				Hollywood2			
	CC↑	NSS↑	AUC-J↑	SIM↑	CC↑	NSS↑	AUC-J↑	SIM↑	CC↑	NSS↑	AUC-J↑	SIM↑
ACLNet [12]	0.434	2.35	0.890	0.315	0.510	2.56	0.897	0.406	0.623	3.08	0.913	0.542
TASED-Net [59]	0.481	2.706	0.894	0.362	0.582	2.920	0.899	0.469	0.646	3.302	0.918	0.507
UNISAL [94]	0.490	2.77	0.901	0.390	0.644	3.38	0.918	0.523	0.673	3.90	0.934	0.542
ViNet [63]	0.521	2.957	0.919	0.388	0.673	3.620	0.924	0.522	0.693	3.730	0.930	0.550
TSFP-Net [76]	0.529	3.009	0.919	0.397	0.685	3.698	0.923	0.561	0.711	3.910	0.936	0.571
STSANet [65]	0.539	3.082	0.920	0.411	0.721	3.927	0.936	0.560	0.705	3.908	0.938	0.579
TMFI-Net [68]	0.554	3.201	0.924	0.428	0.707	3.863	0.936	0.565	0.739	4.095	0.940	0.607
THTD-Net [70]	0.553	3.188	0.924	0.425	0.711	3.840	0.933	0.565	0.726	3.965	0.939	0.585
DiffSal [95]	0.533	3.066	0.918	0.405	0.685	3.483	0.928	0.543	0.765	3.955	0.951	0.610
ViNet-S	0.529	3.008	0.919	0.399	0.673	3.652	0.930	0.530	0.728	3.941	0.941	0.582
ViNet-A	0.525	3.019	0.916	0.399	0.734	4.108	0.940	0.586	0.756	4.119	0.945	0.604
ViNet-E	0.549	3.134	0.922	0.409	0.744	4.156	0.941	0.587	0.766	4.168	0.947	0.609

consists of 32 frames sampled from a window of 64 consecutive frames by selecting every alternate frame. We use the ground truth saliency map of the 33$^{\text{rd}}$ frame for supervision and prediction, akin to action label predictions in STAL models [71]. Both models are trained using the Adam optimizer with a learning rate of 10^{-4} and batch size of 8 for ViNet-S and 6 for ViNet-A.

For evaluating our model on DHF1K, we use the validation set due to unavailable annotations for the test set, as in prior efforts [67, 92]. We use the standard train and test sets provided for training on datasets Hollywood-2, UCF-Sports and DIEM. For Coutrot1, Coutrot2, AVAD, and ETMD, we perform 3-fold cross-validation and report average metrics across the splits. For MVVA, we follow [93] and perform training on a random split.

4.5.2 Loss Function

We utilize a combination of the evaluation metrics mentioned above, a standard technique in saliency tasks [91]. After experimenting with various combinations, we found that for most datasets, the best results were achieved using the following loss function:

$$Loss = KLDiv(P, Q) - CC(P, Q) \tag{4.10}$$

4.6 Results and Discussions

We evaluate the proposed models by comparing them against thirteen different methods from previous research. These include four 3D convolution-based approaches: ViNet [63], TASED-Net [59], STAVIS [73], and TSFP-Net [76]; two methods utilizing recurrent networks: ACLNet [12] and UNISAL [94]; four models employing transformers: STSANet [65], THTD-Net [70], CASP-Net [78], TMFI-Net [68]; one diffusion-based model: DiffSal [95] and a couple of multi-branch network methods: VAM-Net [93]

Table 4.2: Quantitative comparison of model sizes & parameters

Model	Size (MB)	# Params (Million)
ACLNet [12]	250	65.54
TASED-Net [59]	82	21.5
STAViS [73]	79.19	20.76
UNISAL [94]	15.5	4.06
ViNet [63]	124	32.5
TSFP-Net [76]	58.4	15.3
STSA-Net [65]	643	168.56
TMFI-Net [68]	234	61.34
THTD-Net [70]	220	57.67
CASP-Net [78]	196.91	51.62
DiffSal [95]	269	70.54
ViNet-S	36.24	9.5
ViNet-A	147.6	38.69
ViNet-E	183.84	48.19

Table 4.3: Quantitative comparison results on the AVAD, Coutrot1, Coutrot2 and ETMD test sets.

METHOD	Coutrot1				Coutrot2				ETMD				AVAD			
	CC↑	NSS↑	AUC-J↑	SIM↑	CC↑	NSS↑	AUC-J↑	SIM↑	CC↑	NSS↑	AUC-J↑	SIM↑	CC↑	NSS↑	AUC-J↑	SIM↑
ACLNet [12]	0.425	1.92	0.85	0.361	0.448	3.16	0.926	0.322	0.477	2.36	0.915	0.329	0.580	3.17	0.905	0.446
TASED-Net [59]	0.479	2.18	0.867	0.388	0.437	3.17	0.921	0.314	0.509	2.63	0.916	0.366	0.601	3.16	0.914	0.439
STAViS [73]	0.458	1.99	0.861	0.384	0.652	4.19	0.940	0.447	0.560	2.84	0.929	0.412	0.604	3.07	0.915	0.443
ViNet [63]	0.551	2.68	0.886	0.423	0.724	5.61	0.95	0.466	0.569	3.06	0.928	0.409	0.694	3.82	0.928	0.504
TSFP-Net [76]	0.57	2.75	0.894	0.451	0.718	5.30	0.957	0.516	0.576	3.09	0.932	0.433	0.688	3.79	0.932	0.530
CASP-Net [78]	0.561	2.65	0.889	0.456	0.788	6.34	0.963	0.585	0.620	3.34	0.940	0.478	0.691	3.81	0.933	0.528
ViNet-S	0.574	2.876	0.898	0.449	0.754	6.103	0.958	0.547	0.599	3.268	0.941	0.458	0.712	4.090	0.935	0.540
ViNet-A	0.600	3.033	0.900	0.459	0.862	6.8	0.961	0.638	0.623	3.379	0.941	0.458	0.709	4.094	0.933	0.534
ViNet-E	0.614	3.085	0.905	0.465	0.854	6.762	0.962	0.628	0.632	3.437	0.943	0.468	0.729	4.167	0.938	0.547

and VASM [86]. Six of these models (STAVIS, CASP-Net, TSFP-Net, VAM-Net, DiffSal and VASM) additionally employ audio information in their approach. We report results directly from the corresponding papers when available. If the code is publicly available and executable, we compute their results on other datasets.

Visual Only Datasets: Table 4.1 presents results on the visual-only datasets. The model sizes and the number of parameters of the studied models are presented in Table 4.2. We observe that ViNet-E achieves the best performance on UCF-Sports and Hollywood2 datasets [96], while achieving competent results on the DHF1K dataset [12]. Interestingly, ViNet-A also outperforms the previous methods on the UCF-Sports and Hollywood2 datasets. Its strong performance on these two human-centric datasets clearly demonstrates the advantages of using an STAL backbone over an action classification backbone. Notably, all three proposed models, including ViNet-S, consistently surpass the base ViNet model.

The ViNet-S model recovers most of the underlying performance in all the cases while using only a tiny fraction of the parameters. For instance, on the Hollywood2 dataset, the largest SP dataset with 884 videos in the test set, ViNet-S recovers over 98.5% performance on the CC metric compared to the transformer-based SOTA TMFI-Net, while bringing over six-fold reduction in terms of number of parameters (Table 4.2). Interestingly, ViNet-S outperforms TMFI-Net on the AUC-J metric. UNISAL is

Table 4.4: Quantitative comparison results on the DIEM and MVVA test sets.

METHOD	DIEM			
	CC↑	NSS↑	AUC-J↑	SIM↑
ACLNet [12]	0.522	2.02	0.869	0.427
TASED-Net [59]	0.557	2.16	0.881	0.461
STAViS [73]	0.579	2.26	0.883	0.482
ViNet [63]	0.626	2.47	0.898	0.483
TSFP-Net [76]	0.651	2.62	0.906	0.527
CASP-Net [78]	0.655	2.61	0.906	0.543
ViNet-S	0.673	2.732	0.908	0.533
ViNet-A	0.675	2.742	0.908	0.547
ViNet-E	0.701	2.840	0.913	0.566

METHOD	MVVA			
	CC↑	NSS↑	AUC-J↑	KLDIV↓
VASM [86]	0.722	3.976	0.905	0.823
VAM-Net [93]	0.741	4.002	0.912	0.783
TASED-Net [59]	0.653	3.319	0.905	0.970
STAViS [73]	0.77	3.060	0.91	0.80
ViNet [63]	0.81	4.470	0.93	0.75
ViNet-S	0.802	4.617	0.933	0.715
ViNet-A	0.825	4.823	0.934	0.678
ViNet-E	0.828	4.816	0.936	0.663

the only model lighter than the ViNet-S model. However, it consistently underperforms in comparison, possibly due to its recurrent architecture.

Audio Visual Datasets: Table 4.3 and Table 4.4 present results on the audio-visual datasets. The proposed ViNet-S, ViNet-A, and ViNet-E consistently outperform prior models across all six datasets, consistently ranking among the top two models. The videos in the Coutrot2 and MVVA datasets emphasize multi-person interactions. Notably, ViNet-A achieves significant improvements on both datasets, maintaining a consistent performance trend with the other human-centric datasets. On MVVA (the largest audio-visual saliency dataset), while only using the visual modality ViNet-A brings over 20% gains on NSS metric over the complex multi-branch VAM-Net, which uses an explicit combination of motion, texture, face and audio features.

On four out of the six audio-visual datasets, i.e. DIEM, AVAD, Coutrot1, and MVVA, the smaller ViNet-S surpasses all the previous methods. The consistent performance improvements of the ViNet-E model validate the effectiveness of the proposed ensemble strategy, establishing a new SOTA in most datasets. Another notable observation is that incorporating audio information does not appear to provide a significant advantage for the task of SP. Consistent with prior studies [63, 76, 97], we found that several audio-visual models [73, 95], in reality, are not exploiting the audio information. At inference, the models appear agnostic to the audio information, i.e., the results remain the same irrespective of sending the random audio or zero audio. This represents a significant scientific flaw that requires further investigation in future research, and comparisons with their results should be approached with caution. Although ViNet-E outperforms their audio-visual version on several datasets, we limit our comparisons only to their visual only model.

Qualitative comparisons: Figure 4.3 shows the qualitative performance of our ViNet-S, ViNet-A and ViNet-E models on video sequences from three different datasets: DHF1K, UCF-Sports and DIEM. We observe that STAL features efficiently capture the interaction between an actor/object with the context (surrounding) as evident in the strong performance of our model. ViNet-E is consistently closer to the ground truth in different settings than all other models, including STSANet.

Computational load: Table 4.2 compares the different models in terms of models size and number of parameters. The proposed decoder significantly reduces the number of parameters in ViNet-S compared to the original ViNet model. Aside from UNISAL, ViNet-S is the most efficient in terms of model

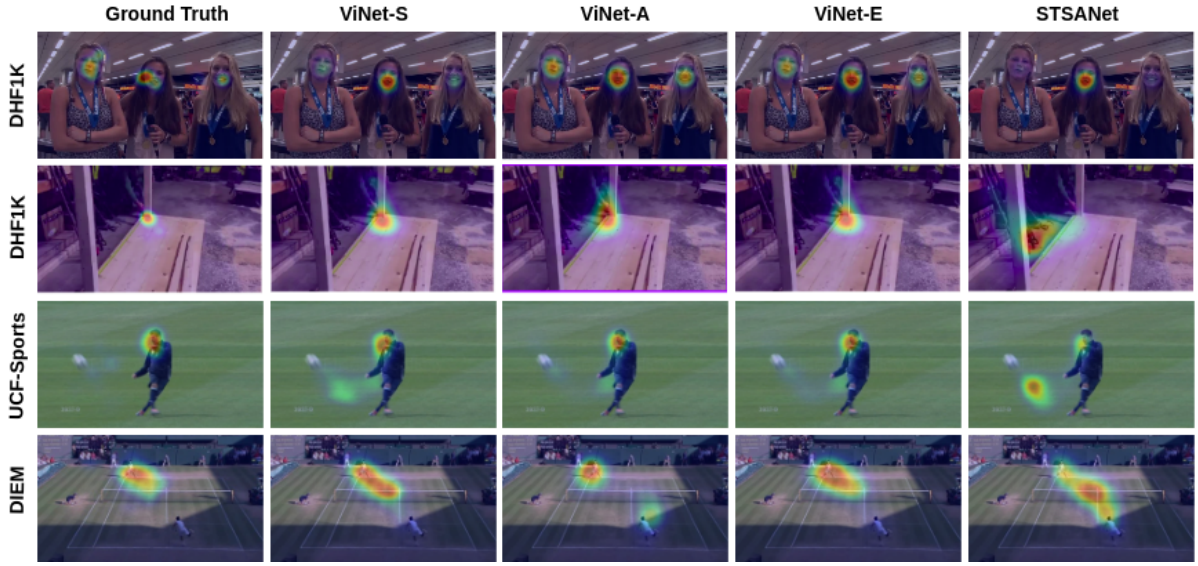


Figure 4.3: Qualitative results: Comparing Ground Truth with the predicted saliency maps of our models and STSANet on three different datasets - DHF1K, UCF-Sports and DIEM.

size and parameters among the compared models. We observe that switching from the S3D backbone in ViNet-S to the SlowFast backbone in ViNet-A leads to significant parameter gains. Notably, ViNet-A's decoder contains only 1.6 million parameters, while the SlowFast backbone accounts for the remaining 37 million. Lastly, the ViNet-E model remains smaller than state-of-the-art transformer-based models (e.g., TMFI-Net and THTD-Net) in both model size and parameter count.

The non-autoregressive design of the proposed ViNet models enables parallel processing, providing a significant advantage over autoregressive models such as UNISAL, which rely on frame-level recurrence. On an Nvidia RTX 4090 GPU, ViNet-S, ViNet-A, and ViNet-E models achieve runtimes of approximately 200fps, 120fps, and 90fps, respectively, in a real-time processing setup (with a batch size of one). With a batch size of eight, ViNet-S reaches an impressive 1070fps.

4.7 Summary

This work introduces two efficient models, ViNet-S and ViNet-A, characterized by their simple architectural design choices. ViNet-S is lightweight yet matches or surpasses most convolutional methods, while ViNet-A, which utilizes localized action features, consistently performs well on human-centric datasets with multiple subjects. ViNet-E, the ensemble model, leverages the complementary nature of action classification and detection to achieve state-of-the-art results on both visual and audio-visual datasets, even without audio cues. By using pixel-wise averaging, it enhances performance, suggesting new avenues for integrating global and localized action features. While this study focuses on model optimization through architecture, future efforts should explore model compression and knowledge distillation.

Chapter 5

EditIQ: Automated Cinematic Editing of Static Wide-Angle Videos via Dialogue Interpretation and Saliency Cues

We introduce *EditIQ*, an entirely automated system for multi-camera video production tailored for staged performances. Our approach utilizes footage from one or more stationary wide-angle cameras and builds upon prior work [5] to generate multiple virtual camera perspectives. The primary focus of our method is to streamline the camera selection process. Choosing the best camera angle requires a deep comprehension of the scene, ensuring essential aspects such as dialogue, the primary speaker, and the performers' actions and reactions are effectively captured. In particular, reaction shots play a crucial role in editing by conveying emotional context, enriching audience engagement, and enhancing scene interpretation. While human vision instinctively tracks significant elements within a scene, replicating this level of perception through automation remains a complex challenge.

EditIQ utilizes Large Language Models to analyze dialogue and identify crucial scene components. In the example depicted in Figure 5.1, a camera tracking only the speaker or audio source would remain fixed on the lead singer, overlooking the non-verbal responses of other musicians during their introductions. In contrast, an LLM discerns that when the lead singer announces, "J T Thomas on the keyboard," the visual focus should transition to the mentioned individual.

Large language models (LLMs) often struggle to infer scene actions that are not explicitly mentioned in dialogue. Additionally, multimodal LLMs generally operate at the frame level, which limits their ability to comprehend temporal dynamics in actions. To address this challenge, we incorporate a saliency prediction framework designed to emulate human gaze patterns, helping to pinpoint crucial areas within a scene. Specifically, we enhance a spatio-temporal action localization model [98] to facilitate video saliency prediction [12]. As illustrated in Figure 5.1, this visual saliency network effectively identifies significant actions, such as the keyboard player's gesture of acknowledgment or the lead singer's expressive hand movement directed toward the bass guitarist.

After the LLM and visual saliency pipelines identify key elements within the scene and their corresponding video shots, we define camera shot selection as a discrete optimization task. At each time frame, an optimal shot is chosen from the available footage for presentation to the viewer. The cost matrix for this selection incorporates three unary potential terms: speaker data, LLM-derived insights, and saliency-based outputs. Similar to [3], these potential terms are integrated with constraints that en-

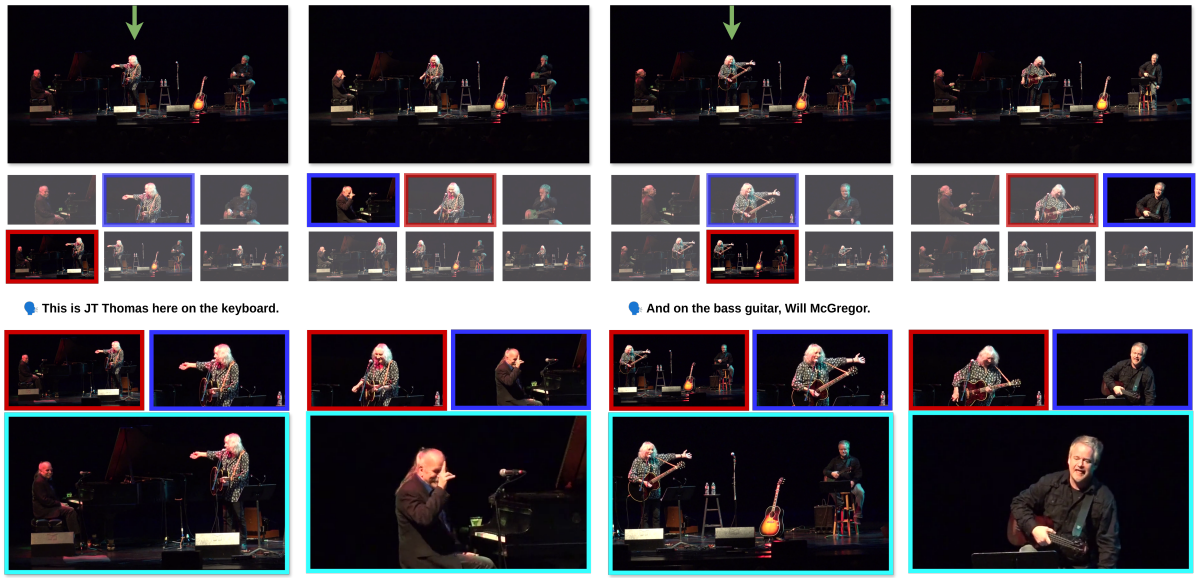


Figure 5.1: We introduce *EditIQ*, an automated video editing system that integrates dialogue comprehension through large language models (LLMs) with visual analysis via video saliency. The first row displays the raw video frames fed into the system, which then produces multiple rushes (shown in the next two rows). A green arrow marks the speaker in the original frames, with the corresponding transcript appearing below the third row. LLMs assess the narrative structure to determine shot selections, highlighted in red on the left side of each frame in the fourth row. Concurrently, saliency detection identifies key visual elements, generating alternative shot choices (depicted in blue on the right side of the fourth row). The fusion of linguistic and visual scene interpretation leads to the optimal video shots presented in the fifth row.

force cinematic editing principles, such as minimizing abrupt jump cuts, maintaining a steady transition rhythm, avoiding fleeting shots, and ensuring appropriate framing to keep actors properly positioned. The optimization problem is ultimately solved using dynamic programming.

To assess the effectiveness of *EditIQ*, we conducted a psychophysical experiment involving 20 participants. The study compared different edited versions of performance recordings sourced from the BBC Old School Dataset (BBC-OSD) [10], which features a quiz show setting. Our approach outperforms multiple alternative editing techniques, such as random shot selection, wide-angle framing, and edits based solely on speaker detection. The results indicate that, on BBC-OSD, the output of *EditIQ* closely aligns with professionally edited footage created by human experts.

5.1 Related Works

5.1.1 Editing through automated crops

Handling high-resolution footage, such as 4K or 8K, unlocks numerous creative opportunities in editing, particularly for zooming and tracking specific regions within a video. Dynamic cropping has proven highly effective in areas like video stabilization and retargeting. Automated video retargeting involves adapting content to a desired aspect ratio by intelligently selecting cropping regions over time. Prior efforts to tackle this challenge have utilized various techniques, including user-provided annotations [99], motion and saliency-based approaches [100, 101], and gaze-tracking data [102]. Grundmann *et al.* [103] introduced a method for generating stabilized videos by applying L1-optimal, constraint-based camera paths to eliminate unwanted motion.

Early research on automated production systems utilizing cropping primarily targeted lecture and presentation settings [104, 105]. These methods generally employed rule-based editing techniques in controlled environments, typically featuring a single speaker in front of a chalkboard or presentation screen. In the domain of sports, Schäfer *et al.* [106] developed a system that allows users to define customized cropping windows within an ultra-high-resolution video feed. Carr *et al.* [107] implemented virtual pan-tilt-zoom (PTZ) cropping over footage from a robotic camera tracking basketball players to refine game editing. Furthermore, virtual PTZ techniques have been leveraged for cinematic editing in panoramic and 360-degree video content [108–110].

Unlike conventional pan-and-scan editing techniques, our approach emulates a multicamera production setup by automating camera shot selection. Gandhi *et al.* [5] developed a technique for generating virtual PTZ cameras by dynamically adjusting a cropping window within wide-angle footage, mimicking camerawork through L1-norm-based optimization [103]. Expanding on this concept, our method incorporates an automated system for selecting between these simulated virtual cameras.

These approaches are closely linked to and enhanced by recent progress in computer vision and machine learning, including areas such as object detection [111], action recognition and localization [71, 87], human tracking [112, 113], pose estimation [114], video saliency forecasting [12, 115], and head orientation estimation [116].

5.1.2 Gaze Based Video Editing

Prior studies have leveraged both participant and viewer gaze to pinpoint prominent areas in videos, establishing eye gaze as a crucial component in video editing. Estimating gaze direction is commonly achieved through head orientation analysis or specialized eye-tracking equipment. For instance, Takemai et al. [117] introduced a video editing approach that utilized participants' gaze during indoor conversations, demonstrating that gaze-based editing more effectively preserves conversational dynamics than traditional speaker-based methods. Likewise, Daigo et al. [118] employed audience gaze to emphasize key moments in basketball footage, whereas Park et al. [119] and Arev et al. [120] determined salient regions by examining how multiple camera perspectives converge.

Recent studies [102, 121] have explored the use of eye gaze data in video retargeting, a process that adapts edited videos for various screen sizes, such as scaling content from a large theater display to a smaller mobile screen. This is generally achieved by shifting a cropping window within the original footage to retain important visual elements. However, these techniques primarily modify the horizontal (x) axis with minimal zoom adjustments based on gaze variation, often resulting in poorly framed shots and unnatural cropping. In contrast, GAZED [3] emulates a multi-camera production and editing system, selecting shots that align with cinematic principles. It enhances composition by adjusting both horizontal and vertical axes, as well as zoom levels, ensuring that key subjects and events remain the focal point of the scene.

5.1.3 Automated Camera Selection

The study of automated camera selection has been extensively explored in 3D settings, particularly in the context of pre-visualization (previz) and video game applications. Early investigations [122, 123] applied cinematic conventions and established shot composition techniques [124] to determine effective camera placements. Another approach frames camera selection as a discrete optimization challenge, employing dynamic programming methods [125–127] to refine shot choices. Merabti et al. [127] specifically focus on dialogue-heavy scenes, leveraging Hidden Markov Models (HMM) for decision-making. Additional elements, including pacing, seamless shot transitions, and continuity principles, are explored in [126]. While our research builds upon these foundations, stage performances introduce distinct constraints, such as fixed camera positions and the absence of detailed spatial information, character tracking, and event metadata typically available in 3D environments.

Extensive research has been conducted on camera selection in the realm of sports broadcasting [128–132]. Wang *et al.* [128] utilize Hidden Markov Models (HMMs) to address this challenge, while [129] focuses on optimizing coverage of key actions. Some approaches [130, 131] take a data-driven perspective, training regression models to assess the relevance of different camera angles in real time. Meanwhile, Pan *et al.* [132] propose an event-centric strategy, first detecting important events and then selecting the most visually engaging perspectives for them.

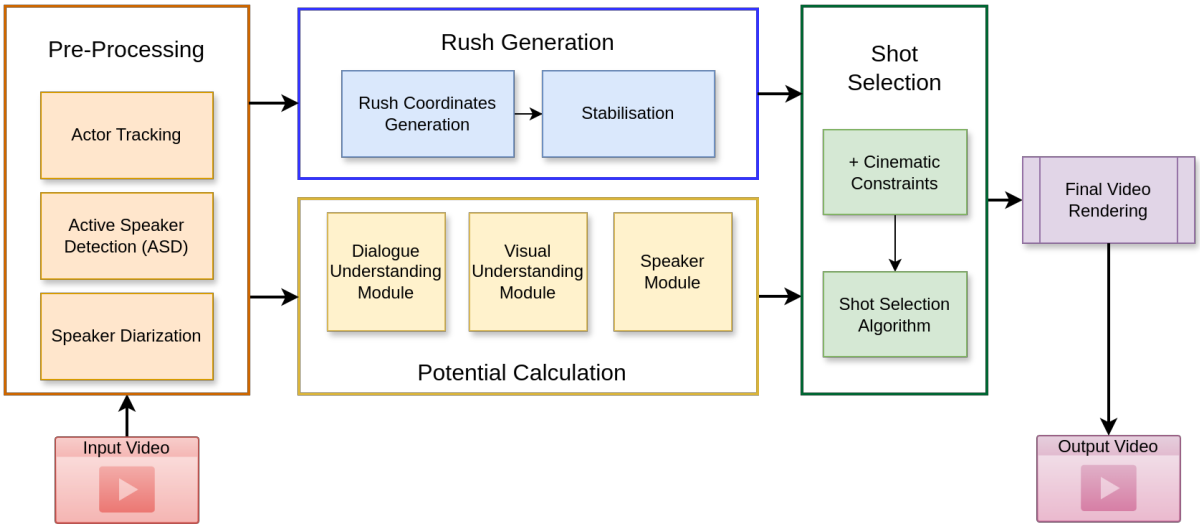


Figure 5.2: *EditIQ* Pipeline: This entirely automated system processes video inputs along with face crops and corresponding IDs, ultimately generating a fully edited video. The different stages of the workflow are illustrated in the figure, with each step building upon the results of the preceding one.

Arev *et al.* [120] present a technique for automatically editing footage from multiple social cameras. Their method employs a trellis graph structure to optimize an objective function that prioritizes capturing essential scene content while following cinematic rules, such as minimizing jump cuts. Content significance is determined by analyzing joint attention across different camera views [133]. In a different approach, Leake *et al.* [134] develop an idiom-based system for editing dialogue-centric scenes. Their framework processes multiple camera angles, alternative takes, and the film script to generate the most effective shot selection for each dialogue segment.

A major difference between our approach and previous methods is the absence of multiple manually controlled camera feeds or alternate takes. Instead, we work with a single wide-angle recording and generate virtual camera views for editing. The closest related work is GAZED [3], which leverages human gaze to detect key elements in a scene, assuming access to actor tracking data. Unlike GAZED, our method operates independently of gaze information and introduces a fully automated editing pipeline that does not depend on actor tracks or predefined cinematic rules. While human gaze serves as a strong indicator of scene relevance, our approach relies on machine learning models to navigate predictive uncertainties inherent in real-world automation.

5.2 EditIQ Overview

This study presents an all-encompassing video editing framework, named *EditIQ* , which streamlines the cinematic editing and production workflow. The system takes as input static, high-resolution footage of a scene and refines it into an engaging, professionally edited video that aligns with fundamental cinematic conventions. Its design is inspired by conventional video production workflows and is organized into four key stages:

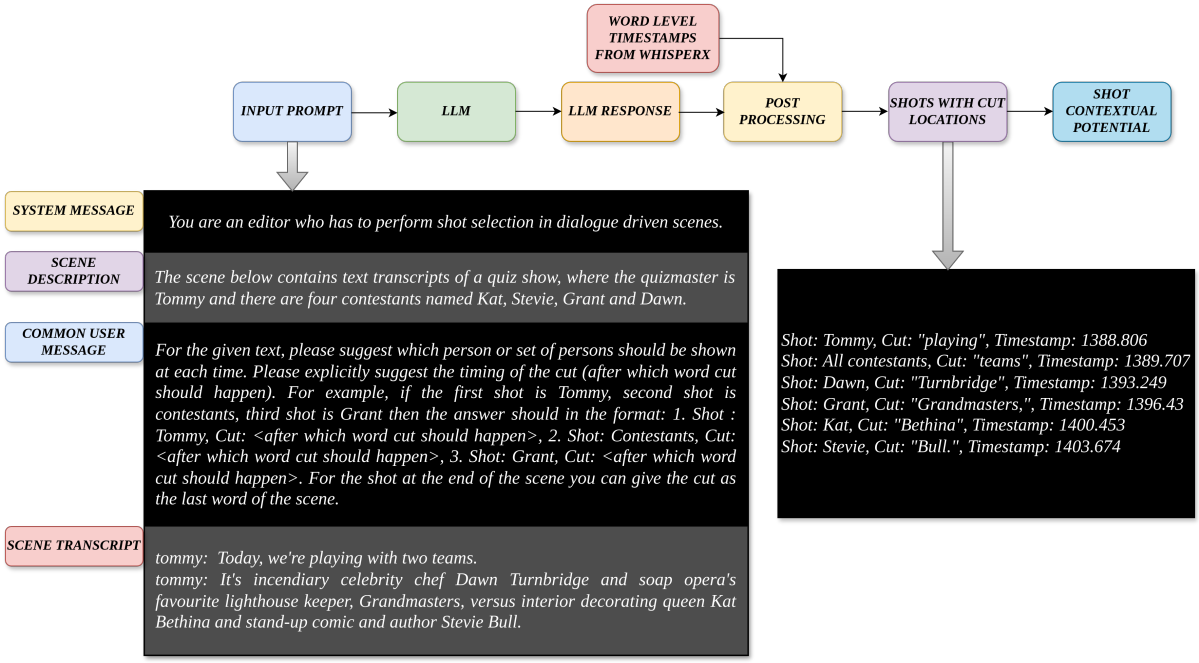


Figure 5.3: The dialogue understanding module extracts Contextual Potential from an LLM for various shots using the scene's transcript. In the diagram above, the post-processing step aligns the LLM output with word-level timestamps (obtained during pre-processing) to determine the cut points.

(i) *Pre-processing:* Preparing the input video for subsequent editing steps. (ii) *Rush Generation:* Producing a set of cinematically appropriate shots that focus on different actors or scene elements. (iii) *Potential Calculation:* Assessing the importance of each shot based on factors such as dialogue context and spatial prominence. (iv) *Shot Selection:* Choosing the most suitable shot at each moment to enhance the storytelling experience.

The different stages of the pipeline are illustrated in Figure 5.2. A detailed discussion of each stage will be provided in the subsequent sections. The pipeline takes the following as inputs:

(i) A high-resolution video recording obtained from fixed-position camera(s) that capture the entire scene. (ii) A concise scene summary along with cast details, which include actor names and a single reference image for each.

5.2.1 Pre-Processing

From this point forward, frames extracted from the original wide-angle input video will be referred to as "*master shots*". Using the master shot, multiple essential features are computed to facilitate the subsequent stages of the pipeline.

1. *Actor Tracks*: In this study, we employ the BoT-SORT [112] model to detect and track individuals. This model generates bounding box coordinates $[x, y, h, w]$ for each identified person in every frame while ensuring identity consistency across frames. Any inaccuracies in tracking were rectified before further processing. These tracked movements are crucial for both Shot Generation

and Video Editing. Notably, this step is not restricted to a specific algorithm, allowing flexibility in substituting other tracking methods if needed.

2. *Character Aware Subtitling*: We apply the WhisperX [135] model to the audio stream of our dataset in order to identify speech segments with word-level timestamps. The recognized words are then combined to form a comprehensive transcript for each video, which is later split into sentences using a sentence tokenizer. Following this, we adopt the approach from [136] to create character-specific subtitles, resulting in a complete dialogue transcription with precise speech timestamps and speaker identification. Specifically, we begin by selecting high-quality examples for each character using TalkNet [30], and then use these examples to categorize all speech segments by speaker identity [137].

5.2.2 Rush Generation

The second phase of the *EditIQ* pipeline involves the shot generation process. In this stage, a simulation technique [5] is used to automatically generate virtual PTZ cameras by adjusting multiple cropping windows to follow a specific actor or group of actors within the master shot. For individual subjects, we typically use medium shots (framing the actor from head to waist), while for scenes with two or more actors, we opt for full shots (showing the subject from head to toe).

For a master shot containing n actors, we create $2^n - 1$ virtual shots. This includes ${}^n\mathrm{C}_1$ individual shots (single actors), ${}^n\mathrm{C}_2$ two-shot compositions (pairs of actors), ${}^n\mathrm{C}_3$ three-shot compositions, and so on, all in a 16:9 aspect ratio. These virtual shots, along with the master shot, are collectively termed "rushes" and are used for subsequent selection and editing. We define S_t as the set of rushes at time t, represented as:

$$S_t = \{A \mid A \subseteq \{x_1, x_2, \ldots, x_n\} \text{ and } A \neq \emptyset\} \cup \{\text{Master Shot}\} \\ \text{where } x_i \text{ is the } i^{th} \text{ actor} \tag{5.1}$$

Tighter framings, such as Medium Shots (MS), focus on an actor's gestures and facial expressions, providing greater emphasis on detail. In contrast, wider framings like Full Shots (FS) encompass the entire actor or group of actors, highlighting their positioning and interaction within the scene. Figure 5.1 illustrates an example of the generated rushes featuring three actors. The rushes consist of the master shot, three individual shots, two two-actor shots, and a single three-actor shot.

Unlike previous approaches [5] that rely on upper-body detection, we employ a person pose estimation model, allowing for greater control over shot composition and a deeper analysis of the subject's posture and orientation. In this study, we utilize YOLOv8-Pose [111, 114] due to its high accuracy and real-time processing capabilities.

Building on the approach in [5], we begin by generating frame-wise shot predictions, which are then refined to produce well-balanced compositions that replicate the smooth motion of professional camera

Figure 5.4: **Saliency potential** of various single-order shots for two frames in a theater video (with potential values displayed alongside each actor's shot). The green arrow marks the speaker, if present.

operators. The virtual PTZ simulation is framed as an optimization problem, aiming to minimize a sum-of-squares term that quantifies the deviation from initial per-frame estimates while incorporating L_1-norm regularization to constrain both velocity and jerk [5, 103].

5.2.3 Dialogue Understanding Module — Contextual Potential

In video editing, dialogue plays a fundamental role in influencing the narrative, character development, and emotional expression. Editors need to grasp not just the direct meaning of the words but also the subtleties carried through factors like tone and emotion. By making strategic decisions about when to cut, how to structure the pacing of conversations, and which reactions to highlight, editors can elevate a basic dialogue scene into a compelling and emotionally charged moment. This, in turn, propels the story and strengthens the viewers' bond with the characters.

Within the Dialogue Comprehension Module, we employ advanced language models to analyze conversations and propose shot sequences. The model processes a scene's transcript alongside a brief instructional prompt to generate a visually structured shot sequence. To enhance contextual understanding, a short scene summary is also included. The transcript, which has undergone preprocessing, provides speaker details, enabling the model to determine who should be in focus at any given moment. Additionally, we request defined cut points, specifying the precise word after which transitions should happen. The generated shot recommendations and cut details are then refined using word-level timestamps to establish exact cut positions. These AI-driven shot suggestions contribute to assessing the contextual significance of each shot at any point in time.

Figure 5.3 demonstrates this workflow with an example from the BBC-OSD dataset, presenting the user prompt, an excerpt from a transcript, and the resulting shot sequence suggestions along with cut points determined through word-level timestamps. An essential guideline for contextual potential is that shots overlapping or encompassing the chosen shot should not be assigned a zero cost; instead, they should retain a minimal cost, acknowledging their partial contribution to the context considered by the language model in its decision-making process.

The contextual potential is determined based on the category of shot chosen by the model:

1. When a first-order shot is selected, it incurs a cost of λ_c. For higher-order shots (p-order shots that include the chosen actor), the cost is reduced to $\frac{\lambda_c}{p}$ since they still represent the context but with less precision compared to a close-up. Any other shots are assigned a cost of zero.

$$C(s_t^X) = \begin{cases} \lambda_c & s_t^X = s_t^{X_s} \\ \frac{\lambda_c}{p} & x_s \in \mathrm{X}, |X| > 1 \\ 0 & \text{for remaining 1-shots} \end{cases} \tag{5.2}$$

   Here, s_t^X represents a shot s occurring at time t that includes a group of actors X, while $s_t^{X_s}$ denotes the first-order shot chosen by the LLM, specifically featuring the actor x_s.

2. When a p-order shot ($p > 1$) is chosen, single-order shots featuring any actors present in the selected shot are assigned a cost of $\frac{\lambda_c}{2^{p-1}}$. According to Equation (5.4), the chosen shot itself carries a cost of λ_c, whereas all remaining shots are assigned a cost lower than λ_c.

$$C(s_t^X) = \begin{cases} \frac{\lambda_c}{2^{p-1}} & \mathrm{X} \in X_s, |X| = 1 \\ 0 & \text{for remaining 1-shots} \end{cases} \tag{5.3}$$

   where s_t^X refers to a shot s at time t that contains a set of actors X and X_s is the higher-order shot selection from LLM with actors $X_s = \{x_i \mid i = 1, 2, \ldots, p\}$

   For shots that include more actors than the selected shot, the contextual potential is determined following the approach outlined in [3]. For instance, consider a two-person shot s_t^X, where $X = \{x_1, x_2\}$ represents the presence of actors x_1 and x_2. The contextual potential for this multi-person shot is expressed in relation to the contextual potentials of the corresponding single-person shots, $C(s_t^{x_1})$ and $C(s_t^{x_2})$, as shown below:

$$C(s_t^{\{x_1,x_2\}}) = C(s_t^{x_1}) + C(s_t^{x_2}) - |C(s_t^{x_1}) - C(s_t^{x_2})| \tag{5.4}$$

   Similarly, contextual potentials of two 2-shots $C(s_t^{\{x_1,x_2\}})$ and $C(s_t^{\{x_2,x_3\}})$ can be used to compute the contextual potential of a 3-shot $C(s_t^{\{x_1,x_2,x_3\}})$, when the actors appear on screen in the order x_1, x_2, x_3 from left to right.

5.2.4 Visual Understanding Module — Saliency Potential

As mentioned earlier, comprehending dialogue plays a vital role in video editing. However, relying solely on contextual potential has its drawbacks, as it does not entirely encapsulate the visual details within a scene. To overcome this limitation, we integrate saliency prediction, which helps identify key visual aspects beyond spoken interactions, highlighting movements, expressions, and the overall flow of the scene. This method provides a more holistic interpretation of both verbal and visual elements in the video.

The *EditIQ* framework utilizes the ViNet-A model, as outlined in Section 4.3.1. Our findings indicate that ViNet-A excels at grasping the broader context of a scene, extending beyond simple motion detection or fundamental semantic elements like facial recognition.

The VSP model generates a saliency map for each video frame, maintaining the original frame dimensions. To refine the output, we impose a threshold of τ_{sal}, filtering out values below this limit. This process enhances the prominence of key features while minimizing background noise.

Next, we determine the saliency score for each actor by averaging the thresholded saliency values within their respective bounding boxes. To maintain consistency across actors, these scores are normalized. The actor with the highest saliency is assigned a score of λ_{Sal}, while the second most prominent actor is given a score of $\lambda_{Sal}/2$:

$$V(s_t^x) = \begin{cases} \lambda_{Sal} & x \text{ is salient actor} \\ \frac{\lambda_{Sal}}{2} & x \text{ is second-most salient actor} \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

Here, s_t^x denotes a shot s at time t featuring a single actor x. As shown in Figure 5.4, the saliency potential captures actions and actor movements, rather than focusing only on the speaker, making it particularly useful in scenes without dialogue. For higher-order shots, the saliency potential is derived from the saliency potentials of the lower-order shots that make up the higher-order shot, following a method similar to Equation (5.4).

5.2.5 Speaker Module — Speaker Potential

The Speaker potential, as described in Section 3.3.3.2, is intended to emphasize the shot associated with the active speaker. Using speaker-aware subtitles and their respective timestamps, the speaker potential (S) is defined as follows:

$$S(s_t^x) = \begin{cases} \lambda_{Sp} & x \text{ is speaker} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

where s_t^x refers to a 1-shot containing actor x at time t.

5.2.6 Cinematic Constraints

Although the potentials mentioned above assign costs based on the significance of different shots, making editing decisions solely based on these costs might lack cinematic coherence. The contextual potential derived from LLMs does not factor in visual information, potentially resulting in issues like overlapping or jump cuts. Moreover, LLMs do not account for shot duration, as this is addressed during the post-processing phase in the Dialogue Understanding Module (Section 5.2.3). Similarly, raw saliency is not tailored for video editing, as it overlooks cinematic guidelines such as minimum shot

length and flow continuity. Hence, it is essential for any video editing system to incorporate cinematic principles alongside these potentials or costs.

We implement the cinematic constraints outlined in Section 2.1.6, in addition to introducing a new penalty referred to as the Misframing Penalty, which is detailed below:

Misframing Penalty: Misframed shots happen when an actor is partially visible in the current frame, leading to a disruption in the shot's composition. For instance, LLM-generated shot suggestions might not consider the actors' spatial positioning, resulting in a cut that places an actor too close to another. To prevent such issues, we define the misframing penalty as follows:

$$M(s_t^i) = \begin{cases} \lambda_{\text{mis}} & \text{if the framing is poor} \\ 0 & \text{otherwise} \end{cases}$$

If a shot s_t^i is determined to be inadequately framed, the penalty λ_{mis} is incorporated into its cost. A shot is considered poorly framed if it extends beyond its defined boundaries and overlaps with actors.

5.2.7 Shot Selection

In the *EditIQ* framework, the subsequent phase entails identifying the most appropriate shot for storytelling at any given moment, considering the various available shots and rushes. This process is modeled as a discrete optimization task, where each shot's relevance is assessed per frame while maintaining cinematic conventions—such as preventing abrupt cuts, excessive transitions, and inconsistent editing rhythms. The significance of a shot at any instance is derived from the factors outlined in Section 5.2.3, Section 5.2.4, and Section 5.2.5, with cinematic rules integrated as penalty constraints. The final edit is determined by computing the optimal trajectory within an editing graph, which, for a scene featuring n actors, comprises $2^n - 1$ nodes per frame. Each node corresponds to a rush, while edges signify either transitions (cuts) or shot continuity (no cut).

For a given series of frames $t = [1, 2, \ldots, T]$ and the corresponding set of produced shots (rushes) S_t (Equation (5.1)), our approach determines an optimal sequence of shots $\epsilon = \{r_t \mid i = 1, 2, \ldots, T\}$, where $r_t \in S_t$, by optimizing the following objective function:

$$E(\epsilon) = \sum_{t=1}^{T} -\ln\Big(U(r_t)\Big) + \sum_{t=2}^{T}\Big[O(r_{t-1}, r_t, \gamma) + R(r_t, r_{t-1}, \tau) + T(r_{t-1}, r_t)\Big] + \sum_{t=1}^{T} M(r_t) \tag{5.7}$$

Here, $U(r_t)$ denotes the unary cost associated with a shot, which reflects its significance. This cost is computed as the aggregate of contextual potential, saliency potential, and speaker potential (Equation (5.8)). The latter two terms correspond to distinct penalty functions, elaborated in Section 5.2.6 and 2.1.6.

$$U(r_t) = C(r_t) + V(r_t) + S(r_t) \tag{5.8}$$

To address Equation (5.7), we employ a dynamic programming approach. Our technique produces a sequence of selected shots for each frame t from a temporally evolving set of generated shots $\{S_t \mid i = 1, 2, \ldots, T\}$. We construct a cost matrix $CM(r_t, t)$, where $r_t \in S_t$ and $t = [1, 2, \ldots, T]$, with its elements computed iteratively using the following recursive formulation:

$$CM(r_t, t) = \begin{cases} -\ln(U(r_t)) + M(r_t) & t = 1 \\ \min_k \Big[CM(r_k, t-1) - \ln(U(r_t)) & \\ +O(r_k, r_t, \gamma) + R(r_t, r_k, \tau) & \\ +T(r_k, r_t) + M(r_t) \Big] & \text{otherwise} \end{cases}$$

The cost matrix is generated through a forward pass along the temporal axis, where each entry is assigned the minimum cost needed to reach it. After populating the matrix, a backtracking step is performed to extract the optimal sequence of shots. In the final edited video, the wide or master shot from the original footage is used as the establishing shot, fixed at a duration of 2 seconds, while optimization is applied solely to the remaining frames.

5.3 Experiments

To evaluate the effectiveness of *EditIQ*, which integrates a dialogue comprehension module and saliency estimation to inform shot selection, we carried out a psychophysical user study. This study aimed to determine whether our approach produces visually engaging and coherent cinematic sequences. A total of twenty participants took part in the evaluation, with the specifics outlined below.

5.3.1 Dataset

We utilize BBC Old School Dataset (BBC-OSD) [10] described in section 3.2.1 for this study.

5.3.2 LLM Configuration & Details

For our inferences based on a large language model (LLM), we employed the Claude 3.5 Sonnet model from Anthropic, using the "claude-3-5-sonnet-20240620" checkpoint [138]. At the time of system development, the model supported a maximum context length of 200K tokens. To ensure reproducibility, we fixed the temperature parameter at 0.

5.3.3 Parameter Selection

Parameters governing cinematic constraints are essential in defining the pipeline's output and serve as a means of personalizing the edits. Most of the parameters in *EditIQ* are either derived from prior research or determined empirically. For example, the rhythm penalty parameter m, which limits the maximum sequence length, is set to 7—aligning with the average shot duration in films from the past twenty years [139]. The minimum allowable shot length, controlled by l, is set to 1 second, with γ_1 assigned a high value (100) to discourage excessively rapid cuts, which can disrupt visual continuity. Likewise, the overlap penalty ν is maintained at an extremely high value (10^6) to strictly prevent jump cuts. For overlap cost settings, α is set to 0.15, as cuts with less than 15% overlap generally do not introduce visual artifacts, while β is fixed at 0.3 to ensure that transitions exceeding 30% overlap are flagged as overly abrupt.

The flexibility of these parameters allows for tailoring the editing style, whether aiming for a rapid or more gradual pacing. The algorithm's efficiency supports interactive content exploration, enabling real-time modifications to fine-tune the final edit according to personal preferences.

5.3.4 Baselines

We evaluated the videos produced by the *EditIQ* pipeline against several alternative video editing approaches, including Random, Wide, and Speaker. These baselines were chosen as they operate without requiring any manual data collection. To ensure a fair assessment, all baseline videos were presented to users with identical resolution and audio quality, preserving the original footage's timeline.

Alongside these baselines, we performed ablation studies using only the LLM-based Contextual Potential and Saliency-based Visual Potential to show that these methods, when used independently, are inadequate for achieving high-quality video editing. Additionally, we incorporated Human Edits as a baseline for the BBC-OSD Dataset to enable a comparison between our approach and professional editing practices.

Random Baseline: The Random baseline (Ran) is a straightforward approach where shots are picked at random from the available footage at various time points, without taking the scene's context into account. After the shots are selected, cinematic guidelines and penalties are applied to enhance the video, as random selections frequently violate these principles, resulting in disjointed and unattractive outcomes. This lack of cohesion makes it the least effective baseline in terms of narrative structure and visual harmony.

Wide Baseline: The Wide baseline draws inspiration from video re-targeting methods and closely resembles the letterboxing technique discussed in previous studies [140]. This strategy aims to select the widest possible shot that encompasses all the performers on stage. It is essentially a zoomed-in version of the master shot, ensuring the entire scene is captured without leaving any actors out of the frame. The primary objective is to guarantee that all performers are visible at all times, favoring coverage over

more detailed or dynamic shots. While this method is straightforward and ensures continuous visibility of all actors, it lacks the adaptability needed to respond to changes in action or focus within the scene.

Speaker Baseline: Speaker cues play a crucial role in editing scenes driven by dialogue, as emphasized in earlier research by Ranjan *et al.* [141] and Leake *et al.* [134], who suggest choosing shots that clearly feature the speaker. Our speaker-based (Sp) editing baseline adopts a similar approach by selecting the shot that most effectively highlights the speaker from the available footage. This decision-making process relies on data from character-aware subtitling (Section 5.2.1). The selected shot remains constant until a new speaker takes over. To avoid jarring transitions, a minimum shot duration is imposed, and if there is a silence lasting longer than 10 seconds, the algorithm switches to a wide shot for the following segment.

LLM-Only baseline: The LLM-based baseline utilizes the capabilities of large language models to choose shots that are consistent with the narrative context. By analyzing the dialogues in the video, this method seeks to select shots that improve storytelling and ensure continuity. The LLM Potential is calculated as described in Section 5.2.3, ensuring that the chosen shots effectively support the narrative. However, since LLMs do not account for factors like shot durations (both long and short cuts) or visual aspects (such as overlapping shots and jump cuts), adjustments based on cinematic principles are applied afterward to maintain smooth visual transitions and adhere to editing conventions. This adjustment process addresses any potential issues from the initial shot selection, ultimately improving the video's overall quality.

Saliency-Only baseline: The Saliency (Sal) baseline employs visual saliency detection to identify the most prominent elements in a video frame, selecting shots that emphasize these critical areas. By focusing on what captures the viewer's attention, this approach aims to enhance the viewer's engagement. The Saliency Potential is computed as described in Section 5.2.4, which assesses how effectively each shot highlights these significant visual features. However, since saliency detection does not account for factors like minimum shot length or narrative coherence, adjustments based on cinematic guidelines are made afterward to ensure smooth storytelling.

Human Edits: These edits were performed by professional video editors at the BBC, providing a valuable point of comparison for our approach against an actual edit (or an established ground truth). Users rated these edits alongside others, unaware that they were created by humans. The videos used for this baseline were taken directly from the BBC-OSD without any modifications, ensuring an accurate representation of professional editing standards.

Human Edits: These edits were created by professional video editors at the BBC, serving as an important benchmark to compare our method against a real-world edit (or a known ground truth). Users evaluated these edits along with others, without knowing they were human-created. The videos for this baseline were sourced directly from the BBC-OSD, untouched, to provide a true reflection of professional editing standards.

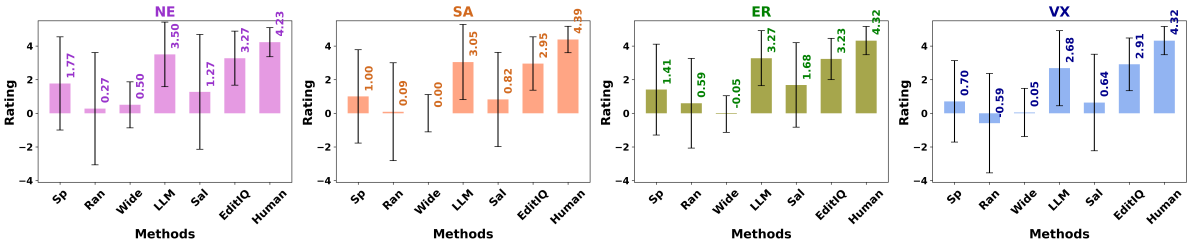


Figure 5.5: **User Study Evaluation:** Bar graphs showing the average user ratings for various editing techniques across four evaluation criteria for the BBC-OSD. The error bars represent the standard deviation. For optimal viewing, please refer to the graph in color and under magnification.

5.4 Evaluation & User Study

5.4.1 Materials & Methods:

To assess the performance of *EditIQ* in comparison to the video editing baselines mentioned earlier, we conducted a psychophysical experiment with 20 participants (ages 20–25, including 2 females). Users were shown both the original and edited versions of 11 videos from the BBC-OSD, as well as the videos produced by all baselines and *EditIQ* . The longest video in this set had a duration of 94 seconds. To ensure a fair comparison, the same parameters were applied for generating all edits using *EditIQ*. After watching the original video, each participant viewed the edited versions[1] in a randomized order to avoid any biases related to the sequence of viewing.

The study was structured so that each participant viewed the original and edited versions of 1-2 videos, ensuring that all 11 recordings were viewed by the 20 users in total. Each participant's session lasted approximately 10 minutes. To maintain consistency, both the original and edited versions of each video received exactly two user ratings. This led to a factor design of 11 (video types) $\times$ 2 (user ratings/video) $\times$ 7 (editing strategies) for the BBC-OSD.

Participants were unaware of the methods used to create the modified versions. They were required to assess each edited version by comparing it to the original and assigning a Likert rating on a scale ranging from -5 to 5 for each of the attributes outlined below. These attributes, derived from [102, 140], were intended to measure how well the modified versions preserved *key scene events* while adhering to event recording limitations. The attributes under consideration included:

(1) **Narrational Effectiveness (NE):** *How effectively did the edited video convey the original narrative?*

(2) **Scene actions (SA):** *How well did the edited video capture actor movements and actions?*

(3) **Actor Emotions and Reactions (ER):** *How well did the edited video capture actor emotions and reactions?*

(4) **Viewing experience (VX):** *How would you rate the edited video for aesthetic quality?*

[1]This includes the six baselines, human-edited versions, and *EditIQ* for BBC-OSD

Participants were introduced to these attributes and provided with an overview of standard cinematic video editing principles before the study began. They were then asked to evaluate questions (1)–(4) *in relation to* a baseline score of '0' assigned to the original video. A *positive* rating indicated that the edited version performed *better* than the original for the given attribute, whereas a *negative* rating signified that the edited version was *inferior* in that regard. The collected responses were aggregated, and average scores were calculated for each attribute and editing approach across all videos (refer to Figure 5.5). The statistical findings and interpretations derived from the user study are detailed below.

5.4.2 Results and Discussion

5.4.2.1 BBC Old School (BBC-OSD):

Bar graphs illustrating the average user ratings for different attributes and editing techniques are shown in Figure 5.5. A two-way balanced ANOVA conducted on the NE, SA, ER, and VX scores across editing approaches indicated significant main effects of *editing strategy* on user perceptions ($p < 0.000001$ for all attributes) and *video category* ($p < 0.005$ for all attributes), with no observed interaction effects. We anticipated that integrating LLM with visual saliency cues through *EditIQ* would produce an engaging, visually dynamic, and aesthetically appealing edit—a hypothesis largely supported by Figure 5.5. As expected, *EditIQ* outperforms the Random, Speaker, Wide, and Saliency baselines across all attributes, though it performs on par with LLM and falls short compared to professionally edited human versions.

Investigating individual attributes, *post-hoc* independent t-tests on NE scores demonstrated a significant distinction between *EditIQ* and Sp ($p < 0.05$), *EditIQ* and Ran ($p < 0.001$), *EditIQ* and Wide ($p < 0.000001$), as well as *EditIQ* and Sal ($p < 0.05$). The NE values for *EditIQ* and LLM were highly similar ($p = 0.6728$), whereas human-edited videos exhibited significantly higher NE scores compared to *EditIQ* ($p < 0.05$). These findings collectively indicate that carefully selecting shots that provide a closer perspective of key scene actors and actions is essential for effective storytelling. The Random baseline, which disregards scene content when selecting shots, performs the worst, followed by the Wide baseline, which captures the full scene context but lacks focus on critical details. The Speaker and Saliency-based approaches, which rely on speech and visual cues for shot selection, yield comparable results ($p = 0.5968$). However, LLM-driven editing, which leverages scene narratives, significantly surpasses visual saliency-based editing ($p < 0.05$), highlighting that visual cues primarily act as a complementary aid to LLM capabilities in automated video editing.

For depicting scene actions, user score trends closely resemble NE scores. *EditIQ* demonstrates a substantial advantage over Ran ($p < 0.0005$), Wide ($p < 0.000001$), Sp ($p < 0.001$), and Sal ($p < 0.05$), while showing no notable difference from LLM ($p = 0.9177$) and falling short in comparison to human editing ($p < 0.005$). Users find visual saliency and speech-driven editing to be similarly effective ($p = 0.7306$), whereas LLM-based editing surpasses both approaches ($p < 0.05$). However, variations emerge when evaluating the portrayal of actor emotions and responses. *EditIQ* receives significantly

higher ratings than Ran ($p < 0.0005$), Wide ($p < 0.000001$), Sp ($p < 0.001$), and Sal ($p < 0.05$), remains on par with LLM ($p = 0.9177$), but does not reach the level of human editing ($p < 0.005$). In this aspect, saliency-based editing achieves stronger results, surpassing Sp ($p < 0.01$) while still lagging behind LLM-based editing ($p < 0.05$). These findings suggest that saliency is particularly effective at capturing visually striking facial expression shifts and reactions, though it struggles to encapsulate overarching scene narratives or essential actions.

Ultimately, similar patterns emerge in terms of the viewing experience. *EditIQ* achieves higher performance than all other automated techniques but remains notably inferior to professional human editing ($p < 0.001$). Saliency-based and speaker-driven editing yield nearly identical results ($p = 0.9326$), while LLM-based editing surpasses both methods ($p < 0.05$).

5.4.2.2 Past-Experience of Participants

Out of the 20 participants, five had prior exposure to video editing, while the rest had no background in using editing tools or techniques. Those with experience exhibited a more discerning approach, consistently assigning notably lower scores to the random baseline than their non-experienced counterparts, who provided more neutral ratings. This suggests their enhanced ability to identify and critique editing flaws. Furthermore, experienced participants showed a stronger preference for high-quality edits, awarding human edits higher ratings—4.4 (NE), 4.8 (SA), 4.6 (ER), 5.0 (VX)—compared to non-experienced participants, who rated them at 4.17 (NE), 4.26 (SA), 4.23 (ER), and 4.11 (VX). This implies that their familiarity with editing enabled them to discern finer details and limitations in the process, leading to a more refined assessment.

5.4.2.3 Discussion Summary

We assessed *EditIQ* in comparison to alternative baselines for a quiz event recorded by the BBC-OSD. In this specific context, where speech cues primarily direct visual focus, the integration of saliency and LLM cues proves to be of limited advantage. Nonetheless, *EditIQ* generally surpasses other automated methods, with professional human editing being the only approach that achieves higher scores for BBC-OSD.

5.5 Conclusion

This work presents *EditIQ*, a system designed for the automated editing of performance videos filmed with unmanned, static, wide-angle, high-resolution cameras. We leverage LLMs for dialogue interpretation, using them to suggest which individual or group should be highlighted at each word timestamp in the character-aware subtitles. In addition, we integrate video saliency prediction techniques to capture actions and visual aspects that are not conveyed through speech. The LLM recommendations, saliency predictions, and speaker data are merged to assess the significance of each shot in the generated rushes at specific times. These shot potentials are then adjusted with cinematic constraints, such as avoiding

jump cuts, rapid transitions, improper framing, and ensuring a smooth rhythm. The final output is a carefully edited sequence that maintains essential content while following cinematic conventions, resulting in a visually appealing video. The performance of *EditIQ* is evaluated against competing baselines in a psychophysical study with twenty participants using the BBC-OSD. *EditIQ* typically outperforms other baselines, only scoring lower when compared to professional human editing for the BBC-OSD.

Chapter 6

Conclusion and Future Work

In conclusion, this thesis analyzes active speaker detection and saliency prediction for automated video editing. First, it introduces a novel audio-based ASD approach that outperforms existing methods in video settings, emphasizing the critical role of speaker information in video editing. The proposed KNN-based method demonstrates superior performance, highlighting the potential for using readily available audio samples and actors' face images to enhance the accuracy of ASD models in practical applications. This work lays the groundwork for future research aimed at achieving fully automated video editing by exploring additional factors that influence this process and developing more robust ASD models.

Second, the thesis introduces two efficient models, ViNet-S and ViNet-A, that advance the state of the art in video saliency prediction. ViNet-S achieves high performance with a lightweight design, while ViNet-A excels in both visual and audio-visual datasets, even without relying on audio cues. The ensemble approach combining these models demonstrates the effectiveness of integrating global and localized features, paving the way for further research in saliency models.

We introduce *EditIQ*, an automated video editing framework that combines two essential components: (1) an LLM-driven dialogue comprehension module that interprets conversational dynamics, and (2) a visual saliency prediction system that highlights important scene elements and camera shots. The editing process is framed as an energy minimization task, where cinematic principles direct the selection of shots, transitions, and overall continuity. In the end, EditIQ produces a visually cohesive and aesthetically pleasing representation of the original narrative, providing a smooth and immersive viewing experience.

Limitations: Using the human gaze to identify important moments in video footage can be very helpful. People naturally understand stories and can recognize the most significant parts of a video. By analyzing where someone's gaze is focused, video editors can gain valuable insights into key moments. However, this method has some drawbacks. It can be subjective since different people might focus on different things, leading to potential bias. Additionally, to effectively use saliency prediction models for automated video editing, it's important to train these models on various complex scenes. For instance, saliency models that consider context and dialogue besides visual cues are necessary to accurately predict salient regions in dialogue-driven scenes.

Another key limitation of the *EditIQ* system is its inability to perform real-time editing, a critical feature for live events. Previous work has established the feasibility of online rush generation, stabilization, and camera selection [5, 142]. Future endeavors will seek to integrate these with a streaming LLM variant, enabling dialogues to be processed incrementally rather than as a complete script.

List of Related Publications

[P1] **Rohit Girmaji**, Sudheer Achary, Adhiraj Deshmukh and Vineet Gandhi, **"Assessing active speaker detection algorithms through the lens of automated editing"**, *IMXw '23: Proceedings of the 2023 ACM International Conference on Interactive Media Experiences Workshops*.

[P2] **Rohit Girmaji**, Siddharth Jain, Bhav Beri, Sarthal Bansal and Vineet Gandhi, **"Minimalistic Video Saliency Prediction via Efficient Decoder & Spatio Temporal Action Cues"**, in proceedings of *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2025.

[P3] **Rohit Girmaji**, Bhav Beri, Ramanathan Subramanian, and Vineet Gandhi **"EditIQ: Automated Cinematic Editing of Static Wide-Angle Videos via Dialogue Interpretation and Saliency Cues"**, in proceedings of *ACM Conference on Intelligent User Interfaces (ACM IUI)*, 2025.

[P4] Sudheer Achary, **Rohit Girmaji**, Adhiraj Anil Deshmukh, and Vineet Gandhi **"Real-Time GAZED: Online Shot Selection and Editing of Virtual Cameras from Wide-Angle Monocular Video Recordings"**, in proceedings of *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[P5] Shreyank Jyoti, **Rohit Girmaji**, Ritvik Agrawal, Sarath Sivaprasad and Vineet Gandhi, **"Salient Face Prediction without Bells and Whistles"**, in proceedings of *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2022.

[P6] Ritvik Agrawal, Shreyank Jyoti, **Rohit Girmaji**, Sarath Sivaprasad, and Vineet Gandhi, **"Does Audio help in deep Audio-Visual Saliency prediction models?"**, in proceedings of *International Conference on Multimodal Interaction (ICMI)*, 2022.

Bibliography

[1] Murch and Walter, *In the Blink of an Eye*. Silman-James Press Los Angeles, 2001, vol. 995.

[2] M. Leake, A. Davis, A. Truong, and M. Agrawala, "Computational video editing for dialogue-driven scenes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 130:1–130:14, 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073653

[3] K. B. Moorthy, M. Kumar, R. Subramanian, and V. Gandhi, "Gazed–gaze-guided cinematic editing of wide-angle monocular video recordings," 2020, pp. 1–11.

[4] Q. Galvane, R. Ronfard, C. Lino, and M. Christie, "Continuity editing for 3d animation," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, B. Bonet and S. Koenig, Eds. AAAI Press, 2015, pp. 753–762. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9662

[5] Gandhi, Vineet, Ronfard, Remi, and M. Gleicher, "Multi-clip video editing from a single viewpoint," in *Proceedings of the 11th European Conference on Visual Media Production*, 2014, pp. 1–10.

[6] V. Gandhi and R. Ronfard, "A computational framework for vertical video editing," in *4th Workshop on Intelligent Camera Control, Cinematography and Editing*. Eurographics Association, 2015, pp. 31–37.

[7] K. Vishnubhotla, A. Hammond, G. Hirst, and S. M. Mohammad, "The emotion dynamics of literary novels," *ACL*, 2024.

[8] Z. Zhou, X. Gu, Y. Zhao, and H. Xu, "Pop-cee: Position-oriented prompt-tuning model for causal emotion entailment," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 4199–4210.

[9] K. Sundar, S. Toshniwal, M. Tapaswi, and V. Gandhi, "Major entity identification: A generalizable alternative to coreference resolution," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 11 679–11 695.

[10] S. Jolly, G. Phillipson, and M. Evans, "Old school: An 8k multicamera shoot to create a dataset for computational cinematography," in *Proceedings of the 2023 ACM International Conference on Interactive Media Experiences Workshops*, ser. IMXw '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 76. [Online]. Available: https://doi.org/10.1145/3604321.3604374

[11] R. Thompson and C. J. Bowen, *Grammar of the Edit*. Taylor & Francis, 2009, vol. 13.

[12] W. Wang, J. Shen, J. Xie, M.-M. Cheng, H. Ling, and A. Borji, "Revisiting video saliency prediction in the deep learning era," vol. 43, no. 1, pp. 220–237, 2021.

[13] J. Roth, S. Chaudhuri, O. Klejch, R. Marvin, A. Gallagher, L. Kaver, S. Ramaswamy, A. Stopczynski, C. Schmid, Z. Xi *et al.*, "Ava active speaker: An audio-visual dataset for active speaker detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4492–4496.

[14] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: speaker diarisation in the wild," *arXiv preprint arXiv:2007.01216*, 2020.

[15] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*. Springer, 2017, pp. 251–263.

[16] G. Sell and D. Garcia-Romero, "Diarization resegmentation in the factor analysis subspace," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4794–4798.

[17] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4930–4934.

[18] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *2018 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5239–5243.

[19] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[20] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[21] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," *Interspeech*, 2019.

[22] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 296–303.

[23] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2020.

[24] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, 2021.

[25] M. Shahid, C. Beyan, and V. Murino, "S-VVAD: visual voice activity detection by motion segmentation," in *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*. IEEE, 2021, pp. 2331–2340. [Online]. Available: https://doi.org/10.1109/WACV48630.2021.00238

[26] F. Patrona, A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Visual voice activity detection in the wild," *IEEE Trans. Multim.*, vol. 18, no. 6, pp. 967–977, 2016. [Online]. Available: https://doi.org/10.1109/TMM.2016.2535357

[27] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *arXiv preprint arXiv:1804.03619*, 2018.

[28] J. L. Alcázar, F. Caba, L. Mai, F. Perazzi, J.-Y. Lee, P. Arbeláez, and B. Ghanem, "Active speakers in context," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 465–12 474.

[29] T. Afouras, A. Owens, J. S. Chung, and A. Zisserman, "Self-supervised learning of audio-visual objects from video," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 208–224.

[30] R. Tao, Z. Pan, R. K. Das, X. Qian, M. Z. Shou, and H. Li, "Is someone speaking?: Exploring long-term temporal features for audio-visual active speaker detection," in *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, H. T. Shen, Y. Zhuang, J. R. Smith, Y. Yang, P. César, F. Metze, and B. Prabhakaran, Eds. ACM, 2021, pp. 3927–3935. [Online]. Available: https://doi.org/10.1145/3474085.3475587

[31] K. Min, S. Roy, S. Tripathi, T. Guha, and S. Majumdar, "Intel labs at activitynet challenge 2022: Spell for long-term active speaker detection."

[32] T. Inoue, K.-i. Okada, and Y. Matsushita, "Learning from tv programs: Application of tv presentation to a videoconferencing system," in *Proceedings of the 8th annual ACM symposium on User interface and software technology*, 1995, pp. 147–154.

[33] Q. Liu, Y. Rui, A. Gupta, and J. J. Cadiz, "Automating camera management for lecture room environments," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2001, pp. 442–449.

[34] Y. Rui, A. Gupta, and J. J. Cadiz, "Viewing meeting captured by an omni-directional camera," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2001, pp. 450–457.

[35] A. Ranjan, J. Birnholtz, and R. Balakrishnan, "Improving meeting capture by applying television production principles with audio and motion detection," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 227–236.

[36] Elson, David, Riedl, and Mark, "A lightweight intelligent virtual cinematography system for machinima production," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 3, no. 1, 2007, pp. 8–13.

[37] C. Lino, M. Chollet, M. Christie, and R. Ronfard, "Computational model of film editing for interactive storytelling," in *Interactive Storytelling - Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011, Vancouver, Canada, November 28 - 1 December, 2011. Proceedings*, ser. Lecture Notes in Computer Science, M. Si, D. Thue, E. André, J. C. Lester, T. J. Tanenbaum, and V. Zammitto, Eds., vol. 7069. Springer, 2011, pp. 305–308. [Online]. Available: https://doi.org/10.1007/978-3-642-25289-1_35

[38] B. Merabti, M. Christie, and K. Bouatouch, "A virtual director using hidden markov models," *Comput. Graph. Forum*, vol. 35, no. 8, pp. 51–67, 2016. [Online]. Available: https://doi.org/10.1111/cgf.12775

[39] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman, "Human focused action localization in video," in *Trends and Topics in Computer Vision: ECCV 2010 Workshops, Heraklion, Crete, Greece, September 10-11, 2010, Revised Selected Papers, Part I 11*. Springer, 2012, pp. 219–233.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[41] I. Mccowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. L. Masson, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus." Methods and Techniques in Behavioral Research, 2005.

[42] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserma, "Spot the conversation : Speaker diarisation in the wild." arXiv : Arxiv2007.01216, 2020.

[43] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third dihard diarization challenge." INTERSPEECH, 2021.

[44] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors." in *Odyssey*, vol. 2018, 2018, pp. 105–111.

[45] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[46] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII*, ser. Lecture Notes in Computer Science, S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., vol. 13682. Springer, 2022, pp. 1–21. [Online]. Available: https://doi.org/10.1007/978-3-031-20047-2_1

[47] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: exceeding YOLO series in 2021," *CoRR*, vol. abs/2107.08430, 2021. [Online]. Available: https://arxiv.org/abs/2107.08430

[48] E. Jain, Y. Sheikh, A. Shamir, and J. K. Hodgins, "Gaze-driven video re-editing," *ACM Trans. Graph.*, vol. 34, no. 2, pp. 21:1–21:12, 2015. [Online]. Available: https://doi.org/10.1145/2699644

[49] K. K. Rachavarapu, M. Kumar, V. Gandhi, and R. Subramanian, "Watch to edit: Video retargeting using gaze," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 205–215, 2018. [Online]. Available: https://doi.org/10.1111/cgf.13354

[50] S. Vijayakumar, J. Conradt, T. Shibata, and S. Schaal, "Overt visual attention for a humanoid robot," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 4, 2001, pp. 2332–2337 vol.4.

[51] J. Driscoll, R. Peters, and K. Cave, "A visual attention network for a humanoid robot," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 3, 1998, pp. 1968–1974 vol.3.

[52] C. Bak, A. Kocak, E. Erdem, and A. Erdem, "Spatio-temporal saliency networks for dynamic saliency prediction," vol. 20, no. 7, pp. 1688–1698, 2018.

[53] Z. Wu, L. Su, and Q. Huang, "Learning coupled convolutional networks fusion for video saliency prediction," vol. 29, no. 10, pp. 2960–2971, 2019.

[54] K. Zhang and Z. Chen, "Video saliency prediction based on spatial-temporal two-stream network," vol. 29, no. 12, pp. 3544–3557, 2019.

[55] S. Gorji and J. J. Clark, "Going from image to video saliency: Augmenting image salience with dynamic attentional push," 2018, pp. 7501–7511.

[56] Q. Lai, W. Wang, H. Sun, and J. Shen, "Video saliency prediction using spatiotemporal residual attentive networks," vol. 29, pp. 1113–1126, 2020.

[57] P. Linardos, E. Mohedano, J. J. Nieto, N. E. O'Connor, X. G. i Nieto, and K. McGuinness, "Simple vs complex temporal recurrences for video saliency prediction," 2019. [Online]. Available: https://arxiv.org/abs/1907.01869

[58] X. Wu, Z. Wu, J. Zhang, L. Ju, and S. Wang, "Salsac: A video saliency prediction model with shuffled attentions and correlation-based convlstm," vol. 34, no. 07, 2020, pp. 12 410–12 417.

[59] K. Min and J. J. Corso, "Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection," 2019, pp. 2394–2403.

[60] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," 2018, pp. 305–321.

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

[62] G. Bellitto, F. Proietto Salanitri, S. Palazzo, F. Rundo, D. Giordano, and C. Spampinato, "Hierarchical domain-adapted feature learning for video saliency prediction," vol. 129, pp. 3216–3232, 2021.

[63] S. Jain, P. Yarlagadda, S. Jyoti, S. Karthik, R. Subramanian, and V. Gandhi, "Vinet: Pushing the limits of visual modality for audio-visual saliency prediction," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3520–3527.

[64] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[65] Z. Wang, Z. Liu, G. Li, Y. Wang, T. Zhang, L. Xu, and J. Wang, "Spatio-temporal self-attention network for video saliency prediction," vol. 25, pp. 1161–1174, 2021.

[66] Y. Wang, Z. Liu, Y. Xia, C. Zhu, and D. Zhao, "Spatiotemporal module for video saliency prediction based on self-attention," vol. 112, p. 104216, 2021.

[67] C. Ma, H. Sun, Y. Rao, J. Zhou, and J. Lu, "Video saliency forecasting transformer," vol. 32, no. 10, pp. 6850–6862, 2022.

[68] X. Zhou, S. Wu, R. Shi, B. Zheng, S. Wang, H. Yin, J. Zhang, and C. Yan, "Transformer-based multi-scale feature integration network for video saliency prediction," vol. 33, no. 12, pp. 7696–7707, 2023.

[69] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," 2022, pp. 3202–3211.

[70] M. Moradi, S. Palazzo, and C. Spampinato, "Transformer-based video saliency prediction with high temporal dimension decoding," *VISIGRAPP*, 2024.

[71] J. Pan, S. Chen, M. Z. Shou, Y. Liu, J. Shao, and H. Li, "Actor-context-actor relation network for spatio-temporal action localization," 2021, pp. 464–474.

[72] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM International Conference on Multimedia*. Association for Computing Machinery, 2017.

[73] A. Tsiami, P. Koutras, and P. Maragos, "Stavis: Spatio-temporal audiovisual saliency network," 2020, pp. 4766–4776.

[74] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" 2018, pp. 6546–6555.

[75] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," vol. 29, 2016.

[76] Q. Chang and S. Zhu, "Temporal-spatial feature pyramid for video saliency detection," *arXiv preprint arXiv:2105.04213*, 2021.

[77] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2017, pp. 2117–2125.

[78] J. Xiong, G. Wang, P. Zhang, W. Huang, Y. Zha, and G. Zhai, "Casp-net: Rethinking video saliency prediction from an audio-visual consistency perceptual perspective," 2023, pp. 6441–6450.

[79] S. Mathe and C. Sminchisescu, "Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition," vol. 37, no. 7, pp. 1408–1424, 2015.

[80] X. Min, G. Zhai, K. Gu, and X. Yang, "Fixation prediction through multimodal analysis," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 1, oct 2016. [Online]. Available: https://doi.org/10.1145/2996463

[81] A. Coutrot and N. Guyader, "How saliency, faces, and sound influence gaze in dynamic social scenes," *Journal of vision*, vol. 14, no. 8, pp. 5–5, 2014.

[82] ——, "Toward the introduction of auditory information in dynamic visual attention models," in *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2013, pp. 1–4.

[83] ——, "An efficient audiovisual saliency model to predict eye positions when looking at conversations," in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 1531–1535.

[84] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson, "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cognitive Computation*, vol. 3, pp. 5–24, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:5736936

[85] P. Koutras, A. Katsamanis, and P. Maragos, "Predicting eyes' fixations in movie videos: Visual saliency experiments on a new eye-tracking database," in *Engineering Psychology and Cognitive Ergonomics*, D. Harris, Ed. Cham: Springer International Publishing, 2014, pp. 183–194.

[86] Y. Liu, M. Qiao, M. Xu, B. Li, W. Hu, and A. Borji, "Learning to predict salient faces: A novel visual-audio saliency model." Springer, 2020, pp. 413–429.

[87] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," 2019, pp. 6202–6211.

[88] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik, "Ava: A video dataset of spatio-temporally localized atomic visual actions," 2018, pp. 6047–6056.

[89] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," 2018, pp. 6848–6856.

[90] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi, "Deep roots: Improving cnn efficiency with hierarchical filter groups," 2017, pp. 5977–5986.

[91] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" vol. 41, no. 3, pp. 740–757, 2019.

[92] F. Hu, S. Palazzo, F. P. Salanitri, G. Bellitto, M. Moradi, C. Spampinato, and K. McGuinness, "Tinyhd: Efficient video saliency prediction with heterogeneous decoders using hierarchical maps distillation," 2023, pp. 2050–2059.

[93] M. Qiao, Y. Liu, M. Xu, X. Deng, B. Li, W. Hu, and A. Borji, "Joint learning of audio-visual saliency prediction and sound source localization on multi-face videos," vol. 132, pp. 2003–2025, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266677960

[94] R. Droste, J. Jiao, and J. A. Noble, "Unified image and video saliency modeling." Springer, 2020, pp. 419–435.

[95] J. Xiong, P. Zhang, T. You, C. Li, W. Huang, and Y. Zha, "Diffsal: Joint audio and video learning for diffusion saliency prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 27 273–27 283.

[96] S. Mathe and C. Sminchisescu, "Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition," *TPAMI*, vol. 37, no. 7, pp. 1408–1424, 2014.

[97] R. Agrawal, S. Jyoti, R. Girmaji, S. Sivaprasad, and V. Gandhi, "Does audio help in deep audio-visual saliency prediction models?" in *Proceedings of the 2022 International Conference on Multimodal Interaction*, 2022, pp. 48–56.

[98] J. Pan, S. Chen, M. Z. Shou, Y. Liu, J. Shao, and H. Li, "Actor-context-actor relation network for spatio-temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 464–474.

[99] P. Krähenbühl, M. Lang, A. Hornung, and M. Gross, "A system for retargeting of streaming video," in *ACM SIGGRAPH Asia 2009 Papers*, ser. SIGGRAPH Asia '09. Association for Computing Machinery, 2009.

[100] Y.-S. Wang, H.-C. Lin, O. Sorkine, and T.-Y. Lee, "Motion-based video retargeting with optimized crop-and-warp," in *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–9.

[101] F. Liu and M. Gleicher, "Video retargeting: automating pan and scan," in *Proceedings of the 14th ACM international conference on Multimedia*, 2006, pp. 241–250.

[102] K. K. Rachavarapu, M. Kumar, V. Gandhi, and R. Subramanian, "Watch to edit: Video retargeting using gaze," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 205–215.

[103] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust l1 optimal camera paths," in *CVPR 2011*, 2011, pp. 225–232.

[104] R. Heck, M. Wallick, and M. Gleicher, "Virtual videography," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 3, no. 1, pp. 4–es, 2007.

[105] C. Zhang, Y. Rui, J. Crawford, and L.-W. He, "An automated end-to-end lecture capture and broadcasting system," *ACM Transactions on multimedia computing, communications, and applications (TOMM)*, vol. 4, no. 1, pp. 1–23, 2008.

[106] R. Schäfer, P. Kauff, and C. Weissig, "Ultra high resolution video production and display as basis of a format agnostic production system," in *Proceedings of International Broadcast Conference (IBC 2010)*, vol. 1, 2010.

[107] P. Carr, M. Mistry, and I. Matthews, "Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 193–202.

[108] Y.-C. Su, D. Jayaraman, and K. Grauman, "Pano2vid: Automatic cinematography for watching 360 videos," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 154–171.

[109] C. Tang, O. Wang, F. Liu, and P. Tan, "Joint stabilization and direction of 360 videos," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 2, pp. 1–13, 2019.

[110] V. R. Gaddam, R. Eg, R. Langseth, C. Griwodz, and P. Halvorsen, "The cameraman operating my virtual camera is artificial: Can the machine be as good as a human?" *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 4, Jun. 2015.

[111] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[112] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *arXiv preprint arXiv:2206.14651*, 2022.

[113] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9686–9696.

[114] R. Varghese and S. M., "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024, pp. 1–6.

[115] S. Jain, P. Yarlagadda, S. Jyoti, S. Karthik, R. Subramanian, and V. Gandhi, "Vinet: Pushing the limits of visual modality for audio-visual saliency prediction," in *IROS*. IEEE, 2021, pp. 3520–3527.

[116] Y. Kao, B. Pan, M. Xu, J. Lyu, X. Zhu, Y. Chang, X. Li, and Z. Lei, "Toward 3d face reconstruction in perspective projection: Estimating 6dof face pose from monocular image," *IEEE Transactions on Image Processing*, vol. 32, pp. 3080–3091, 2023.

[117] Y. Takemae, K. Otsuka, and N. Mukawa, "Impact of video editing based on participants' gaze in multiparty conversation," in *CHI'04 extended abstracts on Human Factors in Computing Systems*, 2004, pp. 1333–1336.

[118] S. Daigo and S. Ozawa, "Automatic pan control system for broadcasting ball games based on audience's face direction," in *Proceedings of the 12th annual ACM international conference on Multimedia*, 2004, pp. 444–447.

[119] H. Park, E. Jain, and Y. Sheikh, "3d social saliency from head-mounted cameras," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[120] I. Arev, H. S. Park, Y. Sheikh, J. Hodgins, and A. Shamir, "Automatic editing of footage from multiple social cameras," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014.

[121] E. Jain, Y. Sheikh, A. Shamir, and J. Hodgins, "Gaze-driven video re-editing," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 2, pp. 1–12, 2015.

[122] D. B. Christianson, S. E. Anderson, L.-w. He, D. H. Salesin, D. S. Weld, and M. F. Cohen, "Declarative camera control for automatic cinematography," in *AAAI/IAAI, Vol. 1*, 1996, pp. 148–155.

[123] L.-w. He, M. F. Cohen, and D. H. Salesin, "The virtual cinematographer: a paradigm for automatic real-time camera control and directing," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 217–224.

[124] D. Arijon, "Grammar of the film language," 1976.

[125] C. Lino, M. Chollet, M. Christie, and R. Ronfard, "Computational model of film editing for interactive storytelling," in *International Conference on Interactive Digital Storytelling*. Springer, 2011, pp. 305–308.

[126] Q. Galvane, R. Ronfard, C. Lino, and M. Christie, "Continuity editing for 3d animation," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[127] B. Merabti, M. Christie, and K. Bouatouch, "A virtual director using hidden markov models," in *Computer Graphics Forum*, vol. 35, no. 8. Wiley Online Library, 2016, pp. 51–67.

[128] J. Wang, C. Xu, E. Chng, H. Lu, and Q. Tian, "Automatic composition of broadcast sports video," *Multimedia Systems*, vol. 14, no. 4, pp. 179–193, 2008.

[129] F. Chen and C. De Vleeschouwer, "Personalized production of basketball videos from multi-sensored data under limited display resolution," *Computer Vision and Image Understanding*, vol. 114, no. 6, pp. 667–680, 2010.

[130] C. Chen, O. Wang, S. Heinzle, P. Carr, A. Smolic, and M. H. Gross, "Computational sports broadcasting: Automated director assistance for live sports," in *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo, ICME 2013, San Jose, CA, USA, July 15-19, 2013*, 2013.

[131] J. Chen, L. Meng, and J. J. Little, "Camera selection for broadcasting soccer games," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 427–435.

[132] Y. Pan, Y. Chen, Q. Bao, N. Zhang, T. Yao, J. Liu, and T. Mei, "Smart director: An event-driven directing system for live broadcasting," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 4, pp. 1–18, 2021.

[133] H. Park, E. Jain, and Y. Sheikh, "3d social saliency from head-mounted cameras," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/1bf2efbbe0c49b9f567c2e40f645279a-Paper.pdf

[134] M. Leake, A. Davis, A. Truong, and M. Agrawala, "Computational video editing for dialogue-driven scenes," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073653

[135] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," *INTERSPEECH 2023*, 2023.

[136] B. Korbar, J. Huh, and A. Zisserman, "Look, listen and recognise: character-aware audio-visual subtitling," 2024.

[137] B. Korbar and A. Zisserman, "Personalised clip or: how to find your vacation videos," in *British Machine Vision Conference*, 2022.

[138] Anthropic, "The claude 3 model family: Opus, sonnet, haiku," Jun. 2024. [Online]. Available: https://assets.anthropic.com/m/61e7d27f8c8f5919/original/Claude-3-Model-Card.pdf

[139] J. Cutting and A. Candan Simsek, "Shot durations, shot classes, and the increased pace of popular movies," *Projections*, vol. 9, pp. 40–52, 12 2015.

[140] E. Jain, Y. Sheikh, A. Shamir, and J. Hodgins, "Gaze-driven video re-editing," *ACM Trans. Graph.*, vol. 34, no. 2, Mar. 2015. [Online]. Available: https://doi.org/10.1145/2699644

[141] A. Ranjan, J. Birnholtz, and R. Balakrishnan, "Improving meeting capture by applying television production principles with audio and motion detection," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 227–236. [Online]. Available: https://doi.org/10.1145/1357054.1357095

[142] S. Achary, R. Girmaji, A. A. Deshmukh, and V. Gandhi, "Real time gazed: Online shot selection and editing of virtual cameras from wide-angle monocular video recordings," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4108–4116.