Title	AI supported haptic design process
Sub Title	
Author	黃, 恪非(Wong, Keh Fei) 南澤, 孝太(Minamizawa, Kouta)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2023
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2023年度メディアデザイン学 第1014号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002023-1014

Master's Thesis
Academic Year 2023

AI Supported Haptic Design Process

Keio University
Graduate School of Media Design

Keh Fei Wong

A Master's Thesis
submitted to Keio University Graduate School of Media Design
in partial fulfillment of the requirements for the degree of
Master of Media Design

Keh Fei Wong

Master's Thesis Advisory Committee:

Professor Kouta Minamizawa	(Main Research Supervisor)
Professor Kazunori Sugiura	(Sub Research Supervisor)

Master's Thesis Review Committee:

Professor Kouta Minamizawa	(Chair)
Professor Kazunori Sugiura	(Co-Reviewer)
Professor Nanako Ishido	(Co-Reviewer)

Abstract of Master's Thesis of Academic Year 2023

AI Supported Haptic Design Process

Category: Design

Summary

The integration of haptic feedback has the potential to enhance the overall multimedia viewing experience. To achieve precise and immersive haptic experiences in conjunction with videos, various authoring methods have been explored in previous research. Manual authoring has been favored for this task, but its labor-intensive nature and time-consuming process pose significant challenges. To address these limitations, this research proposes a multi-model deep learning framework capable of automatically generating haptic audio, which can be seamlessly integrated into manual authoring software to enhance the efficiency of haptic designers and authors. An experimental study was conducted to evaluate the effectiveness of the automatically generated haptic audio in assisting the annotation process. The results demonstrated a significant improvement in Quality of Experience (QoE) when compared to manual authoring, highlighting the enhanced efficiency achieved through the combination of automatic haptic audio generation.

Keywords:

automatic haptic authoring, vibrotactile display, deep learning, sound event detection, object detection

Keio University Graduate School of Media Design

Keh Fei Wong

Contents

Acknowledgements **vii**

1 Introduction **1**
1.1. Enhancing Human-Computer Interaction through Tactile Stimuli 1
1.1.1 The Ultimate Display 1
1.2. Haptic Authoring 2
1.2.1 Manual Haptic Authoring 2
1.2.2 Automatic Haptic Authoring 3
1.2.3 Limitation of automatic Haptic Authoring 3
1.3. Involving Event Identification to Automatic Haptic Authoring . . 4
1.4. Research Goal 5
1.5. Thesis Structure 5

2 Related Works **7**
2.1. Enhancing Experience with Haptic 7
2.1.1 Cutaneous Haptic Stimuli 7
2.1.2 Kinesthetic Haptic Stimuli 8
2.2. Haptic in Multimedia Contents 9
2.2.1 Haptic Stimuli Delivery Mechanism with Multimedia Content 9
2.3. Authoring Methods 10
2.3.1 Manual Haptic Authoring 10
2.3.2 Authoring with Sensor 11
2.3.3 Automatic Authoring 12
2.4. Deep Neural Network in Object Detection 13
2.5. Anomalous Sound Event Detection 16
2.5.1 Monophonic and Polyphonic SED System 16
2.6. Summary 19

3 Concept Design **21**
3.1. Research Objective 21
3.2. Initial system design 22
3.2.1 Training Object Detection Model 22
3.3. Second Prototype 28
3.3.1 Elevating Object Detection Performance 28
3.3.2 Identifying Contact Event 31
3.3.3 Improving Algorithm for Event Detection 32
3.4. System Improvement with Audio Detection 34
3.4.1 First SED model 34
3.4.2 Second SED Model 37
3.5. Final Prototype 39
3.5.1 Generates Haptic Audio Automatically 41

4 Prove of Concept **42**
4.1. Pilot Test 42
4.1.1 Feedback for Pilot Test 43
4.2. Semi-Auto Authoring Experiment 43
4.2.1 Overview 43
4.2.2 Experiment Design 44
4.2.3 Results 47
4.3. Discussion 52

5 Conclusion **53**

References **55**

Appendices **66**
A. Code for Automatic Generating Haptic Video 66
B. Code for Combining Haptic, Video and Audio Components 76

List of Figures

2.1 Force feedback device that simulate grasping rigid objects in VR 8
2.2 Implementing multiple array of vibrotactile actuators inside a jacket 10
2.3 A novel manual authoring tool with automatic sensory effect recognition 11
2.4 Overview Architecture of YOLOv8 Model 14
2.5 Illustration of Monophonic SED system, only one event can be identify within a single audio input 17
2.6 Illustration of Polyphonic SED system, different color tags refers to the onset and offset of a particular event 18

3.1 Prediction from the trained object detection model on testing image 23
3.2 F1 score indicating model accuracy in predicting both positive and negative prediction on sword. 23
3.3 F1 score indicating classification model accuracy in predicting the contact event. 24
3.4 Process of the system in finding the contact events in a single video frame. 25
3.5 F1 score for object detection for racket and shuttle ball. 27
3.6 Illustrate a single frame output from motion saliency algorithm, also used as training data for object detection model. 29
3.7 F1 score for object detection for racket and shuttle ball. 30
3.8 Confusion Matrix for model accuracy on testing video 30
3.9 Two frames prior to the smash being performed, the badminton shuttle is currently positioned out of camera's boundary. 32
3.10 Dataframe including information about the movement direction and changes in direction of shuttle ball. 33
3.11 Audio data manually annotated with Label Studio [1] 35

3.12 Strong labelled audio data in the form of .csv file 35
3.13 Weakly labelled audio data in the form of .csv file 36
3.14 Spectrogram of audio data with event annotations shown in green grid. 38
3.15 Cropped spectrogram with event label to be used as training data. 38
3.16 Cropped spectrogram with event label to be used as training data. 39
3.17 Cropped spectrogram with event label to be used as training data. 39
3.18 Illustration of the overall system framework 40

4.1 Haptic Device 42
4.2 The interface of Ableton Live during the manual annotation task. 44
4.3 The track on above is the manual annotation by participant 3 ; The track on below is the automatic generated haptic audio . . . 45
4.4 The interface of Ableton Live during the semi-auto annotation task. 46
4.5 The collective results comparing the annotation experience in both tasks from the questionnaire (Q1-Q8, See Table 4.1) * $p<0.05$, ** $p<0.01$, *** $p<0.001$. The figure includes scatter plots that visualize the responses of each participant with symbol ($\cdot$) for Manual Authoring, ($\times$) for Semi-Automatic Authoring 48
4.6 The time taken to complete each task was analyzed based on participants' level of experience in video editing. Statistical significance is denoted as * $p<0.05$, ** $p<0.01$, *** $p<0.001$. The figure presents scatter plots that depict the time used by each participant, with different colors representing the order of tasks undertaken in the experiment. 49
4.7 Participants' responses on question 9 and question 10, with mean of each question plot on the end of bar. 50

List of Tables

4.1 Questionnaire used in user study. 47
4.2 Additional question in semi-automatic authoring task's questionnaire. 50

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my family for their unwavering support and for affording me the opportunity to pursue my aspirations. Their encouragement and belief in me have been invaluable.

I would also like to express my sincere appreciation to Professor Minamizawa Kouta for his continuous guidance and support during the entire duration of this research, his expertise and perspicacious insight helped a lot in structuring the research. I am also truly grateful to Visiting Associate Professor Saito Tatsuya for the insightful feedback and encouragement that helps in shaping the development of this study. Furthermore, sincere appreciation to Project Assistant Professor Horie Arata for his invaluable assistance during the later stages of this research. Their support played a crucial role in converging various aspects of the study and providing valuable insights in the design of the experiments.

I would also like to acknowledge the contributions of my peers and the participants who dedicated their time and efforts to this study. Their involvement and valuable insights have been crucial to the successful completion of this research.

Lastly, I would like to express my gratitude to the academic community and all those who have contributed to the field of haptic technology. Their research and advancements have paved the way for innovative discoveries and have served as a source of inspiration throughout my journey.

Thank you to each and every individual who has played a role in the realization of this research. Your contributions and assistance have been invaluable, and I am deeply grateful for your support.

Chapter 1
Introduction

1.1. Enhancing Human-Computer Interaction through Tactile Stimuli

Ever since the introduction of graphical interfaces in computers, tactile stimuli have been employed to enhance user experiences in both computer interaction [1] and multimedia content [2] . In the emerging stage of haptic in human computer interaction, researchers are striving to develop systems that offer feedback to computer users, by reflecting physical sensation based on the inputs from mouse and keyboards [3], the system can increase the intuitive and enjoyment to computer interface [4].

Multimedia content, such as watching a film, becomes more immersive and engaging when users can intuitively interact with it or experience stimuli from the content [5].Physiological studies have also demonstrated the positive impact of cross-modal integration between vision and haptics on tasks involving object interaction, collaborative environments, and target locating [6]. Since visual and haptic inputs provide distinct information, designing a system that cognitively connects these perceptions allows us to convey additional information. This enables the generation of dynamic tactile feedback, including physical attributes of surrounding objects (such as texture and weight), and even precise redirection of the user's visual attention in the scene.

1.1.1 The Ultimate Display

In 1965, Ivan [7] introduce a novel concept 'The Ultimate Display', outlining three essential components for a virtual world in physical world: immersion, interaction, imagination. This concept envisioned a system capable of sensing and tracking

human body's position within physical space, using it as input to control a computer. The displayed objects in computer were not bound by the constraints of physical reality, and users could not only observe these objects but also interact with them as if they were physically present. Thanks to advancements in computer graphics and human-computer interaction, modern VR systems have made significant strides towards achieving the realistic visual and interactive capabilities envisioned by the 'Ultimate Display' concept.

In addition to VR systems, four-dimension (4D) movies offer a captivating experience that combines immersion, interaction, and imagination, exemplifying the successful integration of tactile feedback with visual-audio content [8]. By incorporating elements such as vibration, thermal stimuli, bursts of air, and changes in humidity, recent 4D movies have expanded the viewer's experience, providing users with a heightened sense of realism and immersion, creating a truly engaging experience where they can not only passively observe but actively participate in the narrative, enhancing their overall enjoyment and entertainment value.

1.2. Haptic Authoring

In order to deliver tactile visual-audio content such as 4D movies, it is crucial to synchronize the tactile stimuli with the corresponding visual and auditory cues, including their associated semantics [9]. This alignment between sensory modalities allows for a cohesive and immersive experience, resulting in a more impactful and synchronized multi-sensory presentation.

1.2.1 Manual Haptic Authoring

In the development of haptic content for vibrotactile displays, researchers have dedicated significant efforts to manually annotating haptic feedback for each frame of the video, aiming to create accurate tactile stimuli that synchronize seamlessly with the intended scenes [10–12]. However, this manual editing process is time-consuming and labor-intensive. Haptic designers needs to cycle trough the video-audio content and find the particular moment to add in suitable haptic stimuli. Drawing from my personal experience of manually annotating a video with a monotonous scenario, it took nearly two hours to complete a ten-minute video.

This highlights the significant investment of time and effort involved in the manual haptic authoring process. Supporting this notion, in Li et al. research [13], they indicate that a team of three designers typically spends around 16 days annotating a feature-length 4D film, further emphasizing the demanding nature of the task.

1.2.2 Automatic Haptic Authoring

To address this challenge, researchers have explored semi-automatic [8] or automatic methods for annotating haptic feedback in videos. One common approach is to generate haptic feedback automatically based on the calculation of visual saliency in the scene [14]. However, it is important to note that this approach may not be universally applicable in all scenarios, highlighting one of the limitations of this method. As an example, when considering the generation of tactile feedback, it is important to account for the spatial and contextual aspects of the content. For instance, a moving character positioned closer to the camera may produce stronger tactile sensations compared to an explosion occurring in the background, which is not intuitive for viewers. To tackle this, recent researches in automatic haptic generation take additional elements into consideration, such as audio cues, to ensure more accurate and contextually appropriate tactile feedback for viewers.

1.2.3 Limitation of automatic Haptic Authoring

Despite considerable efforts by researchers in the field of automatic haptic authoring, achieving the same level of accuracy and quality as manual haptic authoring remains challenging. Furthermore, generated haptic feedback may occasionally lack meaning or prove unsuitable in complex scene contexts. Even with the incorporation of audio sources, accurately discerning the intended meaning of the sound presents difficulties. For instance, a booming sound detected by the system may simply be an unintended loud noise from the narrator's microphone. If this sound is translated into haptic sensations, it can cause confusion and misunderstanding for users.

Moreover, the existing research on automatic haptic authoring often fails to generate haptic feedback based on the specific events occurring within a scene. For

example, using the prevalent approach, when two steel balls collide, the calculated saliency may generate haptic feedback based on the trail of the moving ball rather than focusing on the impact that occurs during the crash. As a result, the tactile stimuli produced may differ slightly from what the audience would expect, leading to a potential disparity between the actual experience and the anticipated haptic sensations. As of now, the development of a comprehensive pipeline capable of automatically generating precise tactile feedback for any given video remains an unsolved challenge.

1.3. Involving Event Identification to Automatic Haptic Authoring

As mentioned earlier, the current automatic haptic authoring approach lacks a focus on the events taking place within a scene. However, by leveraging deep learning models to analyze the scene's elements, we can identify the positions of objects of interest and subsequently recognize the actions occurring between these objects. Recent studies by [13, 15] demonstrate the utilization of machine learning models to address visual and audio content separately. By pinpointing the location of the event on the screen, these frameworks generate corresponding haptic feedback on a vibrator array, providing users with a more intuitive haptic experience.

The aforementioned frameworks have demonstrated their ability to identify events within a scene and generate corresponding haptic stimuli. However, these approaches primarily provide haptic feedback from an audience perspective, informing them about the location and intensity of the events on the screen. To further enhance the immersive experience for users, it is crucial to provide haptic stimuli from a first-person perspective, simulating the sensation as if the user themselves were the character experiencing the events within the scene. By adopting this approach, not only can the level of immersion for the audience be significantly enhanced, but it can also foster a greater sense of empathy towards the character in the scene.

1.4. Research Goal

In this study, our objective is to enhance the efficiency of manual haptic authoring by developing a framework that automates the generation of haptic stimuli for specific scenarios based on the events depicted in the video. We aim to provide user with a first-person perspective tactile stimuli, allowing them to experience the events as if they were actively involved in the scene. The resulting output of the system will be a quad channel audio file that encompasses synchronized vibrotactile stimuli aligned with the video content. By utilizing the output audio from our system, haptic authors can benefit from highly accurate, albeit not flawless, haptic annotations. This approach significantly reduces the effort required in the haptic annotation process, as authors will only need to make partial edits to the audio to complete the haptic annotation task effectively.

1.5. Thesis Structure

This paper contains 5 chapter:

- Chapter 1 : This chapter provides an introduction to the background and current state of haptic video authoring, highlighting the existing approaches and their limitations. It sets the stage for the research by identifying the gaps and challenges in the field, and outlines the overall direction that this study aims to pursue.

- Chapter 2 : This chapter serves as an introduction to the field of haptic enhancement in the viewing experience. It explores the current state of research in haptic authoring, discussing various approaches and techniques employed by researchers. Additionally, it provides an overview of deep neural networks, which play a crucial role in this study's methodology.

- Chapter 3 : This chapter introduces the initial prototype that solely relies on object detection for haptic generation. It outlines the iterative process undertaken to develop the final multi-model framework capable of automatically generating haptic feedback for specific scenarios.

- Chapter 4 : This chapter focuses on the evaluation of the automatic haptic generation and its impact on the efficiency of manual haptic authoring tasks. It presents the experiment conducted to assess the effectiveness of the automatic haptic generation approach, also providing an analysis based on the experiment results.

- Chapter 5 : This chapter serves as the conclusion of the research, it also discusses the limitations encountered during the study and presents potential avenues for future research.

Chapter 2
Related Works

2.1. Enhancing Experience with Haptic

Researchers have extensively investigated and demonstrated the effectiveness of haptic feedback in enhancing the overall multimedia experience [10, 16]. When referring to haptic stimuli, it encompasses various tactile sensations, both kinesthetic and cutaneous in nature. These stimuli can range from simple vibrations, commonly employed as notification alerts in mobile devices [17] to more sophisticated systems [18] that provide comprehensive tactile feedback, including thermal stimuli, humidity variations, and changes in air pressure.

2.1.1 Cutaneous Haptic Stimuli

Cutaneous haptic stimuli refer to sensory experiences that are perceived through the skin, it includes a wide range of sensations, including pressure, vibration, texture, temperature, and even pain.

Vibration is a commonly employed method for delivering tactile stimuli. Haptic feedback through vibration can be modulated in terms of intensity, frequency, pattern, and duration to convey a wide range of sensations or information. This type of haptic feedback finds widespread application in our daily lives, including in smartphones, game consoles, and various medical devices. Moreover, researchers have explored the use of vibration haptics to recreate the tactile sensation of different objects, enabling users to perceive textures through haptic feedback [10, 19–21]

Cutaneous haptic stimuli also involve thermal feedback [22, 23] , enabling the simulation of hot or cold sensations within a scene. Furthermore, thermal haptic feedback can also be utilized to elicit the sensation of pain, known as the Thermal

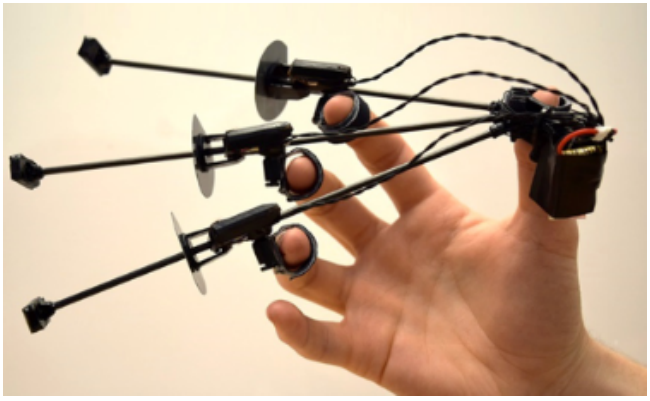
Grill Illusion. This illusion occurs when a specific combination of hot and cold stimuli is provided, and researchers have leveraged this phenomenon to enhance user experiences [24].

Mid-air haptic feedback is another notable technique employed to provide tactile sensations without the need for physical contact with a haptic device. This can be achieved using methods such as air pressure [25, 26], where precise manipulation of air pressure can result in a range of tactile effects.

Another groundbreaking approach involves the use of ultrasonic haptics [27,28], utilizing an array of ultrasonic speakers to create focal points where multiple audio waves superpose, resulting in high-energy tactile feedback at specific locations. Ultrasonic haptic technology has been used to provide intuitive tactile sensations on screens and transparent displays, such as holograms [29] Additionally, research has explored combining mid-air haptics with thermal haptics [25, 30], further expanding the possibilities for immersive tactile experiences.

2.1.2 Kinesthetic Haptic Stimuli

Kinesthetic haptic stimuli refer to the sensory experiences related to movement, position, and forces applied to the body during haptic interactions. It involves the perception and sensation of physical interactions, such as the feeling of resistance, pressure, or force feedback when manipulating objects or performing actions.



(Source: Wolverine, A Wearable Haptic Interface for Grasping in VR [31])

Figure 2.1 Force feedback device that simulate grasping rigid objects in VR

In the context of displaying volumetric data, such as simulating the feeling of a virtual object in virtual reality (VR), force feedback is often utilized. Hand-based haptic devices, designed in the form of gloves, are commonly used to provide users with an intuitive way to interact with objects in the virtual scene [31, 32]. Additionally, there are research efforts exploring larger force feedback systems capable of delivering a sense of displacement to the user [33] .These advancements contribute to a more realistic and immersive haptic experience by enabling users to feel the shape, texture, and physical properties of virtual objects through kinesthetic haptic feedback.

2.2. Haptic in Multimedia Contents

Creating synchronized haptic stimuli is essential in crafting an immersive experience for viewers engaging with audio-visual content. Extensive research has focused on developing efficient approaches to haptic authoring that optimize the user experience.

2.2.1 Haptic Stimuli Delivery Mechanism with Multimedia Content

To ensure precise delivery of haptic sensations, researchers have designed a variety of artifacts to be deployed in conjunction with haptic multimedia content. Cutaneous haptic stimuli are commonly utilized, incorporating vibrotactile actuators placed at different locations to generate diverse tactile effects. Jang et al. [34] provide an interesting idea by employing haptic feedback not only as an output but also as an input mechanism, enabling simple instructions to be conveyed to mobile devices without the need to visually attend to the screen. Swindells et al. [35] demonstrated the utilization of a game controller to display haptic patterns processed by their designed tool. Rahman et al. [12] implemented a rectangular array of actuators within a jacket, enabling different levels of actuation based on the visual content in the video. [13] attached nine haptic actuators to the back of a chair and employed their proposed algorithm to activate the actuators according to event locations within the scene. Kim et al. [10]emphasized the significance of

(Source: Rahman et al. [12])

Figure 2.2 Implementing multiple array of vibrotactile actuators inside a jacket

hands in perceiving haptic stimuli and, accordingly, placed vibrotactile actuators on a glove to provide synchronous vibration stimuli aligned with the film.

These studies demonstrate various strategies for integrating haptic actuators into different physical artifacts, such as jackets, chairs, gloves, etc. By leveraging the specific features of each artifact, researchers have successfully synchronized haptic feedback with multimedia content, enhancing the user's sensory experience.

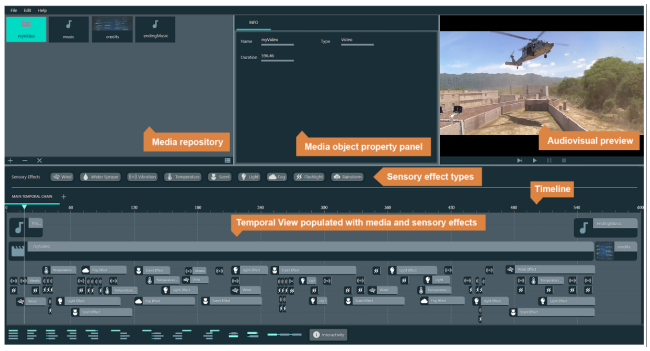
2.3. Authoring Methods

The creation of multimedia content with haptic feedback requires the processing of video data and the activation of vibrotactile actuators in synchronization with the visual scenes. Alternatively, haptic audio can be incorporated by adding it to a separate audio channel, which can be played through specialized speakers or haptic actuators that retrieve data from that channel. Researchers [36]have explored various approaches to authoring haptic audiovisual content, including three main methods: manual authoring [37], sensor based authoring [38] and automatically authoring [15].

2.3.1 Manual Haptic Authoring

Manual authoring is the most common and straightforward haptic authoring method, wherein haptic designers annotate the audiovisual content frame by frame. They add audio that represents different haptic stimuli to the audio track, akin to video editing. This approach enables the creation of high-quality haptic

videos, where the haptic stimuli align intuitively with the audience's perception. By employing manual authoring techniques, haptic designers have the ability to finely tune and synchronize the haptic and audio elements with the visual content, resulting in a compelling and immersive multimedia experience for audience.



(Source: Abreu et al. [39])

Figure 2.3 A novel manual authoring tool with automatic sensory effect recognition

Researches have been done in proposing various efficient haptic authoring tools that can escalate the efficiency of manual authoring [11] Daneieau et al. [40] introduce a manual haptic authoring tool that can easily edit motion effects with haptic, a simplified graphic user interface which can review motion effect allows non expert user to author haptic video with ease. Abreu et al. [39] propose a novel method that integrates automatic sensory effect recognition into authoring tool. By utilizing deep neural network to identify sensory effects and hence allows the whole authoring process to be semi-automatic, greatly increase the efficiency in authoring multimedia content with multiple sensory efects.

2.3.2 Authoring with Sensor

Sensor-based haptic authoring involves capturing haptic motion directly from physical objects using sensors. Accelerometers and other sensors have been utilized to capture motion data and contact forces in various applications [41, 42].

In one scenario, this method can be employed during the filming process by placing sensors at appropriate locations on an actor. This allows for the direct

capture of the actor's motion and vibrations experienced while performing, which can then be recorded and replayed alongside the film. By incorporating this recorded haptic sensation, an intuitive haptic feedback can be provided to the audience, enhancing the overall immersive experience [43].

Another application of this method involves recording the actual haptic sensations of a target object and processing that specific haptic audio during manual authoring. This approach further heightens the realism of the haptic stimuli, resulting in a more authentic sensory experience.

By utilizing sensor-based haptic authoring techniques, researchers and creators can capture and reproduce realistic haptic feedback, leading to enhanced audience engagement and immersion.

2.3.3 Automatic Authoring

Automatic haptic authoring involves the use of systems or frameworks that extract specific features from audiovisual content and generate corresponding haptic feedback. One common approach for automatic haptic feedback generation is by calculating a saliency heatmap based on video analysis [44]. Saliency refers to the identification of moving pixels between frames, and a fast pixel-level adapting background detection algorithm is typically employed to identify the moving elements within the visual image [45].

More recent advancements leverage deep learning techniques to enhance saliency calculation from visual content. Wang et al. [46] propose a method that separately computes static and dynamic saliency using convolutional networks, resulting in more accurate saliency maps with reduced computational load.

In this approach, the motion of objects plays a crucial role in generating haptic feedback, with fast and large movements resulting in stronger haptic sensations, while slower or smaller movements produce milder feedback. However, it is important to note that the haptic stimuli derived solely from object motion may not be appropriate for every scene in an audiovisual context. Consequently, researchers have conducted further investigations to explore alternative factors within the video that can contribute to haptic generation. By considering additional elements in the video, such as audio information [13], more comprehensive and contextually relevant haptic feedback can be achieved.

Enhancing Automatic Haptic Authoring with Audio Integration

Multimedia content typically includes both visual and audio components. By separating the visual content from the audio content, it becomes possible to leverage algorithms that convert the audio into haptic feedback. Zhang et al. [15] take this approach further by categorizing the audio into two types: diegetic and non-diegetic audio. Diegetic audio refers to sounds produced by elements within the visual scene, while non-diegetic audio originates from sources that are not visible, such as a narrator's voice.

To extract haptic information from the diegetic audio, Zhang et al.adopt the approach proposed by Tian et al. [47] for sound source localization. They enhance Tian et al.'s model by incorporating a combination of VGG-19 and VGG-like networks to process the visual features corresponding to each audio frame. This modification allows them to generate a sound localization heatmap that identifies the source of the sound within each frame of the audiovisual content. By incorporating audio information into the haptic feedback generation process, they succeed in automatically generating haptic stimuli precisely by audiovisual content.

Automatic haptic authoring techniques enable the generation of haptic feedback in an automated manner, leveraging computer vision and deep learning algorithms to identify salient visual and audio features. These advancements contribute to the development of efficient and effective systems for creating immersive haptic experiences aligned with audiovisual content.

2.4. Deep Neural Network in Object Detection

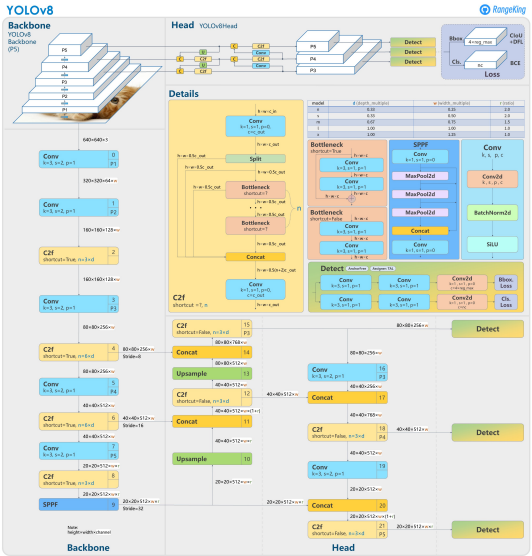
Object detection is a fundamental task in computer vision, aimed at identifying objects within an image and providing bounding box coordinates around the target objects, along with their corresponding class labels. This task extends the concept of object classification and has witnessed significant advancements with the introduction of convolutional neural networks (CNNs) and deep learning techniques.

Existing object detectors can be categorized into three main types: single-stage, two-stage, and transformer-based detectors. Each type employs different architectural designs and strategies to achieve accurate and efficient object detection.

State-of-the-art models exemplifying these types include YOLO (You Only Look Once) [48], Faster R-CNN(Region-based Convolutional Neural Networks) [49], and DeTR(Detection Transformers) [50]. These models have made significant contributions to the field by achieving remarkable performance on benchmark datasets such as Microsoft-COCO [51] and ILSVRC(ImageNet Large Scale Visual Recognition Challenge) [52].

YOLO

YOLO [48], a single-stage detector, stands out for its real-time object detection capabilities, the model is inspired from GoogleNet model which was initially used for image classification. After dividing the input image into grid, YOLO predicts the bounding boxes and class probabilities for each of the grid cell. With this approach, YOLO achieves impressive speed while maintaining reasonable accuracy. This makes it well-suited for applications requiring fast and efficient object detection, such as video analysis and robotics.



(Model Architecture drawn by Github user RangeKing [53])

Figure 2.4 Overview Architecture of YOLOv8 Model

Since the initial development of the YOLO (You Only Look Once) framework

by Joseph Redmon et al., researchers have continued to refine and enhance the YOLO architecture. One notable adaptation is YOLOv2 [54], which introduced significant changes to the backbone architecture. Instead of using the GoogleNet model [55], YOLOv2 employed the DarkNet-19 architecture [56]. This modification allowed YOLOv2 to detect a much larger number of object classes, reaching up to 9000, while simultaneously improving flexibility, speed, and accuracy.

Subsequently, YOLOv4 [57] and YOLOv5 [58] were introduced with notable advancements. YOLOv4 introduced the concept of "bag of freebies" to improve performance without sacrificing inference time. It incorporated various techniques such as self-adversarial training, class label smoothing, and CmBN (Cross mini-Batch Normalization) to achieve significant performance improvements. YOLOv5, while not formally published as a peer-reviewed research, has demonstrated effective results. It adopted several concepts and approaches from YOLOv4, including the new ability to learn the anchor box during the training process.

The most recent iteration of the YOLO model is YOLOv8 [59]. which builds upon the advancements of YOLOv5 with several modifications and improvements. YOLOv8 introduces significant changes to enhance the flexibility and accuracy of the model.

One notable modification in YOLOv8 is the removal of the anchor box approach. The use of predefined anchor boxes in YOLOv5 was believed to limit the model's flexibility, as not every object can be perfectly enclosed by a polygon anchor box. By removing this constraint, YOLOv8 allows for more precise object detection across a wider range of shapes and sizes. Additionally, YOLOv8 incorporates mosaic data augmentation into the model. This approach involves randomly combining multiple input images to form a larger composite image. By training on these mosaic images, the model learns to detect objects that may appear in different positions within an image, enhancing its ability to generalize and handle object variations.

These modifications in YOLOv8 demonstrate a continuous effort to improve the flexibility, adaptability, and detection performance of the YOLO model, pushing the boundaries of real-time object detection capabilities.

2.5. Anomalous Sound Event Detection

The auditory system is a vital sensory function that humans utilize to perceive and extract information from their surroundings. In line with this, researchers have sought to equip machines with similar capabilities by leveraging algorithms and deep learning techniques. Sound Event Detection (SED) emerges as a prominent research area focusing on enabling machines to detect and identify specific sound events. The primary objective of a Sound Event Detection (SED) system is to precisely identify the onset and offset of specific sound events within an audio signal. This approach offers several advantages when compared to solely relying on visual-based object detection methods [60].

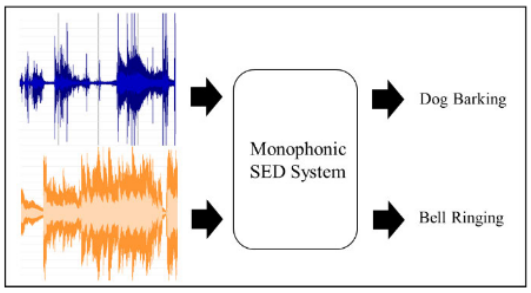
Certain events, such as a car horn honking or a doorbell ringing, are not easily identifiable through visual cues alone. By incorporating audio analysis through SED, machines can effectively recognize and classify these sound events, complementing the information obtained from visual data. Additionally, leveraging audio processing for event detection can be computationally efficient. Analyzing audio data requires fewer computational resources compared to processing visual content, making it a favorable choice for scenarios where computational limitations exist. By exploiting the benefits of audio-based detection, SED systems can provide valuable insights and enhance the overall performance of multimodal perception tasks.

2.5.1 Monophonic and Polyphonic SED System

Monophonic SED

Monophonic SED refers to the task of detecting and classifying individual sound events in a monophonic audio signal. In audio, monophonic signals refer to those containing only a single audio source or sound event at a given time. This means that the input audio for a monophonic SED system should not contain overlapping or concurrent sounds.

The overall detecting task involves analyzing the temporal and spectral characteristics of the audio signal to detect and classify specific sound events of interest. Monophonic SED is commonly applied in various audio processing applications, such as music transcription, speech recognition, and acoustic scene analysis.



(Source: Chan et al. [60])

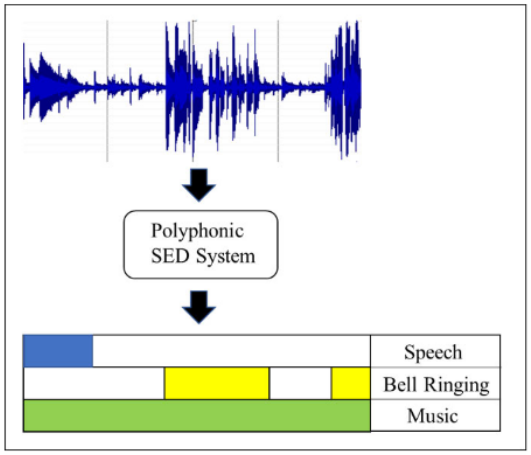
Figure 2.5 Illustration of Monophonic SED system, only one event can be identify within a single audio input

Polyphonic SED

Polyphonic Sound Event Detection (SED) systems are specifically designed to identify and classify multiple sound events within an audio signal that contains overlapping sources of sound. Unlike monophonic signals, which have only one sound source at a given time, polyphonic signals are characterized by the presence of multiple simultaneous sound sources [61, 62].

Designing an effective polyphonic SED system is challenging due to various factors. Firstly, multiple sound events can overlap in time, making it difficult to isolate and distinguish individual sounds. Secondly, sound events with the same label can exhibit significant variations in their acoustic characteristics. For example, different bird species may produce distinct chirping sounds, even though they share the same label of "bird". Additionally, background noise present in the audio further complicates the task of identifying and classifying sound events. The noise can mask or interfere with the target events, reducing the signal-to-noise ratio and affecting the system's performance.

To address the challenges of polyphonic Sound Event Detection (SED), researchers have explored various approaches to develop robust systems. One notable approach is the use of Hidden Markov Models (HMMs) [63], which have been widely used in Automatic Speech Recognition (ASR) tasks. HMMs, implemented with the Expectation-Maximization (EM) algorithm and Gaussian Mixture Models (GMMs) [64], enable the modeling of probability distributions for each sound



(Source: Chan et al. [60])

Figure 2.6 Illustration of Polyphonic SED system, different color tags refers to the onset and offset of a particular event

event. This non-neural network based approach has demonstrated effectiveness in polyphonic SED systems.

With the advancements in deep neural networks, researchers have also leveraged their capabilities in designing SED architectures. One popular framework is the Convolutional Recurrent Neural Network (CRNN), which directly connects convolutional layers to recurrent neural networks. CRNNs excel at learning audio spectrograms that capture both temporal and spectral information. For instance, Cakir et al. [62] proposed a polyphonic SED framework that incorporated Gated Recurrent Units (GRUs) into the CRNN architecture, achieving promising results on the TUT-SED 2016 dataset.

More recent research in polyphonic SED has explored the use of teacher-student frameworks and the integration of weakly labeled data. In one study by Lin et al. [65], a teacher model and a student model were trained by learning from unlabeled data with tags generated by each other. This iterative process allowed the student model to progressively refine the teacher model's performance under the guidance of the teacher model. The proposed approach, described in the study, yielded impressive results on the DCASE 2018 evaluation dataset.

By combining traditional models such as HMMs, advanced deep neural network

architectures like CRNNs, and innovative frameworks like the teacher-student model, polyphonic SED models are achieving better result in various applications, including audio surveillance, acoustic scene analysis, and audio content classification.

2.6. Summary

Through the topics discuss above, we notice that manual authoring of audiovisual content has the potential to create the most captivating and immersive haptic experience in conjunction with the content itself. However, this manual approach is often time-consuming and labor-intensive, posing challenges in terms of efficiency and scalability. As a result, researchers are actively exploring avenues to automatically generate synchronized haptic feedback that enhances the overall audience experience. By leveraging automated techniques, the aim is to achieve a seamless integration of haptic sensations with video content, ultimately providing a more engaging and immersive multimedia experience for viewers.

The current focus of automatic haptic generating frameworks is primarily on generating haptic stimuli corresponding to the location of objects in the scene. However, I argue that this approach, which generates haptic feedback from a third-person perspective, may not provide the most intuitive haptic experience for the audience. For instance, consider the example of two colliding balls. Rather than providing the haptic stimuli based on the balls position and trails, it would be a more immersive experience to provide haptic feedback that simulates the impact when the balls collide, generating haptic stimuli from a first-person perspective as if the audience themselves were in the scene.

While discussing the design of automatic haptic generating frameworks, preceding researches focuses on the visual aspects. Researchers often identify the saliency heatmap in the scene, which indicates the positions of moving objects and provides haptic stimuli based on this information. With the emergence of multimodal approaches, the audio data in multimedia content is also being considered. Researchers adjust the intensity of haptic stimuli based on the audio volume or set thresholds to activate specific haptic feedback. In a few cases, researchers utilize Sound Event Detection (SED) techniques to identify the onset

and offset of events occurring in the scene and generate haptic feedback based on this information.

While there is still considerable progress to be made in the development of a fully automated framework for generating haptic stimuli in audiovisual content, the integration of visual and auditory cues holds great promise for future advancements. By incorporating both sensory modalities, future automatic haptic generating frameworks have the potential to significantly enhance the overall immersive experience for users.

Chapter 3
Concept Design

3.1. Research Objective

Building upon the research discussed in the previous section, this study focuses on the development of an automatic multi-model system for annotating haptic stimuli in audiovisual content. Unlike existing approaches that primarily rely on location of moving object in the scene, the proposed system places emphasis on the timing of events occurring in the scene and provides corresponding tactile feedback to enhance the immersive experience for the audience.

Acknowledging the current limitations in achieving perfect automatic annotation, the aim of this research is to attain an accuracy of at least 60 percent in annotating the onset of events within a 200ms difference compared to manual authoring. The output of the system will be an .mp4 file with haptic audio annotations on a separate channel. This output can be directly utilized by haptic authors or designers, enabling them to efficiently edit the video file and produce a more accurate haptic-enhanced video without the need for starting the annotation process from scratch.

In summary, the primary objective of this study is to automate the generation of haptic audio for videos based on events occurring within the scene. The resulting haptic audio output can seamlessly integrate into manual authoring tools, enhancing efficiency and reducing the labor-intensive nature of the manual annotation process.

3.2. Initial system design

The ultimate objective of this research is to develop a multi-model framework that leverages both visual and audio data to generate haptic feedback based on events within the content. While there have been limited studies exploring this approach, the initial strategy for tackling this task involves employing object detection methods to identify objects in the scene and utilizing a classifier model to determine if objects are in contact with one another.

Choice of Object Detection Model

During the time of this research, the most recent addition to the YOLO family was YOLOv7 [66]. Noteworthy advancements were made in YOLOv7 compared to its predecessor, YOLOv4 [57]. Firstly, YOLOv7 employed the COCO dataset [51] for training its backbone model instead of ImageNet [52]. In terms of computational blocks, Wang et al. introduced the Extended Efficient Layer Aggregation Network (E-ELAN) architecture, which allowed for continuous learning through operations like shuffling, expansion, and merging while preserving the original gradient path.

Additionally, YOLOv7 incorporated a Bag of Freebies (BoF) method known as re-parameterization. The concept of Bag of Freebies refers to techniques that enhance a model's performance or efficiency without incurring significant additional costs in terms of training iterations or computational resources. In the re-parameterization method, multiple models are trained using different input training data but with the same settings. The final model is then obtained as an average of these models.

3.2.1 Training Object Detection Model

To begin the research, an object detection model is trained using the chosen framework, YOLOv7. This framework is selected due to its recent advancements and suitability for the task. In order to evaluate the performance of the model, a specific scenario of sword fighting in a movie scene is selected. A total of 846 images, including scenes featuring sword fighting and close-up shots of swords, are manually labeled and utilized as the training data for the object detection model.

To expedite the training process and enhance the model's performance, transfer learning is employed by initializing the model with the official yolov7.pt weights. This approach allows the model to leverage pre-existing knowledge and accelerate the learning process. The training process is executed for a total of 200 epochs, with a batch size of 8.

Result of Object Detection

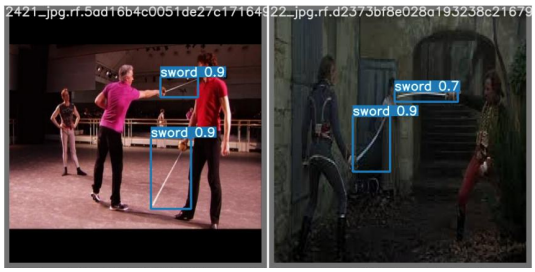


Figure 3.1 Prediction from the trained object detection model on testing image

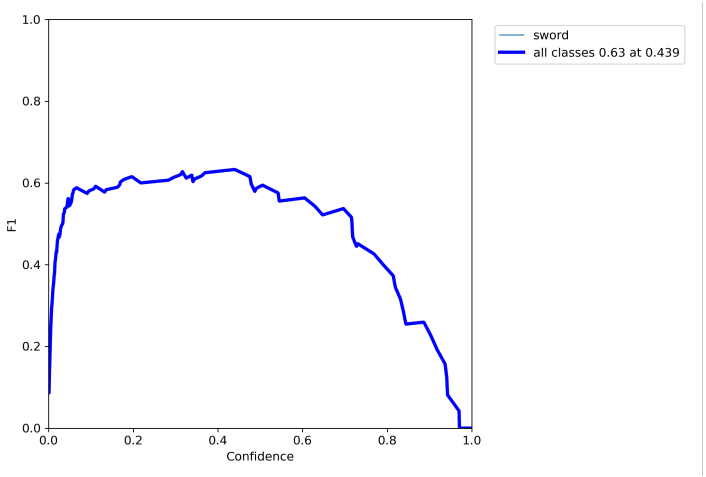


Figure 3.2 F1 score indicating model accuracy in predicting both positive and negative prediction on sword.

After analyzing the testing data, the best F1 score obtained for detecting swords in the scene is 0.63, with a confidence rate of 0.439. Surprisingly, the F1 score

remains consistently above 0.5 even at a confidence level of 0.6, which exceeds initial expectations considering the diverse nature of sword variations.

Moving forward, the next objective is to identify instances where two swords come into contact with each other. Ideally, the system should be capable of directly outputting frames capturing these contact moments. To achieve this, the first step involves locating the positions of the bounding boxes around the swords and cropping them accordingly.

To accomplish this, a white mask is created with the same dimensions as the video. The swords are then extracted based on their respective bounding boxes and pasted onto the white mask. This process generates frames containing only the detected swords, eliminating potential interference from the background environment, which may introduce noise in detecting the contact events.

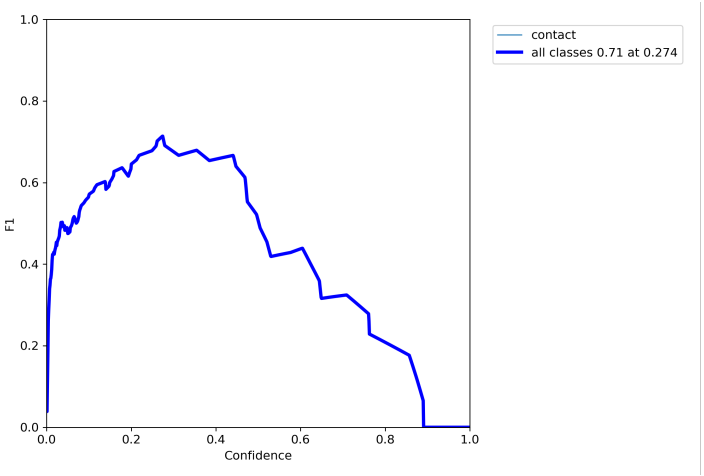


Figure 3.3 F1 score indicating classification model accuracy in predicting the contact event.

Using the cropped sword images with the white background, a classification model is trained based on the YOLOv7 framework, specifically leveraging the pretrained resnet101.pt weights. The training task is same as the object detection model with 200 epochs and batch size of 8. This model's primary task is to determine whether the swords in an image are in contact with each other. If contact is detected, it signifies that the swords in the scene might have collided, indicating the timing for providing haptic feedback.

Result of Classification Model

The classification model for detecting sword contact achieved a best F1 score of 0.71 with a confidence rate of 0.274, and the F1 score remained above 0.6 until a confidence threshold of 0.4. Although the results did not meet the initial expectations, the classification model was still integrated with the object detection model to automate the identification of frames with sword contact events.

To evaluate the overall framework, three commercial videos, each approximately 20 seconds long, were used as input. Figure 3.4 provides an illustration of the process and displays a single-frame result from one of the videos. Based on visual estimation, the accuracy of detecting frames with sword contact events across the three videos is approximately 60 percent in the best-case scenario.

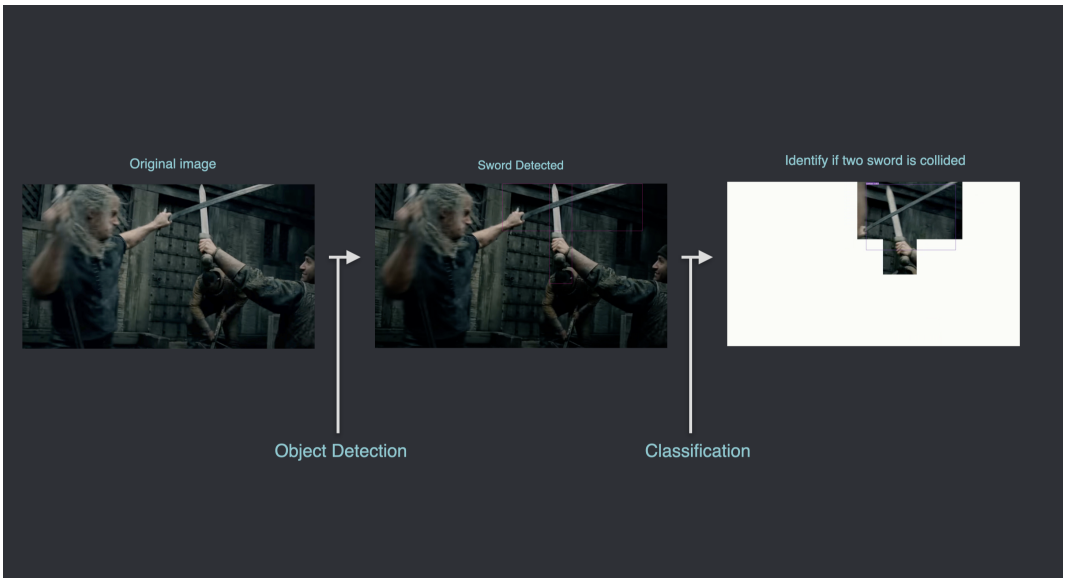


Figure 3.4 Process of the system in finding the contact events in a single video frame.

However, it should be noted that the accuracy is highly dependent on the camera angle. The framework encounters challenges when the camera is positioned behind the main character, obstructing most of the swords and making it difficult for the object detection model to detect them accurately. Additionally, there are instances where white pillars or random white cylinder-shaped objects are

mistaken by the object detection model as swords. This can lead to false positives, where the pillars are detected as contact events, impacting the classification model's accuracy.

These challenges emphasize the significance of considering different camera angles and enhancing the object detection model's capability to differentiate between swords and other objects within the scene. However, upon further reflection, it becomes apparent that finding sword fighting scenes that consistently maintain a perspective enabling clear visual detection of the swords may prove challenging. Consequently, it would be more beneficial to identify scenarios where the camera angles are predominantly fixed, ensuring the system's evaluation is conducted under more controlled conditions. By selecting such scenarios, the accuracy and reliability of the framework can be better assessed and any necessary improvements can be made accordingly.

Different Scenario with Similar Approach

After careful consideration, I have selected the scenario of a badminton match for the second phase of the trial. In most badminton matches, the camera angle remains fixed during rallies, providing an ideal opportunity to capture events where the racket makes contact with the shuttlecock. This scenario allows for the creation of haptic feedback that mimics the experience of being the player hitting the shuttlecock.

To train the object detection model, a dataset of 1120 images containing rackets and shuttlecocks is collected. The model follows a transfer learning approach, utilizing the yolov7.pt weights. However, this time the model is trained to detect two labels: the racket and the badminton shuttlecock. The output from the object detection model is serve as the input for the classification model, which determines whether the racket and shuttlecock have made contact with each other during the match.

The training results for the new object detection model are shown in Figure 3.5, indicating better performance in detecting rackets compared to badminton shuttlecocks. The average F1 score for these two classes is 0.71 with a confidence of 0.361, and an average F1 score above 0.6 is maintained for confidence levels above 0.8. Encouraged by these results, I connected the object detection model to

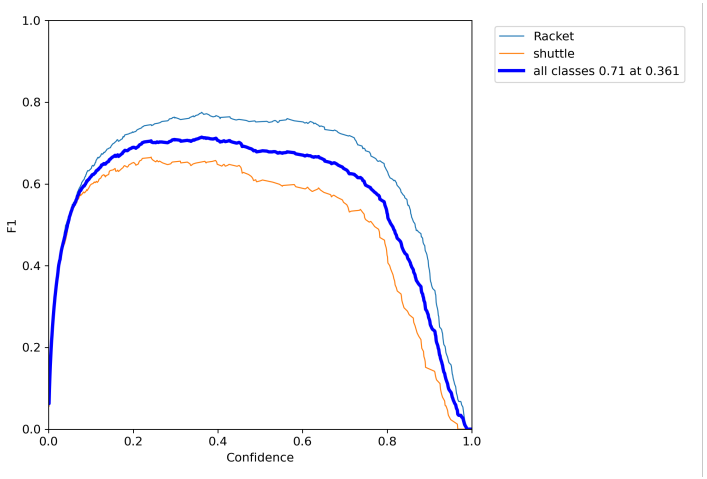


Figure 3.5 F1 score for object detection for racket and shuttle ball.

the classification model and tested it on a 24-second badminton rally video that was not included in the training data.

To my surprise, the output from the classification model was unsatisfactory, indicating that it was not functioning properly. This prompted me to investigate the issue further. After several rounds of trial and error, I discovered that during the frames when the racket and shuttlecock make contact, the shuttlecock is often not visible in the field of view. This is likely due to the high speed at which the shuttlecock moves when struck by the racket. Consequently, the approach of determining contact between the two objects would not work effectively since the model would frequently fail to identify the shuttlecock during these contact events.

After facing disappointing results in both scenarios, I came to the realization that directly inputting raw videos into the machine learning model may not be as effective as I initially thought. Despite the challenges encountered, I still believe that the badminton scenario holds promise for my research. It offers the advantage of easily obtaining relevant training materials, and sports-related activities resonate with people's daily lives. Additionally, many individuals have personal experience playing sports like badminton, making it more relatable and intuitive for them to feel the haptic sensations when watching a badminton rally. Therefore, I have decided to maintain the badminton scenario while exploring

alternative approaches to designing an automatic haptic generating system.

3.3. Second Prototype

After encountering limitations in the initial system design, I conducted an extensive literature review to explore alternative approaches that could be more suitable for my research. Given that the scenario focuses on badminton rally scenes, one of the key challenges is detecting the disappearing badminton shuttle when a racket contact occurs. Through extensive experimentation with over 30 badminton rally videos, I have observed that the badminton shuttle consistently disappears upon contact with the racket. Based on this observation, my hypothesis is that by developing an object detection model that can accurately track the badminton shuttle, when the model lost track of the shuttle ball, I can then identify the frames in which the impact occurs.

3.3.1 Elevating Object Detection Performance

Consequently, my first task is to enhance the object detection model's ability to identify the badminton shuttle. During the development of the second prototype, YOLOv8 [59] was released, offering improvements over YOLOv7 and a more user-friendly interface. Therefore, I decided to train the backbone of my model using YOLOv8. Building upon my previous experience, I realized that directly inputting raw videos into the model would not yield optimal results. To address this, I opted to preprocess the badminton rally videos using image processing techniques.

Video Preprocessing

Firstly, the videos are converted to grayscale, after which an algorithm is applied to calculate the motion saliency of the video frames. The saliency calculation algorithm is adapted from the framework proposed in [45], as frame differencing is applied to identify areas where significant changes or motion occur between frames, the output from motion saliency will focus on the changed pixel across frames and motion energy or magnitute of motion change will result in higher saliency value. This approach benefits in reducing background noises while most

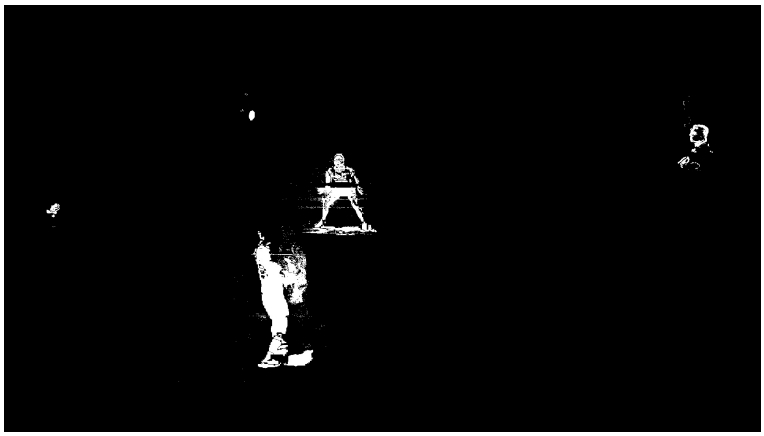


Figure 3.6 Illustrate a single frame output from motion saliency algorithm, also used as training data for object detection model.

importantly, fast moving badminton shuttle that is barely seen by visual will have a strong magnitute in saliency and result in the visability in saliency image.

By leveraging the output frames generated by the motion saliency algorithm, I can establish an automatic haptic generation system similar to the one described in Kim et al.'s research [44] where haptic stimuli are created based on the positions of salient regions in the frame. However, since the ultimate objective of this research is to recreate the sensation of impact when a player strikes the badminton shuttle, the output of the motion saliency algorithm is subsequently fed into an object detection model to track the position of the badminton shuttle in each frame.

Object Detection Training and Result

For training the object detection model, I employ transfer learning using the official YOLOv8 weight file, yolov8n.pt. A total of 2464 saliency images, cropped from six different badminton matches and resembling the example shown in Figure 3.6, are manually labeled and used as training data for the model. The training iterates for 100 epcochs with batch size of 16, and the results of F1 score to confidence is shown in Figure 3.7. This approach of detecting badminton shuttle ball through motion saliency got really good result which maintains a F1 score higher than 0.85 and peaks with 0.89 at confidence 0.351.

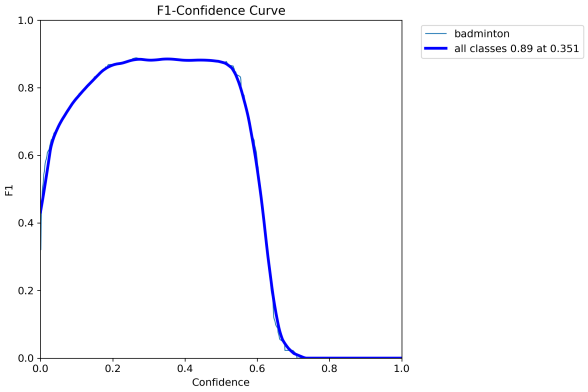


Figure 3.7 F1 score for object detection for racket and shuttle ball.

I then deploy this model to detect a testing video of badminton rally while setting the confidence threshold to be above 0.5. The testing video is also labelled manually by me each frame to draw a bounding box on the badminton to make an evaluation on how accurate the model can get on detecting the badminton shuttle.

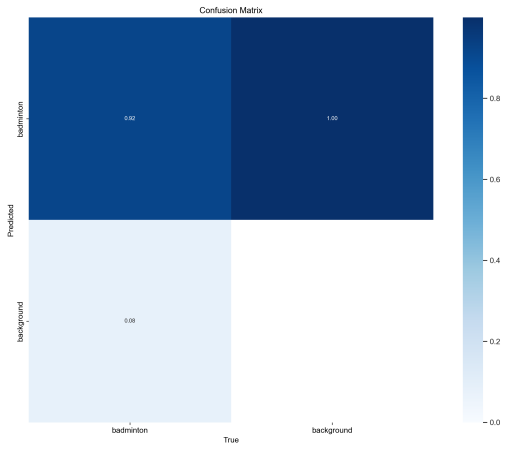


Figure 3.8 Confusion Matrix for model accuracy on testing video

The results of the prediction on the testing video are shown in Figure 3.8 using a confusion matrix. Upon initial observation, the confusion matrix may appear peculiar with a false positive rate of 100 percent. However, this is attributed

to the model only making predictions for a single class, namely the badminton shuttle. Consequently, the model does not predict the background class, resulting in no false negatives but only false positives. Regarding the prediction outcomes, the confusion matrix indicates that the model achieved a 92 percent accuracy in predicting the ground truth labels that were manually annotated. The accuracy of the model got a significant improvement compared to the previous approach that utilized raw video with an object detection model trained with YOLOv7. This increased accuracy demonstrates the effectiveness of the current model. Given these positive results, it is reasonable to proceed to the next step, which involves detecting the contact event.

3.3.2 Identifying Contact Event

In order to identify the frames where the badminton shuttle makes contact, an algorithm was devised that calculated the middle point of the bounding box generated by the object detection model. The algorithm aimed to find the first frame where three consecutive frames did not detect the presence of the badminton shuttle. However, the developed algorithm did not yield the anticipated results despite formulating the hypothesis stated in the beginning of this section. Upon evaluation, I discovered that the quantity of output frames exceeded the actual number of strikes in the badminton rally. To investigate further, I cropped all the output frames produced by the algorithm and identified several factors that I had overlooked.

Factors Leading to Disappointing Results

After observation to the output frames, several issue is being found. Firstly, the camera angle used during the recording was not directly positioned above the players, as depicted in Figure 3.6 Consequently, the badminton shuttle was heavily influenced by the depth of the scene. During movements like a net drop, there was a high probability of the player obstructing the vision of the badminton shuttle, causing the model to lose track of it.

Another significant factor was the player's execution of a high clear, with an example shown in Figure 3.9, where the shuttle is lifted high into the air with an

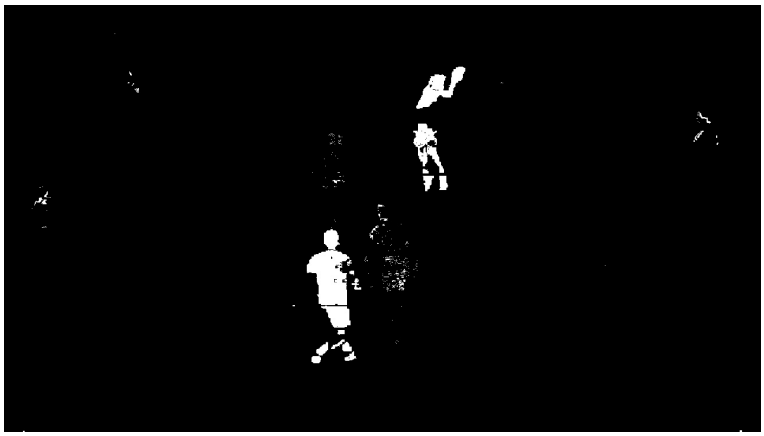


Figure 3.9 Two frames prior to the smash being performed, the badminton shuttle is currently positioned out of camera's boundary.

upward stroke. In such instances, the shuttle ball would also disappear from the scene due to its altitude exceeding the boundaries of the camera's field of view. These factors contributed to the increased number of output frames, which did not align with the expected number of strikes in the badminton rally.

3.3.3 Improving Algorithm for Event Detection

Based on the previous testing, it became evident that solely relying on identifying frames where the badminton shuttle disappears did not exclusively capture the instances of racket-shuttle ball contact. Instead, it includes various scenarios where the ball was missing from the scene. Therefore, further enhancements were necessary to refine the algorithm and isolate the frames with racket-shuttle ball interactions.

To achieve this, additional functionality was introduced to calculate the moving direction of the badminton shuttle. Alongside tracking the position of the shuttle, the algorithm now determines the number of pixels the ball moves in both the horizontal and vertical directions over a span of two frames. If the ball consistently moves in a single direction for more than five frames, the moving direction of shuttle is recorded in a dataframe for further analysis. A change in the shuttle ball's moving direction from consistently upwards to consistently downwards

Frame	Blank	Center_X	Center_Y	Sec	Direction_X	ShiftedDirection_X	Change_Point_X	Direction_Y	ShiftedDirection_Y	Change_Point_Y
120	Badminton	826	237.5	4	Right	Right		Up	Up	
121	Badminton	839	263.5	4.0333333...	Right	Right		Up	Up	
122	Badminton	852	288	4.0666666...	Right	Right		Up	Up	
123	Badminton	852	288.5	4.1	Right	Right		Up	Up	
124	Badminton	861	316	4.1333333...	Right	Right		Up	Up	
125	Badminton	871	341.5	4.1666666...	Right	Right		Up	Up	
126	Badminton	879.5	367.5	4.2	Right	Right		Up	Up	
127	Badminton	886.5	392	4.2333333...	Right	Right		Up	Up	
128	Badminton	893	414.5	4.2666666...	Right	Right		Up	Up	
129	Badminton	893	414.5	4.3	Right	Right		Up	Up	
130	Badminton	899.5	441.5	4.3333333...	Right	Right		Up	Up	
131	Badminton	897.5	430	4.3666666...	Right	Right		Up	Up	
132	Badminton	893	383	4.4	Right	Right		Down	Up	
133	Badminton	882	353	4.4333333...	Left	Right	Changed	Down	Down	Changed
134	Badminton	878.5	327	4.4666666...	Left	Left		Down	Down	
135	Badminton	878.5	327.5	4.5	Left	Left		Down	Down	
136	Badminton	874.5	304	4.5333333...	Left	Left		Down	Down	
137	Badminton	869.5	288.5	4.5666666...	Left	Left		Down	Down	
138	Badminton	864	270	4.6	Left	Left		Down	Down	
139	Badminton	860.5	258	4.6333333...	Left	Left		Down	Down	
140	Badminton	858	248.5	4.6666666...	Left	Left		Down	Down	
141	Badminton	858	248.5	4.7	Left	Left		Down	Down	
142	Badminton	856	242	4.7333333...	Left	Left		Down	Down	
143	Badminton	853.5	236.5	4.7666666...	Left	Left		Down	Down	

Figure 3.10 Dataframe including information about the movement direction and changes in direction of shuttle ball.

indicates either the ball moving out of the camera's boundary and re-entering or the player striking the ball and altering its trajectory. Similarly, in the horizontal direction, a shift from consistent left to consistent right and vice versa suggests a high likelihood of a player hitting the ball but not out of camera's boundary. By combining this directional information with the number of pixels the shuttle moves across each frame, it becomes easier to determine the factors contributing to changes in the shuttle ball's trajectory.

As observed in previous sections, the badminton shuttle is not visible at the moment of contact. Therefore, the final determination of racket-shuttle ball contact is made based on the frame before a significant change in both horizontal and vertical direction. In this context, a significant change is defined as movement exceeding 50 pixels per frame. This value was determined through multiple tests, and it was found to be the most suitable threshold across various badminton matches. Figure 3.10 provides an example, illustrating the dataframe containing information about the shuttle ball's motion state for each frame. It can be observed that in frame 133, there is a change in both the vertical and horizontal

directions, indicating that the badminton shuttle was struck by a player's racket at that moment.

To assess the accuracy of the algorithm's output frames corresponding to racket-shuttle interactions in the badminton rally video, a manual evaluation was conducted. The results revealed an accuracy of 12 out of 23 contacts and 9 out of 16 contacts in two separate testing videos, respectively. This indicates an accuracy rate close to 50 percent. Although there is room for improvement, the overall outcome meets my expectations.

3.4. System Improvement with Audio Detection

To further enhance the system's performance, I aimed to transform it into a multi-model system by incorporating audio data as an additional input. In addition to object detection for tracking the badminton shuttle in the visual domain, I explored the utilization of audio data to identify the moments when the sound of the player striking the shuttle occurs. To process the audio data effectively, I decided to employ the Sound Event Detection (SED) method.

3.4.1 First SED model

During my extensive literature review, I discovered the DESED dataset [67], which is specifically designed for recognizing sound event classes in domestic environments. Among the various approaches evaluated on this dataset, Nam et al. [68] frequency dynamic convolution technique yielded the most impressive performance. This approach adopts a teacher-student framework during model training and leverages both weakly labeled and unlabeled audio data as inputs. Additionally, it introduces a novel method known as frequency dynamic convolution, which employs frequency-adaptive kernels to enable 2D convolution along the frequency axis. This approach differs from the conventional technique of employing shift-invariance on mel spectrograms.

Based on the impressive performance demonstrated in the DESED benchmark, achieving an F1 score of 54, I made the decision to utilize Nam et al.'s framework for training my own sound event detection model, specifically focusing on detecting the onset and offset of badminton sounds.

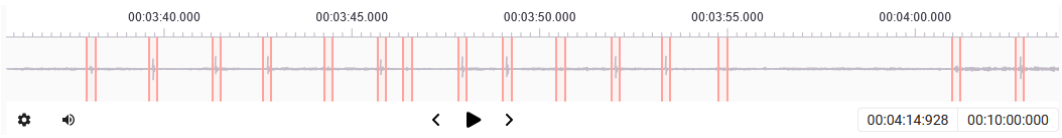


Figure 3.11 Audio data manually annotated with Label Studio [1]

To facilitate the training process, I prepared six training audio files, each with a duration of 10 minutes. These audio files were obtained from six distinct badminton matches and were manually annotated by myself. During the annotation process, events indicating a hit were identified and labeled with a fixed duration of 250 milliseconds, as illustrated in Figure 3.11. This set of labeled data, referred to as strong labels, serves as the ground truth for the onset and offset times of the events and is used for training the deep learning model. The annotations will be output into a .csv file as Figure 3.12 shows, including the onset and offset of an event as well as the filename for the audio.

	A	B	C	D
1	Event	Start	End	File
2	Hit	4.475279	4.788228	Match 1
3	Hit	6.064627	6.582185	Match 1
4	Hit	7.311822	7.668512	Match 1
5	Hit	8.993359	9.304189	Match 1
6	Hit	10.45056	10.67222	Match 1
7	Hit	11.70211	12.0537	Match 1
8	Hit	13.39574	13.66835	Match 1
9	Hit	16.06639	16.37467	Match 1
10	Hit	14.8514	14.99407	Match 1

Figure 3.12 Strong labelled audio data in the form of .csv file

In addition to the strong labels, 150 weakly labeled audio files were incorporated into the training dataset. These files were sourced from the same matches and each had a duration of 10 seconds. In the weak labeling approach, only the occurrence of the badminton sound event, such as the sound of the shuttle being hit, was annotated, without providing precise onset and offset times for each event. The

1 Label Studio : https://github.com/heartexlabs/label-studio/

output csv file is shown in Figure 3.13, providing the path to audio file and the occurence of events. Additionally, another set of 150 unlabeled 10-second audio files was included in the training dataset, meaning that no specific annotations were provided for these files.

	filename	event_labels
2	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.17.0.wav	Badminton_Hit
3	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.18.0.wav	Badminton_Hit
4	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.19.0.wav	Badminton_Hit
5	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.20.0.wav	Badminton_Hit
6	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.21.0.wav	None
7	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.22.0.wav	None
8	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.23.0.wav	Badminton_Hit
9	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.24.0.wav	Badminton_Hit
10	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.25.0.wav	None
11	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.26.0.wav	Badminton_Hit
12	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.27.0.wav	None
13	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.28.0.wav	Badminton_Hit
14	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.29.0.wav	None
15	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.30.0.wav	Badminton_Hit
16	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.31.0.wav	Badminton_Hit
17	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.32.0.wav	None
18	C:/Users/issac/Documents/ML/Badminton_sound/FDY-CRNN/Weak/audio/9.33.0.wav	None

Figure 3.13 Weakly labelled audio data in the form of .csv file

The training commenced with a warm-up phase consisting of 50 epochs and proceeded with an additional 100 epochs. The objective of the model was to accurately determine the onset and offset times for each instance of the badminton hit sound in the scene. However, after conducting multiple tests, the performance of this model proved to be discouraging, as the predicted onset and offset times did not align well with the ground truth of the badminton audio. I made attempts to enhance the accuracy of my manual annotations and even modified the framework to exclusively accept strongly labeled data for training. Regrettably, despite putting significant effort into these endeavors, none of them yielded noticeable improvements in the model's performance.

One possible explanation is that the duration of the badminton hit sound is comparatively short in comparison to the application of this model on other sound events, such as bell ringing, dog barking, or vacuum sounds, which typically range from 500 milliseconds to 10 seconds in the training data. As the duration of badminton sound events usually does not exceed 250 milliseconds, this may have

contributed to the model's difficulty in accurately capturing and predicting the onset and offset times for this particular task.

3.4.2 Second SED Model

Although the best-performing model in the DESED dataset did not meet our requirements for this task, I remain optimistic about the potential of using sound event detection methods to identify the onset and offset times of the badminton hit sound. In a study conducted by Mesaros et al. [69], they provide a comprehensive tutorial on current sound event detection tasks and different approaches employed by researchers in the field.

The paper discusses various aspects of sound event detection, including the representation of audio features and the appropriate window size to achieve a balance between capturing sufficient detail and avoiding high-dimensional representations. In that particular section, they highlight the work of Adavanne et al. [70], which employs multiple spectral representations calculated at different time and frequency resolutions. This approach is particularly beneficial for scenarios involving short sounds, such as a door slam, as analyzing these short sounds with higher time resolution can lead to better results.

In Adavanne's work, they also introduce the concept of using multichannel features from the audio as an input for the convolutional recurrent network and process the features using bi-directional LSTMs(Long Short-Term Memory). Also, in their network, they focus on using low level features such as spectral and time-domain features to learn high-level informations rather than directly utilize high-level features such as deep learning features. With the multiple appraoch they propose, they got a absolute improvement for both dataset TUT-SED 2009 and TUD-SED 2016.

Considering Adavanne et al.'s work aligns better with the task of detecting the sound of the badminton shuttle being hit, the first step is to restructure the training data to suit the model's requirements. Initially, the input audio data, along with the corresponding annotations, are transformed into spectrograms, as depicted in Figure 3.14. A spectrogram provides a visual representation of the signal's frequency content over time, while in the spectrogram, the manually annotated sound events are indicated by green grids.

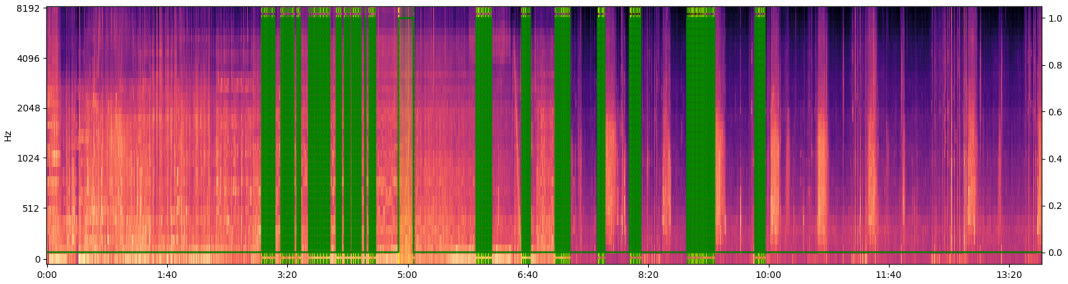


Figure 3.14 Spectrogram of audio data with event annotations shown in green grid.

Next, the spectrograms with annotations are cropped into suitable sizes for the network to learn from, taking into account the short duration of the badminton sound. To accomplish this, a time resolution of 0.1 and a window duration of 0.8 are set, adequately capturing the 250-millisecond badminton events. The resulting cropped spectrograms are illustrated in Figure 3.15, with each window spanning a total time length of 800 milliseconds and containing the 250-millisecond sound events. These cropped spectrograms are then fed into the CRNN network, as described in [70], and trained for 600 epochs using a batch size of 12.

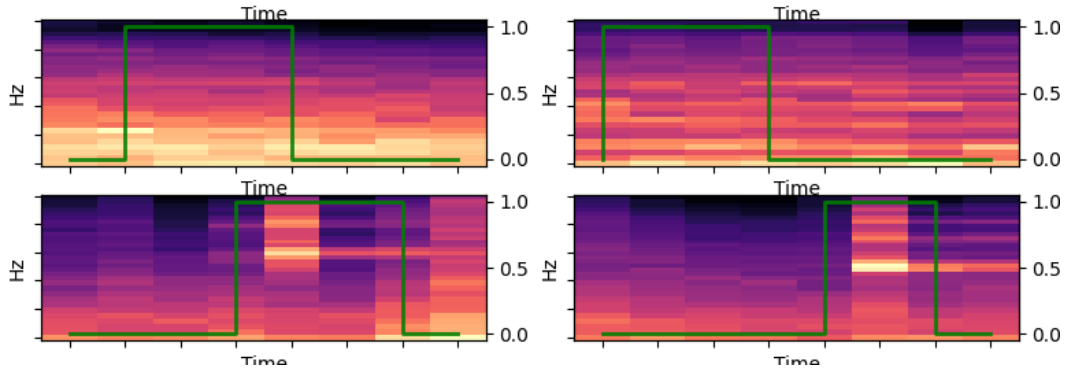


Figure 3.15 Cropped spectrogram with event label to be used as training data.

After training is done, the CRNN based SED model's predictions on a 13-minute testing badminton video are presented in Figure 3.16, the green grid in the spectrogram refers the to ground truth of events happening in the scene. A

zoomed-in version on one specific rally is shown in Figure 3.18 to have a better view for the predictions in a smaller scale.

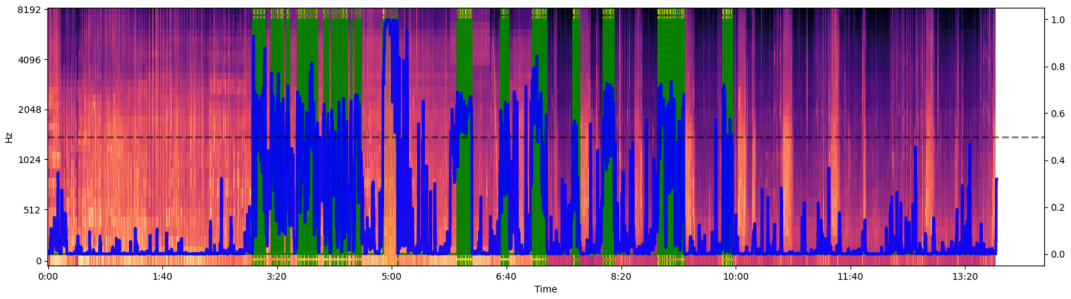


Figure 3.16 Cropped spectrogram with event label to be used as training data.

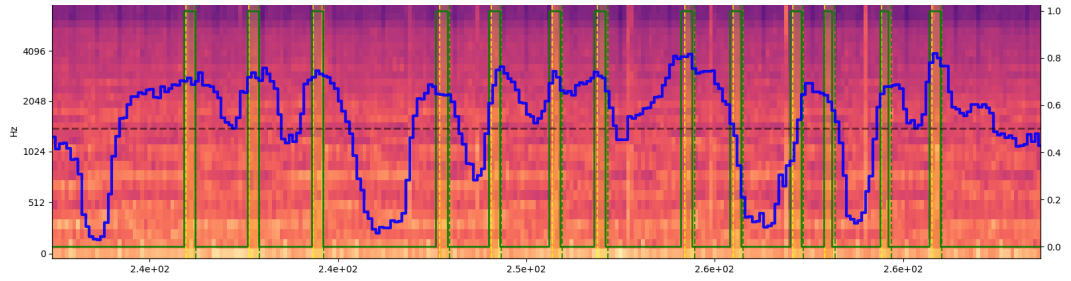


Figure 3.17 Cropped spectrogram with event label to be used as training data.

From the graphs, it is evident that while the model may not precisely predict each sound event in line with the ground truth, it provides information indicating a high likelihood of an event occurring in the surrounding frames. This additional information can be used in conjunction with visual elements to determine if an event has taken place in a given frame.

3.5. Final Prototype

The final prototype integrates both visual and audio processing components to create an audiovisual system, as shown in Fig 3.18

For the visual aspect, the visual scene undergoes image processing techniques, and an object detection model trained with YOLOv8 is employed to track the

badminton shuttle in the scene. The shuttle's coordinates are then utilized to calculate its position and motion state. These parameters are passed into an algorithm that assesses the likelihood of the shuttle being struck by a racket in each frame. Frames with a high probability are identified and recorded in a dataframe for further analysis.

Regarding the audio component, the audio data is first converted into a spectrogram to visualize the audio content. The spectrogram is then divided into smaller segments and fed into a CRNN-based Sound Event Detection (SED) model. This model predicts the likelihood of each frame containing the sound of the badminton shuttle being hit by a racket. Although the predictions may not be precise enough to determine the exact frames with events, frames with a possibility exceeding 50 percent are marked and recorded in the dataframe.

By comparing the markings from both the visual and audio components, frames that have overlapping marks are identified. Furthermore, the three frames preceding and following each identified frame are also considered. Frames that receive markings from the visual, as well as their neighboring frames that receive markings from audio components, are determined to contain the event and are designated for haptic feedback generation. With a comparison to the previous output of only utilizing visual input data, the accuracy of the system on the same testing video as stated in section 3.3.3 increased from 12/23 to 17/23 for the first video and 9/16 to 11/16 in the second video, achieving an accuracy close to 70 percent.

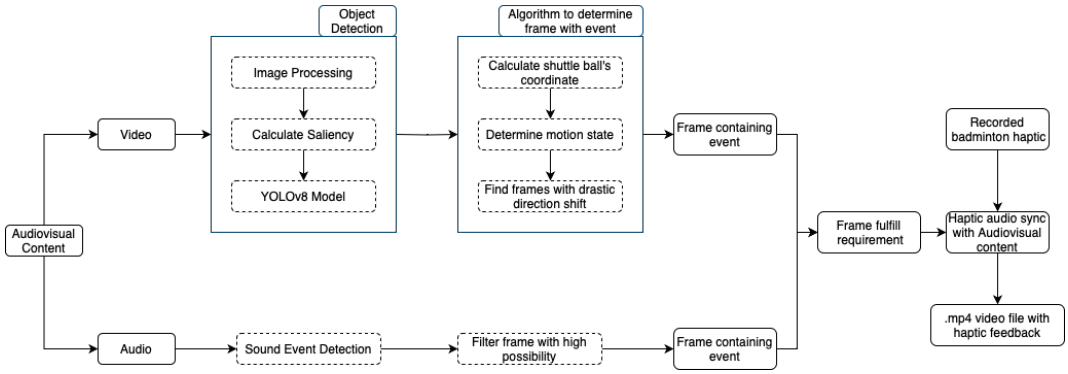


Figure 3.18 Illustration of the overall system framework

3.5.1 Generates Haptic Audio Automatically

Once the algorithm determines the frames that should generate haptic feedback, the next step involves synchronizing the vibration audio of the racket with the corresponding frames. Specifically, pre-recorded audio that represents the haptic sensation of the impact between the racket and shuttle ball is added to the frames identified by the algorithm. For the remaining frames, a silent audio segment is inserted. An audio file with the same sample rate as the badminton video's audio is then created.

In order to get the most realistic haptic audio, Tectile toolkit [71] is integrated into a badminton racket, enabling the transfer of haptic feedback directly from the racket to the computer, where it is recorded. This approach ensures that the haptic audio accurately reflects the impact of the racket striking the shuttlecock, enhancing the overall realism of the haptic experience.

To combine all the components seamlessly, I employ FFMPEG[2], a powerful multimedia framework. FFMPEG automatically merges the original video and audio with the haptic audio generated by my system into a quad-channel mp4 file. The first two channels contain the original audio, while the latter two channels accommodate the haptic audio. This ensures that the haptic feedback is accurately synchronized with the corresponding frames in the video.

2 FFMPEG : https://ffmpeg.org

Chapter 4
Prove of Concept

4.1. Pilot Test

In this pilot test, our objective is to gather user feedback on the accuracy of the automatically generated haptic videos. We utilized two badminton rally videos, one with a length of 16 seconds and the other with 22 seconds, and passed them through our multi-model system to generate haptic feedback for the videos. The haptic videos were then directly provided to the participants for their experience.

To enable participants to feel the haptic feedback from the videos, we provided them with a 3D printed device, as shown in Fig 4.1, which incorporates two haptic modules called "hapStak". These modules enhance the haptic experience for the participants. A total of 4 participants (3 male, 1 female) experienced the haptic videos and provided valuable insights and suggestions for potential improvements based on their experience.

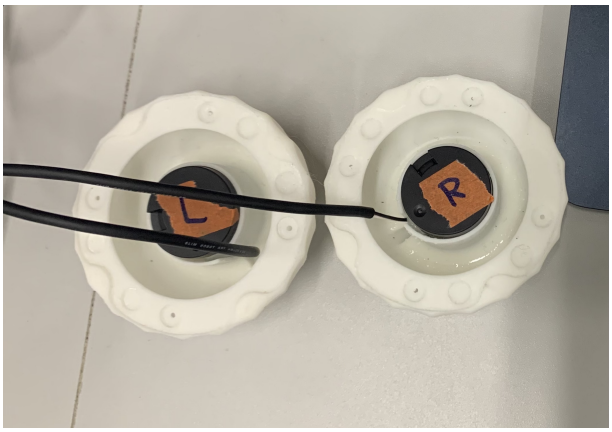


Figure 4.1 Haptic Device

4.1.1 Feedback for Pilot Test

The overall feedback from the participants was positive, with one participant expressing, "The accuracy of the automatic generated haptic feedback exceeded my expectations. While there were a few errors where the haptic device provided feedback when it shouldn't have, I still found these errors to be acceptable. Overall, the haptic feedback did enhanced my immersion with the video."

Three out of four participants provided feedback regarding the improvement of haptic intensity. The participants felt that the volume of the haptic feedback was set too low for the experience, and they suggested increasing the intensity. One participant also suggested that the intensity of the haptic device should vary according to the actions of the players in the scene. For example, a strike should generate a stronger haptic sensation compared to a net drop shot. This feedback provides valuable insights for enhancing the overall haptic experience and offers promising directions for future research in this area.

4.2. Semi-Auto Authoring Experiment

4.2.1 Overview

To assess the effectiveness of the automatic generated haptic feedback in assisting the haptic video annotation task, I propose an experimental study involving participants who will perform two annotation tasks. One task will involve traditional manual authoring, while the other task will incorporate the automatic haptic audio generated by my multi-model system as an assistant. By comparing the two approaches, we aim to evaluate the impact of the new authoring method on the annotation process.

The primary metric for evaluation in this experiment is the time taken by participants to complete each task. Additionally, a questionnaire will be administered to gather feedback on the usability of the system, and a brief interview will be conducted to delve deeper into participants' experiences.

A total of 8 participants, consisting of 4 males and 4 females, took part in the experiment. Among them, three participants (two males and one female) possessed adequate experience in video editing, which adds diversity to the participant pool.

4.2.2 Experiment Design

The video used for annotation in both tasks and across participants is a 21-second badminton rally video. The video contains 22 badminton strikes that are not included in the training data of the object detection and sound event detection models. At the beginning of the experiment, participants will have the opportunity to experience an actual haptic video manually annotated by myself. To facilitate this experience, a 3D printed haptic device, as shown in Fig 4.1, is provided.

For the annotation tasks, participants will use Ableton Live 11 software [3], which will be introduced to them through a tutorial session prior to the actual task. Participants will be given an additional 3 minutes to familiarize themselves with the software before the experiment commences. Two tasks will be conducted: 1) Manual Authoring task and 2) Semi-Auto Authoring task. The order of these tasks will be randomly assigned to participants.

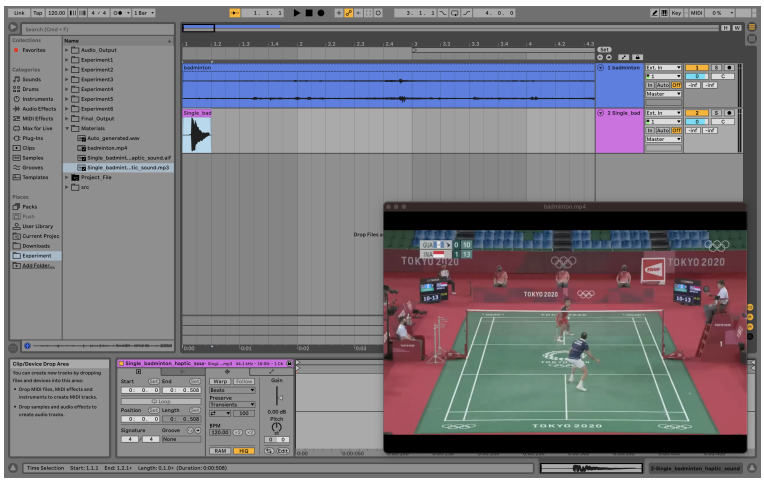


Figure 4.2 The interface of Ableton Live during the manual annotation task.

3 https://www.ableton.com

Manual Authoring Task

In the manual authoring task, participants will be required to create a haptic video from scratch, as depicted in Fig 4.2. To facilitate the annotation process, the visual content will be played synchronously in the bottom right corner of the interface. The first track will contain the audio from the badminton match, while the second track will consist of a mono channel audio file representing the haptic feedback recorded from a badminton racket striking a shuttle ball.

The primary objective of participants in this task is to either duplicate or drag the haptic audio to the desired positions in the video where they feel it intuitively enhances the haptic feedback experience. The time spent by each participant will be recorded, including the final checking phase where participants review the haptic video they created from the beginning and ensure all annotations are correct before submission.

The haptic audio created by each participant will be saved as a stereo mp3 file. Subsequently, the audio will be combined with the visual content and the audio from the badminton rally into an mp4 file, resulting in a final output with four channels of audio.

Semi-Auto Authoring Task

In the semi-auto authoring task, participants will be presented with a track of audio that is automatically generated by the multi-model system. The automatic audio generation achieved an accuracy of 16 correct annotations out of 22 hits. A comparison between the automatically generated audio and the manual annotation by participant 3 is illustrated in Fig 4.3.

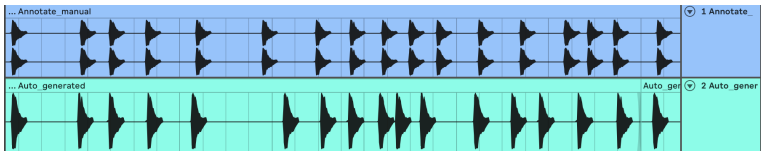


Figure 4.3 The track on above is the manual annotation by participant 3 ; The track on below is the automatic generated haptic audio

The interface for participants during the semi-auto annotation task is depicted

in Fig 4.4. Similar to the manual annotation task, the visual content will be provided to aid the participants.

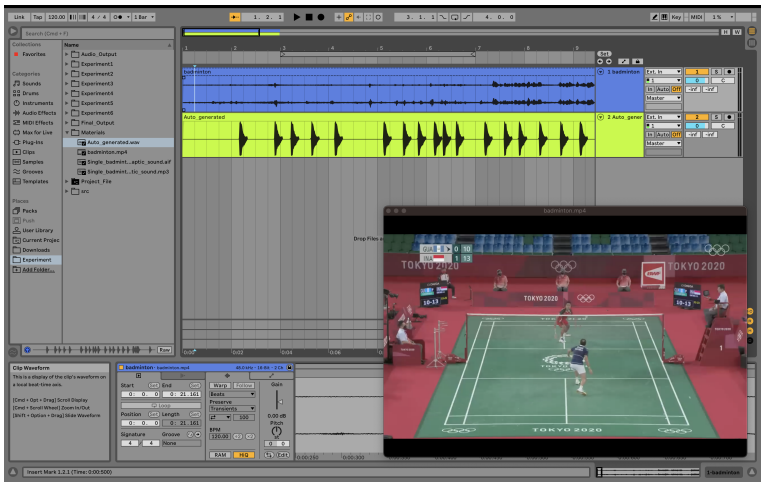


Figure 4.4 The interface of Ableton Live during the semi-auto annotation task.

The main objective for participants in this task is to create a haptic audio that intuitively enhances the experience. However, in this task, participants can utilize the automatically generated audio track, which is the second track in Fig 4.4, as an assistive tool. They will need to adjust the audio track to the correct positions by either removing redundant audio segments or simply dragging the incorrect audio segments to their correct positions. The time taken by each participant to complete the task, including the reviewing phase, will be recorded.

Evaluation

Upon completing each task, participants will be asked to fill out a questionnaire that combines a modified version of the System Usability Scale (SUS) [72] and post-task usability questionnaires [73]. These questionnaires are designed to gather feedback on the usability and user experience of the system.

Once both tasks are finished, the haptic audio created by each participant will be compiled into a haptic video. Participants will then have the opportunity to experience both the haptic video they created in the manual authoring task and the semi-auto authoring task. Subsequently, interviews will be conducted to

gather further feedback on both the authoring tasks and the experience of the haptic videos.

4.2.3 Results

No	Question	Scale
1	I was satisfied with the time spent to complete the task	1 - 5
2	I think that I would like to use this annotation method frequently	1 - 5
3	I found the this annotation method unnecessarily complex	1 - 5
4	I thought the annotation method was easy to use	1 - 5
5	I think that I would need the support of a technical person to be able to use this system	1 - 5
6	I would imagine that most people would learn to use this annotation method very quickly	1 - 5
7	I found this annotation method very inconvenient to use	1 - 5
8	I needed to learn a lot of things before I could get going with this annotation task	1 - 5

Table 4.1 Questionnaire used in user study.

The questionnaire utilized in the experiment is presented in Table 4.1. This questionnaire was administered in both tasks, with slight modifications for the semi-automatic authoring task. In the semi-automatic authoring task, two additional questions were included. The first question was: "I felt that the automatically generated haptic audio significantly improved the efficiency in completing the task," and the second question was:"I am satisfied with the accuracy of the automatically generated haptic audio." Participants were asked to rate their responses on a scale of 1 to 5 for both of these questions.

Analysis

To investigate the impact of the automatically generated haptic audio on the annotation task, a paired t-test was conducted to compare the annotation experiences between the two tasks. Subsequently, a Šidák-Holm multiple comparisons test was employed to identify any significant differences between specific pairs.

Overall Results and Trends

As shown in Fig. 4.5, the utilization of automatic generated haptic audio resulted in a significant improvement in the Quality of Experience (QoE) when compared to manual haptic authoring (Q1-Q4, Q7). Overall, the majority of participants experienced a notable increase in their satisfaction with the time required to

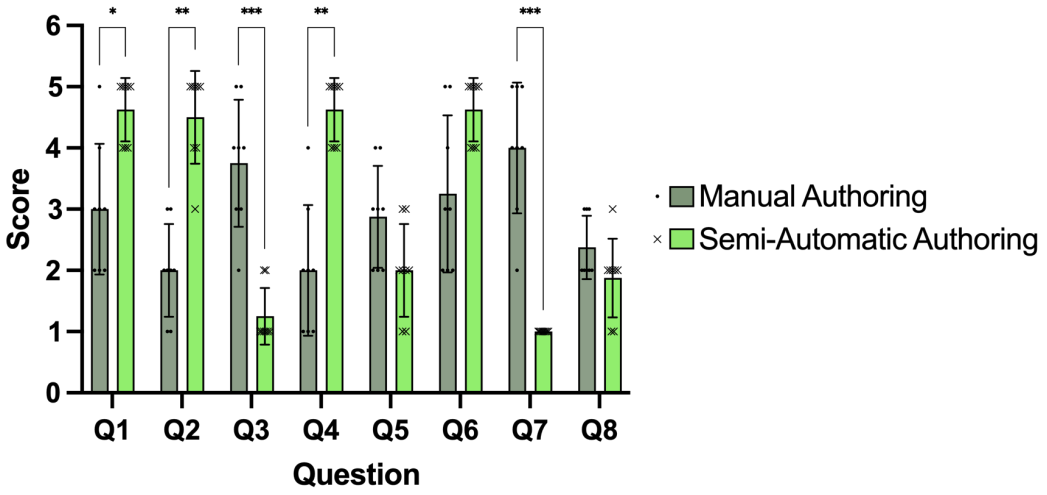


Figure 4.5 The collective results comparing the annotation experience in both tasks from the questionnaire (Q1-Q8, See Table 4.1) * $p<0.05$, ** $p<0.01$, *** $p<0.001$. The figure includes scatter plots that visualize the responses of each participant with symbol (·) for Manual Authoring, (×) for Semi-Automatic Authoring

complete the task in the semi-automatic authoring task ($p < 0.05$). Only one participant provided a score of 5 in both tasks. This participant initially performed the manual authoring task and then the semi-automatic task. During the post-interview session, I inquired about the reason for giving 5 points in both tasks. The participant mentioned being highly experienced in video and audio editing, which contributed to their ability to complete the tasks quickly, and hence a high satisfaction in the time spent. However, they expressed a preference for and greater satisfaction with the time spent in the semi-automatic authoring task, suggesting that starting with the semi-automatic authoring task first would have resulted in giving higher score than the manual authoring task.

In terms of task complexity (Q3, Q7), it is observed that the semi-automatic authoring task was significantly preferred by participants, with a lower score indicating lower complexity compared to the manual authoring task ($p < 0.001$). Interestingly, the score was not significantly affected by the order of the tasks, as seen in the scatter plot where the scores for both these questions in the semi-

automatic authoring task are highly concentrated. Furthermore, participants found the semi-automatic annotation method easier to use (Q4, $p < 0.01$) and expressed a preference for using this annotation method frequently (Q2, $p < 0.01$) compared to manual annotation.

Regarding the ease of learning the annotation method (Q5, Q6, Q8), it was observed that participants found the semi-automatic authoring task to be slightly easier than the manual authoring task, although this difference was not statistically significant. This may be attributed to the fact that participants with prior video editing experience perceived both tasks as relatively easy to learn and navigate.

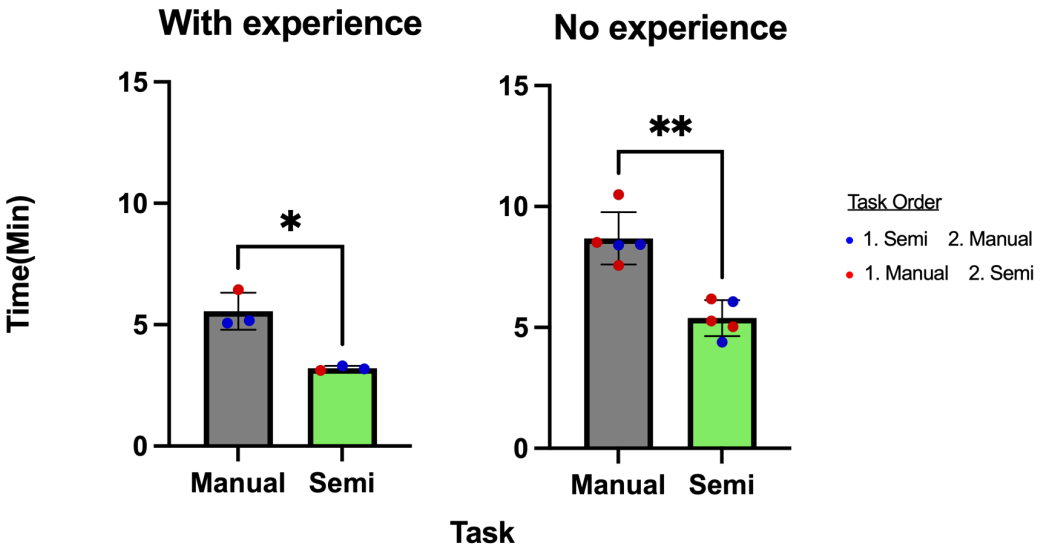


Figure 4.6 The time taken to complete each task was analyzed based on participants' level of experience in video editing. Statistical significance is denoted as * $p<0.05$, ** $p<0.01$, *** $p<0.001$. The figure presents scatter plots that depict the time used by each participant, with different colors representing the order of tasks undertaken in the experiment.

The time taken by each participant to complete each task is presented in Figure 4.6. The results indicate that participants with prior experience in video editing tended to complete both tasks more quickly than participants without much experience in video editing. This can be attributed to the similarity between the

annotation software and other commercial video editing tools, enabling participants with video editing experience to become familiar with the editing tool more efficiently.

When comparing the two groups (participants with and without video editing experience), a significant increase in efficiency was observed between the two tasks. The increase in efficiency was statistically significant at $p < 0.05$ for participants with video editing experience and $p < 0.01$ for participants without video editing experience. Surprisingly, the order of the tasks did not have a significant impact on the time spent for annotation. This suggests that the practice time provided prior to the experiment effectively familiarized participants with the annotation software, regardless of the order in which the tasks were performed.

No	Question	Scale
9	I felt the automatic generated haptic audio greatly increase the efficiency in completing the task	1 - 5
10	I am satisfied with the accuracy of the automatic generated haptic audio	1 - 5

Table 4.2 Additional question in semi-automatic authoring task's questionnaire.

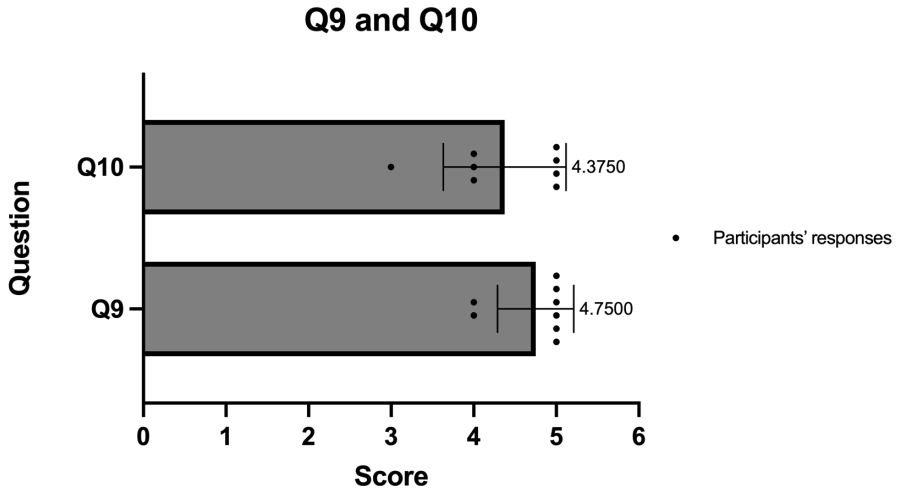


Figure 4.7 Participants' responses on question 9 and question 10, with mean of each question plot on the end of bar.

After the semi-automatic authoring task, two additional questions were presented to the participants, as shown in Table 4.2. These questions aimed to assess

the effectiveness of the automatic generated haptic audio in assisting the annotation process.

Based on the participants' responses, it was evident that they felt the automatic generated haptic audio significantly improved their efficiency in completing the annotation task (Q9), as indicated by a mean value of 4.750. Furthermore, the participants expressed satisfaction with the accuracy of the automatic generated haptic audio (Q10), with a mean score of 4.375.

Feedback

An interview was conducted after compiling the haptic audios annotated by the participants in both tasks and allowing them to experience the feedback. Overall, all participants commented that they found the automatic generated haptic audio to be helpful in completing the annotation task. One participant specifically mentioned, "The semi-automatic authoring is clearly much more efficient than manual authoring. Editing is much easier than creating the audio from scratch, especially when there are frequent repeating events."

Another participant, who had extensive experience in editing video and audio, provided a concise piece of advice, stating, "The automatically generated annotations would greatly increase the efficiency of such work, but only if they have a high level of accuracy. If the accuracy declines too much, despite it being a convenient tool for basic annotation, it becomes frustrating to have to edit and redo them to the point where it becomes more efficient to input them manually. However, I was generally satisfied with the accuracy of the system I used this time." This participant highlighted the importance of accuracy in determining the efficiency of the automatic generated annotations, and in this case, the automatic generated haptic audio surpasses the baseline in terms of accuracy.

The participants' feedback has also provided valuable insights on enhancing the haptic feedback experience. One suggestion is to incorporate player-specific feedback, where the haptic device provides feedback on the corresponding side based on the player's position. For example, if the player on the left side of the scene strikes the shuttlecock, the haptic feedback would be generated on the left side of the haptic device, and vice versa. This approach could significantly enhance the immersive feeling for users, making them feel more engaged in the

scene.

Furthermore, participants expressed a desire for a greater variety of scenes, including other sports such as tennis, soccer, and more. Implementing similar scenarios would offer users a broader range of experiences and increase the versatility of the haptic feedback system.

These feedback suggestions provide valuable directions for further improving the haptic feedback experience and expanding the applicability of the system to different sports and scenarios.

4.3. Discussion

In summary, the results showed that the utilization of automatic generated haptic audio significantly improved the Quality of Experience (QoE) compared to manual authoring. Participants expressed a higher level of satisfaction with the time required to complete the semi-automatic task. Notably, participants with prior video editing experience showed faster completion times overall, and their efficiency was further enhanced when using the semi-automatic authoring method. This suggests that the combination of automatic generated haptic audio and participant experience in video editing resulted in even greater efficiency.

In the study, participants' questionnaire responses highlighted the task to be easier and efficient with the help of automatic generated haptic audio. Moreover, participants expressed satisfaction with the accuracy of the automatic generated haptic audio. Valuable feedback from participants included suggestions for enhancing the haptic feedback experience, such as customizing feedback based on player positions and incorporating a broader range of scenarios. The findings underscored the potential of automatic generated haptic audio to enhance the annotation process, emphasizing the importance of maintaining high accuracy levels for optimal effectiveness.

Chapter 5
Conclusion

With the increasing demand for immersive multimedia experiences, haptic feedback has emerged as a crucial element to enhance the engagement and realism of content. The future is expected to witness a significant rise in the demand for haptic videos, creating a need for skilled haptic designers and editors capable of synchronizing haptic, audio, and visual content while delivering intuitive tactile feedback to the audience.

Manual authoring has traditionally been the preferred method for achieving precise and immersive haptic experiences in videos. However, this approach is time-consuming due to repetitive labor-intensive tasks. To address this challenge, a semi-automatic haptic annotation method is proposed in this research. This method leverages a multi-model deep learning framework to automatically generate haptic audio, thereby increasing the efficiency of the annotation task.

In contrast to previous research, the focus of this study is on generating haptic feedback for specific events occurring within a scene, such as the impact of a badminton racket or shuttle. Previous approaches primarily concentrated on generating trails or overall haptic effects, neglecting the importance of feedback when objects come into contact with the main character. By adopting a first-person perspective, the aim of this research is to provide a more immersive haptic experience where the audience can truly feel a part of the scene.

The results of the annotation experiment indicate that the utilization of automatic generated haptic audio, in combination with manual authoring (referred to as semi-automatic haptic authoring in this research), significantly enhances the efficiency of participants in completing the annotation task, regardless of the order of tasks. This conclusion is supported by both statistical analysis of the questionnaire responses and the observed time taken to complete each task.

Participants provided valuable feedback on the limitations of the current system

prototype. Firstly, the system is currently capable of generating haptic feedback only for the badminton scenario, whereas a wider range of scenarios would be more engaging for the audience. Future studies can address this limitation by increasing the diversity of training data for the object detection model and sound event detection model.

The accuracy of the automatic generated haptic audio was found to significantly impact users' efficiency and preference for the semi-automatic haptic authoring method over manual authoring. While the accuracy achieved in this research met the baseline requirements for participant preference, there is still room for improvement. Enhancements can be made by refining the sound event detection model or exploring alternative approaches to address this task. However, there is currently no accurate baseline to define the accuracy of a haptic annotation. In this research, the measure used to determine the similarity and accuracy in two of the experimental tasks is based on the time difference between each striking event annotated by the participants to the event frame, which should ideally fall within a range of 100 milliseconds.

Improvements can also be made in the haptic feedback experience for users. For example, providing different haptic intensities based on player movements or actions, or delivering haptic feedback according to the location of players within the scene. Such enhancements would further enhance the immersive and enjoyable experience for the audience.

Overall, while the current study presents promising results, further developments and refinements are necessary to expand the range of scenarios, improve accuracy, and enhance the haptic feedback experience in order to maximize the immersion and enjoyment of the audience.

References

[1] Julia Barrett and Helmut Krueger. Performance effects of reduced proprioceptive feedback on touch typists and casual users in a typing task. 13(6):373–381. URL: http://www.tandfonline.com/doi/abs/10.1080/01449299408914618, doi:10.1080/01449299408914618.

[2] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. Haptic rendering: programming touch interaction with virtual objects. In *Proceedings of the 1995 symposium on Interactive 3D graphics - SI3D '95*, pages 123–130. ACM Press. URL: http://portal.acm.org/citation.cfm?doid=199404.199426, doi:10.1145/199404.199426.

[3] Louis Rosenberg and Scott Brave. Using force feedback to enhance human performance in graphical user interfaces. In *Conference companion on Human factors in computing systems common ground - CHI '96*, pages 291–292. ACM Press. URL: http://portal.acm.org/citation.cfm?doid=257089.257327, doi:10.1145/257089.257327.

[4] Timothy Miller and Robert Zeleznik. An insidious haptic invasion: adding force feedback to the x desktop. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 59–64. ACM. URL: https://dl.acm.org/doi/10.1145/288392.288573, doi:10.1145/288392.288573.

[5] Bob G. Witmer and Michael J. Singer. Measuring presence in virtual environments: A presence questionnaire. 7(3):225–240. URL: https://direct.mit.edu/pvar/article/7/3/225-240/92643, doi:10.1162/105474698565686.

[6] K.S. Hale and K.M. Stanney. Deriving haptic design guidelines from human physiological, psychophysical, and neurological foundations. *IEEE Com-*

puter Graphics and Applications, 24(2):33–39, 2004. doi:10.1109/MCG.2004.1274059.

[7] Oliver Staadt. The ultimate display.

[8] Jaebong Lee, Bohyung Han, and Seungmoon Choi. Motion effects synthesis for 4d films. 22(10):2300–2314. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi:10.1109/TVCG.2015.2507591.

[9] Seungmoon Choi and Katherine J. Kuchenbecker. Vibrotactile display: Perception, technology, and applications. 101(9):2093–2104. URL: http://ieeexplore.ieee.org/document/6353870/, doi:10.1109/JPROC.2012.2221071.

[10] Yeongmi Kim, Jongeun Cha, Jeha Ryu, and Ian Oakley. A tactile glove design and authoring system for immersive multimedia. 17(3):34–45. Conference Name: IEEE MultiMedia. doi:10.1109/MMUL.2010.5692181.

[11] Jongeun Cha, Yongwon Seo, Yeongmi Kim, and Jeha Ryu. An authoring/editing framework for haptic broadcasting: Passive haptic interactions using MPEG-4 BIFS. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pages 274–279. IEEE. URL: http://ieeexplore.ieee.org/document/4145187/, doi:10.1109/WHC.2007.20.

[12] Md. Abdur Rahman, Abdulmajeed Alkhaldi, Jongeun Cha, and Abdulmotaleb El Saddik. Adding haptic feature to YouTube. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1643–1646. ACM. URL: https://dl.acm.org/doi/10.1145/1873951.1874310, doi:10.1145/1873951.1874310.

[13] Yaxuan Li, Yongjae Yoo, Antoine Weill-Duflos, and Jeremy Cooperstock. Towards context-aware automatic haptic effect generation for home theatre environments. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11. ACM. URL: https://dl.acm.org/doi/10.1145/3489849.3489887, doi:10.1145/3489849.3489887.

[14] Myongchan Kim, Sungkil Lee, and Seungmoon Choi. Saliency-driven tactile effect authoring for real-time visuotactile feedback. In Poika Isokoski and Jukka Springare, editors, *Haptics: Perception, Devices, Mobility, and Communication*, volume 7282, pages 258–269. Springer Berlin Heidelberg. Series Title: Lecture Notes in Computer Science. URL: http://link.springer.com/10.1007/978-3-642-31401-8_24, doi:10.1007/978-3-642-31401-8_24.

[15] Kai Zhang, Lawrence H Kim, Yipeng Guo, and Sean Follmer. Automatic generation of spatial tactile effects by analyzing cross-modality features of a video. In *Symposium on Spatial User Interaction*, pages 1–10. ACM. URL: https://dl.acm.org/doi/10.1145/3385959.3418459, doi:10.1145/3385959.3418459.

[16] Jongeun Cha, Mohamad Eid, and Abdulmotaleb El Saddik. Touchable 3d video system. 5(4):1–25. URL: https://dl.acm.org/doi/10.1145/1596990.1596993, doi:10.1145/1596990.1596993.

[17] Angela Chang and Conor O'Sullivan. Audio-haptic feedback in mobile phones. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, pages 1264–1267. ACM. URL: https://dl.acm.org/doi/10.1145/1056808.1056892, doi:10.1145/1056808.1056892.

[18] Ping-Hsuan Han, Yang-Sheng Chen, Kong-Chang Lee, Hao-Cheng Wang, Chiao-En Hsieh, Jui-Chun Hsiao, Chien-Hsing Chou, and Yi-Ping Hung. Haptic around: multiple tactile sensations for immersive environment and interaction in virtual reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pages 1–10. ACM. URL: https://dl.acm.org/doi/10.1145/3281505.3281507, doi:10.1145/3281505.3281507.

[19] Heather Culbertson, Juan Jose Lopez Delgado, and Katherine J. Kuchenbecker. One hundred data-driven haptic texture models and open-source methods for rendering on 3d objects. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 319–325. IEEE. URL: http://ieeexplore.ieee.org/document/6775475/, doi:10.1109/HAPTICS.2014.6775475.

[20] Spatial patterns of cutaneous vibration during whole-hand haptic interactions. URL: https://www.pnas.org/doi/10.1073/pnas.1520866113, doi:10.1073/pnas.1520866113.

[21] Anthony Bazelle, Hugo Pourrier-Nunez, Maxime Rignault, and Michael Chang. Simulation of different materials texture in virtual reality through haptic gloves. In *SIGGRAPH Asia 2018 Posters*, pages 1–2. ACM. URL: https://dl.acm.org/doi/10.1145/3283289.3283370, doi:10.1145/3283289.3283370.

[22] Roshan Lalintha Peiris, Wei Peng, Zikun Chen, Liwei Chan, and Kouta Minamizawa. ThermoVR: Exploring integrated thermal haptic feedback with head mounted displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5452–5456. ACM. URL: https://dl.acm.org/doi/10.1145/3025453.3025824, doi:10.1145/3025453.3025824.

[23] Anshul Singhal and Lynette A. Jones. Perceptual interactions in thermotactile displays. In *2017 IEEE World Haptics Conference (WHC)*, pages 90–95. doi:10.1109/WHC.2017.7989882.

[24] Yael Salzer, Tal Oron-Gilad, and Adi Ronen. Thermoelectric tactile display based on the thermal grill illusion. In *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!*, ECCE '07, page 303–304, New York, NY, USA, 2007. Association for Computing Machinery. URL: https://doi.org/10.1145/1362550.1362616, doi:10.1145/1362550.1362616.

[25] Takahiro Arai and Akifumi Inoue. BLASTNEL: Collision sensation display for virtual reality games using highly compressed air. In *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, pages 572–576. ACM. URL: https://dl.acm.org/doi/10.1145/3369457.3369534, doi:10.1145/3369457.3369534.

[26] Sidhant Gupta, Dan Morris, Shwetak N. Patel, and Desney Tan. AirWave: non-contact haptic feedback using air vortex rings. In *Proceedings of the 2013*

ACM international joint conference on Pervasive and ubiquitous computing, pages 419–428. ACM. URL: https://dl.acm.org/doi/10.1145/2493432.2493463, doi:10.1145/2493432.2493463.

[27] Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. UltraHaptics: multi-point mid-air haptic feedback for touch surfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 505–514. ACM. URL: https://dl.acm.org/doi/10.1145/2501988.2502018, doi:10.1145/2501988.2502018.

[28] Graham Wilson, Thomas Carter, Sriram Subramanian, and Stephen A. Brewster. Perception of ultrasonic haptic feedback on the hand: localisation and apparent motion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1133–1142. ACM. URL: https://dl.acm.org/doi/10.1145/2556288.2557033, doi:10.1145/2556288.2557033.

[29] Seki Inoue, Yasutoshi Makino, and Hiroyuki Shinoda. Active touch perception produced by airborne ultrasonic haptic hologram. In *2015 IEEE World Haptics Conference (WHC)*, pages 362–367. IEEE. URL: http://ieeexplore.ieee.org/document/7177739/, doi:10.1109/WHC.2015.7177739.

[30] Yatharth Singhal, Haokun Wang, Hyunjae Gil, and Jin Ryong Kim. Mid-air thermo-tactile feedback using ultrasound haptic display. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11. ACM. URL: https://dl.acm.org/doi/10.1145/3489849.3489889, doi:10.1145/3489849.3489889.

[31] Inrak Choi and Sean Follmer. Wolverine: A wearable haptic interface for grasping in VR. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 117–119. ACM. URL: https://dl.acm.org/doi/10.1145/2984751.2985725, doi:10.1145/2984751.2985725.

[32] Vahid Pooryousef, Ross Brown, and Selen Turkay. Shape recognition and selection in medical volume visualisation with haptic gloves. In *Proceedings of*

the 31st Australian Conference on Human-Computer-Interaction, pages 433–436. ACM. URL: https://dl.acm.org/doi/10.1145/3369457.3369508, doi:10.1145/3369457.3369508.

[33] Takumi Takahashi, Keisuke Shiro, Akira Matsuda, Ryo Komiyama, Hayato Nishioka, Kazunori Hori, Yoshio Ishiguro, Takashi Miyaki, and Jun Rekimoto. Augmented jump: a backpack multirotor system for jumping ability augmentation. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 230–231. ACM. URL: https://dl.acm.org/doi/10.1145/3267242.3267270, doi:10.1145/3267242.3267270.

[34] Sungjune Jang, Lawrence H. Kim, Kesler Tanner, Hiroshi Ishii, and Sean Follmer. Haptic edge display for mobile tactile interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3706–3716. ACM. URL: https://dl.acm.org/doi/10.1145/2858036.2858264, doi:10.1145/2858036.2858264.

[35] Colin Swindells, Seppo Pietarinen, and Arto Viitanen. Medium fidelity rapid prototyping of vibrotactile haptic, audio and video effects. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 515–521. IEEE. URL: http://ieeexplore.ieee.org/document/6775509/, doi:10.1109/HAPTICS.2014.6775509.

[36] Fabien Danieau, Anatole Lecuyer, Philippe Guillotel, Julien Fleureau, Nicolas Mollet, and Marc Christie. Enhancing audiovisual experience with haptic feedback: A survey on HAV. 6(2):193–205. Conference Name: IEEE Transactions on Haptics. doi:10.1109/TOH.2012.70.

[37] Mee Young Sung, Kyungkoo Jun, Dongju Ji, Hwanmun Lee, and Kikwon Kim. Touchable video and tactile audio. In *2009 11th IEEE International Symposium on Multimedia*, pages 425–431. IEEE. URL: http://ieeexplore.ieee.org/document/5366113/, doi:10.1109/ISM.2009.79.

[38] Fabien Danieau, Julien Fleureau, Audrey Cabec, Paul Kerbiriou, Philippe Guillotel, Nicolas Mollet, Marc Christie, and Anatole Lécuyer. Framework for enhancing video viewing experience with haptic effects of motion. In

2012 IEEE Haptics Symposium (HAPTICS), pages 541–546, 2012. doi:10.1109/HAPTIC.2012.6183844.

[39] Raphael Silva de Abreu, Douglas Mattos, Joel dos Santos, Gheorghita Ghinea, and Débora Christina Muchaluat-Saade. Toward content-driven intelligent authoring of mulsemedia applications. 28(1):7–16. Conference Name: IEEE MultiMedia. doi:10.1109/MMUL.2020.3011383.

[40] Fabien Danieau, Jérémie Bernon, Julien Fleureau, Philippe Guillotel, Nicolas Mollet, Marc Christie, and Anatole Lécuyer. H-studio: an authoring tool for adding haptic and motion effects to audiovisual content. In *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology - UIST '13 Adjunct*, pages 83–84. ACM Press. URL: http://dl.acm.org/citation.cfm?doid=2508468.2514721, doi:10.1145/2508468.2514721.

[41] Sile O'Modhrain and Ian Oakley. Adding interactivity: Active touch in broadcast media. In *Proceedings of the 12th International Conference on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, HAPTICS'04, page 293–294, USA, 2004. IEEE Computer Society.

[42] Kai Zhang, Lawrence H Kim, Yipeng Guo, and Sean Follmer. Automatic generation of spatial tactile effects by analyzing cross-modality features of a video. In *Proceedings of the 2020 ACM Symposium on Spatial User Interaction*, SUI '20, New York, NY, USA, 2020. Association for Computing Machinery. URL: https://doi.org/10.1145/3385959.3418459, doi:10.1145/3385959.3418459.

[43] Fabien Danieau, Julien Fleureau, Audrey Cabec, Paul Kerbiriou, Philippe Guillotel, Nicolas Mollet, Marc Christie, and Anatole Lécuyer. Framework for enhancing video viewing experience with haptic effects of motion. In *2012 IEEE Haptics Symposium (HAPTICS)*, pages 541–546, 2012. doi:10.1109/HAPTIC.2012.6183844.

[44] Myongchan Kim, Sungkil Lee, and Seungmoon Choi. Saliency-driven real-time video-to-tactile translation. 7(3):394–404. Conference Name: IEEE Transactions on Haptics. doi:10.1109/TOH.2013.58.

[45] Bin Wang and Piotr Dudek. A fast self-tuning background subtraction algorithm. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 401–404. ISSN: 2160-7516. doi:10.1109/CVPRW.2014.64.

[46] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. 27(1):38–49. Conference Name: IEEE Transactions on Image Processing. doi:10.1109/TIP.2017.2754941.

[47] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. URL: http://arxiv.org/abs/1803.08842, arXiv:1803.08842[cs].

[48] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. arXiv:1506.02640.

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-CNN: Towards real-time object detection with region proposal networks. version: 3. URL: http://arxiv.org/abs/1506.01497, arXiv:1506.01497[cs].

[50] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 213–229, Berlin, Heidelberg, 2020. Springer-Verlag. URL: https://doi.org/10.1007/978-3-030-58452-8_13, doi:10.1007/978-3-030-58452-8_13.

[51] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. URL: http://arxiv.org/abs/1405.0312, arXiv:1405.0312[cs].

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. 115(3):211–252. URL: http://link.springer.com/10.1007/s11263-015-0816-y, doi:10.1007/s11263-015-0816-y.

[53] RangeKing. Model architecture of yolov8. https://github.com/RangeKing, Posted in https://github.com/ultralytics/ultralytics/issues/189, 2023.

[54] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016. arXiv:1612.08242.

[55] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi:10.1109/CVPR.2015.7298594.

[56] Joseph Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016.

[57] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020. arXiv:2004.10934.

[58] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, November 2022. URL: https://doi.org/10.5281/zenodo.7347926, doi:10.5281/zenodo.7347926.

[59] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023. URL: https://github.com/ultralytics/ultralytics.

[60] T. K. Chan and Cheng Siong Chin. A comprehensive review of polyphonic sound event detection. 8:103339–103373. URL: https://ieeexplore.ieee.org/document/9106322/, doi:10.1109/ACCESS.2020.2999388.

[61] Hamidreza Ghafghazi, Amr Elmougy, Hussein T. Mouftah, and Carlisle Adams. Location-aware authorization scheme for emergency response.

4:4590–4608. URL: http://ieeexplore.ieee.org/document/7547379/, doi:10.1109/ACCESS.2016.2601442.

[62] Emre Çakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. 25(6):1291–1303. URL: http://arxiv.org/abs/1702.06286, arXiv:1702.06286[cs], doi:10.1109/TASLP.2017.2690575.

[63] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi:10.1109/5.18626.

[64] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. Acoustic event detection in real life recordings. In *2010 18th European Signal Processing Conference*, pages 1267–1271, 2010.

[65] Liwei Lin, Xiangdong Wang, Hong Liu, and Yueliang Qian. Guided learning for weakly-labeled semi-supervised sound event detection. URL: http://arxiv.org/abs/1906.02517, arXiv:1906.02517[cs,stat].

[66] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. URL: http://arxiv.org/abs/2207.02696, arXiv:2207.02696[cs].

[67] Romain Serizel, Nicolas Turpault, Ankit Shah, and Justin Salamon. Sound event detection in synthetic domestic environments. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 86–90. IEEE. URL: https://ieeexplore.ieee.org/document/9054478/, doi:10.1109/ICASSP40776.2020.9054478.

[68] Hyeonuk Nam, Seong-Hu Kim, Byeong-Yun Ko, and Yong-Hwa Park. Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection. URL: http://arxiv.org/abs/2203.15296, arXiv:2203.15296[eess].

[69] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, and Mark D. Plumbley. Sound event detection: A tutorial. 38(5):67–83. URL: http://arxiv.org/abs/2107.05463, arXiv:2107.05463[eess], doi:10.1109/MSP.2021.3090678.

[70] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen. Sound event detection using spatial features and convolutional recurrent neural network. URL: http://arxiv.org/abs/1706.02291, arXiv:1706.02291[cs].

[71] Kouta Minamizawa, Yasuaki Kakehi, Masashi Nakatani, Soichiro Mihara, and Susumu Tachi. TECHTILE toolkit: a prototyping tool for design and education of haptic media. In *Proceedings of the 2012 Virtual Reality International Conference*, pages 1–2. ACM. URL: https://dl.acm.org/doi/10.1145/2331714.2331745, doi:10.1145/2331714.2331745.

[72] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

[73] Jeff Sauro and Joseph S. Dumas. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 1599–1608, New York, NY, USA, 2009. Association for Computing Machinery. URL: https://doi.org/10.1145/1518701.1518946, doi:10.1145/1518701.1518946.

Appendices

A. Code for Automatic Generating Haptic Video

Calculate Saliency of Input Video

```
Input_video = r"C:\Users\issac\Documents\ML\badminton.mp4"
cap = cv2.VideoCapture(Input_video)
fps = cap.get(cv2.CAP_PROP_FPS)
cap.set(cv2.CAP_PROP_FPS,30)

threshold = 3
print("FPS:", fps)
saliency = None
count = 0
name = r"C:\Users\issac\Documents\ML\output"
# create a motion saliency object
fourcc = cv2.VideoWriter_fourcc(*'mp4v')

ret, frame = cap.read()
height, width, _ = frame.shape
out = cv2.VideoWriter(name+'.mp4',fourcc,fps,(width, height),0)
out.write(frame)
while True:
    # read a frame from the input video

    print("loading",end='\r')
    ret, frame = cap.read()
    if not ret:
        break
    if saliency is None:
```

```
        print(frame.shape[1],',',frame.shape[0])
        saliency = cv2.saliency.MotionSaliencyBinWangApr2014_create()
        saliency.setImagesize(frame.shape[1], frame.shape[0])
        print(frame.shape[1],',',frame.shape[0])
        saliency.init()
        saliency_update = False

    print("Calculating Saliency",end='\r')
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    (success, saliencyMap) = saliency.computeSaliency(gray)
    saliencyMap = (saliencyMap * 255).astype("uint8")
    meannn = np.mean(saliencyMap)

    out.write(saliencyMap)


    print("Calculating Saliency...",end='\r')



out.release()
cap.release()
```

Import Trained YOLOv8 model and Calculate Shuttle's Motion State and Record in Dataframe.

```
model = YOLO("Yolov8/OnlyBadminton.pt")
count =0
frame_num=0
df = pd.DataFrame(columns=['Frame', 'Blank','Center_X','Center_Y','Sec'])
results = model.predict(source=r"C:\Users\issac\Documents\ML\output.mp4", save =False,
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(r'C:\Users\issac\Documents\ML\crop_mask'+'.mp4',fourcc,fps,(width
for result in results:
```

```
    isBlank = "Blank"
    isBadminton = "Badminton"
    ret, frame = cap.read()
    mask = np.ones(result.orig_img.shape,dtype= result.orig_img.dtype)

    mask[:,:] = [255,255,255]

    for bbox in result.boxes.xyxy:
        x1, y1, x2, y2 = bbox[0].item(), bbox[1].item(), bbox[2].item(), bbox[3].item()
        crop_image=result.orig_img[int(y1):int(y2),int(x1):int(x2)]
        mask[int(y1):int(y2),int(x1):int(x2)] = crop_image
        meancrop = np.mean(crop_image)

        center_x = (x1 + x2) / 2
        center_y = (y1 + y2) / 2

    maskmean = np.mean(mask)

    if maskmean ==255.0 :
        df.loc[frame_num] = [frame_num, isBlank,0,0,frame_num/fps]
    else:
        df.loc[frame_num] = [frame_num,isBadminton, center_x,center_y,frame_num/fps]
    frame_num+=1


    out.write(mask)


window_size_ascend = 3
############################Calculate X#################################
mask_X = (df['Center_X'] == df['Center_X'].shift()) | (df['Center_X'] == 0)
df.loc[mask_X, 'Center_X'] = None
df['Center_X'].fillna(method='ffill', inplace=True)
diff_X = df['Center_X'].diff(periods=window_size_ascend)
threshold = 5
df['Direction_X'] = None
```

```
df['ShiftedDirection_x'] = ''
df['Change_Point_X'] = ''
df.loc[diff_X > 0, 'Direction_X'] = 'Right'
df.loc[diff_X < 0, 'Direction_X'] = 'Left'
last_nonzero_direction_X = None
for i, row in df.iterrows():
    if pd.isna(row['Direction_X']):
        df.at[i, 'Direction_X'] = last_nonzero_direction_X
    else:
        last_nonzero_direction_X = row['Direction_X']
df['ShiftedDirection_X'] = df['Direction_X'].shift(1)
change_point_X = df[((df['Direction_X'] == 'Left') & (df['ShiftedDirection_X'] == 'Righ
                    ((df['Direction_X'] == 'Right')& (df['ShiftedDirection_X'] == 'Left
                    |(((df['Center_X'] - df['Center_X'].shift(1)).abs()) > 50)
                    ].index.tolist()
df.loc[change_point_X,'Change_Point_X'] = 'Changed_X'


############################Calculate Y#################################
mask_Y = (df['Center_Y'] == df['Center_Y'].shift()) | (df['Center_Y'] == 0)
df.loc[mask_Y, 'Center_Y'] = None
df['Center_Y'].fillna(method='ffill', inplace=True)
diff_X = df['Center_Y'].diff(periods=window_size_ascend)
threshold = 3
df['Direction_Y'] = None
df['ShiftedDirection_Y'] = ''
df['Change_Point_Y'] = ''
df.loc[diff_X > 0, 'Direction_Y'] = 'Up'
df.loc[diff_X < 0, 'Direction_Y'] = 'Down'
last_nonzero_direction_Y = None
for i, row in df.iterrows():
    if pd.isna(row['Direction_Y']):
        df.at[i, 'Direction_Y'] = last_nonzero_direction_Y
    else:
        last_nonzero_direction_X = row['Direction_Y']
df['ShiftedDirection_Y'] = df['Direction_Y'].shift(1)
```

```
change_point_Y = df[((df['Direction_Y'] == 'Down') & (df['ShiftedDirection_Y'] == 'Up')
df.loc[change_point_Y,'Change_Point_Y'] = 'Changed_Y'
print(df)
out.release()
```

Import Trained Sound Event Detection Model and Predict Sound Events

```
file_count = 1
sr = 16000
n_mels=32
datalist =[]
startlist =[]
endlist = []
durationlist = []
time_resolution = 0.10
batch_size=4
samplerate = 16000
window_duration = 0.801
window_length = int(window_duration / time_resolution)

def next_power_of_2(x):
    return 2**(math.ceil(math.log(x, 2)))

hop_length = int(time_resolution*samplerate)
n_fft = next_power_of_2(hop_length)

audiofile,spec = sed.LoadFromVid(Input_video,sr,n_mels,n_fft,hop_length)
fig, ax = plt.subplots(1, figsize=(20, 5))
plot_spectrogram(hop_length,samplerate,ax,spec)
append_windows = pd.DataFrame()
windows = pd.DataFrame({
    'spectrogram': sed.crop_windows(spec, frames=window_length, step=2),
    'file': audiofile,
})
append_windows = pd.concat([append_windows,windows],axis=0)
```

```
splitData = sed.split_data2(append_windows)

with custom_object_scope( {'weighted_binary_crossentropy':weighted_binary_crossentropy}

    model = load_model(r'C:\Users\issac\Documents\ML\Badminton_sound\model\onlyfinal(60
model.summary()

Xm = np.expand_dims(np.mean(np.concatenate([s.T for s in splitData.spectrogram]), axis=
predictions = predict_spectrogram(model,spec,window_length,Xm)
fig, ax = plt.subplots(1, figsize=(30, 5))
sed.plot_spectrogram(hop_length,samplerate,ax, spec, predictions = predictions)
annotate = events_from_predictions(predictions)
for index, row in annotate.iterrows():
    datalist.append(list(row))
startlist = [row[0] for row in datalist]
endlist = [min(row[1],frame_num-1) for row in datalist]
for i in range(len(startlist)):
    templist = [x for x in range(startlist[i],endlist[i]+1)]
    durationlist.append(templist)
df['Sound_Detect'] = 'No'
df['Haptic'] = 'No'
for i in range(len(durationlist)):
    df.loc[durationlist[i],'Sound_Detect'] = 'Hit'
for index, row in df.iterrows():
    if row['Sound_Detect'] == 'Hit' and row['Blank'] == 'Blank':
        df.loc[index,'Haptic'] = 'Yes'
sounddf = df[['Frame','Change_Point_X','Change_Point_Y']].copy()

sounddf = caluculate_db(Input_video,sounddf)
Create_Audio_Sounddf(sounddf)
EncodeAudioChannel(Input_video)
Combine_Vid_Audio()
plt.show()
```

Utilized Functions for Automatic Generating Haptic Video

```
def LoadCsv(path):
    df = pd.read_csv(path, header=None, skiprows=1, names=['event', 'start', 'end', 'fi
    df['duration'] = df['end'].astype(np.float16) - df['start'].astype(np.float16)
    return df

def LoadAudio(path,Sr,mels,fft,hop_length): #0,1,2,3
    audio_files = lb.util.find_files(path)
    print(audio_files)
    Sdb_List= []
    for audio_file in audio_files:
        y, sr = lb.load(audio_file)
        Spec = lb.feature.melspectrogram(y=y, sr=Sr, n_mels=mels,n_fft=fft,hop_length=h
        Sdb = lb.power_to_db(Spec, ref=np.max)
        print("SDB shape = ",Sdb.shape)
        Sdb_List.append(Sdb)
    return audio_files, Sdb_List

def LoadFromVid(path,Sr,mels,fft,hop_length): #0,1,2,3
    audio_files = lb.util.find_files(path)
    video = VideoReader(path)
    video = mp.VideoFileClip(path)
    audio = video.audio
    audio = video.audio
    temp = "Combine_test/temp.wav"
    audio.write_audiofile(temp)

    y, sr = lb.load(temp, sr=16000)
    Spec = lb.feature.melspectrogram(y=y, sr=Sr, n_mels=mels,n_fft=fft,hop_length=hop_l
    Sdb = lb.power_to_db(Spec, ref=np.max)
    print("SDB shape = ",Sdb.shape)
    return temp,Sdb

    def plot_spectrogram(hop_length,samplerate,ax, spec, events=None, label_activations
```

```
    events_lw = 1.5

    # Plot spectrogram
    lb.display.specshow(ax=ax, data=spec, hop_length=hop_length, x_axis='time', y_axis=

    # Plot events
    if events is not None:
        for start, end in zip(events.start, events.end):
            ax.axvspan(start, end, alpha=0.2, color='yellow')
            ax.axvline(start, alpha=0.7, color='yellow', ls='--', lw=events_lw)
            ax.axvline(end, alpha=0.8, color='green', ls='--', lw=events_lw)

    label_ax = ax.twinx()

    # Plot event activations
    if label_activations is not None:
        a = label_activations.reset_index()
        a['time'] = a['time'].dt.total_seconds()
        label_ax.step(a['time'], a['event'], color='green', alpha=0.9, lw=2.0)

    # Plot model predictions
    if predictions is not None:
        p = predictions.reset_index()
        p['time'] = p['time'].dt.total_seconds()
        label_ax.step(p['time'], p['probability'], color='blue', alpha=0.9, lw=3.0)

        label_ax.axhline(0.5, ls='--', color='black', alpha=0.5, lw=2.0)

def crop_windows(arr, frames, pad_value=0.0, overlap=0.5, step=None):
    if step is None:
        step = int(frames * (1-overlap))

    windows = []
    index = []

    width, length = arr.shape
```

```
    for start_idx in range(0, length, step):
        end_idx = min(start_idx + frames, length)

        # create emmpty
        win = np.full((width, frames), pad_value, dtype=float)
        # fill with data
        win[:, 0:end_idx-start_idx] = arr[:,start_idx:end_idx]  # crop a fix frame from

        windows.append(win)
        index.append(start_idx)

    s = pd.Series(windows, index=index)
    s.index.name = 'start_index'
    return s



def plot_windows( wins,hop_length, samplerate, col_wrap=None, height=4, aspect=1):
    specs = wins.spectrogram

    nrow = 1
    ncol = len(specs)
    if col_wrap is not None:
        nrow = int(np.ceil(ncol / col_wrap))
        ncol = col_wrap

    fig_height = height * nrow
    fig_width = height * aspect * ncol
    fig, axs = plt.subplots(ncol, nrow, sharex=True, sharey=True, figsize=(fig_width, f
    axs = np.array(axs).flatten()

    fig.suptitle(specs.name)
    for ax, s, l in zip(axs, specs, wins.labels):

        l = np.squeeze(l)
```

```
        ll = pd.DataFrame({
            'event': l,
            'time': pd.to_timedelta(np.arange(l.shape[0])*hop_length/samplerate, unit='
        })

        plot_spectrogram(hop_length,samplerate,ax, s, label_activations=ll)

def events_from_predictions(pred, threshold, label='Hit', event_duration_max=1.0,fps =3
    import copy

    event_duration_max = pd.Timedelta(event_duration_max, unit='s')

    events = []
    inside_event = False
    event = {
        'start': None,
        'end': None,
    }

    for t, r in pred.iterrows():
        p = r['probability']

        if not inside_event and p > threshold:
            event['start'] = int(t.total_seconds()  *fps)  #Modify here to calculate wh
            inside_event = True

        elif inside_event and ((p < threshold) or ((t - pd.Timedelta(seconds=event['sta
            event['end'] = int(t.total_seconds() *fps)
            events.append(copy.copy(event))

            inside_event = False
            event['start'] = None
            event['end'] = None
        else:
            pass
```

```
    if len(events):
        df = pd.DataFrame.from_records(events)
    else:
        df = pd.DataFrame([], columns=['start', 'end'], dtype='timedelta64[ns]')
    df['label'] = label
    return df

def predict_spectrogram(model, spec,window_length ,Xm):

    window_hop = 1
    wins = crop_windows(spec, frames=window_length, step=window_hop)
    X = np.expand_dims(np.stack( [ (w-Xm).T for w in wins ]), -1)

    y = np.squeeze(model.predict(X, verbose=False))

    out = merge_overlapped_predictions(y, window_hop=window_hop)

    return out
```

B. Code for Combining Haptic, Video and Audio Components

```
import librosa as lb
import subprocess
import numpy as np

video_path = "/Users/issacfei/Documents /Experiment/Materials/badminton.mp4"
audio_path = cs.EncodeAudioChannel(video_path)
cs.Combine_Vid_Audio(audio_path)

def calculate_dB_frame(frame, ref_level=1e-10):
    spectrogram = lb.stft(frame)
    rms = lb.feature.rms(S=spectrogram)
```

```
    dB = lb.amplitude_to_db(rms,ref=ref_level)
    avg_dB = np.mean(dB)
    return avg_dB
def caluculate_db(path,df):
    video_file = path
    sample_rate = 44100
    duration = 1/30
    db_values = []
    audio, sr = lb.load(video_file, sr=44100)
    for i, row in df.iterrows():
        frame_start = i * int(sample_rate * duration)
        frame_end = (i + 1) * int(sample_rate * duration)
        db_value = calculate_dB_frame(audio[frame_start:frame_end])
        db_values.append(db_value)
    df['dB'] = db_values
    return df

def separate_video_and_audio(path):
    video_file = path
    output_video = "/Users/issacfei/Documents/Experiment/src/badminton_nsound.mp4"
    output_audio1 = "/Users/issacfei/Documents/Experiment/src/output_audio1.aac"
    output_audio2 = "/Users/issacfei/Documents/Experiment/src/output_audio2.aac"

    # Separate video from the input file
    video_command = [
        "ffmpeg",
        "-i", video_file,
        "-c:v", "copy",
        "-an",  # Disable audio
        '-y',
        output_video
   ]
    # Separate audio channels from the input file
    audio_command1 = [
        "ffmpeg",
        "-i", video_file,
```

```
        "-map", "0:a:0",  # Select audio channel 1
        "-c:a", "copy",
        '-y',
        output_audio1
    ]
    audio_command2 = [
        "ffmpeg",
        "-i", video_file,
        "-map", "0:a:1",  # Select audio channel 2
        "-c:a", "copy",
        '-y',
        output_audio2
    ]
    subprocess.run(video_command)
    subprocess.run(audio_command1)




def EncodeAudioChannel(path ):
    original_video = path

    audio1 = "/Users/issacfei/Documents/Experiment/src/output_audio1.aac"
    audio_originalL = "/Users/issacfei/Documents/Experiment/src/left.wav"
    audio_originalR = "/Users/issacfei/Documents/Experiment/src/right.wav"
    haptic_audio = "/Users/issacfei/Documents/Experiment1/Annotated_manual.mp3"  ######
    audio_output ="/Users/issacfei/Documents/Experiment/src/audio_output.wav"

    #Seperate audio and video first then merge
    separate_video_and_audio(original_video)
    #Split the stereo into two mono
    Split_Stereo(audio1)

    ffmpeg_cmd = [
    'ffmpeg',
```

```
    '-i', audio_originalL,
    '-i', audio_originalR,
    '-i', haptic_audio,
    '-i', haptic_audio,
    '-filter_complex', '[0:a][1:a][2:a][3:a]join=inputs=4:channel_layout=quad[a]',
    '-map', '[a]',
    '-y',
    audio_output
]
    subprocess.run(ffmpeg_cmd)

    return audio_output

def Split_Stereo(path):
    ffmpeg_cmd = [
        'ffmpeg',
        '-i', path,
        '-filter_complex', '[0:a]channelsplit=channel_layout=stereo[left][right]',
        '-map', '[left]','-y', "/Users/issacfei/Documents/Experiment/src/left.wav",
        '-map', '[right]','-y', "/Users/issacfei/Documents/Experiment/src/right.wav"
    ]
    subprocess.run(ffmpeg_cmd)
    return


def Combine_Vid_Audio(audio_path):
    output = "/Users/issacfei/Documents/Experiment/Experiment1/Final_Output/Vid_Haptic_
    split_vid = "/Users/issacfei/Documents/Experiment/src/badminton_nsound.mp4"
    audio = audio_path
    ffmpeg_cmd = ['ffmpeg',
                  '-i', split_vid,
                  '-i', audio,
                  '-c:v','copy',
                  '-map',' 0:v:0',
                  '-map', '1:a:0',
                  '-shortest',
```

```
                '-y',
                output]
    subprocess.run(ffmpeg_cmd)
    return
```