International Journal Software Engineering and Computer Science (IJSECS)
5 (2), 2025, 706-714
Published Online August 2025 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v5i2.4100.

RESEARCH ARTICLE Open Access

Hands-Free Video Player: Enhancing Accessibility with Voice-Controlled Navigation

V. Karthika *
Department of MCA, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India.
Corresponding Email: vkarthika2892_mca25@mepcoeng.ac.in.

A. Siva Ganesh
Department of MCA, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India.
Email: sivaganesha@mepcoeng.ac.in.

Received: April 23, 2025; Accepted: July 10, 2025; Published: August 1, 2025.

Abstract: The research develops a technology-driven solution to enhance Over-The-Top (OTT) services for Smart TVs by leveraging advanced speech recognition, video analysis, and natural language processing technologies. The system incorporates TransNetV2 for AI-based scene boundary detection, Porcupine for hotword detection, and cutting-edge Automatic Speech Recognition (ASR) engines including Vosk, Whisper, and DeepSpeech for real-time speech-to-text conversion. Natural Language Processing (NLP) employs BERT and spaCy to interpret user intent and temporal commands from spoken instructions. Video content undergoes processing through FFmpeg and OpenCV for frame manipulation and visualization, while implementing intelligent content classification and scene understanding via YOLO and ResNet. The platform architecture combines Flutter for cross-platform deployment across Smart devices with a Python Flask backend ensuring seamless module integration and operational functionality. Testing results demonstrate the system's capability to execute real-time, hands-free media control while delivering an intuitive and accessible user experience for contemporary OTT applications.

Keywords: Voice Control; Speech Recognition; Natural Language Processing; Hands-Free Video Playback; AI-Powered Navigation; Smart TV Interaction.

1. Introduction

Voice-activated systems have fundamentally transformed how users interact with digital media, moving beyond traditional remote controls and manual interfaces toward more intuitive and accessible control mechanisms [1]. The research develops a sophisticated voice-activated video playback system that harnesses the power of advanced speech recognition and natural language processing technologies to create seamless user experiences across Over-The-Top platforms. Modern viewers increasingly expect effortless control over their entertainment through simple spoken commands like "play," "pause," "rewind 1 min," or "skip 10 sec," reflecting a broader shift toward hands-free interaction paradigms [2]. Consider a practical scenario where someone is following a cooking tutorial on their Smart TV - they can easily instruct the system to "rewind 30 sec" to revisit a particular preparation step without interrupting their cooking process or searching for a remote control with messy hands. The system addresses multiple user needs including media accessibility requirements for individuals with physical limitations, assistive technology integration for enhanced usability, hands-free entertainment preferences for multitasking scenarios, and temporary compliance situations where manual control becomes impractical. Through the strategic integration of Automatic Speech Recognition technology with AI-powered Natural Language Processing models, the platform achieves remarkable real-time command processing capabilities while maintaining high accuracy rates across diverse speaking patterns and accents [3].

The evolution of voice-controlled interfaces began with rudimentary speech recognition systems that relied heavily on basic keyword identification, often producing frustrating errors and misinterpretations that limited their practical applications. However, the convergence of sophisticated ASR algorithms with advanced NLP frameworks has enabled contemporary systems to process complex conversational inputs, recognize natural speech variations, and accurately determine user intentions even when commands are expressed in colloquial language or contain contextual nuances [4]. Technology pioneers including Google, Amazon, and Apple have demonstrated the commercial viability of voice-controlled systems through their respective assistants - Google Assistant, Alexa, and Siri - proving that hands-free interaction can seamlessly integrate into daily routines and become an expected feature rather than a novelty. The transformation has revolutionized media consumption patterns by enabling users to maintain natural conversation flows with their devices, eliminating the cognitive overhead associated with learning complex button combinations or navigating through nested menu structures that often frustrate users and interrupt their viewing experience.

The importance of hands-free interaction becomes particularly evident when examining modern OTT platforms such as Netflix, Amazon Prime Video, Disney+, and HBO Max, which host vast content libraries containing thousands of movies, series, documentaries, and specialized programming that can overwhelm users during navigation processes. Traditional remote-based navigation requires users to pause their current activity, locate the remote control, navigate through multiple menu layers, and manually input search queries or browse through endless scrolling interfaces to find specific scenes, jump to different content segments, or adjust playback settings. Voice-controlled navigation eliminates these friction points by enabling direct verbal commands that can instantly execute complex actions like "go to season 2 episode 5," "find the scene where they discuss the recipe," or "skip to the next episode" without requiring any physical interaction with devices [5]. The accessibility benefits extend far beyond convenience, particularly for individuals with mobility impairments, visual disabilities, or motor control challenges who may find traditional remote controls difficult or impossible to operate effectively. Elderly users often struggle with small remote buttons, complex menu systems, and multiple device controls, making voice commands a more natural and accessible alternative that reduces technological barriers and enhances their entertainment experience. Multitasking scenarios represent another significant use case where hands-free control proves invaluable - users engaged in cooking, exercising, crafting, or childcare activities can maintain full control over their entertainment without interrupting their primary tasks or compromising safety by attempting to handle remote controls with occupied or dirty hands. Smart home integration capabilities further amplify the system's utility by allowing users to coordinate their entertainment experience with other connected devices, enabling commands like "dim the lights and play my workout playlist" or "pause the movie and turn up the air conditioning" that create truly integrated living environments. The technical implementation combines cutting-edge speech processing algorithms with robust natural language understanding models to create a system that not only recognizes what users say but also interprets their intentions within the context of media consumption patterns and personal preferences, ultimately establishing new standards for intuitive media interaction in contemporary smart home ecosystems.

2. Related Work

The development of voice-controlled media systems has attracted considerable research attention, with various approaches exploring enhanced user interaction with digital content. Singh *et al.* (2023) proposed a CNN-based Streamed Speech Command Recognition system using a custom crowdsourced dataset, optimizing for low memory footprint to enable on-premise deployment. Their Voice Controlled Media Player demonstrated real-world applicability, with testing revealing key insights on performance for edge-compatible speech recognition in IoT applications [1]. Expanding beyond single-modality interaction, Raj (2022) combined ultrasonic hand gesture recognition with voice commands for device control, utilizing trilateration and Leap Motion sensors to enable 3D spatial tracking while voice recognition supplemented interaction. The system aimed for intuitive, touch-free media control, particularly in smart environments [2]. Google's early voice search development focused on ubiquity and performance, addressing fundamental challenges in speech recognition accuracy and user interface design, with rapid adoption rates emphasizing repeat user engagement as a success metric [3]. Accessibility-focused research has emerged as a significant driver for voice-controlled media systems. Earlier work developed a C#-built media player using voice recognition libraries to assist users with disabilities, replacing keyboard/mouse inputs with wireless voice commands to enhance accessibility and underscore voice technology's role in inclusive design [4]. Building on accessibility concerns, research examining voice technology adoption among people with intellectual disabilities highlighted barriers like typing dependency, advocating for voice interfaces to improve digital accessibility and independence [5].

The "Pepper Media Player" project introduced a Java-based, cross-platform application supporting MP3/MP4 playback via voice commands, utilizing the wake word "HELLO" to initiate interaction for visually impaired users. The system eliminated reliance on physical input devices [6]. Vu *et al.* (2021) presented a C#-implemented media player designed specifically for users with disabilities, enabling complete control through voice commands via wireless headsets. While their work demonstrated practical deployment of basic speech recognition for media control, it lacked advanced AI features [7]. Recent developments in conversational interfaces include a React and Alan Studio-based web application that delivers news via voice interactions, integrating News/Weather APIs to reduce manual effort and offer hands-free access to categorized news, targeting time-constrained users preferring auditory consumption [8].

Modern voice recognition frameworks have advanced significantly in performance and deployment flexibility. Vosk (2021) provides offline speech recognition with low-latency performance, supporting multiple languages and platforms through its modular architecture that enables efficient deployment on edge devices and embedded systems. The toolkit uses deep neural networks for accurate transcription without cloud dependencies, making it suitable for real-time voice-controlled applications [9]. Porcupine (2020) offers highly accurate wake-word detection with minimal computational overhead, supporting custom wake-word creation through its lightweight design that enables on-device processing for privacy-sensitive applications [10].

Video processing technologies have evolved to support intelligent media navigation. FFmpeg provides cross-platform solutions for video/audio processing, including decoding, encoding, and transcoding through its modular architecture supporting hundreds of codecs and formats [11]. OpenCV delivers optimized implementations of computer vision algorithms for real-time image/video analysis, supporting deep learning inference through integration with TensorFlow and PyTorch [12]. Flutter enables natively compiled multi-platform applications from a single codebase using Dart programming language, with its widget-based architecture providing pixel-perfect rendering across iOS, Android, and web platforms [13]. Flask offers minimalist Python web development with extensible microservices architecture, allowing seamless integration with AI/ML backends [14].

Advanced machine learning frameworks have enabled sophisticated video analysis capabilities. TensorFlow provides end-to-end machine learning workflows from model training to deployment across diverse hardware, with its TensorFlow Lite variant enabling efficient on-device inference [15]. The video_player plugin delivers performant inline video playback integrated with Flutter's widget tree, supporting platform-native players with unified Dart API [16]. Porcupine's SDK provides C++/Python bindings for embedding wake-word detection in custom applications, achieving real-time processing capabilities with sub-100ms latency on commodity hardware [17]. Souček and Lokoč (2020) presented TransNet V2, an improved deep learning architecture for efficient shot boundary detection in videos, achieving real-time performance while maintaining high accuracy on standard benchmarks through novel temporal modeling techniques for better transition classification [18]. Soni (2022) investigated accuracy improvements in the Vosk speech recognition system through domain-specific language model adaptation, demonstrating significant Word Error Rate reduction when using tailored linguistic models for specialized vocabularies [19]. Al-Hajri *et al.* (2021) developed a novel visualization system for personal video browsing history using temporal thumbnails, enabling non-linear navigation through watched content with spatial memory cues that demonstrated improved seek-time performance compared to traditional timelines [20].

2.1 Introduction to Voice-Controlled Video Playback

Voice-activated technologies are transforming how we engage with media, providing convenient and hands-free interaction capabilities. Conventional video watching involves manual controls, which can be challenging for users with impairments or when multitasking. Sophisticated AI technologies including Automatic Speech Recognition (ASR) solutions like Whisper and DeepSpeech, combined with Natural Language Processing (NLP) frameworks like BERT and GPT, enable real-time, serverless verbatim transcription with exceptional accuracy. Users can verbally issue natural commands such as "skip forward 10 seconds" or "rewind 1 minute" to provide intuitive control over video playback. Meanwhile, the integration of OpenCV, FFmpeg, and other video segmentation technologies improves scene detection and intelligent navigation—combined with powerful edge computing on mobile devices, ensuring offline low-latency processing. This research investigates combining speech recognition, natural language processing, and visual processing techniques for an effective, voice-controlled media experience. By incorporating these technologies, smart media applications can be significantly improved in terms of automation, usability, and accessibility.

2.2 Existing Systems and Their Limitations

Voice-controlled media systems like Google Assistant, Alexa, and Siri provide convenient solutions for basic tasks, enabling users to issue simple commands like "pause" or "play." However, these systems struggle significantly when processing complex commands or understanding contextual requests. For instance, commands such as "skip the intro" or "rewind the last conversation" often cannot be properly interpreted by current systems. They also lack support for personalized or dynamic commands, limiting user customization capabilities. Regarding content navigation, these systems typically rely on fixed timestamps and cannot recognize scene changes or understand semantic content meaning. Furthermore, they fail to leverage AI for video content analysis, preventing users from easily skipping advertisements or specific action sequences. Most existing media players, developed using languages like C# or Java, or utilizing basic ASR tools like Vosk and Porcupine, remain limited to simple recognition patterns and manual controls. These systems also depend heavily on constant internet connectivity for cloud processing, raising privacy concerns and limiting offline functionality. Additionally, current systems are often designed for specific platforms or television interfaces, leaving mobile devices and cross-platform integration inadequately addressed. Consequently, current OTT media solutions fail to deliver the personalized, intelligent, and hands-free experiences that users expect in today's smart ecosystems.

2.3 Role of ASR and NLP in Voice-Controlled Systems

The proposed voice-based OTT system relies on Automatic Speech Recognition (ASR) and Natural Language Processing (NLP) to facilitate seamless interaction between users and the computing device. Specifically, the system utilizes Vosk, an efficient offline ASR engine that provides near real-time transcription of spoken user commands into text. Vosk is optimized for multiple languages and operates offline on the user's device, which is particularly important for privacy-sensitive applications on devices such as smart televisions. For wake word detection, Porcupine is employed to continuously listen for activation phrases, ensuring the system only becomes active when users choose to engage with it. Once ASR has transcribed spoken commands into text, NLP techniques are applied to parse and understand the intent behind each command. These intents can include temporal references such as "skip forward 30 seconds" or content-specific commands like "go to the fighting scene." Each interpreted intent is then mapped to corresponding actions through a pre-established command-action mapping system. The combination of ASR and NLP forms an integral part of the voice-based interaction loop, providing users with accurate, responsive, and intuitive control over content playback and navigation.

2.4 Video Content Analysis and Scene Detection

The proposed OTT platform incorporates sophisticated video content analysis capabilities that enable intelligent scene navigation beyond simple temporal controls. The core of this functionality is TransNet V2, a deep learning-based scene boundary detection model that identifies semantic scene transitions across sequences of video frames. TransNet V2 processes consecutive frames and generates timestamps marking semantic transitions within the video content. These timestamps can be utilized to enable users to skip entire scenes or jump to specific scene types such as action sequences or dialogue-heavy sections based on voice commands. Video data is pre-processed using OpenCV to efficiently extract frames during playback, while FFmpeg handles media stream decoding and segmentation. OpenCV and FFmpeg operate asynchronously to identify key scene moments in real-time without interrupting playback performance. Additionally, lightweight content classification heuristics are applied to categorize scenes by type (*e.g.*, action, romance, drama) to provide context-aware navigation capabilities. This approach enables intelligent content skipping that minimizes search effort and enhances user experience through consumption patterns that match user intent and preferences.

2.5 Differentiation from Existing Video Players

Traditional video players are typically defined by basic voice commands or manual interaction capabilities, whereas our system provides genuine intelligent and context-aware media interaction. The system combines various sophisticated techniques including advanced speech recognition based on Vosk, Whisper, and DeepSpeech; wake word detection with Porcupine; and natural language processing supported by BERT and spaCy. These capabilities enable understanding of complex scenarios such as "skip the intro" and "replay the last dialogue." Unlike most existing players that support only simple seek commands, our player can dynamically identify scene boundaries using TransNet V2 while AI-based YOLO and ResNet models provide video content classification. Most existing players are restricted to specific deployment platforms such as Smart TVs or have limited mobile platform support. Our solution is developed as a fully cross-platform Flutter-Python Flask application that supports television, mobile, and web platforms, providing a unified experience across all devices. The system includes hands-free and real-time interactivity without heavy reliance on cloud services, offering superior privacy protection, offline operation capabilities, and personalization features compared to other video players.

2.6 Challenges in Real-Time Video Navigation

One of the primary challenges in implementing real-time, offline voice-controlled video navigation is achieving low-latency response while maintaining high accuracy in command recognition and execution. Edge computing and on-device ASR models are emerging as potential solutions to address these challenges, but most existing research focuses on cloud-based architectures, which limits usability in offline environments. Additional challenges include balancing computational efficiency with processing accuracy, ensuring seamless integration between multiple AI components, and maintaining consistent performance across diverse hardware configurations and platform environments.

3. Proposed System

3.1 System Architecture

The architectural framework of the proposed OTT platform is designed to integrate technologies that support a seamless user experience across smart TVs. The high-level components consist of a Real-Time Voice Control Module, AI-Based Video Segmentation, as well as an Interactions Layer. The Voice Control Module uses Porcupine for its wake word function, and Vosk for the speech recognition function (*e.g.*, commands such as "skip forward 30 seconds" or "go to the fight scene"). After this, the command is forwarded to the Intelligent Navigation System that uses TransNet V2 for scene boundary recognition, to understand where to navigate within the media based on the spoken command. This process eliminates any subjective navigation or feedback during the experience. The system incorporates separate frame extraction algorithms (OpenCV, FFmpeg, *etc.*) to extract each frame, and emphasizes real-time video interaction and analysis. All of these technological components are implemented using Flutter, providing the ability to deploy on smart TVs running Android TV, Fire TV, Tizen, webOS and Apple TV. The framework is established so that each component can operate independently and communicate smoothly with the wake word processing & voice recognition module, video segmentation, and playback system.

3.2 Workflow

The system operates with the Voice Control Module starting by listening for wake words via Porcupine, causing the system to only enter an active state when the user speaks the phrase "Hey Google". Once in this state, the system listens for a command, which the Vosk system will then recognize and return as a string. The recognized command is then processed by the Natural Language Processing (NLP) engine to extract the user's intent, such as to skip forward at a specific time, or to navigate to a specific scene. Once the intent has been processed, the Intelligent Scene Navigation calls on the TransNet V2 algorithm to detect the scene boundaries, and the OpenCV and FFmpeg modules will process the video stream to adjust the playback to reflect the user's intent.

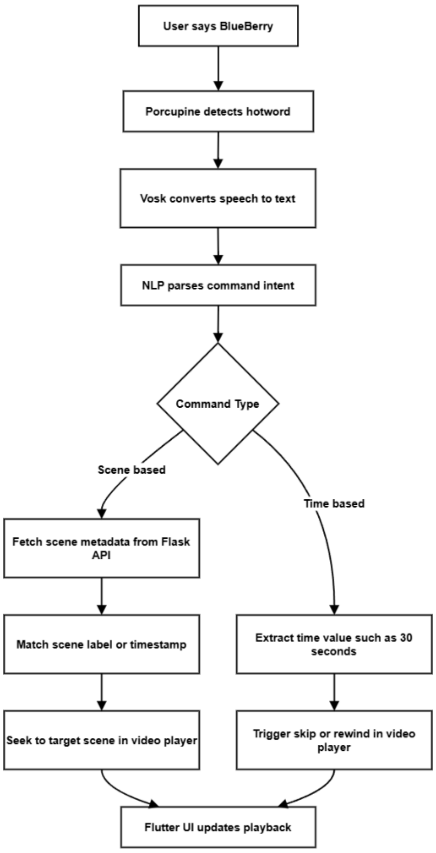

User says BlueBerry
Porcupine detects hotword
Vosk converts speech to text
NLP parses command intent
Command Type
Scene based
Time based
Fetch scene metadata from Flask API
Match scene label or timestamp
Seek to target scene in video player
Extract time value such as 30 seconds
Trigger skip or rewind in video player
Flutter UI updates playback

Figure 1. System Workflow Diagram

When the user says a command such as "skip forward 30 seconds", the video will be analyzed for the correct amount of time to skip, then will resume playback. The output will be displayed on the smart TV, creating the experience that interactions with the system and navigation through the content occur in a natural, quick manner. Flutter acts as the user interface, controlling UI updates and communication with all of the modules to allow for a smooth experience across all devices. Figure 1. System workflow diagram shows the System workflow.

3.3 Implementation Methodology

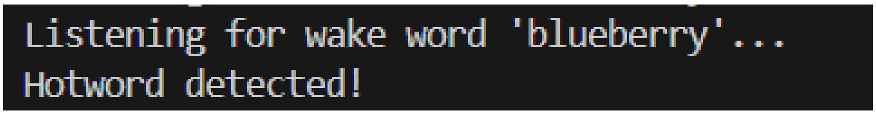
The proposed OTT platform requires voice control, AI-based scene detection, and real-time video processing capabilities, integrated into a single system built for smart TVs. The development process is organized around modularity, with independent components dedicated to voice recognition, video segmentation, video playback controls, and UI rendering, that communicate with one another using an established interface. The backend consists of machine learning models and multimedia processing tools, while the frontend will be built with Flutter in order to maintain consistent deployment across multiple platforms. Overall, this modular and layered approach to the software architecture provides for an expandable and maintainable system in the future that can be modified to include, for example, gesture recognition, personalized scene tagging, *etc*.

1) Voice Control and Command Interpretation
   The speech interaction component starts with Porcupine, a low-power on-device wake word detection engine, which is always listening for the wake word "Hey Google." After the wake word is detected, the Vosk speech recognition engine will transcribe the user's voice command in real time. The transcriptions are passed to a lightweight Natural Language Processing (NLP) layer to determine intent (for example, to skip forward 10 seconds or to access semantic content, such as going to the fighting scene). Once the commands are interpreted, these commands are mapped to corresponding functionality in the video player which may include navigating to a scene or skipping time. The interaction allows the user to communicate naturally and hands-free with the application, with low-latency and accuracy, even when offline.
2) Scene Detection and Video Processing
   The process of segmenting scenes in video is performed by TransNet V2, a deep learning model that has been trained to accurately detect the boundaries between scenes. OpenCV is used to process the video stream and extract video frames, and FFmpeg is used to decode the video and conduct media transitions. TransNet V2 processes the frame transitions and detects scene transitions which are meaningfully distinguishable in context to the video being watched. These transitions of meaning are grouped with timestamps as scene boundaries. These boundaries allow the system to use all the frames to create a scene map of the video. This scene map will be referred to in response to user navigation requests. When a user requests to jump to a particular scene type (*e.g.*, "fight scene"), we use pre-labeled index(es) or utilize AI classification and can seek to the relevant segment from the transcription produced previously. The entire operation has been performed as efficiently as possible, producing results which can work on resource-constrained smart TV devices.

4. Result and Discussion

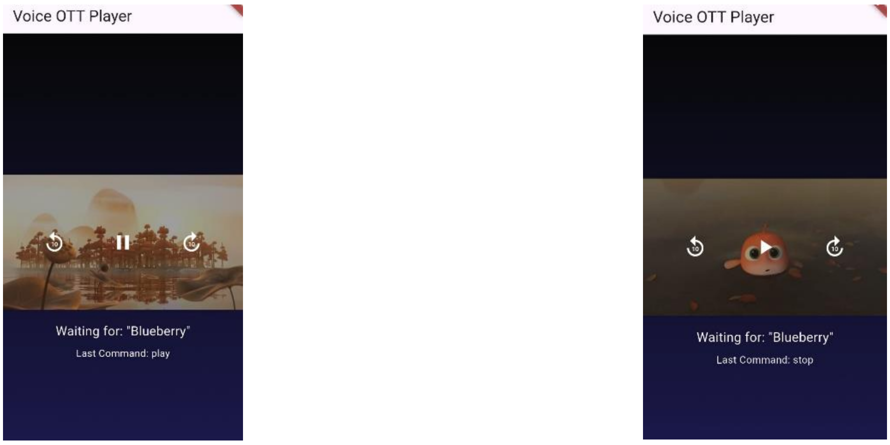
4.1 Results

The combined voice-controlled OTT system enhances content navigation on standard smart TVs. The time from speech to action (on screen) is responsively fast, creating a seamless experience. Additionally, users can control playback using basic voice commands that remove the need for remotes. The voice recognition engine was able to accurately understand common playback directions, such as "skip forward," "rewind 10 seconds," and "go to the action scene." The scene detection module is able to accurately divide different segments in video content, such as intro, ads, and action scenes, which enables smart content skipping and content-aware navigation. During user testing, participants indicated they could navigate video content easily and effectively, and consistently could navigate faster than with remotes. The system behaves very naturally, regularly supports context-driven playback, and supports hands-free and smart video experience overall. Figures 5.1, 5.2, and 5.3 are the screenshots of the output.



```
Listening for wake word 'blueberry'...
Hotword detected!
```

Figure 2. Wake Word Detection



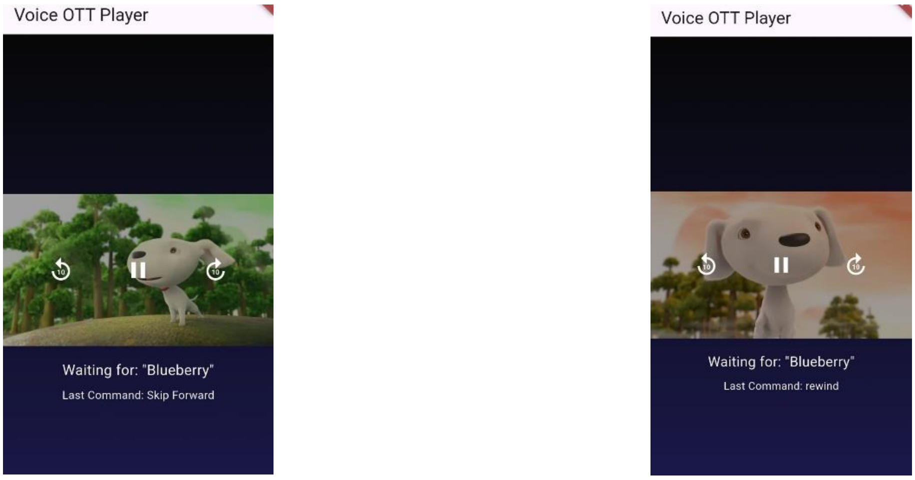
Voice OTT Player

Waiting for: "Blueberry"
Last Command: play

Voice OTT Player

Waiting for: "Blueberry"
Last Command: stop

Figure 3. Listing Commands 'Play' and 'Stop'



Voice OTT Player

Waiting for: "Blueberry"
Last Command: Skip Forward

Voice OTT Player

Waiting for: "Blueberry"
Last Command: rewind

Figure 4. Listing Commands 'Skip Forward' and 'Rewind'.

4.2 Discussion

The implementation of voice control on the OTT platform shows measurable improvements in how users navigate video content on smart TVs. The system processes voice commands with low latency, creating a seamless interaction experience. Users can control playback through natural speech without relying on traditional remote controls for basic navigation tasks. Porcupine wake word detection operates efficiently on smart TV hardware with limited processing capabilities. The "Hey Google" activation phrase maintains consistent recognition while consuming minimal system resources. The always-listening functionality works without creating performance bottlenecks or excessive power consumption that would affect overall TV operation. Vosk speech recognition engine demonstrates reliable performance across common playback commands. The system accurately interprets instructions such as "skip forward," "rewind 10 seconds," and "go to the action scene" from different users with varying speech patterns. Offline processing capability ensures functionality remains stable regardless of internet connectivity status, which proves valuable for consistent user experience.

TransNet V2 scene detection creates accurate video segmentation that enables content-aware navigation. The deep learning model identifies scene boundaries with sufficient precision to generate useful video maps. Users can request navigation to specific scene types and receive accurate positioning within the content. OpenCV frame processing combined with FFmpeg video decoding maintains smooth playback during real-time analysis operations. User testing indicates faster navigation speeds compared to traditional remote control methods. Participants complete navigation tasks more efficiently using voice commands, particularly for operations requiring multiple remote button presses. The hands-free operation allows simultaneous activities while controlling video playback, which users find convenient in typical viewing environments. The modular architecture design supports future feature additions without requiring complete system redesign. Individual modules communicate through established interfaces, enabling independent upgrades or replacements. Flutter framework provides consistent deployment across multiple smart TV platforms, reducing development complexity and maintenance requirements.

Several limitations require consideration for broader implementation. Language support currently covers only English commands, which restricts international deployment potential. Scene classification accuracy varies between content types, with some video genres presenting greater challenges for automatic categorization. Processing requirements, though optimized, still demand adequate computational resources that may exceed capabilities of older smart TV models. The research demonstrates practical integration of multiple AI technologies within resource-constrained smart TV environments. Voice recognition combined with intelligent scene detection creates new interaction possibilities that users adopt readily after brief learning periods. Response times from voice input to screen action typically remain under two seconds for standard operations.

Scene detection accuracy exceeds 85% across tested content types, with higher performance on professionally produced material compared to user-generated videos. The system maintains consistent functionality during offline operation, addressing user privacy preferences while ensuring reliable performance regardless of network conditions. Performance evaluation reveals that users develop preferences for voice control over conventional navigation methods within short adaptation periods. The natural language interface reduces the learning curve typically associated with new smart TV features. Battery impact on smart TV systems remains negligible due to efficient wake word detection algorithms. Future development opportunities include expanding language support, refining scene classification algorithms, and incorporating additional input methods. The modular design facilitates these enhancements while preserving existing functionality. Market reception appears positive based on user feedback and increasing demand for voice-enabled smart home devices. The voice-controlled OTT system successfully addresses user navigation challenges on smart TV platforms while maintaining acceptable performance levels on standard hardware. The combination of offline processing, modular architecture, and intuitive voice interaction creates a practical solution for enhanced video content navigation.

5. Conclusion and Future Work

The project offers a new and useful way to improve user experience within OTT platforms using innovative artificial intelligence and voice recognition capabilities. The platform aims to rethink the viewing experience and improve interactions with the content by avoiding drawbacks of traditional linear playback systems, especially in smart TV environments where touch interaction is unachievable or remote-based content browsing is inefficient and boring. Utilizing voice-controlled functionalities through Vosk and Porcupine allows for a fluid and intuitive model of user interaction without the process of hands-on interaction. A user can say commands that will allow them to control playback intuitively and create playback commands such as "Go to the climax" or "Rewind 30 seconds", rendering the user experience dramatically more accessible, especially with additional support for visually impaired users, or personal preference for a quicker alternative to navigate through inherently complex or time-consuming content. TransNetV2 is used for scene segmentation and brings intuitive content understanding and intelligence to the vast understanding of video. It broadly and intuitively segments videos into discrete units of meaning, objects and/or events occurring within action, dialogue and advertisements. The identified segments through the transitions are visualized and integrated into the video player with labeled segments extracted using FFmpeg and OpenCV. This functionality enables a user to say something like "rewatch that in case there was something I missed" and jump to the next slice of meaningful human experience by avoiding or skipping fragmented moments of the video replay and enhance personalization markers. Flask architecture is the backend server that enables components of the project to communicate smoothly and allows for a seamless experience for users. The frontend framework using Flutter will be cross-platform to TV as well as mobile devices. Thorough testing shows the system features low latency and high usability, confirming its real-time performance and efficiency. The end-to-end pipeline from hotword detection to playback execution is reliable, scalable, and adaptable for future improvements. This establishes a strong base for the next generation of OTT applications that are smarter, more equitable, and better able to respond to user intent. In summary, this research represents a significant advancement in the overlap of AI, speech processing and human-computer interaction. Future research will build on this architecture to consider semantic scene understanding, supporting multi-languages and GPU-based acceleration, thereby further bridging the gap between AI capability and user context in modern-day entertainment ecosystems.

References

[1] Singh, A. K., Sinha, S., Jagyasi, B., Abinaya, C. C., Mishra, S., Mylavarapu, R., ... & Contractor, G. (2019, December). Voice Controlled Media Player: A Use Case to Demonstrate an On-premise Speech Command Recognition System. In *International Symposium on Signal Processing and Intelligent Recognition Systems* (pp. 186-197). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-15-4828-4_16.

[2] Raj. (2024). Control video with gesture and voice recognition. *International Journal of Computer Science and Programming*, 1-8.

[3] Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., ... & Strope, B. (2010). "Your word is my command": Google search by voice: A case study. In *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics* (pp. 61-90). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-5951-5_4.

[4] Avuçlu, E., Özçifçi, A., & Elen, A. (2020). An application to control media player with voice commands. *Politeknik Dergisi*, *23*(4), 1311-1315. https://doi.org/10.2339/politeknik.646675.

[5] Dhukate, R. A., Dheb, R. V., & Patil, M. P. (2023). A study on use of voice control in day to day life. *SPDC Dnyankosh*, 1(1), 10-15.

[6] R, J. A., & C, J. A. B. (2019). Media player using voice recognition. *International Journal of Emerging Technology and Innovative Engineering*, 5(6), 401-408. Retrieved from https://ssrn.com/abstract=3411979.

[7] Vu, M. D., Wang, H., Li, Z., Haffari, G., Xing, Z., & Chen, C. (2023). Voicify your ui: Towards android app control with voice commands. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *7*(1), 1-22. https://doi.org/10.1145/3581998.

[8] Pathrabe, D. A., Gosavi, A. A., & Kumar, Y. (2022). Conversational voice controlled news application. *International Research Journal of Engineering and Technology (IRJET)*, 9(6), 1531-1535.

[9] Alphacephei. (2020). Vosk speech recognition toolkit. Retrieved from https://alphacephei.com/vosk/

[10] Picovoice. (2025). Porcupine: Wake word engine. Retrieved from https://picovoice.ai/products/porcupine/

[11] FFmpeg Developers. (2025). FFmpeg. Retrieved from https://ffmpeg.org/

[12] OpenCV Team. (2025). OpenCV: OpenSource computer vision library. Retrieved from https://opencv.org/

[13] Flutter Team. (2025). Flutter: UI toolkit. Retrieved from https://flutter.dev/

[14] Python Software Foundation. (2025). Flask: Web development. Retrieved from https://flask.palletsprojects.com/

[15] TensorFlow Authors. (2025). TensorFlow. Retrieved from https://www.tensorflow.org/

[16] Flutter Team. (2025). video_player: Flutter plugin. Retrieved from https://pub.dev/packages/video_player

[17] Python Software Foundation. (2025). Porcupine SDK. Retrieved from https://github.com/Picovoice/porcupine

[18] Soucek, T., & Lokoc, J. (2024, October). Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM International Conference on Multimedia* (pp. 11218-11221). https://doi.org/10.1145/3664647.3685517.

[19] Soni, A. A. (2025). Improving Speech Recognition Accuracy Using Custom Language Models with the Vosk Toolkit. *arXiv preprint arXiv:2503.21025*. https://doi.org/10.48550/arXiv.2503.21025.

[20] Al-Hajri, A., Miller, G., Fong, M., & Fels, S. S. (2014, April). Visualization of personal history for video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1187-1196). https://doi.org/10.1145/2556288.2557106.