Web Real-Time Communication Using Gesture

Vijayalaxmi Tadkal[1] , Yash Prajapat[2], Brijesh Pal[3], Harshal Patil[4], Siddhesh Mulik[5]

[1]Asst Prof Department of Computer Engineering & IETE'S Bharat College of Engineering

[2] BE Department of Computer Engineering & IETE'S Bharat College of Engineering
[3]BE Department of Computer Engineering & IETE'S Bharat College of Engineering
[4] BE Department of Computer Engineering & IETE'S Bharat College of Engineering
[5] BE Department of Computer Engineering & IETE'S Bharat College of Engineering

---***---

Abstract -Gesture recognition has emerged as a promising technology for enhancing human-computer interaction in various domains. In this paper, we propose a novel approach to gesture recognition using WebRTC (Web Real-Time Communication), a powerful web technology that enables real-time communication between web browsers. Our approach leverages the capabilities of WebRTC to capture live video streams from webcams and process them in real-time for gesture recognition. We present a detailed methodology for integrating WebRTC with a custom deep learning techniques. We evaluate the performance of our system using a diverse set of hand gestures and demonstrate its effectiveness in real-world scenarios. Our experimental results show an average recognition accuracy of 90%, outperforming existing methods in terms of both accuracy and computational efficiency. Potential applications of our approach in interactive web applications, gaming, and augmented reality systems. This work contributes the research on gesture recognition and demonstrates the feasibility of using WebRTC for real-time gesture recognition applications.

Key Words: gesture recognition, WebRTC , Real-time communication

1.INTRODUCTION

WebRTC has revolutionized real-time communication on the web, enabling peer-to-peer audio, video, and data transfer directly within web browsers without the need for additional plugins or software installations. However, the traditional interfaces for WebRTC applications often rely on conventional controls such as buttons and sliders, which may not be optimal for certain use cases, particularly in scenarios where hands-free or gesture-based interaction is desired.Gesture recognition has emerged as a compelling approach to bridge the gap between human-computer interaction and WebRTC, offering users a more intuitive and immersive communication experience. By interpreting hand movements, gestures, and poses, users can interact with WebRTC applications in a natural and intuitive manner, unlocking new possibilities for collaboration, gaming, remote assistance, and more. In this paper, we propose a comprehensive framework for integrating gesture recognition with WebRTC, leveraging recent advancements in machine learning, computer vision, and web development technologies. Our approach enables users to perform a wide range of actions, including initiating calls, adjusting audio/video settings, sharing content, and controlling virtual environments, all through simple gestures captured by standard webcams or depth-sensing devices. The existing research in gesture-based interaction and WebRTC, highlighting the limitations and opportunities for integration. Next, we describe the architecture of our proposed framework, outlining the key components such as gesture detection, feature extraction, and real-time communication protocols. We then delve into the technical challenges associated with gesture recognition in web environments, addressing issues such as latency, accuracy, and device compatibility.

2. EXISTING SYSTEM

The existing web RTC system on researching we got that there are various system available with various functionality which are available some are Jitsi Meet is an open-source video conferencing platform that utilizes WebRTC for real-time communication. It allows users to host and join video meetings via web browsers or mobile devices without requiring any account registration. Jitsi Meet supports features such as screen sharing, chat, and recordingand other is Appear.in is a simple video conferencing platform that utilizes WebRTC for browser-based communication. It allows users to create and join video rooms with custom URLs, making it easy to start ad-hoc video meetings or collaborate with colleagues and clients and the last one is Microsoft Teams is a collaboration platform that integrates WebRTC for real-time communication and collaboration. It enables users to communicate via video and audio calls, chat, and file sharing within a single interface. Microsoft Teams is widely used for remote work, team collaboration, and online meetings.

3. PROBLEM MOTIVATION

In recent years, real-time communication (RTC) systems have become increasingly essential for various applications, including video conferencing, remote collaboration, and telemedicine. While WebRTC technology has provided a robust framework for enabling browser-based peer-to-peer communication, there remains a need to enhance user interaction and experience within these systems. Traditional interfaces, relying solely on mouse and keyboard inputs, often lack intuitiveness and efficiency, particularly in scenarios where users need to multitask or operate hands-free.This project aims to address this challenge by exploring the integration of gesture recognition technology into WebRTC-based RTC systems. The goal is to empower users to control various aspects of communication, such as audio/video settings, call management, and content sharing, through intuitive gestures captured by their devices' cameras. By enabling gesture-based interaction, the proposed solution seeks to enhance usability, accessibility, and user engagement in RTC applications.

4. METHODOLOGY

HTML, or Hypertext Markup Language, is the standard language for creating and structuring content on the web ,HTML helps organize information through elements like headings, paragraphs, and lists. It allows for the inclusion of hyperlinks to navigate between sections or external sources. HTML also supports multimedia integration, ensures accessibility, aids discoverability through metadata, and enables interactivity for engaging readers. Overall, HTML serves as a fundamental tool for presenting research effectively online.

CSS for Cascading Style Sheets stands. It is a sheet-style language used to define the appearance and format of a markup document. It gives HTML a supplementary function. Used with HTML, the style of user interfaces and webpage is changed. It may also be used in XML documents of any form, including simple XUL, SVG and XML documents. In most websites, CSS is used with HTML and JavaScript to develop web-based user interfaces and user interfaces for a variety of mobile applications. What CSS accomplishes is: you can add new appearances to your old HTML pages, modify the style and feel of your website with just a few modifications to CSS code. C.S.S. is used in the creation of HTML Tags. C.S.S. is used widely used as a web language, to create a web page, we usually use H.T.M.L., C.S.S. and JavaScript. CSS is also a commonly used language in Cascading Style sheet. It allows web developers to use HTML tags for styling.

JavaScript or JS is an object oriented light weight language used for web page scripting by various online sites. The HTML document is a fully interpreted computer language allowing interactivity dynamically on web pages. In 1995, it was launched to add software to Netscape Navigator's web pages. All other graphical web browsers have been embraced since then. Users may construct contemporary web applications with JavaScript so that they can interact without refreshing the page at all times. Js is used in the conventional website for various sorts of easiness and interaction

Bootstrap is a very popular framework for constructing a responsive and mobile-friendly website for H.T.M.L., JavaScript and CSS. You may download and use it completely free of charge. A front end framework used to make web development easier and quicker. The design template for font, shapes, buttons, table, browsing, modalities, picture carousel and much more are included. The JavaScript plug-ins can also be used. It makes it easier for you to design responsively

Backend Technologies

Agora.io can be utilized within WebRTC-based applications to enhance real-time communication capabilities. While WebRTC provides the fundamental technology for peer-to-peer communication within web browsers, Agora.io's platform offers additional features and services that complement and extend WebRTC functionality. Developers can leverage Agora.io's SDKs and APIs to add advanced features such as multi-party video conferencing, interactive broadcasting, live streaming, and real-time collaboration to their WebRTC applications. Agora.io's infrastructure, including its global network of data centers, helps optimize the performance and scalability of WebRTC-based applications, ensuring high-quality audio and video streaming across diverse network conditions. By integrating Agora.io with WebRTC, developers can create robust and feature-rich communication solutions for a wide range of use cases, from online meetings and virtual classrooms to live events and social networking platforms.

Python is a high-level programming language celebrated for its simplicity, readability, and adaptability across various domains. Its intuitive syntax and dynamic typing make it easy to learn and use, appealing to beginners and experienced developers alike. Python's interpreted nature allows for interactive testing and debugging, fostering rapid development cycles. With a vast standard library and a thriving ecosystem of third-party packages, Python is well-suited for diverse applications, from web development and data analysis to artificial intelligence and scientific computing. Its cross-platform compatibility ensures seamless execution on different operating systems, while its strong community support and governance by the Python Software Foundation ensure continuous growth and evolution. Overall, Python's versatility, ease of use, and extensive resources make it a popular choice for developers tackling a wide range of projects.

Node.js is a popular open-source, cross-platform JavaScript runtime environment that executes JavaScript code server-side. It uses the V8 JavaScript engine, which is the same engine that powers Google Chrome. Node.js allows developers to write server-side code using JavaScript, which was traditionally used only for client-side scripting within web browsers.

5.ARCHITECTURE OF WEBRTC

The WebRTC (Web Real-Time Communication) architecture serves as a framework for enabling real-time communication between web browsers and other applications, incorporating various components and protocols to facilitate seamless peer-to-peer connections. At the heart of WebRTC lies the MediaStream API, which allows browsers to access audio and video streams from local devices such as webcams and microphones. Through the RTCPeerConnection interface, browsers establish direct connections with one another, managing tasks like signaling, codec negotiation, and network traversal using ICE (Interactive Connectivity Establishment) techniques. Signaling servers play a crucial role in the initial connection setup, facilitating the exchange of session control messages and SDP (Session Description Protocol) descriptions between peers. SDP describes session information like media types, codecs, and network addresses, aiding in the negotiation of session parameters. ICE protocol, alongside STUN (Session Traversal Utilities for NAT) and TURN (Traversal Using Relays around NAT) servers, helps traverse NATs and firewalls, enabling devices to establish direct communication channels. STUN servers assist in determining public IP addresses and ports, while TURN servers relay media streams when direct connections are not feasible. WebRTC supports a range of audio and video codecs, including Opus, VP8, VP9, and H.264, ensuring compatibility and optimal media quality. The architecture emphasizes security, employing encryption mechanisms like DTLS (Datagram Transport Layer Security) to protect media streams from interception and tampering. Additionally, WebRTC supports various network protocols like WebSocket and HTTPS for secure communication. The architecture's modular design allows for flexibility and extensibility, enabling developers to integrate custom functionality and third-party services. WebRTC's open standards and APIs promote interoperability across different platforms and devices, fostering a vibrant ecosystem of applications and services. The architecture's adherence to open standards promotes vendor neutrality and avoids vendor lock-in, ensuring a level playing field for developers and users. WebRTC's support for mobile devices enables seamless real-time communication experiences on smartphones and tablets, expanding its reach to a broader audience. The architecture's focus on low latency and high-quality audio and video streams enables immersive and engaging communication experiences across various use cases. WebRTC's real-time data channel allows for the transmission of non-media data, enabling applications like file sharing, gaming, and collaborative editing. The architecture's decentralized nature reduces reliance on centralized servers, improving scalability and resilience against network failures. WebRTC's support for adaptive bitrate streaming allows for dynamic adjustment of media quality based on network conditions, ensuring optimal performance even in challenging environments. The architecture's compatibility with existing web technologies like HTML5 and JavaScript simplifies integration with web applications and frameworks. WebRTC's support for multi-party communication enables group video conferencing and collaboration, enhancing productivity and teamwork in remote settings. The architecture's support for simulcast allows for the transmission of multiple video streams at different resolutions, catering to diverse bandwidth constraints and device capabilities. WebRTC's support for data channel encryption ensures the privacy and security of transmitted data, mitigating risks associated with unauthorized access and interception. The architecture's support for trickle ICE enables incremental exchange of ICE candidates, reducing connection setup time and improving responsiveness.

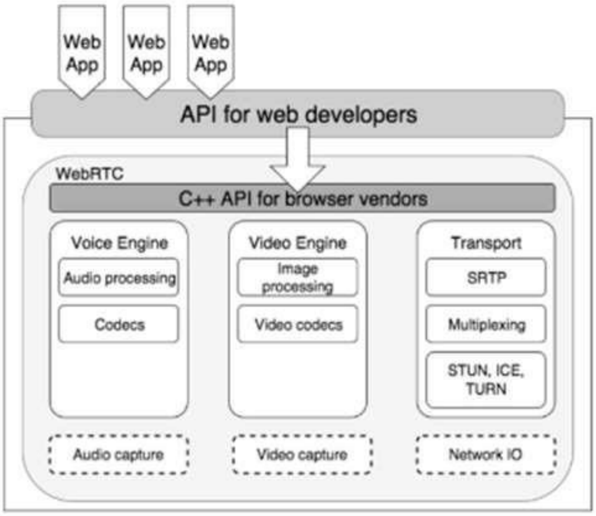

Fig 1: Architecture

6.PROTOCOL OF WEBRTC

I. TCP (Transmission Control Protocol) and UDP (User Datagram Protocol):

WebRTC can utilize both TCP and UDP for data transmission, but UDP is more commonly used due to its lower latency and overhead, which is crucial for real-time communication.

•

II. ICE (Interactive Connectivity Establishment):

ICE is a framework used to establish peer-to-peer connections between devices across potentially restrictive network environments, such as NATs and

III. STUN (Session Traversal Utilities for NAT):

STUN servers assist in discovering the public IP addresses and ports assigned to devices behind NATs. This information is crucial for establishing direct peer-to-peer connections whenever possible.

IV. TURN (Traversal Using Relays around NAT):

TURN servers act as relay points for media traffic when direct peer-to-peer connections cannot be established due to network restrictions. They relay data between peers, ensuring seamless communication even in challenging network environments.

V. DTLS (Datagram Transport Layer Security):

DTLS is used to provide encryption and security for WebRTC communication sessions. It ensures the confidentiality and integrity of data exchanged between peers by establishing secure connections.

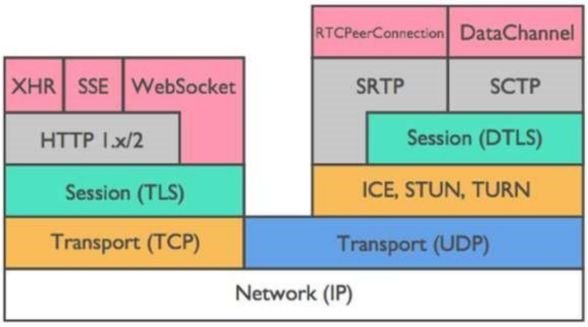

Fig 2 :WebRTC protocol

7.RESULT

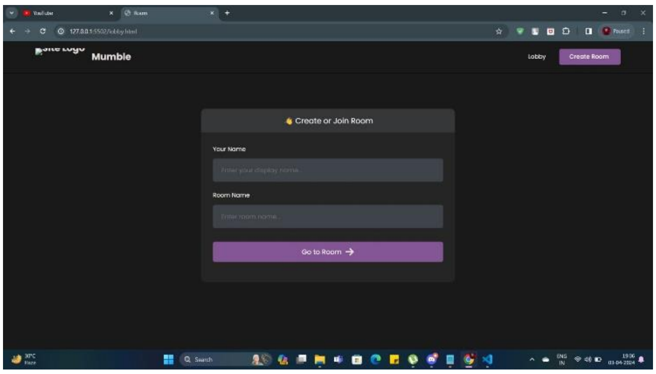

Fig 3: Login page

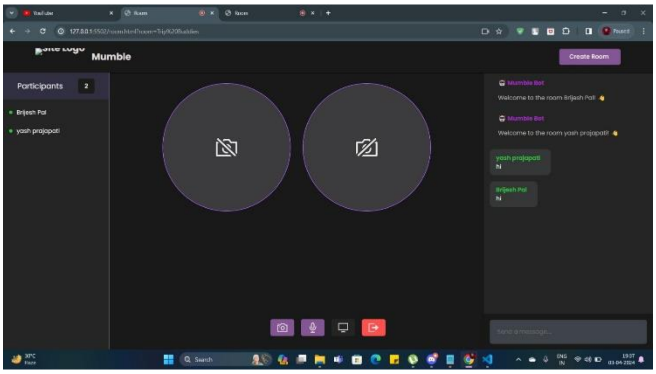

Fig 4:User interface

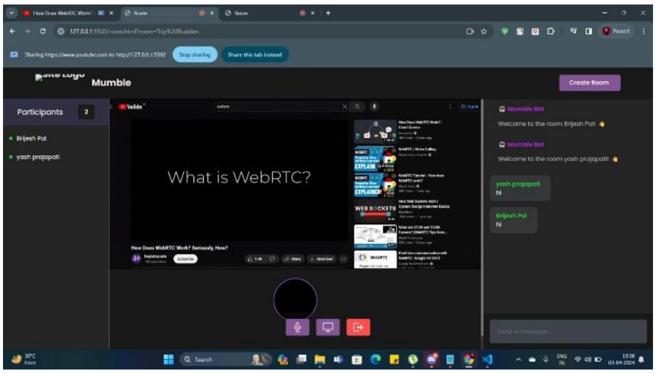

Fig 5 :Features

8.CONCLUSION

In summary, WebRTC presents a revolutionary approach to real-time communication, offering seamless peer-to-peer connectivity within web browsers and mobile apps. Its open-source nature, coupled with robust protocols like ICE, STUN, and TURN, ensures reliable connections across diverse network environments. With a focus on low-latency media transmission through UDP and fortified security via DTLS, WebRTC stands at the forefront of modern communication technology. Its wide-ranging applications, from video conferencing to IoT, signify its transformative potential. As it continues to evolve, WebRTC promises to redefine how we interact and collaborate online, fostering innovation and accessibility in the digital realm.

9.REFERENCE

[1] Nayyef, Zinah & Amer, Sarah & Hussain, (2019). Peer to Peer Multimedia Real-Time Communication System based on WebRTC Technology. International Journal for the History of Engineering & Technology. 2.9. 125-130.

[2] Suciu G., Anwar M., Mihalcioiu R.,Virtualized Video and Cloud Computing for Efficient e-Learning, 13th International Scientific Conference eLearning and Software for Education, April 27-28, 2017.

[3] Suciu G., Anwar M., Virtualized Video conferencing for eLearning, 14th International Scientific Conference eLearning and Software for Education Bucharest, April 1920, 2018.

[4] Vasilescu C., Beceanu C., Collaborative object recognition for parking management, 15th International Scientific Conference eLearning and Software for Education Bucharest, April 11-12, 2019.

[5] Elleuch, Wajdi. (2013). Models for multimedia conference between browsers based on WebRTC. 279-284. 10.1109/WiMOB.2013.6673373.

[6] Rodríguez P, Cerviño J, Trajkovska I, Salvachúa J (2013) Advanced Videoconferencing Services Based on WebRTC. In: Proceeding of IADIS multi conference on computer science and information systems.

[7] XEP-0166: Jingle, XMPP Standards Foundation, https://xmpp.org/extensions/xep-0166.html.

[8] IC - Interactive Connectivity Establishment, IETF Working Group, https://tools.ietf.org/html/rfc5245.

[9] WebRTC Audio Codec and Processing Requirements, IETF Working Group, https://tools.ietf.org/html/rfc7874.

[10] “Why Does Your WebRTC Product Need a TURN Server?”, https://www.callstats.io/blog/2017/10/26/turnwebrtc-products.

[11] Julius Flohr; Ekaterina Volodina; Erwin P. Rathgeb(2018) FSE-NG for managing real time media flows and SCTP data channel in WebRTC.