



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## AI-ENABLED VIDEO CONFERENCE

<sup>1</sup>Deepak K.N, <sup>2</sup>Anand Ramesh, <sup>3</sup>Manu T.M, <sup>4</sup>Suraj K.S, <sup>5</sup>Vaishnavu M.V

<sup>1</sup>Assistant Professor, Department of Computer Science and Engineering, Universal Engineering College Vallivattom, Thrissur, India.

<sup>2,3,4,5</sup>B.Tech Student, Department of Computer Science and Engineering, Universal Engineering College Vallivattom, Thrissur, India.

**Abstract:** Video conference has recently become increasingly popular in all fields. With the rise in demand of the video conference it is necessary to improve this further. So, a hand gesture recognition system is proposed that allows the user to control the video conference using hand gestures. The hand gestures can be used to control the platform of video conference, in addition to this, hand gestures can be used to enable air writing in video conference. Hand detection is done by using media pipe. A specially designed software identifies meaningful gestures from a predetermined gesture library where each gesture is matched to a computer command for the conference. This way hand gestures become the input to obtain the outputs of the video conference. The proposed system also provides air writing feature using hand gesture recognition. Writing in air can be defined as to write digits or character in a virtual space by using a finger. The proposed system uses an air-writing character recognition using a convolutional neural network (CNN). For this, an air canvas will be built on which we can draw anything by just capturing the motion of a finger-tip using a camera.

**Index Terms -** Video Conference, Hand gesture recognition, Air-writing

### I. INTRODUCTION

The COVID-19 pandemic and the resulting stay-at-home orders have led to significant changes in the way people work. One of these changes involves increased use of video conference in the field of education. Video conference boosts productivity, saves times, reduce travel expenses, and overall promotes collaboration. The advantage of video conferencing is the ability to facilitate all of those benefits without requiring constant travel for face-face communication. Furthermore, it is likely that the use of video conference will continue long after the pandemic ends. The proposed system makes video conference more interactive and easier to use without any physical contact with the system.

Video conferencing stands out from other forms of telecommunication because you can see the other participants. Face-to-face interactions generally develop stronger rapport and relationships. The conferences can be conducted from anywhere, at any time, with the advent of mobile devices such as laptops, tablets, desktop computers, and even smartphones. Software-based video communication platforms are used to transmit the participants' communication over the internet. Video conferencing works through two stages: Compression: First, the video and audio are captured, and the data is compressed into waves of frequencies and amplitudes, which capture the sound and visual data. The data is compressed further into digital packets, which allow the information to be transferred more quickly over a network connection. Transfer: Then, during the transfer, the data is transferred and received by a computer, and the data is decompressed and converted back into audio and video, displaying on the recipient's screen.

Hand gesture recognition is one of the active research areas in the field of human-computer interface due to its flexibility and user friendliness. The gesture recognition technique is used to develop a system that can be used to convey information among disabled people or for controlling a device. Major challenges for the development of an efficient hand gesture recognition technique are illumination variation, nonuniform backgrounds, diversities in the size and shape of a user's hand, and high interclass similarities between hand gesture poses.

Writing with a finger on a touch-based interface is intuitive because it follows the metaphor of pen-based writing. Recent advances of tracking technology make it possible to track hand and finger motions without user worn devices, and writing motion is no longer restricted on a physical plane. Air-writing provides a viable alternative interface for text input, particularly when conventional input devices, such as a keyboard or a mouse, are not available or suitable. Compared to other non-traditional input methods such as typing with a virtual keyboard or similar schemes, air writing offers the advantage of "eye-free" execution, requiring minimum attention focus

This system integrates the three i.e., video conference, hand gesture recognition and air writing to make usage of video conference more interesting and easier.

## II. REVIEW OF LITERATURES

Here we introduce each paper based on the technologies used in the AI-Enabled Video Conference and they are arranged in technologies bases

In the paper “A P2P-MCU Approach to Multi-Party Video Conference with WebRTC,” proposed by Kwok-Fai Ng, Man-Yan Ching, Yang Liu, Tao Cai, Li Li, and Wu Chou, a P2P-MCU approach is proposed for multi-party video conferencing that efficiently supports both ordinary smart mobile phones and PCs. By this approach, a MCU module is integrated into the browser to mix and transcode the video & audio streams in real time. And when the browser acts as the MCU, the node leaves the conference session without notice, another candidate browser can take over the control immediately, and the ongoing WebRTC conference can be seamlessly recovered with an MCU selection algorithm. In addition to this, the proposed system works under the 3G symmetric NAT networks by using some UDP hole punching method. This P2P-MCU solution reduces 64% CPU usages and 35% bandwidth consumptions for each participant compared to the mesh-network solution in an eight-party WebRTC conference experiments. Although the P2P-MCU module may introduce some delay (<500ms), the delay is stable and perceptually almost neglect able.

In the paper “Audio and Video Mixing Method to Enhance WebRTC,” proposed by Dongming Tang, and Liquan Zhang, the WebRTC protocol is restricted to a small number of peers because there is no simple way to mix real-time streams from multiple peers and then distribute the mixed stream to a large number of participants. For example, it is necessary to mix audio and video streams from peers in a conversation and broad cast the real-time mixed stream to more than 10k participants in the conference. This paper proposes a method for the synchronized mixing of real-time audio/video streams from multiple peers while minimizing latency. Compared to the previous WebRTC architecture, an additional quasi-peer is added to the WebRTC gateway. The quasi-peer functions as a normal peer in that it collects media streams from other peers, but it does not generate any media streams for other peers. After the audio/video data are collected, the quasi-peer mixes the data and ultimately distributes the output (the final mixed audio/video) via ICE, a content delivery network to the participants. This method enables the implementation of an online live conversation system that is able to mix live conversation streams from multiple peers and then rebroadcast the mixed stream to a large number of participants. In an enterprise network, it is common to have hundreds of video conferences held simultaneously such that a large number of video streams need to be transmitted between participants in different geographical locations. For such a large transmission demand, an advance reservation (AR) system deployed for the video conferencing system can provide quality-of-service (QoS) guarantees to users and improve the resource utilization of the network.

In the paper “Elastic Timeslot-based Advance Reservation Algorithm for Enterprise Video Conferencing Systems,” proposed by Zhiwen Liao, and Ling Zhang, the authors propose an algorithm called the elastic timeslot-based advance reservation algorithm (ETARA), which aims at improving the resource utilization and reducing the computational complexity. The Advanced Reservation (AR) algorithm makes use of the topology management module (TMM) to obtain the topology of the network and the traffic engineering database (TED) to obtain the available bandwidth on each link. Combined with the request time, the AR algorithm generates resource matrices with the time attribute. As the request is processed, the AR algorithm makes a decision based on whether there insufficient bandwidth in resource matrices to ensure a certain QoS level. The results show that with the same acceptance ratio, the runtime of ETARA can be up to 57 times lower than that of the flexible time slot-based approach. Though ETARA has a slightly longer run time than the dynamic approach, the acceptance ratio of ETARA can be twice as high as that of the dynamic timeslot-based approach.

In the paper “Scheduling Dynamic Multicast Requests in Advance Reservation Environment for Enterprise Video Conferencing Systems,” proposed by Zhiwen Liao, and Ling, the authors has proposed a method for maximizing the number of admitted Dynamic Multicast requests in the Advance Reservation environment (MDMAR) for the enterprise video conferencing system. The abilities to reserve resources in advance, as well as effective dynamic multicast when participants can join and leave the conference at any time, are essential in the distributed multiparty video conferencing systems. However, the effective advance reservation strategies of the dynamic multicast requests for a heavy traffic case still remains open. For these two path schemes of a fixed path and variable paths, as well as a heterogeneous bandwidth reservation model is taken into account. The MDMAR problem is NP-complete and formulate it mathematically as an integer linear program (ILP) for small networks. Then, greedy algorithms and simulated annealing (SA) algorithms for enterprise networks is developed. Comparative simulations are performed to evaluate the heuristic algorithms for both small networks and enterprise networks. From that it is found that the SA algorithms can provide within 6% lower optimal solutions than the ILP algorithms for small network, and up to 10% improvement over the greedy algorithms for the large campus or enterprise network.

The main aim of this paper “Dealing with User Heterogeneity in P2P Multi-Party Video Conferencing: Layered Distribution Versus Partitioned Simulcast,” proposed by EymenKurdoglu, Yong Liu, and Yao Wang, is to maximize the received video quality for both systems under uplink-downlink capacity constraints, while constraining the number of hops the packets traverse to two. One way to deal with user bandwidth heterogeneity is employing layered video coding, generating multiple layers with different rates, whereas an alternative is partitioning the receivers of each source and disseminating a different non-layered video version within each group. Here authors have proposed an algorithm that solves for the number of video layers, layer rates, and distribution trees for the layered system. For the partitioned simulcast system, an algorithm is developed to determine the receiver partitions along with the video rate and the distribution trees for each group. Through numerical comparison, we show that the partitioned simulcast system achieves the same average receiving quality as the ideal layered system without any coding overhead for the four-user systems simulated, and better quality than the layered system when the layered coding overhead is only 20%. The two systems perform similarly for the six-user case if the layered coding overhead is 10%.

According to the paper “Faces in the Clouds: Long-Duration, Multi-User, Cloud-Assisted Video Conferencing,” proposed by Richard G. Clegg, Raul Landa, David Griffin, Miguel Rio, Member, Peter Hughes, Ian Kegel, Tim Stevens, Peter Pietzuch, and Doug Williams, increasingly end-hosts in a multi-user video conference are assisted by cloud-based servers that improve the quality of experience for end users. For this, a proposed system is introduced to evaluate the impact of strategies for placement of such servers on user experience and deployment cost. The authors consider scenarios based upon the Amazon EC2 infrastructure as well as future scenarios in which cloud instances can be located at a larger number of possible sites across the planet. The proposed system is driven by real data to create demand scenarios with realistic geographical user distributions and diurnal behaviour. According to the proposed system it is found that on the EC2 infrastructure a well-chosen static selection of servers performs well but as more cloud locations are available a dynamic choice of servers becomes important.

The paper “Motion-Based Rate Adaptation in WebRTC Videoconferencing using Scalable Video Coding,” proposed by Gonca Bakar, RizaArdaKirmizioglu, and A. Murat Tekalp, proposes methods for rate adaptation by motion-based spatial and temporal resolution selection in both mesh-connected and selective-forwarding-unit (SFU) connected WebRTC videoconferencing using scalable video coding. In the mesh-connected case, the proposed motion-adaptive spatial/temporal layer selection allows each peer to send video to different peers with different terminal types and network rates at different rates using a single encoder. In the SFU-connected case, motion-adaptive rate control is used both at peers to adapt to the network rate between the sending peer and SFU by spatiotemporal resolution adaptation and at the SFU by layer selection to adapt to the network rate between the SFU and receiving peer. Experimental results show that our proposed motion-based rate adaptation achieves better perceptual video quality with sufficiently high frame rates and lower quantization parameter for video with high motion; and high spatial resolution and lower quantization parameter for video with low motion compared to simple rate-distortion model-based layer selection that does not use motion complexity, at the same rate.

In the paper “Hand Gesture Recognition Using Multiple Acoustic Measurements at Wrist,” proposed by Nabeel Siddiqui, and Rosa H. M. Chan, a device consists of 40 microphones to be worn at the wrist is introduced. The gesture recognition performance is evaluated through the identification of 36 gestures in American sign language (ASL), including 26 ASL alphabetical characters and 10 ASL numbers. The optimal area for sensor band placement (distal/proximal) is examined to reveal the location of the highest discrimination accuracy. Ten subjects are recruited to perform over ten trials for each set of hand gestures. The results of the paper shows that the intra subject average classification accuracy above 90% using the two features with all 40 microphones, while the average classification accuracy exceeding 84% is obtained using ten microphones. These results indicate that acoustic signatures from the human wrist can be utilized for hand gesture recognition, while the use of few, simple features, with low computational requirements is sufficient to characterize some hand gesture.

The paper “A Hand Gesture Recognition Sensor Using Reflected Impulses,” proposed by Seo Yul Kim, Hong Gul Han, Jin Woo Kim, Sanghoon Lee, and Tae Wook Kim, introduces a hand gesture recognition sensor using ultra-wideband impulse signals which are reflected from a hand. The reflected waveforms in time domain are determined by the reflection surface of a target. Thus, every gesture has its own reflected waveform. Hence the proposed system uses machine learning approach such as convolutional neural network (CNN) for the gesture classification. The CNN extracts its own feature and constructs classification model then classifies the reflected waveforms. Six hand gestures from American Sign Language (ASL) are used for an experiment and the result shows more than 90% recognition accuracy. For fine movements, a rotating plaster model is measured with 10° step. An average recognition accuracy is also above 90%.

In the paper “Recognizing Hand Gestures with Pressure Sensor based Motion Sensing,” proposed by YuFei Zhang, Bin Liu, and Zhiqiang Liu, a prototype system, including a wearable gesture sensing device with four pressure sensors and the corresponding algorithmic framework, is developed to realize real-time gesture-based interaction. With the device worn on the wrist, the user can interact with the computer using 8 predefined gestures. Experimental results show that the delay of gesture recognition is about 100ms, with the average accuracy of 95.28% in the experienced-user test and 86.20% in the inexperienced-user test. Finally, the system is evaluated by a mouse-controlling interaction task and performs well. Both experienced and inexperienced people can easily and quickly complete interactive tasks. These results demonstrate that a pressure-sensor based wristband can be used to classify hand gestures well and to control the mouse interaction. This approach provides an interactive way to replace the mouse for decreasing the risk of the carpal tunnel syndrome (CTS).

In the paper “Hand Gesture Based Remote Control System Using Infrared Sensors and a Camera,” proposed by FatihErden and A. EnisÇetin, Fellow, a multimodal hand gesture detection and recognition system using differential Pyroelectric Infrared (PIR) sensors and a regular camera is described. Any movement within the viewing range of the differential PIR sensors are first detected by the sensors and then checked if it is due to a hand gesture or not by video analysis. If the movement is due to a hand, one-dimensional continuous-time signals extracted from the PIR sensors are used to classify/recognize the hand movements in real-time. Classification of different hand gestures by using the differential PIR sensors is carried out by a new winner-take all (WTA) hash-based recognition method. Jaccard distance is used to compare the WTA hash codes extracted from 1-D differential infrared sensor signals. It is experimentally shown that the multimodal system achieves higher recognition rates than the system based on only the on/off decisions of the analog circuitry of the PIR sensors.

In the paper “Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation,” proposed by Munneer Al-Hammadi, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaimani, Mohammed A. Bencherif, Tareq S. Alrayes, Hassan Mathkour, and Mohamed Amine Mekhtiche, a novel system is proposed for dynamic hand gesture recognition using multiple deep learning architectures for hand segmentation, local and global feature representations, and sequence feature globalization and recognition. The proposed system is evaluated on a very challenging dataset, which consists of 40 dynamic hand gestures performed by 40 subjects in an uncontrolled environment. The results show that the proposed system outperforms state-of-the-art approaches, demonstrating its effectiveness.

In the paper, “Micro Hand Gesture Recognition System Using Ultrasonic Active Sensing,” proposed by Yu Sang, Laixi Shi, and Yimin Liu, we propose a micro hand gesture recognition system and methods using ultrasonic active sensing. This system uses micro dynamic hand gestures for recognition to achieve human-computer interaction (HCI). The implemented system, called hand-ultrasonic gesture (HUG), consists of ultrasonic active sensing, pulsed radar signal processing, and time-sequence pattern recognition by machine learning. A lower frequency (300kHz) ultra-sonic active sensing to obtain high resolution range-Doppler image features is adopted here. Using high quality sequential range-Doppler features, a state-transition-based hidden Markov model is proposed for gesture classification. This method achieves a recognition accuracy of nearly 90% by using symbolized range-Doppler features and significantly reduces the computational complexity and power consumption. Furthermore, to achieve higher classification accuracy, we utilize an end-to-end neural network model and obtain a recognition accuracy of 96.32%.

In this paper, “Deep Learning Based Real-Time Recognition of Dynamic Finger Gestures Using a Data Glove,” proposed by Minhyuk Lee, and Joonbum Bae, a real-time dynamic finger gesture recognition using a soft sensor embedded data glove is presented, which measures the metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joint angles of five fingers. In the gesture recognition field, a challenging problem is that of separating meaningful dynamic gestures from a continuous data stream. To solve the problem of separating meaningful dynamic gestures, the authors have proposed a deep learning-based gesture spotting algorithm that detects the start/end of a gesture sequence in a continuous data stream. The gesture spotting algorithm takes window data and estimates a scalar value named gesture progress sequence (GPS). Moreover, to solve the gesture variation problem, a sequence simplification algorithm and a deep learning-based gesture recognition algorithm is proposed here. The proposed three algorithms (gesture spotting algorithm, sequence simplification algorithm, and gesture recognition algorithm) are unified into the real-time gesture recognition system and the system was tested with 11 dynamic finger gestures in real-time. The proposed system took only 6 ms to estimate a GPS and no more than 12 ms to recognize the completed gesture in real-time.

In the paper “Deep-Learning Methods for Hand-Gesture Recognition Using Ultra-Wideband Radar,” proposed by SruthySkaria, Akram Al-Hourani, and Robin J. Evans, a framework, using deep-learning techniques, is proposed here to classify hand-gesture signatures generated from an ultra-wideband (UWB) impulse radar. The signals of 14 different hand-gestures are extracted and represent each signature as a 3-dimensional tensor consisting of range-Doppler frame sequence. These signatures are passed to a convolutional neural network (CNN) to extract the unique features of each gesture, and are then fed to a classifier. Four different classification architectures to predict the gesture class, namely; fully connected neural network (FCNN), k-Nearest Neighbours (k-NN), support vector machine (SVM), (iv) long short-term memory (LSTM) network is compared here. The shape of the range-Doppler-frame tensor and the parameters of the classifiers are optimized in order to maximize the classification accuracy. The classification results of the proposed architectures show a high level of accuracy above 96% and a very low confusion probability even between similar gestures.

In the paper “Writing in the Air with WiFi Signals for Virtual Reality Devices,” proposed by Zhangjie Fu, Jiashuang Xu, Zhuangdi Zhu, Alex X. Liu and Xingming Sun, the CSI (channel state information) derived from wireless signals to realize the device-free air-write recognition called Wri-Fi is utilized for air writing purpose. Compared to the gesture recognition, the increased diversity and complexity of characters of the alphabet make it challenging. The PCA (Principle Component Analysis) is used for denoising effectively and the energy indicator derived from the FFT (Fast Fourier Transform) is to detect action continuously. The unique CSI waveform caused by unique writing patterns of 26 letters serve as feature space. From the experiments conducted in the laboratory the average accuracy of the Wri-Fi is 86.75% and 88.74% in two writing areas, respectively.

In the paper “Air-writing via Receiver Array Based Ultrasonic Source Localization,” proposed by Hui Chen, TarigBallal, Ali H. Muqabel, Xiangliang Zhang, and Tareq Y. Al-Naffouri, an air-writing system using acoustic waves is proposed. The proposed system consists of two components: a motion tracking component, and a text recognition component. For motion tracking, we utilize direction-of-arrival (DoA) information. An ultrasonic receiver array tracks the motion of a wearable ultrasonic transmitter by observing the change in the DoA of the signals. A novel 2-D DoA estimation algorithm is proposed that can track the change in the direction of the transmitter using measured phase-differences between the receiver array elements. The proposed phase-difference projection (PDP) algorithm can provide accurate tracking with a 3-sensor receiver array. The motion tracking information is passed next for text recognition. To this end, and in order to strike the desired balance between flexibility, processing speed, and accuracy, a training-free order restricted matching (ORM) classifier is designed. The proposed air-writing system, which combines the proposed DoA estimation and text recognition algorithms, achieves a letter classification accuracy of 96.7%. The utility, processing time, and classification accuracy are compared with four training-free classifiers and two machine learning classifiers to demonstrate the efficiency of the proposed system.

The work of the paper “Vision-Based Mid-Air Unistroke Character Input Using Polar Signatures,” proposed by Lalit Kane and Pritee Khanna, presents a prototype framework for vision-based mid-air unistroke character input, which can be adapted as an interface for the IRS. At first, an acquisition module is developed which effectively spots the legitimate gesture trajectory by implementing pen-up and pen-down actions using depth thresholding and velocity tracking. The extracted trajectory is recognized through a novel, fast, and easy to implement the equipolar signature (EPS) technique. Apart from resistance to rotation, scale, and translation variations, EPS exhibits neutrality to stroking directions as well. On the three self-collected datasets comprising of digits, alphabets, and symbols, the EPS scheme obtains over 96.5% accurate results with an average of 30-ms running time. The proposed scheme is also validated on an open dataset DAIR (Dataset for AIR Handwriting), where it achieves 95.5% mean accuracy with 24.3-ms recognition time per gesture. The developed approach is compared with benchmark schemes to justify its accuracy and speed.

In the paper “Air-Writing Recognition—Part I: Modeling and Recognition of Characters, Words, and Connecting Motions,” proposed by Mingyu Chen, GhassanAlRegib, and Biing-Hwang Juang, recognition of characters or words is accomplished based on six-degree-of freedom hand motion data. Air-writing are of two levels: motion characters and motion words. Isolated air-writing characters can be recognized similar to motion gestures although with increased sophistication and variability. For motion word recognition in which letters are connected and superimposed in the same virtual box in space, statistical models are built for words by concatenating clustered ligature models and individual letter models. A hidden Markov model is used for air-writing modelling and recognition. The proposed system achieves a word error rate of 0.8% for word-based recognition and 1.9% for letter-based recognition.

The paper “Air-Writing Recognition—Part II: Detection and Recognition of Writing Activity in Continuous Stream of Motion Data,” proposed by Mingyu Chen, GhassanAlRegib, and Biing-Hwang Juang, addresses detecting and recognizing air-writing activities that are embedded in a continuous motion trajectory without delimitation. Detection of intended writing activities among superfluous finger movements unrelated to letters or words presents a challenge that needs to be treated separately from the traditional problem of pattern recognition. At first a dataset that contains a mixture of writing and non-writing finger motions is made. The LEAP from Leap Motion is used for marker-free and glove-free finger tracking. A window-based approach is proposed that automatically detects and extracts the air-writing event in a continuous stream of motion data, containing stray finger movements unrelated to writing. Consecutive writing events are converted into a writing segment. The recognition performance is further evaluated based on the detected writing segment. The proposed system achieves an overall segment error rate of 1.15% for word-based recognition and 9.84% for letter-based recognition.

### III. METHODOLOGY

#### A. Video Conference

This section describes how the WebRTC-related protocols used for video conference interact with one another in order to create a connection and transfer data and/or media among peers. WebRTC API helps to establish a peer-to-peer connectivity to other web browsers easily. Unfortunately, WebRTC can't create connections without some sort of server in the middle. This process is called as the signal channel or signalling service.

Here Peer A who will be the initiator of the connection, will create an Offer. They will then send this offer to Peer B using the chosen signal channel. Peer B will receive the Offer from the signal channel and create an Answer. They will then send this back to Peer A along the signal channel. This way the call establishment proceeds between the two peers and multimedia data is streamed between them. Additional functionalities are supported such as messaging, air-writing and controlling of the platform using hand gestures. The system has a user-friendly UI that can provide an easy understanding of the system and all the user requires to establish connectivity with the other peer is the link to the website.

When a user starts a WebRTC call to another user, a special description is created called an offer. This description includes all the information about the caller's proposed configuration for the call. The recipient then responds with an answer, which is a description of their end of the call. In this way, both devices share with one another the information needed in order to exchange media data. This exchange is handled using Interactive Connectivity Establishment (ICE), a protocol which lets two devices use an intermediary to exchange offers and answers even if the two devices are separated by Network Address Translation (NAT).

Generally, there are two kinds of IP addresses: private and the public IP address. A private IP address is a non-Internet facing IP address that the router assigns to each device connected to it. A public IP address is an Internet facing IP address that is assigned to the device each time it connects to the Internet. The NAT (Network Address Translation) is a device that generally performs the translations between the public and private IP address spaces. This provides unique identification for devices that are within your home network. Since the peers need to know the each other's IP address to communicate this is done with the help of the STUN (Session Traversal Utilities for NAT). This server can be defined as an entity that can handle the requests made by the clients. It is primarily made use by clients to know their public IP address. It is a client-server protocol and can work across TCP and UDP connections. The aim of the STUN server is to overcome issues associated with lack of standardized behaviour in NATs. The web client first contacts this server which sends back the public IP address back to the client. The same operation is performed by the other client and hence the connectivity between them is established behind the NAT. The web clients then contact the server where the web application is stored by sending an HTTP request. Once the two peers are connected to the same “channel”, the peers are able to communicate and negotiate session information through SDP (Session Description Protocol).

The configuration of an endpoint on a WebRTC connection is called a session description. The description includes information about the kind of media being sent, its format, the transfer protocol being used, the endpoint's IP address and port, and other information needed to describe a media transfer endpoint. This information is exchanged and stored using Session Description Protocol (SDP). Each peer, then, keeps two descriptions on hand: the local description, describing itself, and the remote description, describing the other end of the call.

As well as exchanging information about the media (SDP), peers must exchange information about the network connection. This is known as an ICE candidate and details the available methods the peer is able to communicate (directly or through a TURN server). The TURN (Traversal Using Relay NAT) server is used when peer to peer communication fails. It has

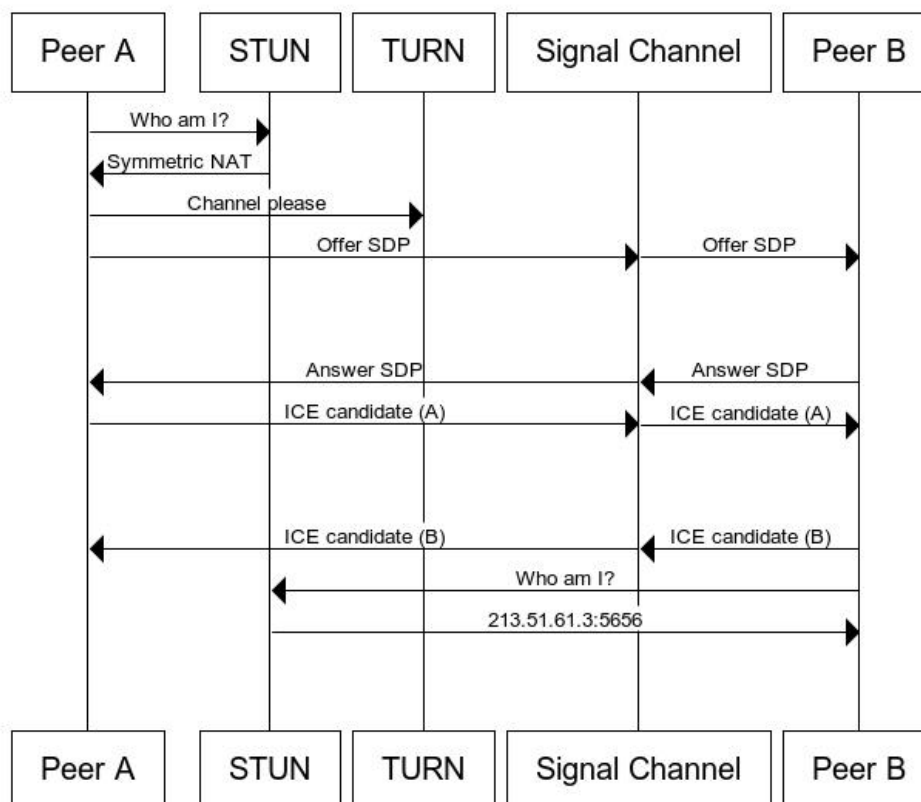
advantageous over STUN since it has the ability to traverse symmetric NATs. The TURN server acts a fallback mechanism when the STUN server fails since they are not reliable in nature especially when it comes to the handling of heavy media traffic and due to low bandwidth supported by them. However, the TURN server has a critical disadvantage when it comes to cost, maintenance, and huge bandwidth usage when HD video stream is being delivered. Typically, each peer will propose its best candidates first, making their way down the line toward their worse candidates. Ideally, candidates are UDP (since it's faster, and media streams are able to recover from interruptions relatively easily), but the ICE standard does allow TCP candidates as well.

The ICE layer selects one of the two peers to serve as the controlling agent. This is the ICE agent which will make the final decision as to which candidate pair to use for the connection. The other peer is called the controlled agent. The controlling agent not only takes responsibility for making the final decision as to which candidate pair to use, the controlled agent then just waits to be told which candidate pair to use.

It's important to keep in mind that a single ICE session may result in the controlling agent choosing more than one candidate pair. Each time it does so and shares that information with the controlled agent, the two peers reconfigure their connection to use the new configuration described by the new candidate pair. Once the ICE session is complete, the configuration that's currently in effect is the final one, unless an ICE reset occurs. At the end of each generation of candidates, an end of candidates notification is sent in the form of `RTCIceCandidate` whose `candidate` property is an empty string. This candidate should still be added to the connection using `addIceCandidate()` method, as usual, in order to deliver that notification to the remote peer.

When there are no more candidates at all to be expected during the current negotiation exchange, an end-of-candidates notifications is sent by delivering a `RTCIceCandidate` whose `candidate` property is null. This message does not need to be sent to the remote peer. It's a legacy notification of a state which can be detected instead by watching for the `iceGatheringState` to change to complete, by watching for the `iceGatheringStateChange` event.

1. The caller captures local Media via `MediaDevices.getUserMedia`
2. The caller creates `RTCPeerConnection` and calls `RTCPeerConnection.addTrack()` (Since `addStream` is deprecating)
3. The caller calls `RTCPeerConnection.createOffer()` to create an offer.
4. The caller calls `RTCPeerConnection.setLocalDescription()` to set that offer as the *local description* (that is, the description of the local end of the connection).
5. After `setLocalDescription()`, the caller asks STUN servers to generate the ice candidates
6. The caller uses the signaling server to transmit the offer to the intended receiver of the call.
7. The recipient receives the offer and calls `RTCPeerConnection.setRemoteDescription()` to record it as the *remote description* (the description of the other end of the connection).
8. The recipient does any setup it needs to do for its end of the call: capture its local media, and attach each media tracks into the peer connection via `RTCPeerConnection.addTrack()`
9. The recipient then creates an answer by calling `RTCPeerConnection.createAnswer()`.
10. The recipient calls `RTCPeerConnection.setLocalDescription()`, passing in the created answer, to set the answer as its local description. The recipient now knows the configuration of both ends of the connection.
11. The recipient uses the signaling server to send the answer to the caller.
12. The caller receives the answer.
13. The caller calls `RTCPeerConnection.setRemoteDescription()` to set the answer as the remote description for its end of the call. It now knows the configuration of both peers. Media begins to flow as configured.



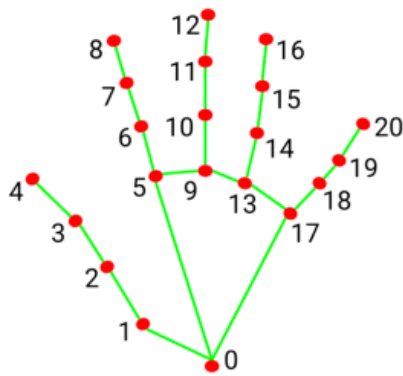
For the working of chatting, socket.io library can be used. Socket.IO is a JavaScript library that facilitates improving work with Web Sockets. It consists of parts – server part (for Node.JS) and sender or receiver part (for web browsers). Socket.IO allows using additional functions consisting of sending message to large number of sockets at the same time (broadcasting) or storing the messages. While a sender wants to send a message, socket emit function can be used to emit chat message along with the name of the sender and the message to the server. The sent message will also be displayed on the sender's browser. The emitted chat message received via the server will be broadcasted to all of the other users another emit function. This way exchange of chat messages takes place between the users.

The screen sharing feature works as follows. Firstly, the initiator needs to obtain the local display streams with the GetUserMedia API. The local screen streams can be collected from different resources, e.g., the entire display of the computer and the active tabs. Once selected, the raw screen stream needs to be clipped to a scale before being attached to the local PeerConnection object for a too-large frame may cause the error. For the participants, the remote PeerConnection object needs to be set up accordingly, and attach the remote screen streams received to it so that they can see the screen from the initiator.

### B. Hand detection

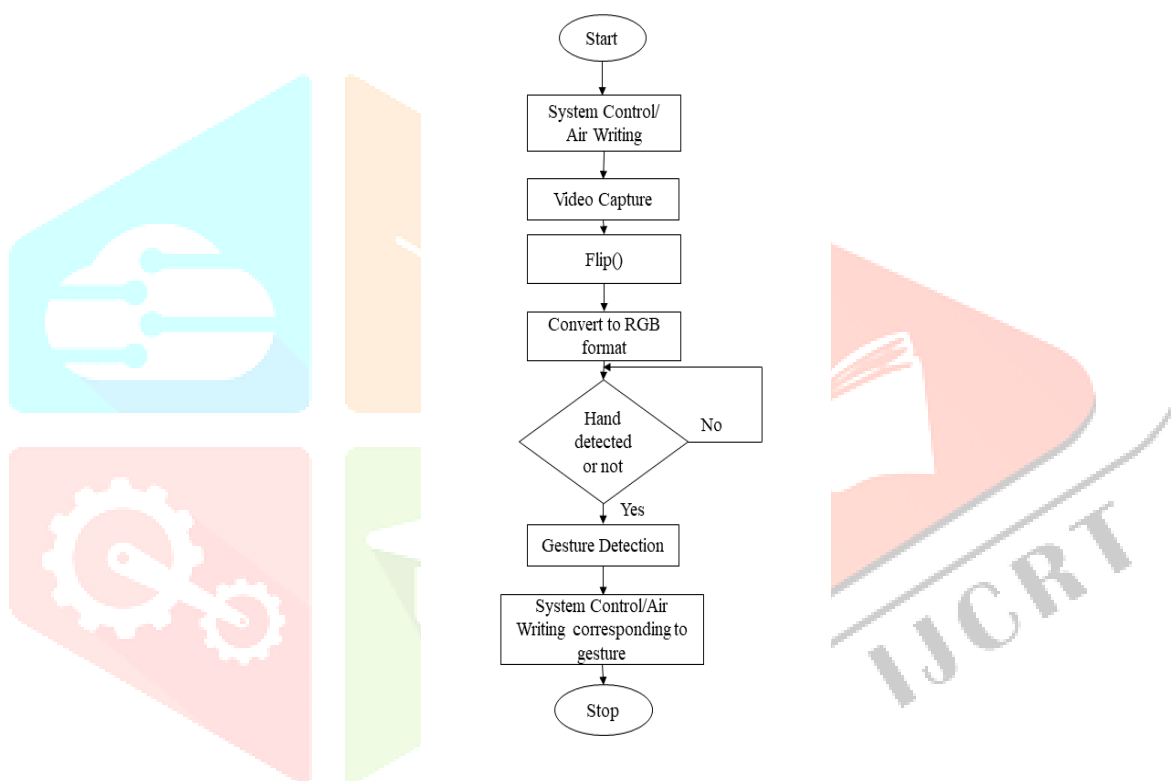
The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms. Media Pipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame.

In this project, we've built a hand gesture recognizer using OpenCV and python. The MediaPipe framework is used for the detection and gesture recognition respectively. Media Pipe recognize the hand and the hand key points and returns a total of 21 key points for each detected hand. By default, media pipe detects a maximum number of two hands, but here we detect only one hand at a time in this project.



- |                       |                       |
|-----------------------|-----------------------|
| 0. WRIST              | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC          | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP          | 13. RING_FINGER_MCP   |
| 3. THUMB_IP           | 14. RING_FINGER_PIP   |
| 4. THUMB_TIP          | 15. RING_FINGER_DIP   |
| 5. INDEX_FINGER_MCP   | 16. RING_FINGER_TIP   |
| 6. INDEX_FINGER_PIP   | 17. PINKY_MCP         |
| 7. INDEX_FINGER_DIP   | 18. PINKY_PIP         |
| 8. INDEX_FINGER_TIP   | 19. PINKY_DIP         |
| 9. MIDDLE_FINGER_MCP  | 20. PINKY_TIP         |
| 10. MIDDLE_FINGER_PIP |                       |

These key points will be fed into a pre-trained gesture recognizer network to recognize the hand pose. The model can recognize 10 different gestures. ['okay', 'peace', 'thumbs up', 'thumbs down', 'call me', 'stop', 'rock', 'live long', 'fist', 'smile']. Using this hand detection our "air writing module and system controlling works.



Flow diagram

For both air writing and system controlling the above hand detection is used. First system reads each frame from the webcam and these frames are thus flipped. MediaPipe works with RGB images but OpenCV reads images in BGR format. So, system converts frame to RGB format. The process function takes an RGB frame and returns a result class. Then we check if any hand is detected or not, then we loop through each detection and store the coordinate.

### 1. Air writing

Method of air writing can be achieved using OpenCV and machine learning technology. This is possible by detecting the hand using the library Media Pipe in python hence detecting the tip of the index finger [8]. The above hand detection method is used here. The additional functions to draw circle, rectangle and line drawing are included. And the activating of this additional functions through hovering the index finger in the respective positions for a specified time period.

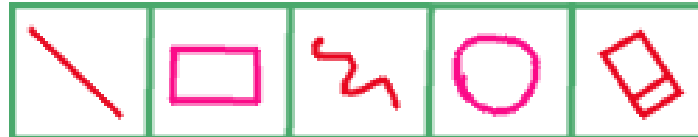


Steps


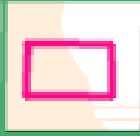



1. Reads each frame from the webcam
2. Convert the frame to RGB format
3. Check if any hand is detected or not.
4. Loop through each detection and store the coordinate on an array.
5. Finally draw the points stored in an array on the frames and canvas.

Select the tool

The below image set is included in the user interface to select the suitable tools. It is placed at a specific distance from the left margin.



When the index finger hovers the tool area then it get activated, from that onwards pointing finger helps to move position and combination of middle finger and index finger

	Line Tool
	Rectangle tool
	Draw Tool
	Circle Tool
	Eraser Tool

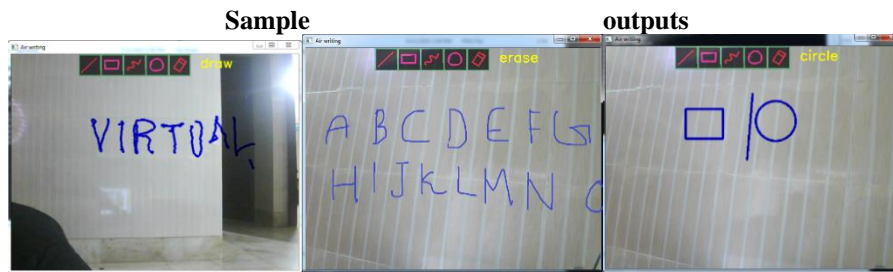
Trained gestures



To select tools



To draw



**Command prompt output**

```

Administrator: Command Prompt - python virtual_paint_app.py
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

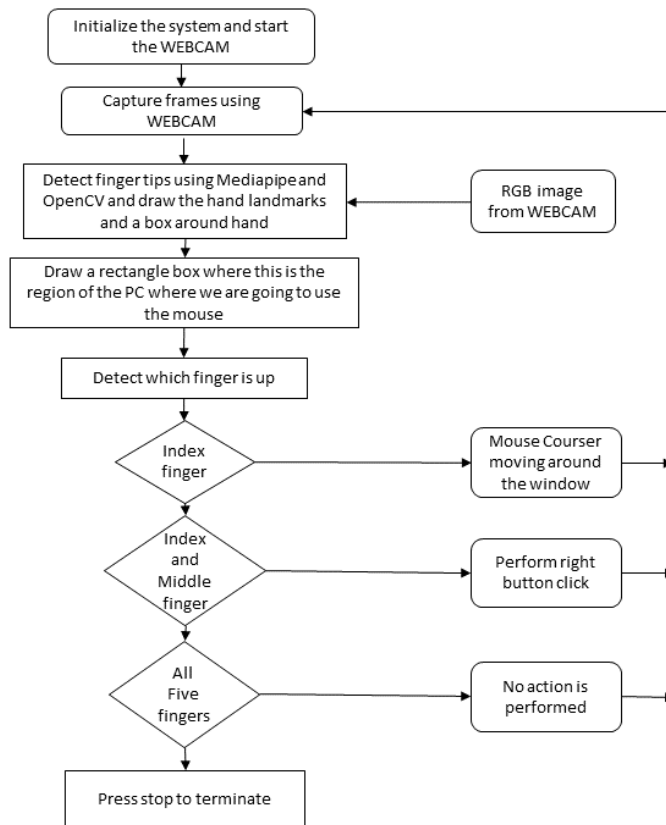
C:\Users\Admin>cd C:\virtual-paint-main

C:\virtual-paint-main>python virtual_paint_app.py
INFO: Created TensorFlow Lite XNNPACK delegate for GPU.
your current tool set to : line
your current tool set to : erase
your current tool set to : draw
your current tool set to : rectangle
your current tool set to : erase
your current tool set to : erase
your current tool set to : rectangle
your current tool set to : line
your current tool set to : circle
    
```

**2. System controlling**

Here in this module also we use hand detection model. In addition, with that an AutoPy library is included. This module contains functions for getting the current state of and controlling the mouse cursor. Unless otherwise stated, coordinates are those of a screen coordinate system, where the origin is at the top left.

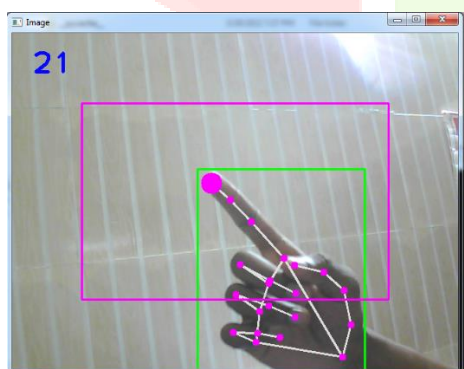
Real time video captured by the Webcam is processed and only the two finger tips are extracted. With the use of those two fingers, we can use our mouse. Their centers are measured by using the system webcam fingertip moments, and therefore the action to be taken is decided. The index finger is used to move the mouse cursor and the combination of index finger and index finger will output the mouse event.



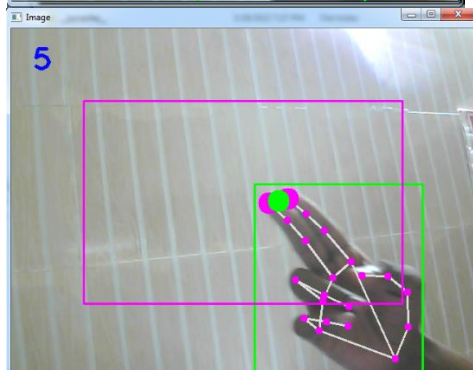
Flow diagram

The flow diagram shows the working of system with different functions. The system will first take the input from the system camera and converts into frames. The frame is resized as to increase smoothness. After detecting whether the finger is up, the cursor will move around the window according to the movement of the index finger. If the detected finger in combination of index and middle finger it will perform click action.

Sample output



This image shows the index finger tip which helps to move the cursor all over the window



This image shows how to click on any point of the screen

#### IV. CONCLUSION

This project is based on the idea of using hand gesture recognition to make the video conferencing more interactive and simpler to use. Basically, it is a mix of video conference and hand gesture recognition. In this project, the users can not only use hand gestures to control video conference but it also provides an air writing feature. This project mainly focuses on online education, the air writing feature and hand gesture recognition to control video conference make teachers more comfortable to take class. This way the online class will be more similar to live classes in offline as the teachers are taking classes with hands free. The proposed technology mainly uses image processing techniques to detect hands and recognize the features of gestures. The working of the project is like when a user enters a video conference the user can enable the hand gesture recognition feature to switch to control the video conference using gestures. This is done by capturing the images of hand through a camera and match the features of hand gestures to the features that are stored in database to identify the which gesture the user is showing. Once the gestures are identified corresponding output will be performed. This output is defined earlier within the system, like this for other gestures also has specified outputs defined to it within the system. This output may consist of muting, disable camera, or selecting participants etc. This way the video conference becomes more interactive and easier to use. In future as the technology becomes more developed the user interaction to the technology needs to be more efficient and easier to use. By using the hand gestures as input to any system the human interaction with the system will be improved. Hence the AI-Enabled Video Conference can become the stepping stone for the development of future AI technologies

#### REFERENCES

- [1]. Kwok-Fai Ng, Man -Yan Ching, Yang Liu, Tao Cai, Li Li, and Wu Chou, "A P2P-MCU Approach to Multi-Party Video Conference with WebRTC," *International Journal of Future Computer and Communication*, Vol. 3, No. 5, October 2014.
- [2]. Dongming Tang, and Liqun Zhang "Audio and Video Mixing Method to Enhance WebRTC," *IEEE Access*, April. 2020.
- [3]. Zhiwen Liao, and Ling Zhang "Elastic Timeslot-based Advance Reservation Algorithm for Enterprise Video Conferencing Systems," *IEEE Access*, vol. 4, pp 1-17 ,2019.
- [4]. Zhiwen Liao, and Ling, "Scheduling Dynamic Multicast Requests in Advance Reservation Environment for Enterprise Video Conferencing Systems," *IEEE Access*, vol. 4, 2016.
- [5]. EymenKurdoglu, Yong Liu, and Yao Wang, "DealingWithUserHeterogeneityinP2PMulti-Party Video Conferencing: Layered Distribution Versus Partitioned Simulcast," *IEEETransactions on Multimedia*, vol.18, No.1, pp 90-101, Jan 2016.
- [6]. Richard G. Clegg, Raul Landa, David Griffin, Miguel Rio, Member, Peter Hughes, Ian Kegel, Tim Stevens, Peter Pietzuch, and Doug Williams, "Faces in the Clouds: Long-Duration, Multi-User, Cloud-Assisted Video Conferencing," *IEEE Transactions on Cloud Computing*, 2016, DOI 10.1109/TCC.2017.2680440
- [7]. Gonca Bakar, RizaArdaKirmiziloglu, and A. Murat Tekalp, "Motion-Based Rate Adaptation in WebRTC Videoconferencing using Scalable Video Coding," *IEEE Transactions on Multimedia*, DOI 10.1109/TMM.2018.285662
- [8]. Nabeel Siddiqui, and Rosa H. M. Chan, "Hand Gesture Recognition Using Multiple Acoustic Measurements at Wrist," *IEEE Transactions on human-machine systems*, vol. 51, No. 1, pp 56-62, Feb. 2021.
- [9]. Seo Yul Kim, Hong Gul Han, Jin Woo Kim, Sanghoon Lee, and Tae Wook Kim, "A Hand Gesture Recognition Sensor Using Reflected Impulses," *IEEE Sensors Journal*, DOI 10.1109/JSEN.2017.2679220.
- [10]. YuFei Zhang, Bin Liu, and Zhiqiang Liu, "Recognizing Hand Gestures with Pressure Sensor based Motion Sensing," *IEEE Transactions on Biomedical Circuits and Systems 1*, DOI 10.1109/TBCAS.2019.2940030.
- [11]. FatihErden and A. EnisÇetin, Fellow, "Hand Gesture Based Remote Control System Using Infrared Sensors and a Camera," *IEEE Transactions on Consumer Electronics*, Vol. 60, No. 4, November 2014,pp 675-680.
- [12]. Munneer Al-Hammadi, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaimani, Mohammed A. Bencherif, Tareq S. Alrayes, Hassan Mathkour, and Mohamed Amine Mekhtiche "Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation," *IEEE Access*, vol. 8, Nov. 2020.
- [13]. Yu Sang , Laixi Shi , and Yimin Liu, "Micro Hand Gesture Recognition System Using Ultrasonic Active Sensing," *IEEE Access*, vol. 6,3-28 Sept 2018
- [14]. Minhyuk Lee, and Joonbum Bae, "Deep Learning Based Real-Time Recognition of Dynamic Finger Gestures Using a Data Glove," *IEEE Access*, vol. 8, 17-19 Nov , 2020.
- [15]. SruthySkaria, Akram Al-Hourani, and Robin J. Evans, "Deep-Learning Methods for Hand-Gesture Recognition Using Ultra-Wideband Radar," *IEEE Access*, vol. 8, 10-19 Nov, 2020.
- [16]. Zhangjie Fu, Jiashuang Xu, Zhuangdi Zhu, Alex X. Liu and Xingming Sun, "Writing in the Air with WiFi Signals for Virtual Reality Devices," *IEEE Transactions on Mobile Computing*, DOI 10.1109/TMC.2018.2831709.
- [17]. Hui Chen, TarigBallal, Ali H. Muqaibel, Xiangliang Zhang, and Tareq Y. Al-Naffouri, "Air-writing via Receiver Array Based Ultrasonic Source Localization," *IEEE Transactions on Instrumentation and Measurement*,2020, DOI 10.1109/TIM.2020.299157.
- [18]. Lalit Kane and Pritee Khanna, "Vision-Based Mid-Air Unistroke Character Input Using Polar Signatures," *IEEE Transactions on Human-Machine Systems*, 2017. DOI 10.1109/THMS.2017.270669.
- [19]. Mingyu Chen, GhassanAlRegib, andBiing-Hwang Juang, "Air-Writing Recognition—Part I: Modeling and Recognition of Characters, Words, and Connecting Motions," *IEEE Transactions on Human-Machine Systems*,2015, DOI 10.1109/THMS.2015.2492598.
- [20]. Mingyu Chen, GhassanAlRegib, and Biing-Hwang Juang, "Air-Writing Recognition—Part II: Detection and Recognition of Writing Activity in Continuous Stream of Motion Data," *IEEE Transactions on Human-Machine Systems*, 2015, DOI 10.1109/THMS.2015.2492.