JOURNAL OF CRITICAL REVIEWS
ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

VIDEO VISION BASED BROWSER ACTION CONTROL USING HAND GESTURES

[1]Bhuvaneshwari R, [2]Kavin Pragadeesh K, [3]Kailash J.P

Department of Computer Science and Engineering, Anna University, Chennai

bhuvi.saakshu@gmail.com, kavinswealthvalley@gmail.com, kailashjp977@gmail.com

Received: 14 Feb 2020 Revised and Accepted: 25 March 2020

ABSTRACT: Human Computer Interaction (HCI) plays a vital role in todays era that incorporates many new technological developments which are feasible and convenient for human access. Natural User Interface is a system which paves the way for HCI using natural day to day human behaviors or gestures. The proposed work tracks the hand gestures wherein the process relies on working of a webcam video using video vision techniques. The objective of this paper is to develop a novel extension of chrome which uses the live videos for further processing of hand gestures being manipulated using Handtrack.Js. This script helps in tracking hand gestures in video processing. Using the hand gestures, five different browser actions are controlled like creating and closing a tab, forwarding, backwarding and refreshing the current screen. The paper explains the methods in which the above process was developed to ease human work and also allows humans to access the chrome from a distance. The model imparts an outstanding appeal for the impaired community with the right setup and precise data points.

Keywords: Natural User Interface, Human Computer Interaction, Hand Gestures, Video Vision, Handtrack.Js

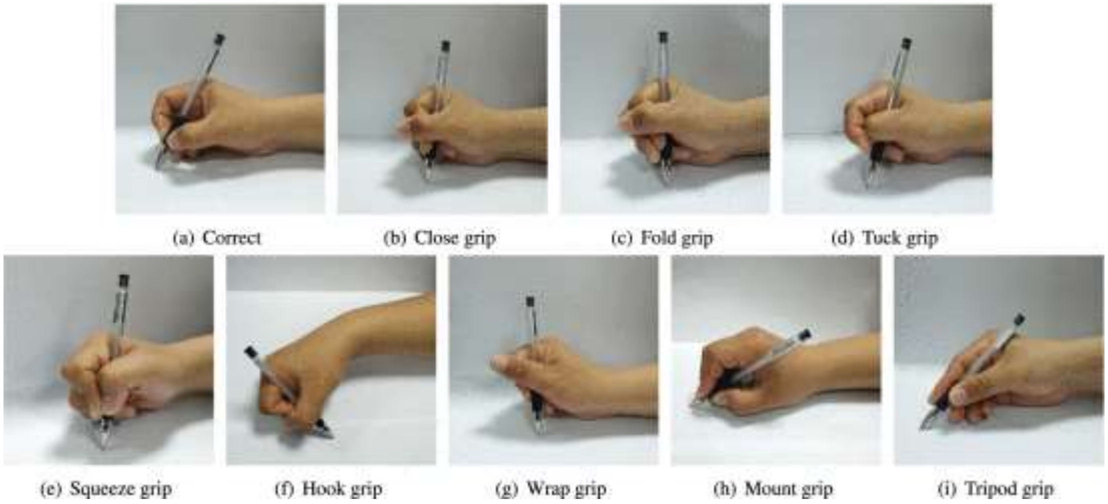
1 INTRODUCTION

Technical advancements like Artificial Intelligence, Machine Learning, Deep Learning, Natural Language Processing etc. rule the world from early morning alarms to nightlamps. Moreover, these data are digitally tabulated for further processing. The data got from a resource is processed several times and the data sets were used to automate the devices. One such example is image and video vision. It plays a major role in modern society. The open Image and video were processed and the information is given as data sets. (JSON Objects). Many model providers were working on this to make work easy. The paper applies Natural User Interface (NUI) which is a leading domain to deal with image and video processing. It is an interface that is designed to feel as natural as possible to the users, While touchscreens and motion-based virtual reality games are the major fields of NUI, several others exist as well. For example, a voice recognition interface like Apple's Siri assistant on the iPhone is considered a natural user interface since it responds to naturally spoken commands and questions. Virtual reality devices also rely under the same category, since they emulate a real-world experience. Likewise, hand tracking and processing also plays a major role in NUI systems. In recent years new methods of Human Computer Interaction (HCI) are being developed. Real time 3D Interaction tools [3] helps in embedding this feature in many devices. Human Computer Interaction, Gesture provides a way for computers to understand human body language and react according to that. Hand Gesture Controlled Browser uses this stack to interact with chrome and do browser actions like forward, backward, refresh, open new tab and close current tab. An extension is developed to control the webcam [12]. This hand gesture controlling is done via the physical webcam in our device. Hand Gesture recognition is handled through Handtrack.Js which is a third-party library using mathematical algorithms and methods to track hands and emit out result objects.

2 RELATED WORKS

2.1 Identifying Pen Holding Gesture using SmartWatch:

Hongliang Bi et.al [1] talks about identifying the posture and holding gestures of hand while writing. It is used to obtain a legible handwriting in English and Chinese. It uses multiple images as a dataset and deduces a good algorithm for good handwriting like English and Chinese. Figure 1 shows the hand posture and motions are tracked by the smartwatch in another hand while writing. Its advantage is that it helps kids to mould their handwriting from first and give good practice. SmartGe can identify pen-holding gestures as up to 98.3%. The mathematical algorithms were used to track the finger and hand positions. It's very useful for people to provide a lightweight system for monitoring the writing quality and habits continuously, but having a smartwatch always while writing becomes a drawback here.



(a) Correct (b) Close grip (c) Fold grip (d) Tuck grip

(e) Squeeze grip (f) Hook grip (g) Wrap grip (h) Mount grip (i) Tripod grip

Figure 1: Different Pen holding Hand Gestures and their identification

2.2 Sign Language using 3DCNN:

Muneer Al-Hammadi et.al [2] had given a detailed proof about identifying hand gestures for sign language detection. 3DCNN is used to train the spaciotemporal gestures. They have used 3D Convolutional Neural Network for gesture detection and have obtained 84.38%, 34.9%, and 70% of success rate on three different datasets. It consists of linguistic content that carries a large amount of information in sign language. These descriptors code enough information for the inter-frames temporal dependency, as well as the hand position, shape and orientation in each frame. The computed features should be able to minimize the effect of different circumstances like background clutter and occlusions. It also plays a pivotal role in human-computer interaction (HCI) systems.

2.3 Vision-Based Multimodal Head-Hand Interaction:

Anupam Agarwal et.al [9] explains the modes of interaction with the computer such as hand, head, facial recognition, touch. Etc. Relying on just one of them reduces the accuracy of the whole human computer interaction. Thus, using the user interface of the software, a user can control any interactive application (say VLC player, image browser etc) using hand and head gestures which in turn are automatically mapped to the mouse and keyboard events through windows API.

2.4 Hand Data Glove:

The author uses a glove to capture current position and angle of hand and fingers, and further it has been classified using the k-NN classifier algorithm. The gestures have been classified as clicking, dragging, rotating, pointing and ideal position. Recognizing these gestures relevant actions are taken, such as air writing and 3D sketching by tracking the path. Piyush Kumar et.al [10] studied and described the above work which is mainly used in controlling the image browser tool. The results show that the data glove used for interaction is better than the keyboard and mouse, since the interaction process is more accurate and natural.

2.5 TV Remote Control via Wearable Smart Watch Device:

Vinod Keshav Seetharaman.et.al [8] have developed a wearable smart watch with user customized gestures used to control a web browser on a smart TV running the Tizen operating system. The smartphone is used to translate the smart watch commands into a format that the TV can understand, as well as perform any other processing tasks. Alternatively, the mobile phone itself can also be held in the user's hand instead of the smart watch, and used for making gestures. Thus, this deals with controlling TV operations (Tizen OS) using hand gestures processed through smart watches. The main disadvantage is that maintenance of a good remote connection from quite a long distance from television is difficult.

2.6 Double Hand-Gesture Interaction Through VR Environment:

The proposal emphasizes the role of hand gesture tracking in VR environmental development system. It uses the finger(thumb) control to move the user right and left. The main stack used here is double hand-gesture interaction (DHGI) method. Compared with the results of the joystick input device and portal method using oculus rift touches, the objective and subjective findings of this study indicate that DHGI is intuitive, easy to learn, easy to use, and causes low fatigue. Fan Zhang et.al [7] works had a greater impact and results when compared with joystick inputs and Oculus Rift touches.

2.7 Hand Gesture Based Motion Control for Robotic Arm using Stereo Camera:

The device uses a stereo camera to observe the Robot’s arm movements. The robot’s movements were analyzed in 3D motion and sent to the processor as data sets via TCP/IP Connection. Then the data is processed and the response data is used to control the robot’s arm in Movement. Handri Toar et.al [4] explains about that the system that will track the pitch, yaw and roll motion of the hand. Remote control of the arm is handled by a separate LAN connection and the Leap Motion Controller is used for physical motion which could be understood with figure 2. Some of the difficulties are handling dimension, self-occlusion, speed manipulation, torque handling, uncontrolled environment, etc.,

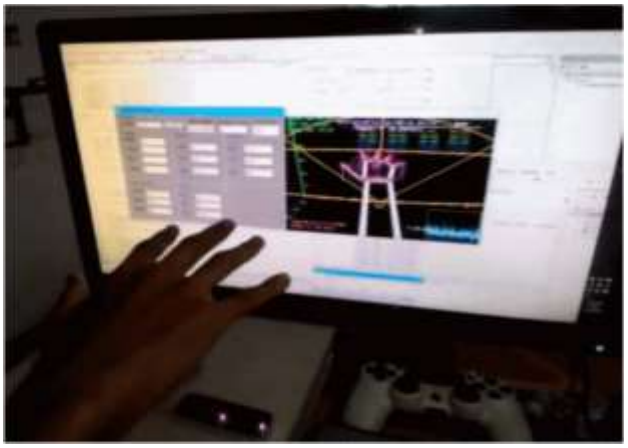


Figure 2: Hand Gestures Tracking System for Browser using Handtrack.Js

2.8 Posteriori Processing Optimization Method:

Gesture recognition plays an important role in the applications of smart medicine technologies. Generally, in an aseptic operation theatre room, doctors cannot access the patient's image data directly and they have to let the nurse operate the system under their instructions. After the nurses finish their task, they also need to re-dress the surgical gown, disinfect their hands, then they can go back to the operating room. If the doctor could access the image directly by gestures during the operation, they would be able to save a lot of time and work more efficiently. Perri Zhang et.al [5] proposes a template matching algorithm, called dynamic time warping (DTW) based on posteriori processing for gesture recognition.

3 PROPOSED WORK: Handtrack.Js for Hand Gesture Recognition

Based on the above foreseen works on gesture concepts and the technology stack, an application which is an extension to the chrome browser is proposed. It would be launched in the chrome extension store and it could be made public accessible. After importing it to chrome, the popup window of the extension opens up a video window which records user media of the user (i.e) the webcam. The backend work is done with Handtrack.Js which is an exporting module which tracks hand movements and its resolution on the screen. The hand position coordinates (x,y) are used to differentiate browser action and occur as datasets. These datasets and positions were used to redirect the extension to do a necessary browser action correspondingly based on gesture. Whenever the new tab or browser action calls up, the screen is refreshed and the extension opens up on the current page. The entire functionality is explained in figure3.

Thus, the browser can be controlled using gestures remotely. The main advantage is that it minimizes human work by controlling the browser remotely. This technique can also be extended for many applications in mobile, tablet, PC for education, medicine and also defenses.

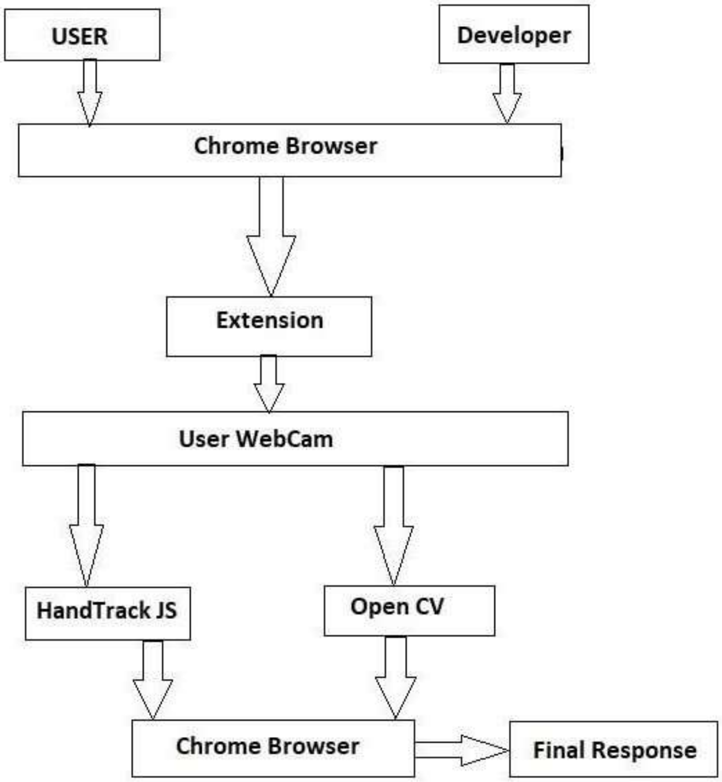


Figure 3: Hand Gestures Tracking System for Browser using Handtrack.Js

a). Creating extension: Basically, chrome extensions consist of three console i.e popup console, webpage console and background console. The popup console deals with the window appearing in the popup and objects seen in it. The webpage console deals with the objects seen in the actual webpage over which the popup is visible. The background console deals with the whole chrome browser (opening, closing, etc). The code developed for the application on the whole is split up into three components as above and appropriately given in the three files (popup.js, content.js and background.js). The manifest.json file holds up the important jargons for the chrome extension and it is important to include before the extension is launched. Appropriate functions should be used to navigate between the three layers of the extension. Some remote APIs and min.js files will not have permissions in the extension and it could be overcome by using "options.js" file.

b). Webcam: Webcam should be triggered with the help of usermedia codes in javascript. Thus, the video recorded by the webcam is stored in the variable. This variable is given as input to the Handtrack.Js. It processes the video and the returned response is manipulated.

c). Handtrack.Js: Handtrack.Js is an important module which can be implemented for capturing handtrack information. This module is implemented in the JS file and it tracks a video given as input. It tracks the position, clearness, etc. of the hand(s) in the video. It also notes the number of hands appearing on the screen. It returns a JSON object as a response which contains score, x coordinate, y coordinate and number of hands, etc., The X and Y coordinate is regularly (every one second) manipulated and analyzed. Based on the position of hands on the screen, the corresponding browser action is performed.

3.1 Design Workflow:

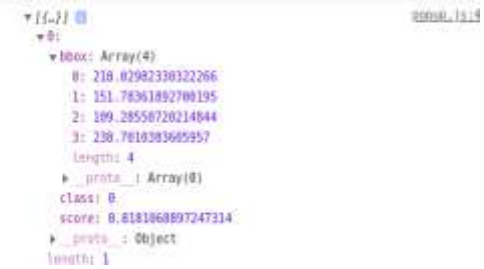
1. Design a layout for video streaming.
2. Enable and start the device web camera.
3. Stream the webcam video in the layout created in STEP 1.
4. Process the video streamed in STEP 3 using handTrackJS library.
5. Study the response output (coordinates) every second.
6. Initialize the above setup in a chrome extension base.
7. Embed the above sections of code in the PopupJS entity of the extension.
8. Pass the output (coordinates x and y) as arguments to the contentJS entity.
9. If $0 < x < 150$ *and* $y > 170$ then do backward action.
10. If $400 < x < 550$ *and* $y > 170$ then do forward action.
11. If $0 < x < 150$ *and* $y < 170$ then do new tab action.
12. If $400 < x < 550$ *and* $y < 170$ then do close tab action.
13. If $150 < x < 300$ *and* $100 < y < 300$ then do refresh or reload action.
14. Thus, the browser action is achieved by hand gestures.

4 EXPERIMENTAL RESULTS:

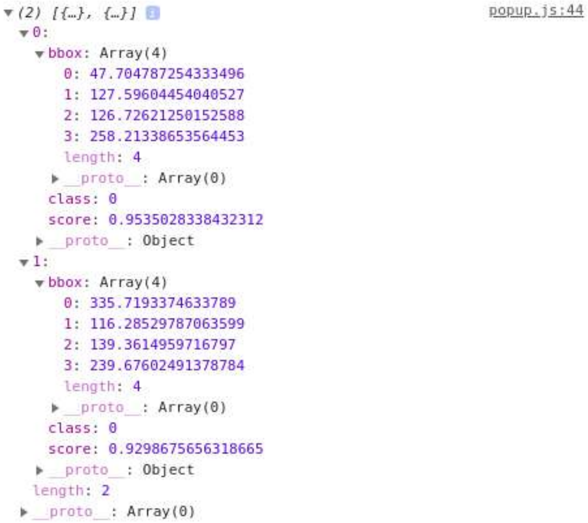
4.1 DATASETS:

Scores and positions of the hands were obtained as datasets. The positions (x,y coordinates) were used to do the necessary browser action (Backward , Forward , Refresh , New tab and Close Tab) . The below figures 4

& 5 indicates the output datasets for one hand and two hands recognition. Only one hand gesturing was used in the process. The table1 shows the mapping of dataset for a specific browser action.



```
▼ [{…}]                                popup.js:4
  ▼ 0:
    ▼ bbox: Array(4)
        0: 218.02982330322266
        1: 151.78361892700195
        2: 109.28558720214844
        3: 238.7010383605957
        length: 4
      ▶ __proto__: Array(0)
      class: 0
      score: 0.8181068897247314
    ▶ __proto__: Object
    length: 1
```



```
▼ (2) [{…}, {…}]                                 popup.js:44
  ▼ 0:
    ▼ bbox: Array(4)
        0: 47.704787254333496
        1: 127.59604454040527
        2: 126.72621250152588
        3: 258.21338653564453
        length: 4
      ▶ __proto__: Array(0)
      class: 0
      score: 0.9535028338432312
    ▶ __proto__: Object
  ▼ 1:
    ▼ bbox: Array(4)
        0: 335.7193374633789
        1: 116.28529787063599
        2: 139.3614959716797
        3: 239.67602491378784
        length: 4
      ▶ __proto__: Array(0)
      class: 0
      score: 0.9298675656318665
    ▶ __proto__: Object
    length: 2
  ▶ __proto__: Array(0)
```

Figure 4: One hand Datasets
Figure 5: Two hand Datasets

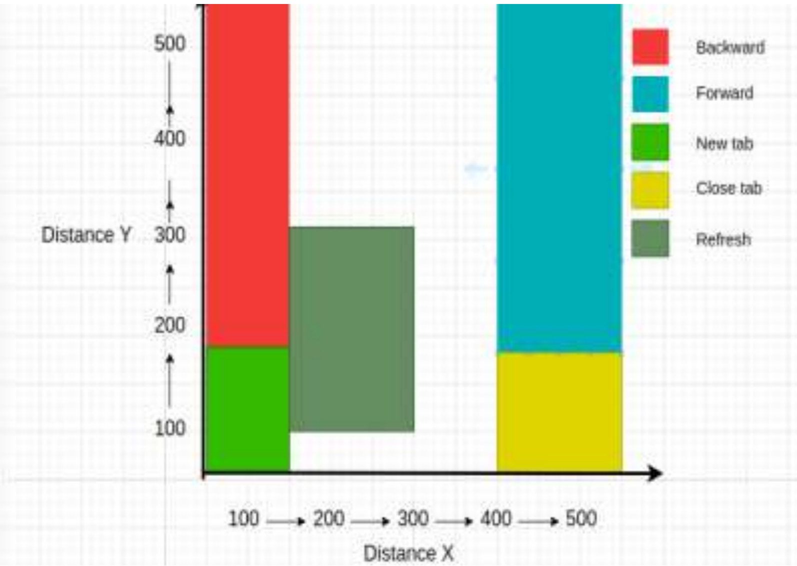
Action	Position in window	X value	Y value
Backward	Left bottom corner	*0 < x < 150*	*y > 170*
Forward	Right bottom corner	*400 < x < 550*	*y >170*
New tab	Left top corner	*0 < x < 150*	*y < 170*
Close Tab	Right top corner	*400 < x < 550*	*y < 170*
Refresh or Reload	Middle of window	*150<x<300*	*100 < y < 300*

Table 1: Datasets mapped for a specific browser Action

4.2 RESULTS:

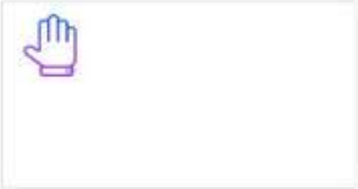
4.2.1 COORDINATES EXTRACTION:

The coordinates X and Y of the hand are extracted from the given video (webcam) using handTrack.JS. It returns a JSON object which contains many keys about the hand tracked. Coordinates are being extracted for the action. The above given chart is used to perform the necessary actions.

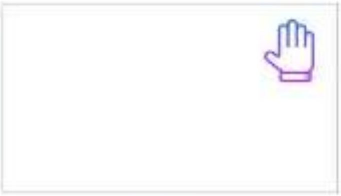


Graph 1: X and Y coordinates depicted in graph

The X and Y coordinate hand positions are clearly depicted in the above graph 1. The dataset outputs is equivalent to the scale values in the graph. The following section explains the proposed work with real time examples.

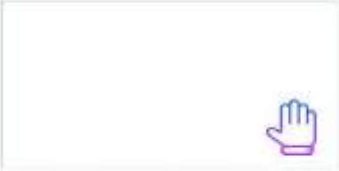


NEW TAB GESTURE
(Left side top of the window)



CLOSE TAB GESTURE
(Right side top of the window)

BACKWARD ACTION
(Left side down of the window)



FORWARD ACTION
(Right side down of the window)

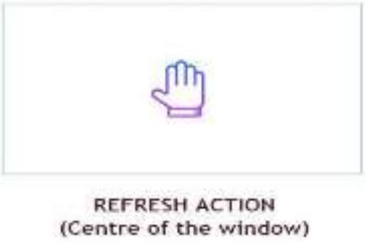


Figure 6: Visual Representation of Hand Gestures on Browser Wind

The above visual representations in figure 6 denotes the position of hands in front of the screen or monitor to do a respective action. The below series portrays the actual functioning of the Hand Gesture Extension. These are the sample snapshots of the gestures to do certain browser actions (back, close tab, new tab, refresh and forward respectively). When there is no hand in the popup screen no action will happen.

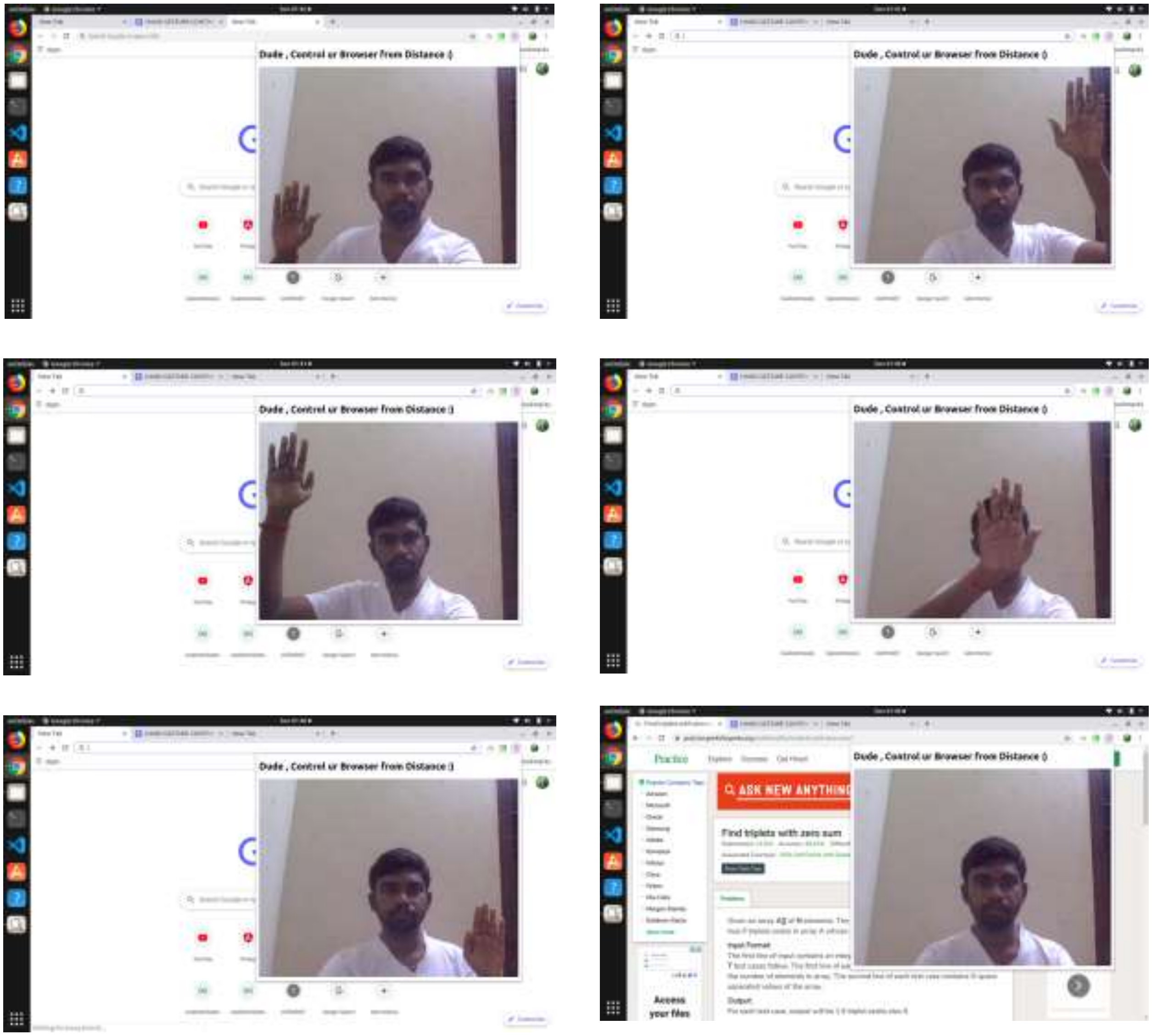
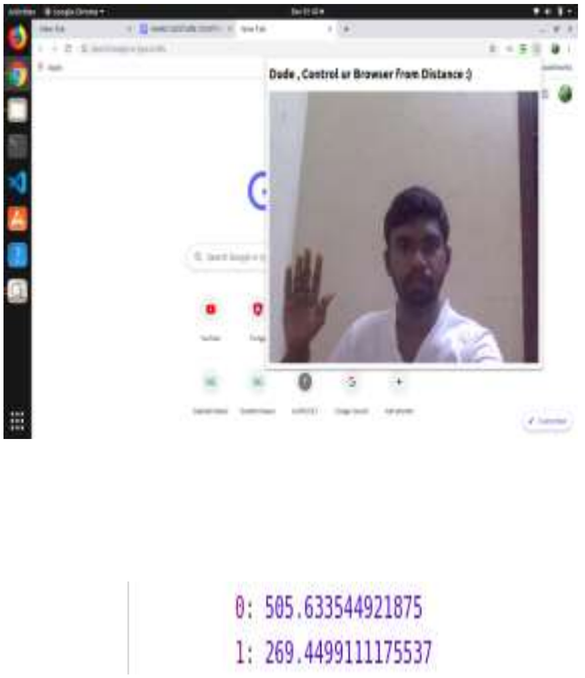

Dude , Control ur Browser from Distance :)
Dude , Control ur Browser from Distance :)
Dude , Control ur Browser from Distance :)
Dude , Control ur Browser from Distance :)
Dude , Control ur Browser from Distance :)
Dude , Control ur Browser from Distance :)
Practice
ASK NEW ANYTHING
Find triplets with zero sum
Access your files

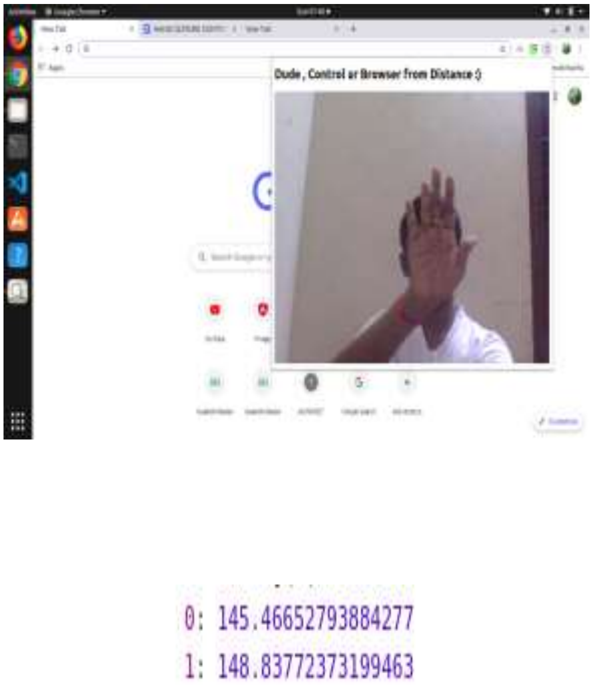
Figure 7: Snapshots of Hand Tracking Chrome Extension (Forward, New Tab, Close Tab, Reload, Backward and No action)

The real time screenshots in figure 7 depicts chromes extension popup displaying different states of gestures actions such as forwarding the screen, creation of new tab screen, closing the tab, reloading, backwarding the screen and no action.



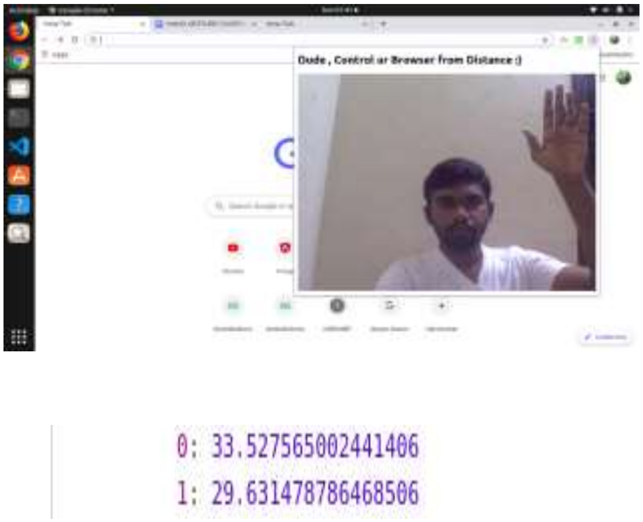
```
0: 505.633544921875
1: 269.4499111175537
```

Figure 8: x and y coordinates for Forward Action



```
0: 145.46652793884277
1: 148.83772373199463
```

Figure 9: x and y coordinate for Refresh Action



```
0: 33.527565002441406
1: 29.631478786468506
```

Figure 10: x and y coordinate for New Tab Action

As per our results figure 8 shows the forward gesture and the dataset (x and y value) which refers to the table 1 *$400 < x < 450$ and $y > 170$*, followed by the figure 9 which shows the reload gesture and the according dataset *$150 < x < 300$ and $100 < y < 300$* and the figure 10 shows the new tab gesture with dataset (x and y value) *$0 < x < 150$ and $y < 170$*. Thus, the experimental values for each coordinate are depicted.

4.2.2 BROWSER ACTION:

The datasets i.e the X and the Y coordinates are obtained and the next process is to render a necessary browser action based on table 1 given data. For this operation the basic javascript commands are used in table 2. As the webcam video is processed every second and the appropriate datasets are returned to the console, when a dataset is obtained, it is processed to do the programmed browser action.

a). JavaScript commands:

Action	**JS Command**
Backward	window.history.back();
Forward	window.history.forward();
New Tab	window.open(“www.google.com”,”_blank”);
Close Tab	window.close();
Refresh	document.location.reload();

Table 2: JavaScript Commands for the Browser Actions

Thus, the appropriate browser action is tendered. The chrome extension would automatically be enabled in the new window and can continuously perform another browser action. Then the extension will be available in the extension store explicitly.

5 CONCLUSION AND FUTURE WORKS:

The paper portrays the concepts of Natural User Interface (HandTrack.Js) to implement hand gesture tracking and use it to do necessary browser action. It finally resulted in the successful design of the chrome extension which helps chrome users to easily navigate through back, forward and tabs via their hand gestures. The main advantage of this work is that it minimizes human work easily by controlling the browser remotely. This can also play a major role for security purposes. This technique can be extended in many applications like TV, mobile, tablet for education, medicine and for defence. It would be useful to people if it is implemented not only in browsers but also in many offline applications available [1][8].

The control of zoom in and zoom out of web pages, image sliding, gesture control platform for YouTube, online virtual music makers and many more applications can also be extended. The HandTrack library also provides room for developing virtual games and gesture-controlled products. Similarly, OpenCV models, CNN, k-NN models [4]-[6] can also be used to define logic for Natural User Interface.

6 REFERENCES:

1. Hongliang Bi, Jian Zhang and Yanjiao Chen, 20 Jan 2020, ”SmartGe: Identifying Pen Holding Gesture with SmartWatch”, IEEE Hand Gesture, vol. 8.

2. Muneer Al-Hammadi, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, Mohamed A. Bencherif, Mohamed Amine Mekhtiche, 27April 2020, ”Hand Gesture Recognition for Sign Language Using 3DCNN”, IEEE Hand Gesture ,vol. 8.
3. Yunlong Che and Yue Qi, 17 February 2020, ”Embedding Gesture Prior to Joint Shape Optimization Based Real-Time 3D Hand Tracking”, IEEE Hand Gesture, vol. 8.
4. Handri Toar, Budi Sugandi, Ary Alfianto, 3-4 Oct 2018, “Hand gesture-based motion control for robotic arm using stereo camera”, International Conference on Applied Engineering (ICAE).
5. Perri Zhang, Lingzhi Yang, Zhishuai Han, Yuxing Gao, Xiaojuan Ban, 15 March 2018, “A posteriori processing optimization method for gesture interaction in aseptic operation room”, Hand Gesture, vol.6.
6. Takuya Sakamoto, Xiaomeng Gao, Ehsan Yavari, Ashikur Rahman, Olga Boric-Lubecke, Victor M. Lubecke, 21 August 2018, “Hand gesture recognition using a radar echo i–q plot and a convolutional neural network”, Hand Gesture, vol.2.
7. Fan Zhang,Shaowei Chu, Ruifang Pan, Naye Ji, Lian Xi, 29 June 2017, “Double hand-gesture interaction for walk-through in VR environment”, IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS).
8. Vinod Keshav Seetharamu, Joy Bose, Sowmya Sunkara, Nitesh Tigga, 11-13 Dec 2014, “TV remote control via wearable smart watch device”, Annual IEEE India Conference (INDICON).
9. Anupam Agrawal, Rohit Raj, Shubha Porwal, 11-12 April 2013, “Vision-based multimodal human-computer interaction using hand and head gestures”, IEEE Conference on Information & Communication Technologies.
10. Piyush Kumar, Anupam Agarwal, Siddharth S. Rautaray, 15-17 March 2012, “Hand data glove: A new generation real-time mouse for Human Computer Interaction”, 1st International Conference on Recent Advances in Information Technology (RAIT).
11. Yu Shi, Ronnie Taib, Serge Lichman, 5-8 Dec 2006, “GestureCam: A smart camera for gesture recognition and gesture controlled web navigation”, 9th International Conference on Control, Automation, Robotics and Vision.