Integrated System Controls Manager Using Hand Gestures

K. Bala Chandra Reddy,
M.Tech, (Ph.D.)
Assistant Professor,
Department of CSE,
JNTUACEP, Pulivendula.
kakarla505@gmail.com

Ramavath Balaji Naik
20191A0552,
Student,
Department of CSE,
JNTUACEP, Pulivendula.
yuvibalaji015@gmail.com

Sunkara Chaithanya Krishna
20191A0555
Student,
Department of CSE,
JNTUACEP, Pulivendula.
chaithanyak155@gmail.com

Abstract:

In this paper, we have developed an Integrated System Controls Manager, utilizing hand gestures as input for system manipulation. OpenCV and Media-Pipe modules form the basis of this implementation to interpret gestures. The system utilizes a webcam to capture images and videos, allowing users to control a range of functions seamlessly. These functions include adjusting volume and brightness, navigating mouse movements and clicks, scrolling through content, and locking the cursor for precise interactions. Additionally, users can manage tabs by switching between them, minimizing or closing them, and perform text-related actions such as copying and pasting through intuitive gestures. Furthermore, the system enables zooming in and out of windows or PDF files for enhanced viewing flexibility. To distinguish between the many gestures, we have introduced a 5-bit binary number-based approach, facilitating the implementation of multiple gestures simultaneously Using real-time gesture recognition, individuals can interact with a computer by making hand gestures in front of a camera connected to the system. This approach enables system control via hand gestures, reducing reliance on keyboard and mouse controls to some extent.

Keywords:

Gesture recognition, human-machine interaction (HMI), OpenCV, media pipe, 5-bit Binary Encoding.

I. INTRODUCTION

There have been numerous recent advancements in human-computer interaction. Among these, hand gestures stand out as one of the most effective and expressive forms of communication, akin to a universal language understood by both the hearing and the speech-impaired. Throughout history, gestures such as handshakes, hand is raised and moved from side to side for “hi” have played integral roles in human communication. It's widely recognized that gestures offer a straightforward means of interaction. So, why not extend this natural mode of communication to our machines? In this endeavour, we are aimed to demonstrate real-time gesture recognition.

The objectives of our research include addressing the discomfort and inconvenience often associated with touchpads on most laptops, and streamlining the pre-processing of data to enhance its interpretability and processing efficiency. Additionally, we aim to reduce hardware costs. Through this research, our goal is to develop cost-effective hand gesture recognition software compatible with PCs and laptops equipped with a built-in web-cam. This project encompasses a hand recognition application capable of performing various functions, including volume and brightness adjustment based on bounding box area and preset features, as well as mouse pointer movement, position locking, clicking, text selection along with copy and paste of text, tab minimization, maximization, and closure, zooming in, and zooming out. These actions are categorized based on distinct hand gestures associated with each feature mentioned above.

II. LITERATURE SURVEY

Various authors have undertaken research in the field mentioned, and their contributions are outlined below:

In the paper "Design And Implementation of Virtual Mouse With Volume/Brightness Control Using Hand Gestures" in 2023 on Emerging Trends in Engineering and Management by Jaspreet Kaur, Silky Khurana, Kamal, Computing & Intelligent Systems, 45-55. They have introduced a method for manipulating the cursor's position, adjusting volume, and controlling brightness solely through hand gestures, eliminating the need for electronic devices.[1]

V. Kavitha, G. Amrutha, N. Mahithosh, N. Venkata Subba Rao, and K. Shalem Raj conducted research on "AI Virtual Mouse using Hand Gestures" (Volume 11, Issue 5, No. 115, May 2022). Their study focuses on virtually replicating mouse features such as left clicks, right clicks, scrolling up, and scrolling down using a chatbot interface to initiate the virtual AI mouse. However, they encountered challenges with precise left-right clicks at specific positions and dragging and text selection using the cursor.[2]

Arzoo Poswal, Martendra Pratap and Eshu presented research on volume control using Hand Gesture Recognition in the International Journal of Research Technology and Innovative Science on May 2022 in Issue 5, Volume 7. Their project enables recognition of human hand gestures to control devices through real-time gesture recognition, employing OpenCV and Python.[3]

J. Jaya, S. Shriram, P. Ajay, B. Nagaraj, S. Shankar elaborated on an AI virtual mouse system using head detection fingers and hand gestures published on 2021, Article ID 8133076 in the Hindawi Journal of Healthcare Engineering.[4]

Vishal Jayaswal, Vijay Kumar Sharma, Md. Iqbal, Vimal Kumar, Sachin Tawara, and discussed a hand gesture based virtual mouse control system on Volume 7, Issue 12 of 2020 in the GIS Science Journal. Their proposal allows cursor manipulation solely through hand movements without electronic devices. The system relies on a webcam used as an input device, with operations performed i.e. clicking and dragging of files and icons, displaying output of the camera on the system screen.[5]

Ali Alnaji, Munir Oudah and Javaan Chahal discussed computer vision based hand gesture recognition in the J. Imaging 2020, 6, 73 publication. They covered various hand gesture techniques, analyzing OpenCV methods' performance, hand segmentation techniques, algorithms, drawbacks, types of gestures, and datasets.[6]

Namratha V, Nidhishree Arun, Shreenivas B and Ananya Dutta showcased gesture recognition for hand and controlling volume using Python and OpenCV in the International Journal of Creative Research Thoughts (IJCRT). Their system simplifies handling sound devices through efficient hand segmentation techniques and custom functions for gesture detection.[7]

Mokhtar M. Hasan and Pramod K. Mishra published a research paper within Vol. 4, Issue 5 of the International Journal of Image Processing, detailing a brightness controller using hand gestures. Their approach involves using the HSV colour module for edge detection and segmentation for data storage in images, detection algorithms, feature extractions, and matching templates. To address limitations, they increased the no. of samples per gesture, thereby expanding the size of database size and it's processing time.[8]

Rafiqul Zaman Khan and Noor Adnan Ibrahim did a study about a hand gesture recognition system. They published their work within Volume 3, Number 4, in July 2012 on the International Journal of Artificial Intelligence and Applications. In their study, they used a method called Gaussian distribution with multiple variables. to recognize hand gestures.[9]

R. Alami's research, an approach based on methodology that links gesture classification to discerning human intentions within the realm of human-robot interaction. She proposed a methodology for linking gesture classification to human intention identification in human-robot interaction. The study investigates how gestures can be classified to infer human intentions during interactions with robots.[10]

These studies contribute significantly to the advancement of hand gesture recognition systems and their applications in various fields.

III. OVERVIEW OF SYSTEM:

A. *Existing Systems:*

computer mouse functions as an input device facilitating pointing and interaction with on-screen elements. With significant technological A advancements, wireless mice have emerged to streamline movement and enhance accuracy. However, despite improvements in mouse precision, inherent limitations persist. As a hardware input device, issues such as malfunctioning mouse clicks can arise due to wear and tear, necessitating replacement.

Similarly, the use of keyboards may also encounter issues over time, such as certain command keys ceasing to function properly. Consequently, periodic replacement of keyboards becomes necessary to ensure optimal performance, particularly when critical command keys become unusable.

Also there exist a lot of projects with individual features i.e.

Modules	**Existing features**
Volume controller	Controlling volume using Pinch gesture
Brightness Controller	Controlling brightness using Pinch gesture
Mouse	Curser Movement, Right click, Left Click, Scroll-up & Scroll-down

B. *Proposed System:*

The system which we proposed aims to address real world challenges by enabling virtual execution of multiple system controls concurrently. Situations may arise where physical mouse usage is impractical, or integrated laptop keyboards experience malfunctioning function keys. Moreover, in scenarios akin to the COVID-19 pandemic, where touch-based interactions pose infection risks, adopting touchless devices becomes imperative. Thus, the proposed system offers a viable solution. By leveraging hand and fingertip gestures detected via the built-in webcam or mobile camera with the assistance of Droid-Cam, the system facilitates PC or laptop mouse function control and certain keyboard functionalities.

The newly proposed features include:

Modules	**Proposed features**
Tabs controller	Tab Minimization, Tabs Swapping, Tab closing
Zoom Controller	Window Zoom in-Zoom out, Document or Pdf Zoom in-Zoom out.
Copy & Paste	Copying of images or text, Pasting of images or text.

The modifications added for existing features are:

Modules	**Proposed features**
Volume controller	Addition of gesture detection based on hand's bounding box area, Set volume based on pinky finger
Brightness Controller	Addition of gesture detection based on hand's bounding box area, Set brightness based on [11111] gesture
Mouse	Locking Curser , Text Selection.

IV. METHODOLOGY

The system which we proposed is a combination of 7 subsections which are clubbed together with implementation similar to Rock-Paper-Scissor model as the user has to show rock sign each time he wants to implement the one functionality, such that the rock sign helps in switching between modules. (Note that this many subsections are feasible with only two hands with left-right hand classification with help of media pipe.) The 7 subsections are virtual mouse with text Selection& cursor locking, Copy& Paste, Zoom controller, volume controller, brightness controller, tab controls and scrolling.

Working of Implementation Block Diagram for our model:

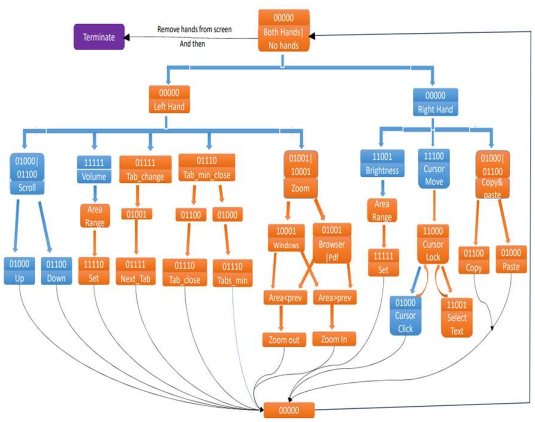


Fig (a): Model of Implementation

Understanding of above block diagram involves mapping of specified 5-bit binary numbers with respect to each hand gesture i.e.

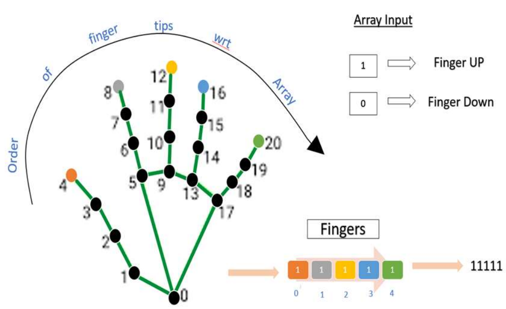


Fig (b): Hand Gesture Encoding

Either it is left or right hand the Fingers array indexing start from thumb to Pinky finger. We call this as encoding of hand fingers into a 5-bit binary number.

Ex: encoding for left hand == {Left, [11111]}.

To identify whether finger is up(1) or down(0), the approach is as follows:

a)For the thumb finger, if x-distance(node(4)) > x-distance(node(3)), then finger is up and vice versa.

b)But for remaining fingers, if y-distance(node(u)) > y-distance(node(v)), then finger is up and vice versa.

Where as,

$\forall u \in \{8,12,16,20\}$, $\forall v \in \{6,10,14,18\}$,

"x-distance" means distance of node(u) or node(v) from x co-ordinate,

"y-distance" means distance of node(u) or node(v) from y co-ordinate.

Mapping of each node() label with respect to fingers be:

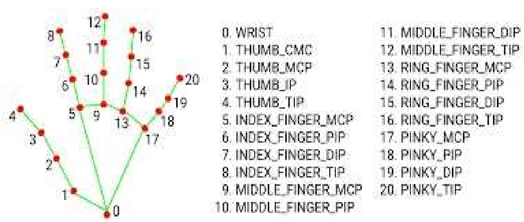


Fig (c): Hand Land-Marks

Also in our model we added a feature for detecting bounding box around hand having area range to detect gesture at a specific distance from screen, to make model more feasible and effective virtually i.e.

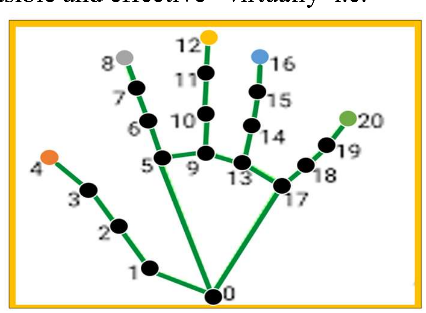


Fig (d): Model of Implementation

Ex: In above diagram We will consider the extreme points for nodes {4(left most point), 12(top most point), 20(right most point), 0(bottom most point)}. Point looks similar to (x_i, y_i). Now using extreme values we find corner points for the rectangle that can be drawn around hand. Finally using corner points, we can find length and breadth of rectangle and finally area using derived length, breadth of rectangle using formula:

Area = Length * Breadth

V. IMPLEMENTATION

The approach used in implementation of each gesture of the combined system is explained separately. Each resultant operation contains at most 3 steps to interpret small transition in a gesture to perform each system control(observe step no. on each window within one system-control's image under Results & Analysis section later). These are the following subsections:

A. Tabs Controller:

The approach to implement controlling of tabs using hand gestures:

A transition of encoding i.e. {Left, [01111]} -> {Left, [01001]} -> {Left, [01111]} will switch to the next tab.

A transition of encoding i.e. {Left, [01110]} -> {Left, [01001]} -> {Left, [01111]} will close the current tab.

A transition of encoding i.e. {Left, [01111]} -> {Left, [01001]} -> {Left, [01111]} will minimize all the current tabs.

B. Zoom in-out Controller:

Controlling window or PDF file zoom in/out using hand gestures can be achieved using computer vision libraries like OpenCV and gesture recognition techniques.

Mapped encoding for zoom in-out for Window is {Left, [10001]}, while for pdf file zoom in-out is {Left, [01001]}.

Based on area of Bounding box created around hand gesture. The algorithm works as :

If area(current frame bounding box) < area(previous frame bounding box)

then Perform zoom out.

Elif area(current frame bounding box) > area(previous frame bounding box)

then Perform zoom in.

C. Copy & Paste Controller:

The core of implementing copy& paste exist only when we determine how users will select text or images to copy and where they will paste them. This could be within a specific application or across different applications. Copy& Paste might involve simulating keyboard shortcuts (e.g., Ctrl + C for copy, Ctrl + V for paste) or using system-level APIs for clipboard manipulation.

Gesture encoding for the copy function i.e. {Right, [01100]}, while for paste function i.e. {Right, [01000]}.

D. Virtual mouse:

The virtual mouse system utilizes an algorithm that translates fingertip coordinates from the webcam screen to the entire window screen of computer, enabling the control of mouse.

Gesture encoding for the cursor movement function i.e. {Right, [11100]}, while for cursor locking function i.e. {Right, [11000]}, also for the text selection i.e. {Right, [11001]}, finally for , click& Double click functions i.e. {Right, [01000]} or {Right, [01100]}.

E. Volume controller:

Controlling volume using OpenCV be accomplished in three simple steps:

• Step -1: Identify the hand gesture, such as {Left, [11111]}.

• Step -2: Based on the area range, detect step 3.

• Step -3: Determining the distance from the tip of the thumb to the tip of the index finger.

• Step -4: Utilize Pinky finger gesture to adjust the volume, for example, {Left, [11110]}.

• Step -5: Associate the distance between the thumb's and index finger's tip with the range of volume. In this instance, the range spans from (15 to 220)mm, correlating with a volume range of (-63.5 to 0.0)dB.

Typically, volume adjustments should not rely on percentages due to their assumption of linearity. However, by converting them into percentages using NumPy arrays, the percentages become applicable, aligning with dB values where 0% corresponds to -63.5dB and 100% corresponds to 0dB.

Also there exist area range for the hand's bounding box such that it can only detect volume changes, when Step 2 is performed within area range. In this case, area range is 250 to 1000 mm^2.

Finally Pinky-finger helps in stabilizing the volume set such that it will be comfortable to set volume.

F. Brightness controller:

The method closely resembles that of volume control. Initially, landmarks of the hand are detected along with gesture encoding, such as {Right, [11001]}, followed by the calculation of the distance between the thumb's and the index finger's tip. Subsequently, this distance is correlated with luminance range. In this scenario, the distance spans from 15mm to 220mm, with a minimum distance of 15mm and a maximum of 220mm. The luminance range is set between 200 and 500 nits, where 1 nit equals one candela per square meter ($1cd/m^2$). Additionally, an area range of 100 to 1000 mm^2 is employed to detect changes in brightness. Ultimately, the brightness level can be adjusted to the desired setting by pausing the gesture with {Right, [11111]}.

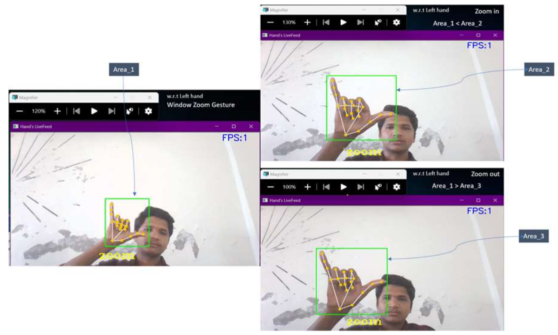
G. Null Gesture:

This gesture is similar to the reset function as it helps in switching between various gestures.

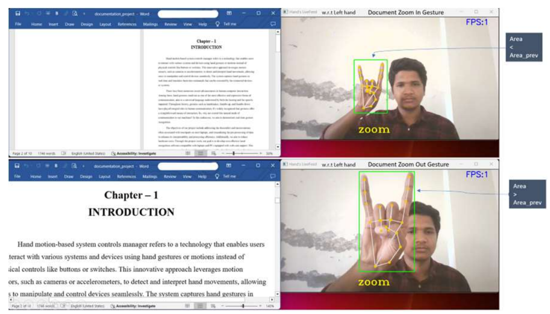
VI. RESULT AND ANALYSIS

A)Zoom Controller:

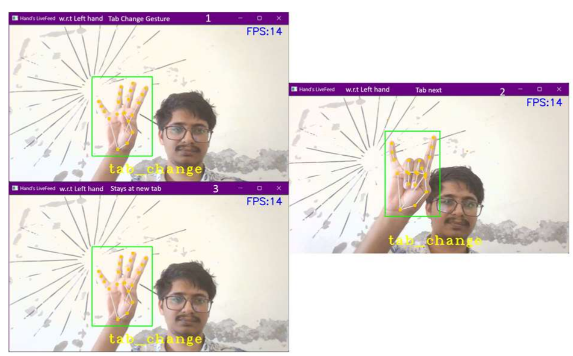
(i)Window Zoom in-out {Left, [10001]}:



(ii) Document Zoom in-out {Left, [01001]}:

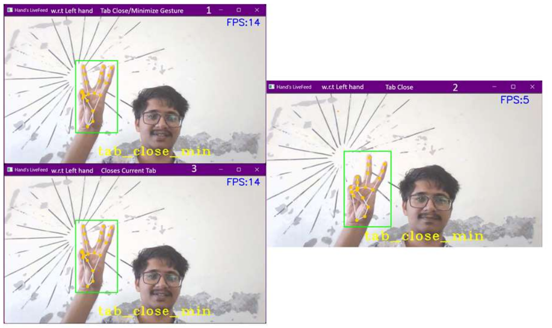


B) Tab Change {Left, [01111]}:

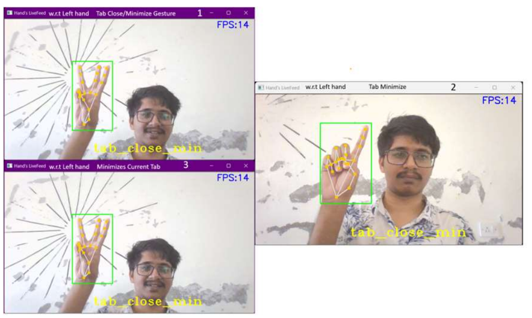


C)Tab Close/Minimization{Left, [01110]}:

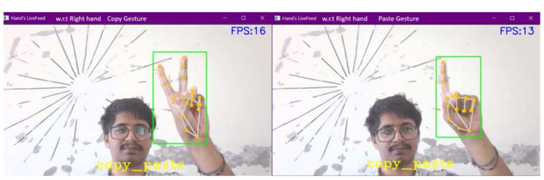
(i)Close Current Tab{Left, [01100]}:



(ii)Minimize Current Tabs{Left, [01100]}:

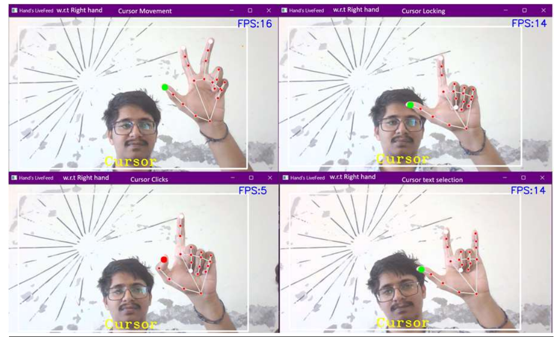


D)Copy{Right,[01100]}&Paste{Right, [01000]}:

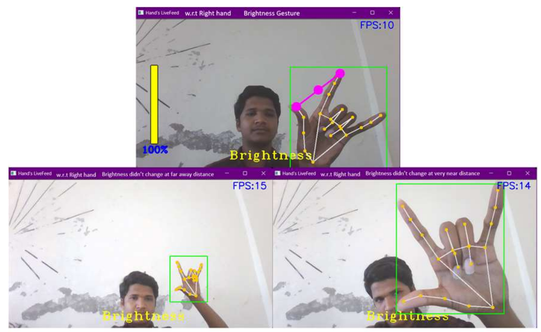


E)Scroll-Up{Left,[01100]}&Scroll-Down{Left,[01000]}:

F)Virtual Mouse{Right, [11100]}:



G)Brightness Controller{Right, [11101]}:



H)Volume Controller{Left, [11111]}:

I)Null Gesture{Right/Left, [00000]}:

VII. CONCLUSION

The Integrated System Controls Manager offers a more intuitive approach to managing a maximum number of system controls, such as mouse functionalities like cursor movement, text selection with the cursor, cursor locking, volume and brightness control with set features, and area-based detection. Additionally, it includes special features like tab controls for closing, minimizing, and swapping between tabs, also zoom in-out of window as well as documents or PDFs. With text selection at our fingertips, we can perform copy and paste activities using hand gestures instead of relying solely on traditional mouse and keyboard controls.

Based on the results of the project, we can conclude that the proposed Integrated System Controls Manager has performed well and demonstrates better accuracy compared to existing models. Furthermore, the model successfully addresses most limitations present in current existing systems. However, the model does have some limitations, such as a slight decrease in accuracy when transitioning from tab control features, like minimizing or closing tabs, to scrolling features, as their respective gestures differ slightly i.e. {Left, [01110]} -> {Left, [01100]} or {Left, [01000]}.In our future work we will work on improving and classifying the gestures better, to achieve maximum accuracy.

VIII. FUTURE SCOPE

Also, We have thought about another feature to implement i.e. replacing keyboard with a speech to text model such that the feature works like taking microphone input from user with a hand gesture and records audio and processes it and gets converted into text, then it will automatically be copied to clipboard. As we already can paste a copied text, such that we can virtualize typing with a speech to text model. For now we couldn't find any feasible pretrained model works well with our proposed system, so this will be implemented in our future work with a feasible solution.

REFERENCES

[1] "Design and Implementation of Virtual Mouse With Volume/Brightness Control Using Hand Gestures" by Jaspreet Kaur, Silky Khurana, Kamal in 2023 on Emerging Trends in Engineering and Management, 45 to 55.

[2] V. Kavitha, G. Amrutha, N. Mahithosh, N. Venkata Subba Rao, and K. Shalem Raj on "AI -Virtual Mouse using Hand Gestures" (Volume 11, Issue 5, No. 115, May 2022)

[3] Eshu Yadav, Arzoo Poswal, Martendra Pratap Singh on May, 2022 at International Journal of Innovative Science and Research Technology under Volume 7, Issue 5.

[4] S. Shankar, B. Nagaraj, S. Shriram, J. Jaya and P. Ajay on 2021 at Hindawi Journal of Healthcare &Engineering.

[5] "A novel method for controlling the cursor on a computer screen using hand gestures", Vimal Kumar ,Vijay Kumar Sharma, Sachin Tawara, Vishal Jayaswal, Md. Iqbal at GIS Science journal within vol.7, Issue 12 on 2020.

[6] "Computer vision-based hand gesture recognition" , Ali-Alnaji, Munir Oudah and Javaan Chahl at J. Imaging on 2020 within volume 6, issue 73.

[7] Namratha V , Nidhishree Arun, Sreenivas B, Ananya Dutta at International Journal of Creative Research Thoughts, "Controlling volume using Python and OpenCV".

[8] Pramod K. Mishra, Mokhtar M. Hasan "hand gesture brightness control system" within volume 4, issue 5 at international Journal of Image Processing.

[9] "Hand gesture recognition system" by and Noor Adnan Ibraheem and Rafiqul Zaman Khan on International Journal of Artificial Intelligence and Applications, Vol.3, No.4, July2012.

[10] "A methodological approach related to the classification of gestures for identification of human intent in the context of human-robot interaction", by R. ALAMI, Chrystopher L. Nehaniv, K. Dautenhahn, in IEEE International journal, on 3 October 2005.