www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

# VOLUME AND BRIGHTNESS CONTROL VIA HAND GESTURES

***Mrs.K.Sheetal[1]., Puram Ashritha[2]***
*1 Assistant Professor, Department of CSE, Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India*
*2, B.Tech CSE (21RG1A05Q0),*
*Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India*

## ABSTRACT

We are developing a volume and brightness controller in which we are using hand gestures as the input to control the system , Opencv module is basically used in this implementation to control the gesture. This system basically uses the web camera to record or capture the images /videos and accordingly on the basis of the input, the volume and brightness of the system is controlled by this application. The main function is to increase and decrease the volume and brightness of the system. The project is implemented using Python, OpenCV.

**Keywords**:Gestures, Convolutional Neural Networks, Opencv,Mediapipe, Human Computer Interaction

## I. INTRODUCTION

Hand gestures are unprompted and also robust transmission mode for Human Computer Interaction (HCI). Keyboard, mouse, joystick or touch screen are some input devices for connection with the computer but they don't provide appropriate interface whereas, the current system will contain either desktop or laptop interface in which hand gesture can be done by wearing data gloves or web camera used for snapping hand image. The first step towards this gesture recognition is hand capturing and analyzing. Sensors are used in Data-Glove methods for initializing fingers movement and other sensor will program hand movements. In comparison the vision-based method only needs a camera and hence identifying the actual interaction between human and computer without using any other devices. The challenges of this system are constant background, sometimes person and lighting also. Different procedure and algorithms which are used in this system are elaborated here along with the recognition techniques. The method of searching a connecting region in the picture with particular specification, being it color or intensity, where a pattern and algorithm is adjustable is known

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

as segmentation. Vision Based method requires a web camera, so that one can realize natural interaction between humans and computer without using any other devices. The challenging part in these systems is background images or videos which is recorded or captured during taking the inputs i.e. hand gesture by the user, also sometime lightning effect the quality of the input taken which creates the problem in recognizing the gestures. Process to find a connected region within the image with some of the property such as color ,intensity and a relationship between pixels i.e. pattern is termed as segmentation. And have used some important packages which have OpenCv-python, tensorflow, numpy, mediapipe, imutils, scipy, numpy.

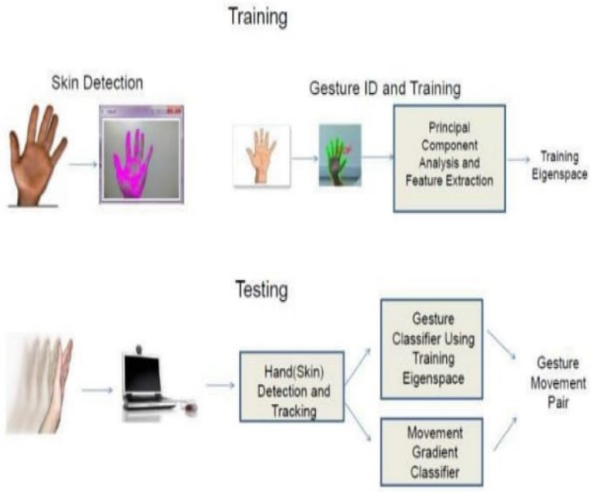



**Fig:1 System Architecture**

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

## II.RELATED WORK

| S. No | AUTHORNAME | PROBLEMS IDENTIFIED | TECHNIQUES USED | ACCURACY | DRAWBACKS |
|---|---|---|---|---|---|
| 1. | Mahmoud E and Bernd M | To Recognize the Isolated and Meaningful Hand Gesture | Hidden Markov Model | 93.84 | It is having high Computational consuming |
| 2. | Hasan | Gesture Recognition based on brightness factor matching | Trimming and Scaling Normalization Technique | 95% | It uses the complete frame or complete web cam. |
| 3 | Robust | Gesture recognition for robotic control | Dynamic Gesture recognition | 93.2% | Using lowest pixels cameras |

## III.SYSTEM ANALYSIS

A DSP, DM6437 of Texas Instruments, is used for our portable hand gesture recognition system. For the real-time hand gesture sensing and recognition system, we propose a finger skin pixel algorithm to quickly and easily distinguish the hand in a complex image and use the region of interest to reduce the amount of computation. The hand trace direction can be found easily using a hand gesture center point in our system. Finally, our system is applied to TV channel and volume control using hand gestures and hand tracing.

Applied scaled normalization for gesture recognition based on brightness factor matching. The input image with is segmented using thresholding technique where the background is black. Two methods are used for extraction the features; firstly, by using the edge mages, and secondly by using normalized features where only the brightness values of pixels are calculated and other black pixels are neglected to reduce the length of the feature vector . The database consists of 6 different gestures, 10 samples per gesture are used, samples for training and 5 samples for testing.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

We are developing a volume and brightness controller in which we are using hand gestures as the input to system.

A vision-based hand Gesture system that does not require any special markers or gloves and can operate in real-time on a commodity PC with low-cost cameras.

## IV. IMPLEMENTATION

### Conventional Neural Network

Step1: Convolutional operation

The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out

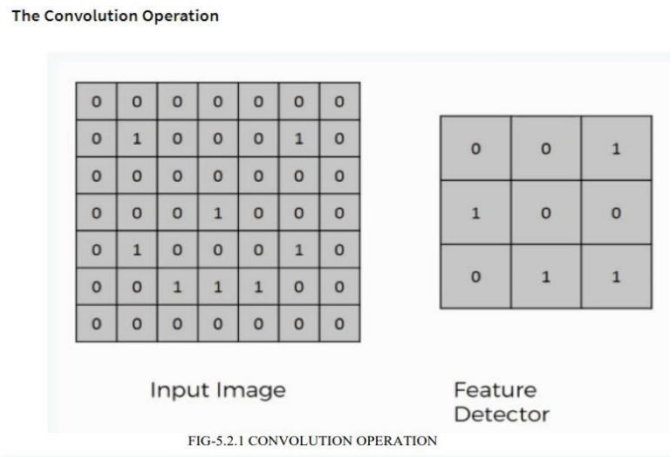
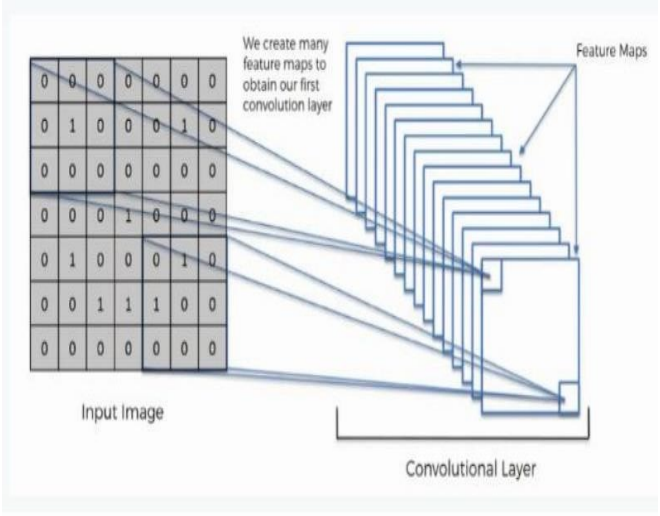
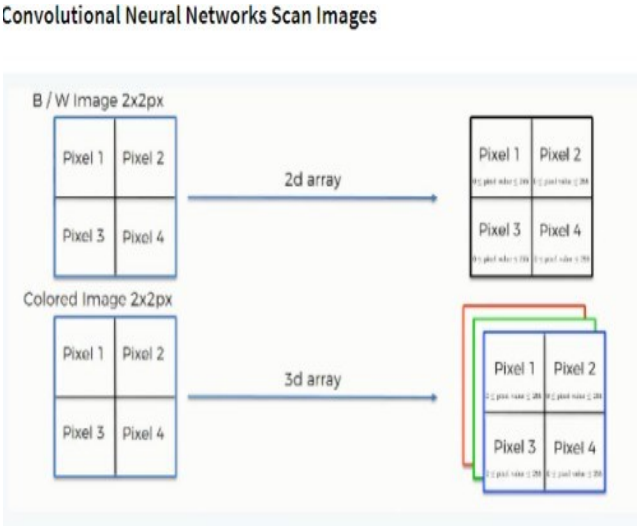



FIG-5.2.1 CONVOLUTION OPERATION




Step (1b): Relu Layer

The second part of this step will involve the Rectified Linear Unit or Relook. We will cover Relook layers and explore how linearity functions in the context of Convolutional Neural Networks. Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.




Step 2: Pooling Layer

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.

Step 3: Flattening

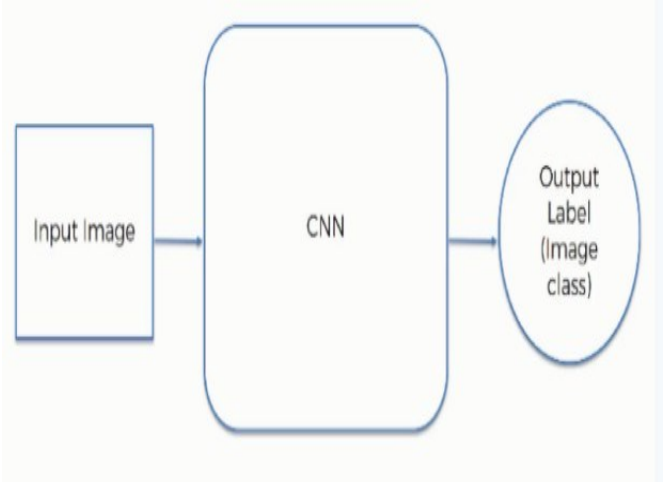
This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

Step 4: Full Connection

In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.

Summary

In the end, we'll wrap everything up and give a quick recap of the concept covered in the section. If you feel like it will do you any benefit (and it probably will), you should check out the extra tutorial in which Soft ax and Cross-Entropy are covered. It's not mandatory for the course, but you will likely come across these concepts when working with Convolutional Neural Networks and it will do you a lot of good to be familiar with them.




## V.MODULE DISCRIPTION

**OpenCV:**

OpenCV is a library of programming capacities mostly focused on ongoing PC vision. Initially created by Intel, it was later bolstered by Willow Garage then Itseez . The library is cross-stage and free for use under the open-source BSD permit.

OpenCV underpins a few models from profound learning structures like TensorFlow, Torch, PyTorch (in the wake of changing over to an ONNX model) and Caffe as indicated by a characterized rundown of upheld layers. It advances Open Vision Capsules. , which is a versatile configuration, perfect with every other organization. Authoritatively propelled in 1999 the OpenCV venture was at first an Intel Research activity to propel CPU- concentrated

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

applications, some portion of a progression of undertakings including constant beam following and 3D show dividers. The principle supporters of the undertaking remembered various improvement specialists for Intel Russia, just as Intel's Performance Library Team.

OpenCV is written in C++ and its essential interface is in C++, yet it despite everything holds a less far reaching however broad more seasoned C interface. There are ties in Python, Java and MATLAB/OCTAVE. Since variant 3.4, OpenCV.js is a JavaScript official for chose subset of OpenCV capacities for the web stage.

The entirety of the new turns of events and calculations in OpenCV are presently evolved in the C++ interface.

OpenCV runs on the accompanying work area working frameworks: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the accompanying portable working frameworks: Android, iOS, Maemo, BlackBerry 10. The client can get official discharges from SourceForge or take the most recent sources from GitHub. OpenCV utilizes CMake.

**Advantages:**

- ❖ OpenCV is accessible liberated from cost.
- ❖ As OpenCV library is written in C/C++ it is very quick.
- ❖ Low RAM use .
- ❖ It is versatile as OpenCV can run on any gadget which runs on C

As a human, handling pictures is a characteristic procedure as we cooperate with numerous individuals and it is simple for us to perceive in our day by day lives. PCs, then again need to follow by their own interesting procedures so as to break down huge measures of media information. All together for profound learning PC vision to flourish, thousands on a great many photographs, recordings, and different pictures should be incorporated for a viable AI to get valuable

The PC arranges a picture dependent on pixel esteems, it isolates a picture into an enormous framework of boxes - also called pixels - and allots a number to each case .

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. learning algorithms. identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

**NumPy:**

NumPy is a Python library, including support for enormous, multi-dimensional clusters and frameworks, alongside a huge assortment of significant level numerical capacities to work on these exhibits. The predecessor of NumPy, Numeric, was initially made by Jim Hugunin with commitments from a few different designers. In 2005, Travis Oliphant made NumPy by consolidating highlights of the contending Numarray into Numeric, with broad alterations. NumPy is open-source programming and has

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

numerous givers. Such clusters can likewise be sees into memory cradles distributed by C/C++, Cython, and Fortran expansions to the CPython mediator without the need to duplicate information around, giving a level of similarity with existing numerical libraries. This usefulness is misused by the SciPy bundle, which wraps various such libraries . NumPy has worked in help for memory-mapped ndarrays.

NumPy focuses on the CPython reference execution of Python, which is a non-streamlining bytecode translator. Numerical calculations composed for this rendition of Python regularly run much more slow than assembled counterparts. NumPy addresses the gradualness issue incompletely by giving multidimensional clusters and capacities and administrators that work productively on exhibits, requiring changing some code, generally internal circles utilizing NumPy.

Utilizing NumPy in Python gives usefulness practically identical to MATLAB since they are both deciphered, and the two of them permit the client to compose quick projects as long as most activities take a shot at exhibits or frameworks rather than scalars. In examination, MATLAB brags a huge number extra tool compartments, remarkably Simulink, while NumPy is inherently coordinated with Python, an increasingly present day and complete programming language. Additionally, reciprocal Python bundles are accessible; SciPy is a library that includes more MATLAB-like usefulness and Matplotlib is a plotting bundle that gives MATLAB-like plotting usefulness. Inside, both MATLAB and NumPy Inside, depend on BLAS and LAPACK for effective direct variable based math calculations.

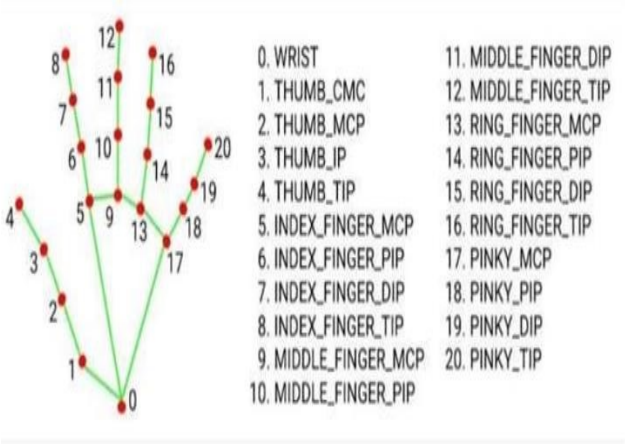
Python ties of the broadly utilized PC vision library OpenCV use NumPy exhibits to store and work on information. Since pictures with various channels are essentially spoken to as three-dimensional clusters, ordering, cutting or concealing with different exhibits are exceptionally proficient approaches to get to explicit pixels of a picture. The NumPy cluster as widespread information structure in OpenCV for pictures, extricated include focuses, channel pieces and a lot more limitlessly streamlines the programming work process and troubleshooting.

The center usefulness of NumPy is its "ndarray", for n-dimensional exhibit, information structure. Rather than Python's worked in list information structure, these clusters are homogeneously composed: all components of a solitary exhibit must be of a similar kind.

Such clusters can likewise be sees into memory cradles distributed by C/C++, Cython, and

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Fortran expansions to the CPython mediator without the need to duplicate information around, giving a level of similarity with existing numerical libraries. This usefulness is misused by the SciPy bundle, which wraps various such libraries . NumPy has worked in help for memory-mapped ndarrays.




**Mediapipe**

MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano.

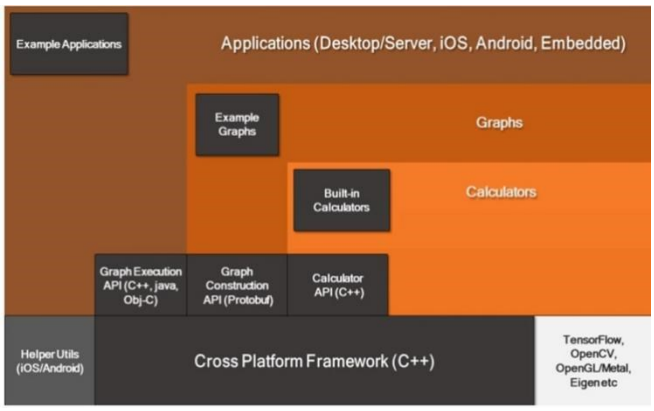
MediaPipe Toolkit comprises the Framework and the Solutions. The following diagram shows the components of the MediaPipe Toolkit.

Framework: The Framework is written in C++, Java, and Obj-C, which consists of the following APIs.

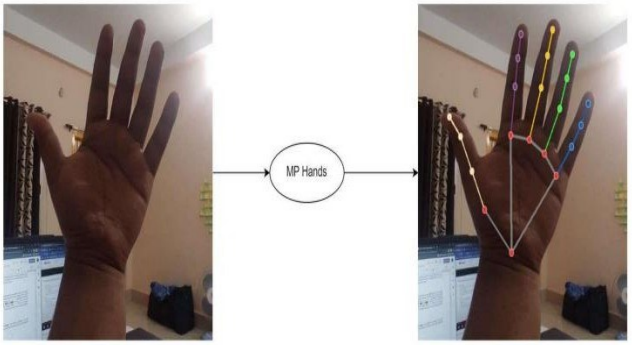
1.Calculator API (C++).

2.Graph construction API (Protobuf).

3.Graph Execution API (C++, Java, Obj-C)

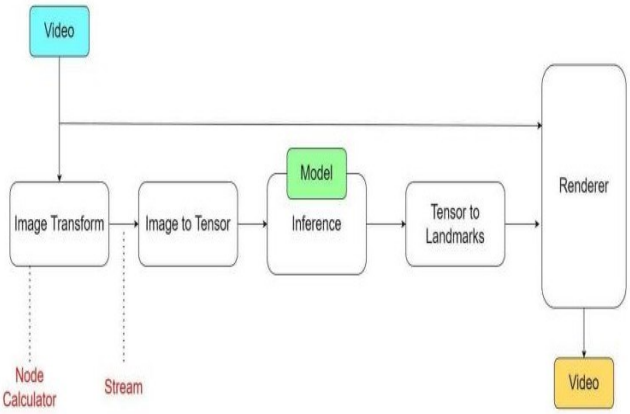



**Graphs:** The MediaPipe perception pipeline is called a Graph. Let us take the example of the first solution, Hands. We feed a stream of images as input which comes out with hand landmarks rendered on the images.




•The flow chart below represents the MP (Abbr. MediaPipe) hand solution graph.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86




MediaPipe hands solution graph

In computer science jargon, a graph consists of Nodes connected by Edges. Inside the MediaPipe Graph, the nodes are called Calculators, and edges are called Streams. Every stream carries a sequence of Packets that have ascending time stamps.

In the image above, we have represented Calculators with rectangular blocks and Streams using arrows.

MediaPipe Solutions : Solutions are open-source pre-built examples based on a specific pre-trained TensorFlow or TFLite model.

•The solutions are available in C++, Python, JavaScript, Android, iOS, and Coral. As of now, majority of the solutions are available only in C++ (except KNIFT and IMT) followed by Android, with Python not too far behind.

•The other wrapper languages, too, are growing fast with a very active development state. As you can see, even though MediaPipe Framework is cross-platform, that does not imply the same for the solutions. MediaPipe is currently at alpha version 0.7. We can expect the solutions to get more support with the beta releases. Following are some of the solutions provided by MediaPipe.

•Synchronization and Performance Optimization

MediaPipe supports multimodal graphs. To speed up the processing, different calculators run in separate threads. For performance optimization, many built-in calculators come with options for GPU acceleration. Working with time series data must be in proper synchronization; otherwise, the system will break. The graph ensures this so that flow is handled correctly according to the timestamps of packets. The Framework handles synchronization, context sharing, and inter-operations with CPU calculators

## V. RESULTS AND DISCUSSION

### CONTROLLING VOLUME WITH LEFT HAND(MIN):

When the thumb and index fingers distance is minimum the volume is minimum

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

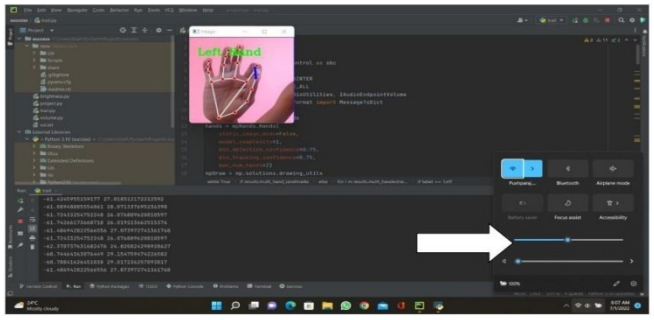


**Fig-1 Right Hand**

**CONTROLLING VOLUME WITH LEFT HAND(MAX):**

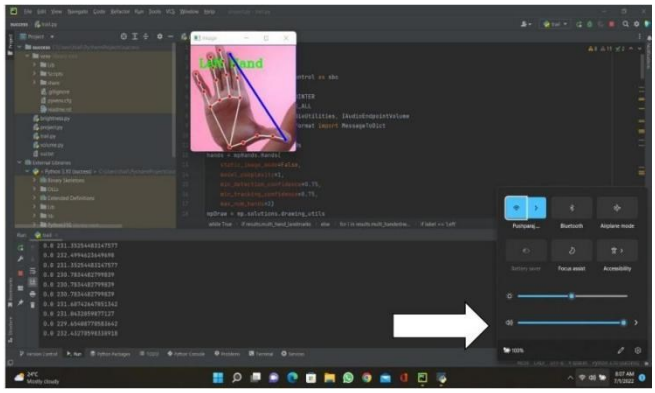


**Fig-.2 Left Hand**

**CONTROLLING VOLUME WITH LEFT HAND:**

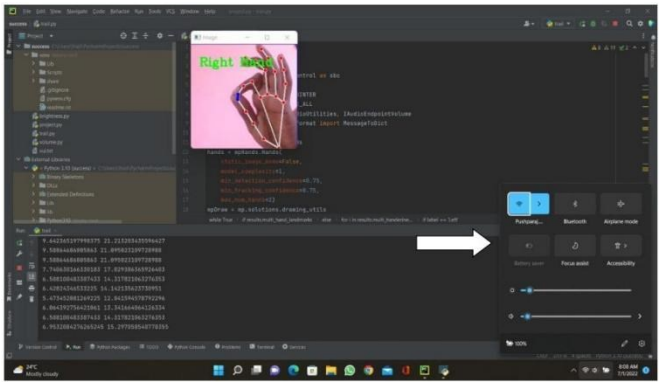


**Fig-.3 Data Collection Page**

## VI. CONCLUSION

The project presented a program that allowed user to perform hand gestures for easy software control. A vision-based hand Gesture system that does not require any special markers or gloves and can operate in real-time on a commodity PC with low-cost cameras. Specifically, the system can track the tip positions of the counters and index finger for each hand. The motivation for this hand Gesture was a desktop- based

Hand gesture analysis can be divided into two main approaches, namely, glove- based analysis, vision-based analysis . The glove-based approach employs sensors (mechanical or optical) attached to a glove that acts as transducer of finger flexion into electrical signals to determine hand posture.

## FUTURE ENHANCEMENTS:

- In future we can use this feature TVs and Mobile phones also, Currently its for PCs and Laptops.
- It is also going to help blind people to control volume by hand gestures.

## REFERENCES

1.M.-C. Roh, S.-J. Huh and S.-W. Lee, "A virtual mouse interface based on a two-layered bayesian network", Proc. WACV, 2009.

2.T. B. Moeslund, M. Störring and E. Granum, "A natural interface to a virtual environment through computer vision-estimated pointing gestures", Gesture and Sign Language in Human-Computer Interaction, pp. 59-63, 2002.

3.R. ALAMI, "A methodological approach relating the classification of gesture to identification of human

intent in the context of human-robot interaction", pp. 371-377, 2005.

4.M. KRUEGER, Artificial reality II, Ma:Addison-Wesley Reading, 1991.

5.H.A JALAB, Static hand Gesture recognition for human computer interaction, 1-72012.

6."Hand tracking and gesture recognition for human-computer interaction", 2005.

7.R. Merletti and D Farina, Surface Electromyography: Physiology Engineering and Applications, New York, NY, USA:John Wiley & Sons, 2016.

8.Z. Zhang, Y. Wu, Y. Shan and S. Shafer, "Visual panel: Virtual mouse keyboard and 3d controller with ordinary piece of paper", Proceedings of Perceptual User Interfaces, 2001.

9.W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition", International workshop on automatic face and gesture recognition, 1995.

10.G. R. S. Murthy and R. S. Jadon, "A Review of Vision Based Hand Gestures Recognition", International Journal of Information Technology and Knowledge Management, vol. 2, no. 2, 2009.

11.Mokhtar M. Hasan and Pramoud K. Misra, "Brightness Factor Matching For Gesture Recognition System Using Scaled Normalization", 2011.

12.T. Hesselmann, S. Flöring and M. Schmitt, "Stacked half-pie menus: navigating nested menus on interactive tabletops", Proc. ITS, 2009.

13.M. Kranz, S. Freund, P. Holleis, A. Schmidt and H. Arndt, "Developing gestural input", Proc. IWSAWC, 2006.

14.C. Kray, D. Nesbitt, J. Dawson and M. Rohs, "User-defined gestures for connecting mobile phones public displays and tabletops", Proc. MobileHCI, 2010.

15.R. Liang and M. Ouhyoung, "A Real-Time Continuous Gesture Recognition System for Sign Language", Proc. FG, 1998.

16.T. Masui, K. Tsukada and I. Siio, "MouseField: A Simple and Versatile Input Device for Ubiquitous Computing", Proc. UbiComp, 2006.