**CHAPTER-1**

**INTRODUCTION OF PROJECT**

**1.1 INTRODUCTION**

In this work, we tried to control mouse cursor movement and click events using a webcam. Processing techniques involve an image subtraction algorithm to detect hand. Once the hand are detected the system performs control actions .no additional hardware i s required by the system other than standard webcam which is provided in every laptop computer.

As the technologies are developing day by day the devices becoming compact in size. Some devices have gone wireless, some of them gone latent. This paper proposes a system that could make some the devices go latent in the future that is the future of HCI (Human-Computer Interaction). The proposal is to development of a Virtual Mouse using Gesture Recognition. The aim is to control mouse cursor functions using only a simple camera instead of a traditional or regular mouse device. The Virtual Mouse works as a medium of the user and the machine only using a camera. It helps the user to interact with a machine without any mechanical or physical devices and control mouse functions. In this gesture recognition system, it is very possible to capture & track the fingertip of hand with a webcam or built-in cam which is bearing a color cap or color sticky note paper and the system track the color and movement of the hand & move cursor with it.

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

As the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

As the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

As the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

s the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wit

As the technologies are developing day by day the devices

becoming compact in size. Some devices have gone wireless,

some of them gone latent. This paper proposes a system that

could make some the devices go latent in the future that is the

future of HCI (Human-Computer Interaction). The proposal is

to development of a Virtual Mouse using Gesture

Recognition. The aim is to control mouse cursor functions

using only a simple camera instead of a traditional or regular

mouse device. The Virtual Mouse works as a medium of the

user and the machine only using a camera. It helps the user to

interact with a machine without any mechanical or physical

devices and control mouse functions. In this gesture

recognition system, it is very possible to capture & track the

fingertip of hand with a webcam or built-in cam which is

bearing a color cap or color sticky note paper and the system

track the color and movement of the hand & move cursor wi

**1.2 HISTORY**

The first pointing device gesture, the "[drag](https://en.wikipedia.org/wiki/Click_and_drag)", was introduced by [Apple](https://en.wikipedia.org/wiki/Apple_Inc.) to replace a dedicated "move" button on mice shipped with its Macintosh and Lisa computers. Dragging involves holding down a pointing device button while moving the pointing device; the software interprets this as an action distinct from separate clicking and moving behaviour. Unlike most pointing device gestures, it does not involve the tracing of any particular shape. Although the "drag" behaviour has been adopted in a huge variety of software packages, few other gestures have been as successful.

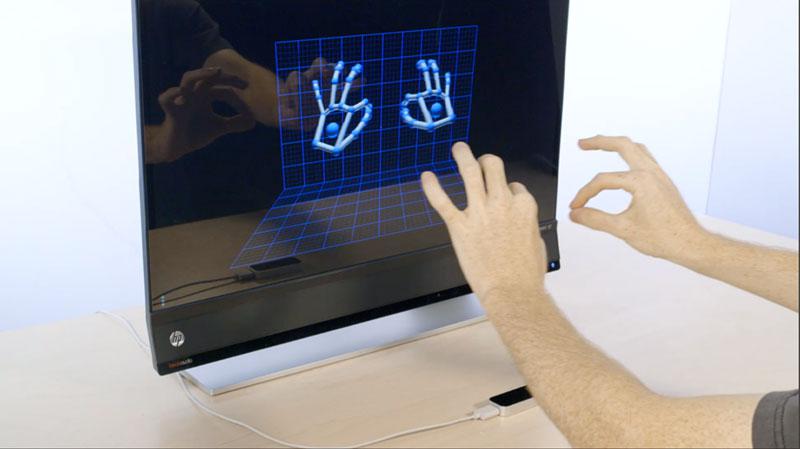


Figure 1.1

**1.3 OBJECTIVE OF THE SYSTEM**

A virtual mouse using hand gesture recognition is a system that **allows users to give mouse inputs to computer without using an actual mouse hardware.** To the extreme, it can also be called as hardware because it uses a camera for tracking hands. A virtual mouse can usually be operated with multiple input devices,

**1.4 JUSTIFICATION AND NEED FOR THE SYSTEM**

**Gesture recognition** is a topic in [computer science](https://en.wikipedia.org/wiki/Computer_science) and [language technology](https://en.wikipedia.org/wiki/Language_technology) with the goal of interpreting human [gestures](https://en.wikipedia.org/wiki/Gesture) via mathematical [algorithms](https://en.wikipedia.org/wiki/Algorithm).[[1]](https://en.wikipedia.org/wiki/Gesture_recognition#cite_note-Kobylarz-1) It is a subdiscipline of [computer vision](https://en.wikipedia.org/wiki/Computer_vision). Gestures can originate from any bodily motion or state but commonly originate from the [face](https://en.wikipedia.org/wiki/Face) or [hand](https://en.wikipedia.org/wiki/Hand). Current[*[when?](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Dates_and_numbers" \l "Chronological_items" \o "Wikipedia:Manual of Style/Dates and numbers)*] focuses in the field include [emotion recognition](https://en.wikipedia.org/wiki/Emotion_recognition) from face and hand gesture recognition. Users can use simple gestures to control or interact with devices without physically touching them. Many approaches have been made using cameras and [computer vision](https://en.wikipedia.org/wiki/Computer_vision) algorithms to interpret [sign language](https://en.wikipedia.org/wiki/Sign_language). However, the identification and recognition of posture, gait, [proxemics](https://en.wikipedia.org/wiki/Proxemics), and human behaviors is also the subject of gesture recognition techniques.[[2]](https://en.wikipedia.org/wiki/Gesture_recognition#cite_note-2) Gesture recognition can be seen as a way for computers to begin to [understand human body language](https://en.wikipedia.org/wiki/Computer_processing_of_body_language), thus building a richer bridge between machines and humans than primitive [text user interfaces](https://en.wikipedia.org/wiki/Text_user_interface) or even [GUIs](https://en.wikipedia.org/wiki/GUI) (graphical user interfaces), which still limit the majority of input to keyboard and mouse and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at this point will move accordingly. This could make conventional input on devices such and even redundant.

**1.5 ADVANTAGES OF THE SYSTEM**

[Computing](https://en.wikipedia.org/wiki/Computing), a **pointing device gesture** or **mouse gesture** (or, simply, **gesture**) is a way of combining [pointing device](https://en.wikipedia.org/wiki/Pointing_device) or finger movements and [clicks](https://en.wikipedia.org/wiki/Point-and-click) that the [software](https://en.wikipedia.org/wiki/Software) recognizes as a specific [computer event](https://en.wikipedia.org/wiki/Event_(computing)) and responds in a manner particular to that software. They can be useful for people who have difficulties typing on a [keyboard](https://en.wikipedia.org/wiki/Computer_keyboard). For example, in a [web browser](https://en.wikipedia.org/wiki/Web_browser), a user can navigate to the previously viewed page by pressing the right pointing device button, moving the pointing device briefly to the left, then releasing the button.

The system consists of the generic mouse and trackpad system of monitor controlling and the non availability of a hand gesture system. The remote accessing of monitor screen using the hand gesture is unavailable. Even-though it is largely trying to implement the scope is simply restricted in the field of virtual mouse. The existing virtual mouse control system consists of the simple mouse operations using the hand recognition system, where we could perform the basic mouse operation like mouse pointer control, left click, right click, drag etc. The further use of the hand recognition is not been made use of. Even-though there are a number of systems which are used for hand recognition, the system they made used is the static hand recognition which is simply recognition of the shape made by hand and by defining an action for each shape made, which is limited to a number of defined actions and a large amount of confusion.

**1.6 PREVIOUS WORK OR RELATED SYSTEM**

How they are used. Before we begin a new system it is important to study the system that will be improved or replaced (if there is one). We need to analyze how this system uses hardware, software, network and the people resources to convert into fast and useful device

**Gesture recognition** is a topic in [computer science](https://en.wikipedia.org/wiki/Computer_science) and [language technology](https://en.wikipedia.org/wiki/Language_technology) with the goal of interpreting human [gestures](https://en.wikipedia.org/wiki/Gesture) via mathematical [algorithms](https://en.wikipedia.org/wiki/Algorithm). It is a sub discipline of [computer vision](https://en.wikipedia.org/wiki/Computer_vision). Gestures can originate from any bodily motion or state but commonly originate from the [face](https://en.wikipedia.org/wiki/Face) or [hand](https://en.wikipedia.org/wiki/Hand). Currentfocuses in the field include [emotion recognition](https://en.wikipedia.org/wiki/Emotion_recognition) from face and hand gesture recognition. Users can use simple gestures to control or interact with devices without physically touching them. Many approaches have been made using cameras and [computer vision](https://en.wikipedia.org/wiki/Computer_vision) algorithms to interpret [sign language](https://en.wikipedia.org/wiki/Sign_language). However, the identification and recognition of posture, gait, [proxemics](https://en.wikipedia.org/wiki/Proxemics), and human behaviors is also the subject of gesture recognition techniques. Gesture recognition can be seen as a way for computers to begin to [understand human body language](https://en.wikipedia.org/wiki/Computer_processing_of_body_language), thus building a richer bridge between machines and humans than primitive [text user interfaces](https://en.wikipedia.org/wiki/Text_user_interface) or even [GUIs](https://en.wikipedia.org/wiki/GUI) (graphical user interfaces), which still limit the majority of input to keyboard and mouse and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at this point will move accordingly. This could make conventional input on devices such and even redundant.

**1.7 FUNCTIONS OF VIRTUAL MOUSE**

* Quick Working
* Able to work as mouse
* Simple to use
* Quick Start
* Uses Calibration
* Uses HSV
* Manage color
* Recognize the color
* Automatically find Mid point
* Navigate and Arrange
* Assign and Custom settings
* Able to Maximize and Minimize windows
* Exercise aerobic anaerobic charts
* Recognize the Color.
* WebCam

**1.8 SOFTWARE REQUIREMENTS**

**FOR WINDOWS-**

PYTHON is one of the programming languages designed for the Common Language Infrastructure. It was developed by Microsoft and download Python as your windows requirement (32 bit, 64 bit).

There are some required libraries –

* OpenCV (Open Source Computer Vision Library)
* Numpy
* WxPython
* PynPut
* pyautogui

**FOR LINUX –**

PYTHON is one of the programming languages designed for the Common Language Infrastructure. It was developed by Microsoft and download Python as your Linux requirement.

There are some required libraries –

* OpenCV (Open Source Computer Vision Library)
* Numpy
* WxPython
* PynPut
* pyautogui

**FOR MAC-**

PYTHON is one of the programming languages designed for the Common Language Infrastructure. It was developed by Microsoft and download Python as your Mac requirement .

There are some required libraries –

* OpenCV (Open Source Computer Vision Library)
* Numpy
* WxPython
* PynPut
* pyautogui

**1.9 HARDWARE REQUIREMENTS**

* Intel Pentium D processor 1.8 GHz or AMD Athlon X2 processor X2 processor 1.8 GHz or higher
* 3 GB RAM
* 5 GB HDD space
* Peripheral webcam at least 30 frames/second, 640x480 resolution.

**CHAPTER-2**

**FEASIBILITY STUDY**

Nowadays actions are increasingly being handled in electronic ways, instead of physical interaction. From earlier times biometrics is used in the authentication of a person. It recognizes a person by using a human trait associated with it like eyes (by calculating the distance between the eyes) and using hand gestures, fingerprint detection, face detection etc. Advantages of using these traits for identification are that they uniquely identify a person and cannot be forgotten or lost. These are unique features of a human being which are being used widely to make the human life simpler. Hand gesture recognition system is a powerful tool that supports efficient interaction between the user and the computer. The main moto of hand gesture recognition research is to create a system which can recognise specific hand gestures and use them to convey useful information for device control. This paper presents an experimental study over the feasibility of principal component analysis in hand gesture recognition system. PCA is a powerful tool for analyzing data. The primary goal of PCA is dimensionality reduction. Frames are extracted from the Sheffield KInect Gesture (SKIG) dataset. The implementation is done by creating a training set and then training the recognizer. It uses Eigen space by processing the eigenvalues and eigenvectors of the images in training set. Euclidean distance with the threshold value is used as similarity metric to recognize the gestures. The experimental results show that PCA is feasible to be used for hand gesture recognition system.

Gestures originating from face and hand are most commonly used in the applications that work in Human Gesture Recognition technology. These techniques have already established hand as complex and important structure to be used by recognition technology. But now a days hand gestures also have gained interests of different device manufacturers apart from their application in Telerobotics, sign language recognition and Human Computer Interaction (HCI). Face recognition systems although very popular has some disadvantages.

1. Problem with false rejection during recognition when people change their hair style, grow or shave a beard or wear glasses.
2. (ii) It can’t tell the difference between identical twins. There are two types of gestures namely static and dynamic. Static gesture means user with certain fix pose or gesture.

Dynamic means gesture with some strokes they can be either pre stroke, stroke or post stroke phases. For static gestures, images are processed directly after converting them to vector form but for videos or dynamic gestures the frames are first extracted from videos then processed after converting them to vector form .

**2.1 Technical Feasibility**

As the technologies are developing day by day the devices becoming compact in size. Some devices have gone wireless, some of them gone latent. This paper proposes a system that could make some the devices go latent in the future that is the future of HCI (Human-Computer Interaction). The proposal is to development of a Virtual Mouse using Gesture Recognition. The aim is to control mouse cursor functions using only a simple camera instead of a traditional or regular mouse device. The Virtual Mouse works as a medium of the user and the machine only using a camera. It helps the user to interact with a machine without any mechanical or physical devices and control mouse functions. In this gesture recognition system, it is very possible to capture & track the fingertip of hand with a webcam or built-in cam which is bearing a color cap or color sticky note paper and the system track the color and movement of the hand & move cursor with it.

**2.2 Operational Feasibility**

Once the changeover takes place from the existing manual system to the computerized systems, depending upon the complexity of the installed computerized system, high quality manpower should be employed for the smooth functioning of the system. So the platform and tools should be selected in such way that, once the system is up and running, getting the right manpower for that system to function successfully, should not be a cause to worry.

This project is operationally feasible as it has a very user friendly environment which is very easy to use by the users. Also, the operations of every type of user can be easily studied by the design of the site. So a new web developer can easily understand the project in a short time.

**2.3 Behavioural Feasibility**

This aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to get used to the system which must be accepted as a necessity. The level of acceptance by the user solely depend on the developed system and has a modest technical requirement as only minimal or null changes are required for implementing the system.

**2.4 LEGAL FEASIBILTIY**

In Legal Feasibility study project is analyzed in legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall it can be said that Legal Feasibility Study is study to know if proposed project conform legal and ethical requirements.

**2.5 Economical Feasibility**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. Ifbenefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

**2.5.1 Cost based study**

This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system.

**2.5.2 Time based study**

This is an analysis of the time required to achieve a return on investments. The benefits derived from the system. The future value of a project is also a factor.

This project is economically feasible also as development costs and operating costs are not too high. Also it is very much used by the users so many a sponsors will be there for the site. So, benefits from the application will definitely be more than the cost involved in making or maintaining the application.

**2.6 Schedule Feasibility**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. It should be determined whether the deadlines are mandatory or desirable.

Schedule Feasibility is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. If a project has a high probability to be completed on-time, then its schedule feasibility is appraised as high.

**CHAPTER-3**

**SOFTWARE REQUIREMENT AND SYSTEMSPECIFICATION**

**3.1 INTRODUCTION**

A **Software Requirements Specification** (**SRS**) is a complete description of the behavior of the system to be developed. It fully describes what the software will do and how it will be expected to perform. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated.

**3.1.1 Developer responsibilities overview**

The roles and responsibilities that people can assume in the project are based on merit. Everybody can help no matter what their role. Those who have been long term or valuable contributors to the project obtain the right to vote and commit directly to the source repository.

**USER**

Users are the people who use the products of the Project. People in this role aren't contributing code, but they are using the products, reporting bugs, making feature requests, and such. This is by far the most important category of people as, without users, there is no reason for the Project. When a user starts to contribute code or documentation patches, they become a developer.

**DEVELOPERS**

Developers are the people who write code or documentation patches or contribute positively to the project in other ways. A developer's contribution is always recognized. In source code, all developers who contribute to a source file may add their name to the list of authors for that file.

**COMMITTERS**

Developers who give frequent and valuable contributions to a subproject of the Project can have their status promoted to that of a "Committer" for that subproject. A Committer has write access to the source code repository and gains voting rights allowing them to affect the future of the subproject.

**3.1.2 Information description**

This section of the SRS provides a detail of the problem that software must solve. It should describe the core of the application i.e. THE DATA. The information the software is going to work is the most basic part of the software. The description of each data or information entity is described here. It also gives details of the relationship between the data elements of the software. The information description helps the software designers in their designing purpose.

**3.1.3 Functional requirement**

This section of the SRS describes the each function to solve the problem. It emphasizes on the core of the software on which the data will be processed-i.e. design constraints, and performance characteristics. The DFD or any other graphical diagram can also be added to describe the functionality of the system.

**3.1.4 Logical design**

Logical design of an information system shows the major features and also how they are related to one another. The first step of the system design is to design logical design elements. This is the most creative and challenging phase and important too. Design of proposed system produces the details of the state how the system will meet the requirements identified during the system analysis that is, in the design phase we have to find how to solve the difficulties faced by the existing system. The logical design of the proposed system should include the details that contain how the solutions can be implemented. It also specifies how the database is to be built for storing and retrieving data, what kind of reports are to be created and what are the inputs to be given to the system. The logical design includes input design, output design, and database design and physical design.

**3.1.5 Input design**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The system needs the data regarding the asset items, depreciation rates, asset transfer, and physical verification for various validation, checking, calculation and report generation. The error raising method is also included in the software, which helps to raise error message while wrong entry of input is done.

**3.1.6 Output design**

Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the system's relationship with the user and help in decision making. Since the reports are directing referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy. The options for the output reports are given in the appendix.

**3.1.7 Physical design**

The process of developing the program software is referred to as physical design. We have to design the process by identifying reports and the other outputs the system will produce. Coding the program for each module with its logic is performed in this step. Proper software specification is also done in this step.

**3.1.8 Behavioral description**

This section of the SRS describes the behavior of the software will exhibit. It is based on the definitions of the events and the operations that it will perform because of events.

**3.1.9 Validation criteria**

This section of the SRS contains the details of the tests that should be performed to validate functions, performance, and behavior of the software. It is one of the most important aspect of the software which decides how much robust our software is.

**3.2 TECHNOLOGY SPECIFIC REQUIREMENTS**

**3.2.1 Hardware requirements**

1. Window, Linux, Mac are important to perform “VIRTUAL MOUSE”.
2. A smart phone with android to run the program on in the room with the following specifications

* Processor : snapdragon 865
* RAM : 2GB
* Storage : 4GB
* Display : For all Laptops

**3.2.2 Software requirement**

* Platform : By User
* The Operating System : Windows
* Framework : Window, Python code

**CHAPTER-4**

**TECHNOLOGICAL ENVIRONMENT**

**4.1 SOFTWARE OVERVIEW**

**4.1.1 Operating system:-**

This project work is done on the windows 10 operating system. As operating system is a set of software tools designed to make it easy for people or programmers to make optimum use of the computer. Windows 10 is an operating system released by Microsoft. Windows 10 introduced a lot of changes, and many people found it difficult to use. Windows 10 is very similar to windows 8.1, but it addresses some of the problems people had with windows 10. The new updating features of windows 10 are

* New Microsoft edge
* More lock screen and start screen options
* Better search
* 3D printing support
* Refreshed Xbox music
* Dual screen with security

**4.1.2 Language and tools used:-**

* **PYTHON—**

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) as well as its [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.[[30]](https://en.wikipedia.org/wiki/Python_%28programming_language%29#cite_note-AutoNT-7-30)

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), object-oriented and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

[Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) began working on Python in the late 1980s, as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)), and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension) and a garbage collection system using [reference counting](https://en.wikipedia.org/wiki/Reference_counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility) and much Python 2 code does not run unmodified on Python 3. Python 2 was discontinued with version 2.7.18 in 2020.

Currently Python release the 3.9.5 version in 3 series 2021.Python consistently ranks as

one of the most popular programming languages in world.

**4.2 CHARACTERISTICS OF PYTHON**

Important characteristics make Python practical nature possible:

* Easy to code
* Free and Open Source
* Object-Oriented.
* GUI Programming Support.
* High-Level Language
* Extensible feature
* Python is Portable Language
* Python is Integrated Language
* Interpreted Language
* Large Standard Libraries
* Dynamically Typed Language
* Platform Independent
* Python supports POP and OOP
* Expressive Languages

**4.3 ARCHITECTURE**

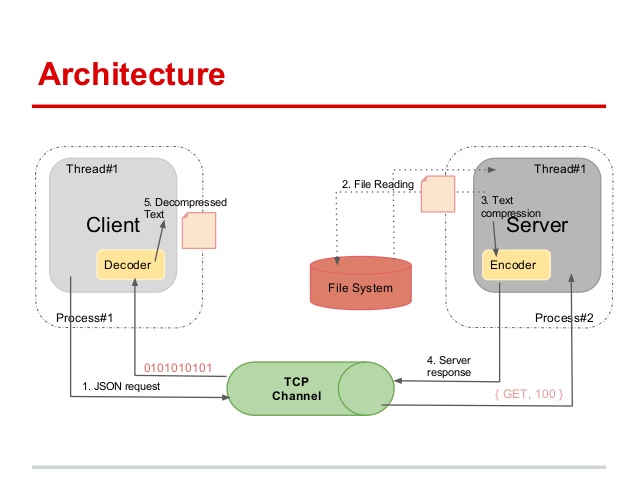


Fig 4.1:- Architecture of Python

**4.3.1 Features of Python:**

* Easy Language. Python is an easy language. It is easy to read, write, learn and understand. Python has a smooth learning curve.
* Readable. The Python language is designed to make developers life easy. Reading a Python code is like reading an English sentence.
* Interpreted Language. Python is an interpreted language. It comes with the IDLE (Interactive Development Environment). This is an interpreter and follows the REPL structure (Read-Evaluate-Print-Loop).
* Dynamically-Typed Language. Python is not statically-typed like Java. You dont need to declare data type while defining a variable.
* Object-Oriented. Python is object-oriented but supports both functional and object-oriented programming. Everything in Python is an object.
* Popular and Large Community Support. Python has one of the largest communities on StackOverflow and Meetup. If you need help, the community will answer your questions.
* Open-Source. Python is open-source and the community is always contributing to it to improve it. It is free and its source code is freely available to the public.
* Large Standard Library. The standard library is large and has many packages and modules with common and important functionality.
* Platform-Independent. Python is platform-independent. If you write a program, it will run on different platforms like Windows, Mac and Linux.
* Extensible and Embeddable. Python is extensible. You can use code from other languages like C++ in your Python code. It is also embeddable.
* GUI Support. You can use Python to create GUI (Graphical User Interfaces). You can use tkinter, PyQt, wxPython or Pyside for this.
* High-level Language. Python is a high-level language and C++ is mid-level. Python is easy to understand and closer to the user.

**4.3.2 Working of PYTHON applications:**

Python has several built-in modules and functions for handling files. These functions are spread out over several modules such as os, os.path, shutil, and pathlib, to name a few. This article gathers in one place many of the functions you need to know in order to perform the most common operations on files in Python.

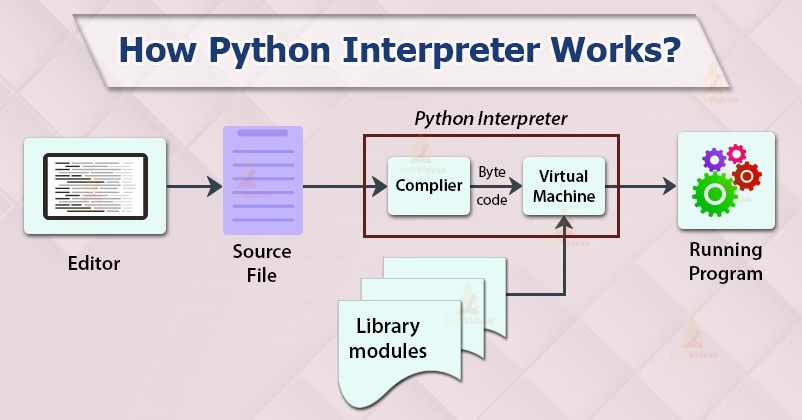


Figure:-4.2

**4.4 VARIABLE AND DATA STRUCTURE IN PYTHON**

Variable is just like a container for data. We can say we store data in variable. A Python variable is a**reserved memory location to store values**. In other words, a variable in a python program gives data to the computer for processing.

Every value in Python has a datatype. Different data types in Python are Numbers, List, Tuple, Strings, Dictionary, etc. Variables in Python can be declared by any name or even alphabets like a, aa, abc, etc.

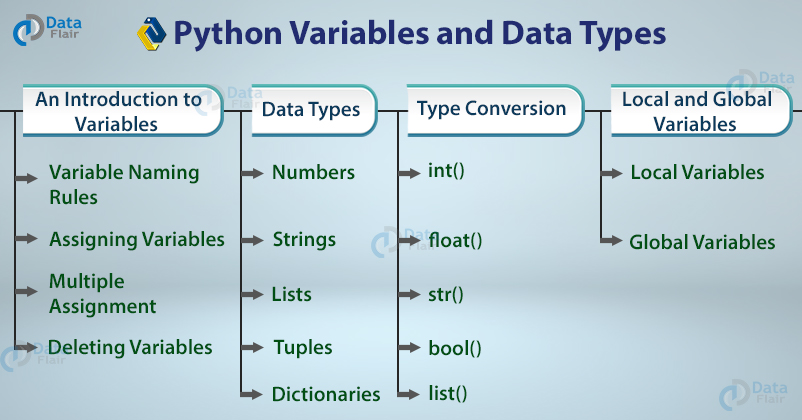
Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

## **Assigning Values to Variables**

Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.

The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

  
 **Figure 4.3**

In Python, variables are the containers for storing data values. They are reference, or pointers, to an object in memory which means that whenever a variable is assigned to an instance, it gets mapped to that instance. Unlike other languages like C/C++/JAVA, Python is not “statically typed”. We do not need to declare variables before using them or declare their type. A variable is created the moment we first assign a value to it

## **Scope of variable**

The location where we can find a variable and also access it if required is called the **scope of a variable**.

#### Global and local variables

Global variables are the ones that are defined and declared outside any function and are not specified to any function. They can be used by any part of the program.

**4.5 DATA TYPES IN PYTHON**

All programming languages’ aim is to create some data or to do some actions on the data and process the data. The data can be categorized into different types like integers, float numbers, boolean, string, etc. Categorizing the data is important in order to perform the operations on it. That classification, which states the type of data stored in the variable is called data type. Each and every programming language provides certain data types such that all the types of data can be stored in those .

Generally, data types can be classified into two types:

1. **Built-in data types:** Data types that are already available in python are called built-in data types
2. **User-defined data types:** Data types that are created by the programmer

##### **Built-in data types in Python:**

1. Numeric types (int, Float, Complex)
2. bool
3. None
4. Str
5. Bytes
6. Bytearray
7. Tuple
8. List
9. Range
10. Set
11. Dict

##### **Numeric types in Python:**

**int:** The int data type represents values or numbers without decimal values. In python, there is no limit for the int data type. It can store very large values conveniently.

##### **Float Data Type in Python:**

The float data type represents a number with decimal values. floating-point numbers can also be written in scientific notation. e and E represent exponentiation. where e and E represent the power of 10 such numbers are also treated as floating-point numbers.

##### **Complex Data Type in python:**

The complex data type represents the numbers that are written in the form of a+bj or a-bj, here a is representing a real part of the number and b is representing an imaginary part of the number. The suffix small j or upper J after b indicates the square root of -1. The part “a” and “b” may contain integers or floats.

**CHAPTER – 5**

**DESIGN AND COMPONENTS**

**5. DESIGN**

Design is a meaningful engineering representation of something that is to be built. It can be traced to a customer’s requirements and at the same time assessed for quality against asset of predefined criteria for the good design. Thus it is the blue print of the system. In the software engineering context design concentrates on three factors, data, architecture, interfaces and components.

The data and architectural design focuses on patterns as they apply to the application to be built. At the interface level human ergonomics often dictate our design approach. At the component level a programming approach leads to the effective data and procedural designs. Design begins with requirements model and is processed to transform it into four levels of design details; data structure, system architecture, interface representation and the component level. Finally a design specification is produced. This specification consists of design models that describe data, architecture, interfaces and components. At each stage, software design work products are renewed for clarity, correctness, completeness and consistency with the requirements and with one another.

**OVERVIEW:**

The “VIRTUAL MOUSE” is perform as hand gesture mouse, because as we know that we can perform works in virtual manner.Microsoft design “VIRTUAL REALITY” gadget this is also work in the gesture terms . According to it we should moves hand virtually and in any directions , this is help to move things any directions, lift the things in any directions ,as we make this “VIRTUAL GESTURE MOUSE” for making life creative and easy.

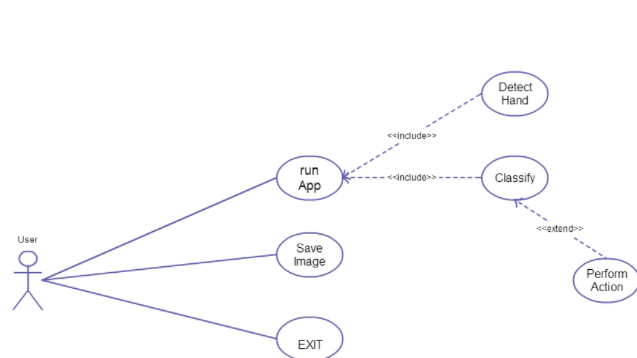


Fig:5.1:-Overview Of Project

**EXISTING FUCTION**

The existing system consists of the generic mouse and trackpad system of monitor controlling and the nonavailability of a hand gesture system. The remote accessing of monitor screen using the hand gesture is unavailable. Even-though it is largely trying to implement the scope is simply restricted in the field of virtual mouse. The existing virtual mouse control system consists of the simple mouse operations using the hand recognition system, where we could perform the basic mouse operation like mouse pointer control, left click, right click, drag etc. The further use of the hand recognition is not been made use of. Even-though there are a number of systems which are used for hand recognition, the system they made used is the static hand recognition which is simply recognition of the shape made by hand and by defining an action for each shape made, which is limited to a number of defined actions and a large amount of confusion.

**PURPOSED SYSTEM**

Using the current system even-though there are a number of quick access methods available for the hand and mouse gesture for the laptops, using our project we could make use of the laptop or web-cam and by recognizing the hand gesture we could control mouse and perform basic operations like mouse pointer controlling, select and deselect using left click, and a quick access feature for file transfer between the systems connected via network LAN cable. The project done is a “Zero Cost” hand recognition system for laptops, which uses simple algorithms to determine the hand, hand movements and by assigning an action for each movement[2]. But we have mainly concentrated on the mouse pointing and clicking actions along with an action for the file transfer between connected systems by hand action and the movements. The system we are implementing which is been written in python code be much more responsive and is easily implemented since python is a simple language and is platform independent with a flexibility and is portable which is desirable in creating a program which is focused in such an aim for creating a Virtual Mouse and Hand Recognition system. The system be much more extendable by defining actions for the hand movement for doing a specific action. It could be further modified to any further extent by implementing such actions for the set of hand gestures, the scope is restricted by your imagination

**USE OF PROPOSED WORK**

This Virtual Mouse Hand Recognition application uses a simple color cap on the finger without the additional requirement of the hardware for the controlling of the cursor using simple gestures and hand control. This is done using vision based hand gesture recognition with inputs from a webcam

**COMPONENTS**

The components used in this project can‟t be specific, since this project is a prototype for all computers. As such, certain prerequisites are as follows:

**Webcam**

Webcam is a necessary component for detecting the image. Sensitivity of mouse is directly proportional to resolution of camera. If the resolution of camera is good enough, an enhanced user experience is guaranteed. The webcam serves the purpose of taking real time images whenever the computer starts. On the basis of gestures and motion of fingers, system will decide the respective action



Fig: 5.2 –Web Cam

**MATLAB**

MATLAB version greater than 2012a is used for making this project. Image Acquisition toolbox and Image processing toolbox is necessary for developing the firmware. Along with this, an updated version of Java is required.

**SDK tool and .NET Framework**

SDK and .NET is required in order to create standalone applications for windows based system.

**SYSTEM DEVELOPMENT**

In the object tracking application one of the main problems is object detection. Instead of finger tips, a color pointer has been used to make the object detection easy and fast. A circle blue sticker is used as a color pointer in this study. To simulate the click events of the mouse, three fingers serving as three color pointers has been used. The basic algorithm is as follows:

• Set a pointer in the image

• Detect the pointer using the defined color information

• Define the region and the center of the pointer and draw a bounding box around it

• Track the motion of the pointer

• Move the cursor according to the position of the center of the pointer

• Simulate the single and the double left click and the right click of the mouse

**METHODOLOGY**

In the Methodology, the method used in each component of the system will be explained separately.

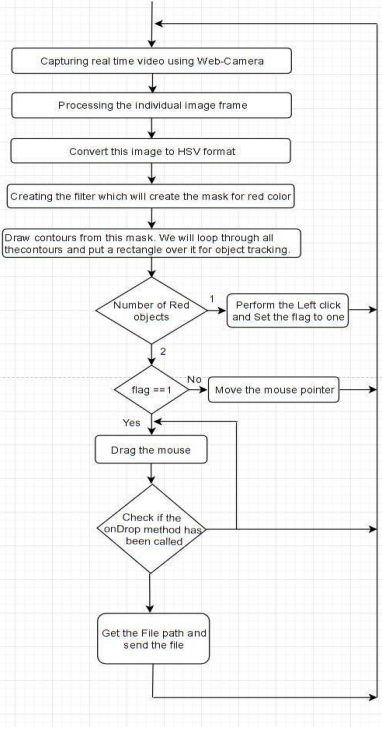
They are following subsections:

**Camera Settings**

The runtime operations are managed by the webcam of the connected laptop or desktop. To capture a video, we need to create a Video Capture object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera.Since we only use a single camera we pass it as ‘0’. We can add additional camera to the system and pass it as 1,2 and so on. After that, you can capture frame-by-frame. But at the end, don’t forget to release the capture. We could also apply color detection techniques to any image by doing simple modifications in the code.

**Capturing frames**

The infinite loop is used so that the web camera captures the frames in every instance and is open during the entire course of the program. We capture the live feed stream, frame by frame. Then we process each captured frame which is in RGB(default) color space to HSV color space. There are more than 150 color-space conversion methods available in OpenCV. But we will look into only two which are most widely used ones, BGR to Gray and BGR to HSV.



**Masking technique**

The mask is basically creating some specific region of the image following certain rules. Here we are creating a mask that comprises of an object in red color. After that we perform a bitwise AND operation on the Input image and theThreshold image, which result in only the red colored objects are highlighted. This result of the AND operation is stored in res. We then display the frame, res and mask on 3 separate windows using imshow() function

**Display the frame**

The imShow() is a function of HighGui and it is required to call the waitKey regulerly. The processing of the event loop of the imshow() function is done by calling waitKey. The function waitKey() waits for key event for a “delay” (here, 5 milliseconds). Windows events like redraw, resizing, input event etc. are processed by HighGui. So we call the waitKey function, even with a 1ms delay.

**Mouse Movement**

We have to first calculate the center of both detected red object which we can easily do by taking the average of the bounding boxes maximum and minimum points. now we got 2 co-ordinate from the center of the 2 objects we will find the average of that and we will get the red point shown in the image. We are converting the detected coordinate from camera resolution to the actual screen resolution. After that we set the location as the mouse\_position. but to move the mouse pointer it will take time. So we have to wait till the mouse pointer reaches that point. So we started a loop and we are not doing anything there we are just waiting will the current mouse location is same as assigned mouse location. That is for the open gesture.

**Clicking**

The next step is to implement the close gesture. The operation is performed by clicking the object and dragging it. It is similar to the open gesture, but the difference is we only have one object here so we only need to calculate the center of it. And that will be placed on the location where we will position our mouse pointer. Instead of mouse release operation we will be performing a mouse press operation.

**Drag**

In order to implement the dragging we introduce a variable ‘pinch flag’. It will be set to 1 if it was clicked earlier. So after clicking whenever we find the open gesture we check if the pinch flag is set to 1. If it is set to one then Drag operation is performed otherwise the mouse move operation is performed. DnD Frame First we create the My File Drop Target class. Inside that we have one overridden method, On Drop Files. This method accepts the x/y position of the mouse along with the file that are dropped.

**CHAPTER – 6**

**LIBRARIES**

**6. LIBRARIES**

We know that a module is a file with some Python code, and a package is a directory for sub packages and modules. But the line between a package and a Python library is quite blurred.

A Python library is a reusable chunk of code that you may want to include in your programs/ projects.

Compared to languages like C++ or C, a Python libraries do not pertain to any specific context in Python. Here, a ‘library’ loosely describes a collection of core modules.

Essentially, then, a library is a collection of modules. A package is a library that can be installed using a package manager like rubygems or npm.

A Python library defines lines of code that can be reused in other programs. It is basically a collection of modules. Their usefulness lies in the fact that new codes are not required to be written every time the same process is required to run. **Libraries in Python** play an important role in areas of data science, machine learning, data manipulation applications, etc.

### **Python Standard Library**

The Python Standard Library is a collection of exact syntax, token, and semantics of Python. It comes bundled with core Python distribution. We mentioned this when we began with an introduction.

It is written in C, and handles functionality like I/O and other core modules. All this functionality together makes Python the language it is.

More than 200 core modules sit at the heart of the standard library. This library ships with Python.

But in addition to this library, you can also access a growing collection of several thousand components from the Python Package Index (PyPI). We mentioned it in the previous blog.

### **Important Python Libraries**

Next, we will see twenty Python libraries list that will take you places in your journey with Python.

These are also the Python libraries for Data Science.

* [Matplotlib](https://www.upgrad.com/blog/libraries-in-python-explained/#1_Matplotlib)
* [Scipy (Scientific Python)](https://www.upgrad.com/blog/libraries-in-python-explained/#4_Scipy_Scientific_Python)
* [.SQLAlchemy](https://www.upgrad.com/blog/libraries-in-python-explained/#5_SQLAlchemy)
* [Scrapy](https://www.upgrad.com/blog/libraries-in-python-explained/#6_Scrapy)
* [PyTorch](https://www.upgrad.com/blog/libraries-in-python-explained/#13_PyTorch)
* [Theano](https://www.upgrad.com/blog/libraries-in-python-explained/#14_Theano)
* [SymPy](https://www.upgrad.com/blog/libraries-in-python-explained/#15_SymPy)
* [Caffe2](https://www.upgrad.com/blog/libraries-in-python-explained/#16_Caffe2)
* [NuPIC](https://www.upgrad.com/blog/libraries-in-python-explained/#17_NuPIC)
* [Pipenv](https://www.upgrad.com/blog/libraries-in-python-explained/#18_Pipenv)
* [PyBrain](https://www.upgrad.com/blog/libraries-in-python-explained/#19_PyBrain)
* [MILK](https://www.upgrad.com/blog/libraries-in-python-explained/#20_MILK)
* OpenCV
* NumPY
* PynPut
* WxPython
* PyAutoGUI

Now we discuss some libraries which are using in “VIRTUAL GESTURE MOUSE”.

**1.NUMPY**

Deviating towards the scientific computation areas, NumPy is the most used open-source packages offered by python. It supports large matrices and multidimensional data and has inbuilt mathematical functions for easy computation. The name “NumPy” defines “Numerical Python”. It can be used in linear algebra, random number capability, etc., and can act as a multi-dimensional container for generic data. Python NumPy Array is an object defining N-dimensional array in the form of rows and columns.

[NumPy](https://www.upgrad.com/blog/python-numpy-tutorial/) is preferred over lists in python because of:

* Less memory
* Fast
* Convenient

**Installation**

The installation of the NumPy package is done through typing the command ““pip install numpy” on the command prompt. Importing of the package in the IDE can be done through the command “import numpy as np”.

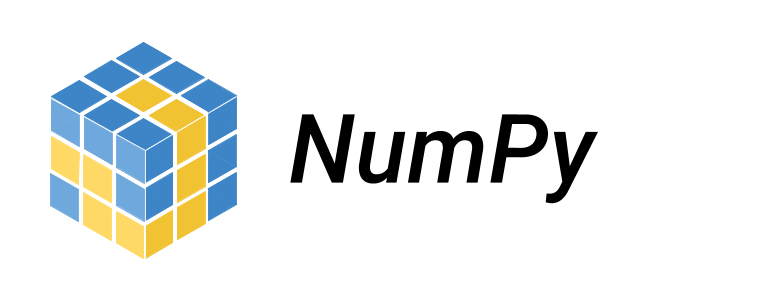


Fig 6.1: Numpy

**2.OpenCV**

**Open Source Computer vision** is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

## **OpenCV.**

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpuy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both **academic** and **commercial** use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.



Fig 6.2: Open CV

**3.PynPut**

The package pynput.keyboard contains classes for **controlling and monitoring the keyboard**. Use pynput.keyboard.Controller like this: A keyboard listener is a threading. Thread, and all callbacks will be invoked from the thread. Call pynput.keyboard .Listener. stop from anywhere, raise Stop Exception or return False from a callback to stop the listener.

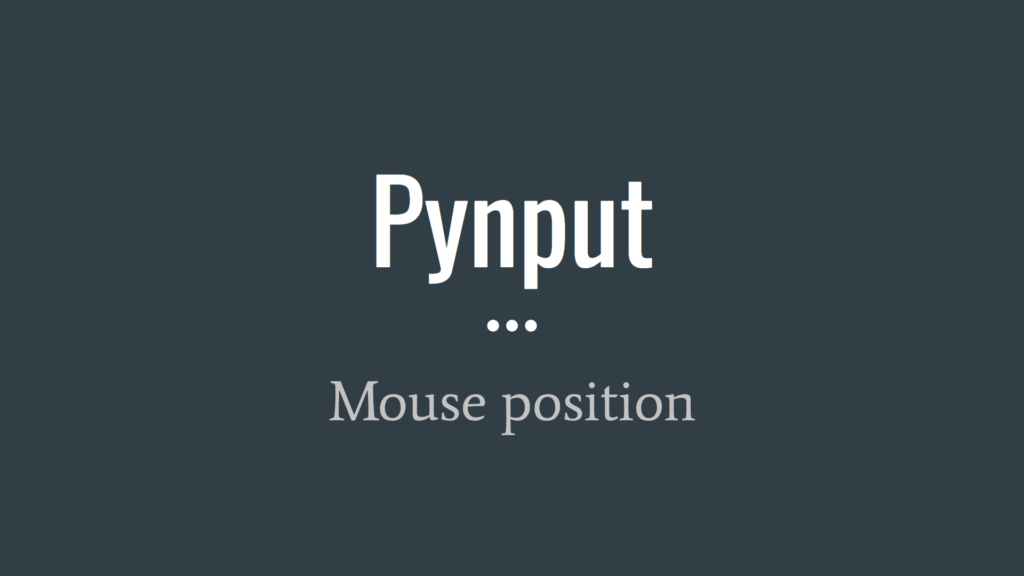


Fig 6.3- Pynput

**4.PyAutoGUI**

PyAutoGUI is a cross-platform GUI automation Python module for human beings. Used to programmatically control the mouse & keyboard.

pip install pyautogui

# **Dependencies**

PyAutoGUI supports Python 2 and 3. If you are installing PyAutoGUI from PyPI using pip:

Windows has no dependencies. The Win32 extensions do not need to be installed.

macOS needs the rubicon-objc module installed (in that order).

Linux needs the python3-xlib (or python-xlib for Python 2) module installed.

Pillow needs to be installed, and on Linux you may need to install additional libraries to make sure Pillow's PNG/JPEG works correctly. See:

If you want to do development and contribute to PyAutoGUI, you will need to install these modules from PyPI:

* pyscreeze
* pymsgbox
* pytweening

## **Keyboard and Mouse Control**

The x, y coordinates used by PyAutoGUI has the 0, 0 origin coordinates in the top left corner of the screen. The x coordinates increase going to the right (just as in mathematics) but the y coordinates increase going down (the opposite of mathematics). On a screen that is 1920 x 1080 pixels in size, coordinates 0, 0 are for the top left while 1919, 1079 is for the bottom right.

Currently, PyAutoGUI only works on the primary monitor. PyAutoGUI isn't reliable for the screen of a second monitor (the mouse functions may or may not work on multi-monitor setups depending on your operating system and version).

All keyboard presses done by PyAutoGUI are sent to the window that currently has focus, as if you had pressed the physical keyboard key.



Fig 6.4 Pyautogui

**FUNCTIONS**

**1.CV2.CONTOUR**

ou can use threshold() method of cv2 library to separate an object from the background in the image. To use cv2 library, you need to import cv2 library using import statement.

Now let’s see the syntax and return value of cv2 threshold() method first, then we will move on the examples.

**Python**

|  |  |
| --- | --- |
|  | cv2.threshold(src, thresholdValue, maxVal, thresholdingTechnique) |

### **Parameters**

You need to pass four parameters to cv2 threshold() method.

1. src:Input Grayscale Image array.
2. thresholdValue: Mention that value which is used to classify the pixel values.
3. maxVal: The value to be given if pixel value is more than (sometimes less than) the threshold value.
4. thresholdingTechnique: The type of thresholding to be applied.

There are 5 different simple thresholding techniques are :

* 1. cv2.THRESH\_BINARY: If pixel intensity is greater than the set threshold, value set to 255, else set to 0 (black).
  2. cv2.THRESH\_BINARY\_INV: Inverted or Opposite case of cv2.THRESH\_BINARY.<li.
  3. cv2.THRESH\_TRUNC: If pixel intensity value is greater than threshold, it is truncated to the threshold. The pixel values are set to be the same as the threshold. All other values remain the same.
  4. cv2.THRESH\_TOZERO: Pixel intensity is set to 0, for all the pixels intensity, less than the threshold value.
  5. cv2.THRESH\_TOZERO\_INV: Inverted or Opposite case of cv2.THRESH\_TOZERO.

### **Return Value**

This method return a tuple of 2 values in which 1st value is given threshold value and 2nd value is modified image array.

## cv2 threshold() Method examples

Now Let’s see the Python code :

### Example 1: Using cv2.THRESH\_BINARY thresholding technique.

**Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | #Python  # import computer vision library(cv2) in this code  import cv2    # main code  if \_\_name\_\_ == "\_\_main\_\_"        # mentioning absolute path of the image      img\_path = "C:\\Users\\user\\Desktop\\flower.jpg"        # read/load an image in grayscale mode      grey\_img = cv2.imread(img\_path,0)        # show the Input image on the newly created image window      cv2.imshow('Input',grey\_img)        # applying cv2.THRESH\_BINARY thresholding techniques      ret, thresh\_img = cv2.threshold(grey\_img, 128, 255, cv2.THRESH\_BINARY)        # show the Output image on the newly created image window      cv2.imshow('Output',thresh\_img) |

**Output:**

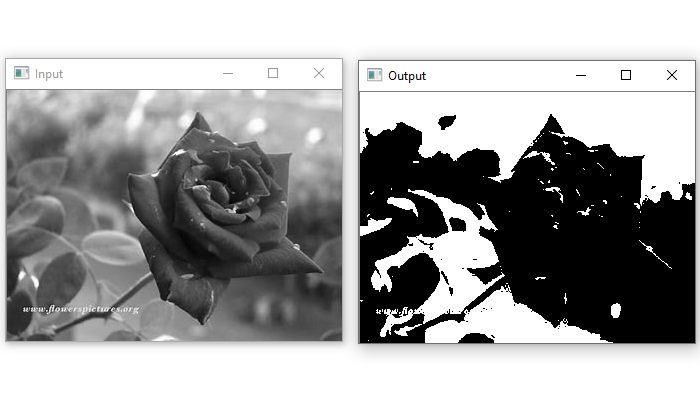


Fig 6.5 HSB Image

### Example 2 : Using cv2.THRESH\_BINARY\_INV thresholding technique.

**Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | #Python  # import computer vision library(cv2) in this code  import cv2    # main code  if \_\_name\_\_ == "\_\_main\_\_" :        # mentioning absolute path of the image      img\_path = "C:\\Users\\user\\Desktop\\flower.jpg"        # read/load an image in grayscale mode      grey\_img = cv2.imread(img\_path,0)        # show the Input image on the newly created image window      cv2.imshow('Input',grey\_img)        # applying cv2.THRESH\_BINARY\_INV thresholding techniques      ret, thresh\_img = cv2.threshold(grey\_img, 128, 255, cv2.THRESH\_BINARY\_INV)        # show the Output image on the newly created image window      cv2.imshow('Output',thresh\_img) |

**Output:**



Fig 6.6- Calibration Image

Similarly, you can apply other given thresholding techniques and see their results.

**HSB**

RGB - RGB (Red, Green, Blue) describes what kind of light needs to be emitted to produce a given color. Light is added together to create form from darkness. RGB stores individual values for red, green and blue. RGB is not a color space, it is a color model. There are many different RGB color spaces derived from this color model, some of which appear below.

HSV - (hue, saturation, value), also known as HSB (hue, saturation, brightness), is often used by artists because it is often more natural to think about a color in terms of hue and saturation than in terms of additive or subtractive color components. HSV is a transformation of an RGB colorspace, and its components and colorimetry are relative to the RGB colorspace from which it was derived.

**Sample Solution**:-

**Python Code:**

def rgb\_to\_hsv(r, g, b):

r, g, b = r/255.0, g/255.0, b/255.0

mx = max(r, g, b)

mn = min(r, g, b)

df = mx-mn

if mx == mn:

h = 0

elif mx == r:

h = (60 \* ((g-b)/df) + 360) % 360

elif mx == g:

h = (60 \* ((b-r)/df) + 120) % 360

elif mx == b:

h = (60 \* ((r-g)/df) + 240) % 360

if mx == 0:

s = 0

else:

s = (df/mx)\*100

v = mx\*100

return h, s, v

print(rgb\_to\_hsv(255, 255, 255))

print(rgb\_to\_hsv(0, 215, 0))

Copy

Sample Output:

(0, 0.0, 100.0)

(120.0, 100.0, 84.31372549019608)

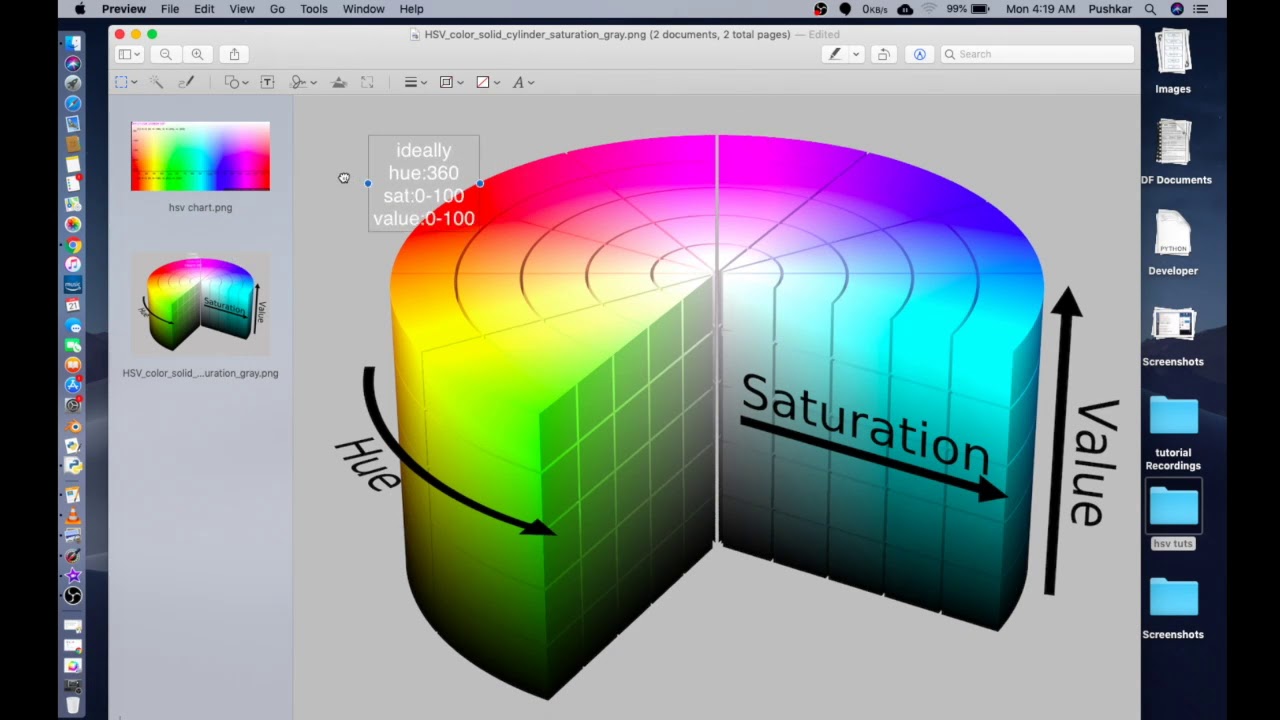


Fig 6.7 Saturation Image

**CALIBRATION**

A camera is an integral part of several domains like robotics, space exploration, etc camera is playing a major role. It helps to capture each and every moment and helpful for many analyses. In order to use the camera as a visual sensor, we should know the parameters of the camera. **Camera Calibration** is nothing but estimating the parameters of a camera, parameters about the camera are required to determine an accurate relationship between a 3D point in the real world and its corresponding 2D projection (pixel) in the image captured by that calibrated camera.

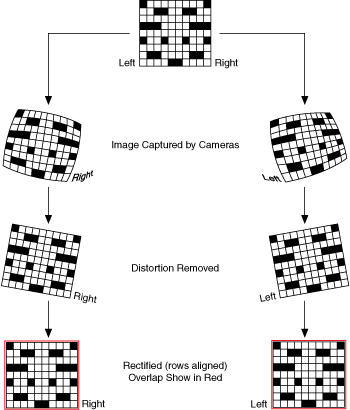
We need to consider both internal parameters like focal length, optical center, and radial distortion coefficients of the lens etc., and external parameters like rotation and translation of the camera with respect to some real world coordinate system.

### **Camera Calibration can be done in a step-by-step approach:**

* **Step 1:** First define real world coordinates of 3D points using known size of checkerboard pattern.
* **Step 2:** Different viewpoints of check-board image is captured.
* **Step 3:** *findChessboardCorners()* is a method in *OpenCV* and used to find pixel coordinates *(u, v)* for each 3D point in different images
* **Step 4:** Then *calibrateCamera()* method is used to find camera parameters.

It will take our calculated*(threedpoints, twodpoints, grayColor.shape[::-1], None, None)*as parameters and returns list having elements as *Camera matrix, Distortion coefficient, Rotation Vectors*, and *Translation Vectors.*

*Camera Matrix* helps to transform 3D objects points to 2D image points and the *Distortion Coefficient* returns the position of the camera in the world, with the values of *Rotation* and *Translation* vectors



**REMOVE NOISE**

**Morphological operations** are a set of operations that process images based on shapes. They apply a structuring element to an input image and generate an output image.  
The most basic morphological operations are two:**Erosion and Dilation**  
**Basics of Erosion:**

* Erodes away the boundaries of foreground object
* Used to diminish the features of an image.

**Working of erosion:**

1. A kernel(a matrix of odd size(3,5,7) is convolved with the image.
2. A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded (made to zero).
3. Thus all the pixels near boundary will be discarded depending upon the size of kernel.
4. So the thickness or size of the foreground object decreases or simply white region decreases in the image.

**Basics of dilation:**

* Increases the object area
* Used to accentuate features

**Working of dilation:**

1. A kernel(a matrix of odd size(3,5,7) is convolved with the image
2. A pixel element in the original image is ‘1’ if atleast one pixel under the kernel is ‘1’.
3. It increases the white region in the image or size of foreground object increases

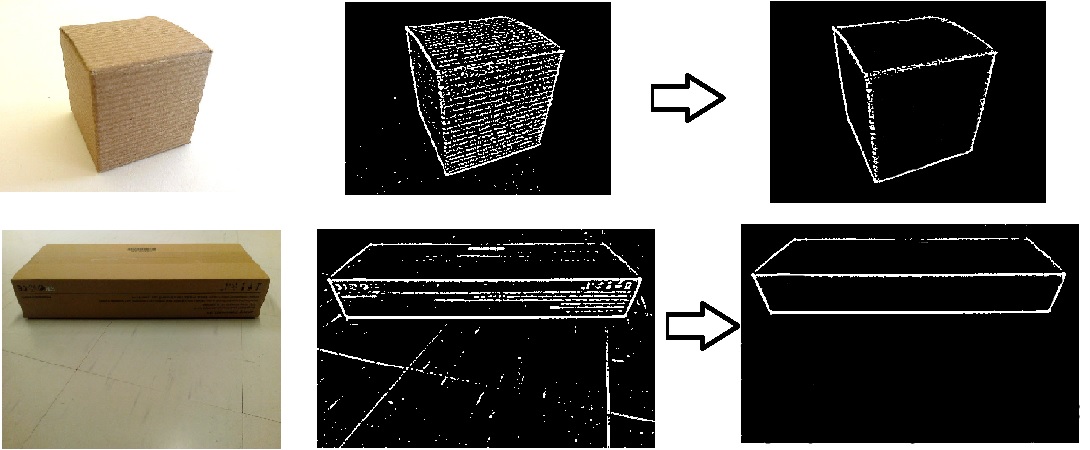


Fig 6.8- Remove Noise

**CHAPTER – 7**

**CODING**

**The first thing that we do is convert the captured video into HSV format.**

**code :**

# All packages needed for the program are imported ahead

import cv2

cap = cv2.VideoCapture(0)

while(1):

# Capture frame-by-frame

\_, frameinv = cap.read()

# flip horizontaly to get mirror image in camera

frame = cv2.flip( frameinv, 1)

# Our operations on the frame come here

hsv = cv2.cvtColor( frame, cv2.COLOR\_BGR2HSV)

# Display the resulting frame

cv2.imshow('Frame', hsv)

k = cv2.waitKey(10) & 0xFF

if k == 27:

break

cap.release()

cv2.destroyAllWindows()

Now the user gets to calibrate the colour ranges for three of his fingers individually. This is done by calling the **calibrateColor()** function thrice right at the beginning of the program.

The user has an option to use the default settings as well.

mport cv2

import numpy as np

def nothing(x):

pass

# Create a black image, a window

kernel = np.zeros((300,512,3), np.uint8)

name = 'Calibrate'

cv2.namedWindow(name)

# create trackbars for color change

cv2.createTrackbar('Hue', name, 0, 255, nothing)

cv2.createTrackbar('Sat', name, 0, 255, nothing)

cv2.createTrackbar('Val', name, 0, 255, nothing)

# create switch for ON/OFF functionality

switch = '0 : OFF \n 1 : ON'

cv2.createTrackbar(switch, name,0,1,nothing)

while(1):

cv2.imshow(name,kernel)

k = cv2.waitKey(1) & 0xFF

if k == 27:

break

# get current positions of four trackbars

hue = cv2.getTrackbarPos('Hue', name)

sat = cv2.getTrackbarPos('Sat', name)

val = cv2.getTrackbarPos('Val', name)

s = cv2.getTrackbarPos(switch,name)

if s == 0:

kernel[:] = 0

else:

kernel[:] = [hue,sat,val]

cv2.destroyAllWindows()

Depending on the calibrations, only the three fingertips are extracted from the video, one by one, using the **cv2.inRange()**function. In order to remove noise in the video feed, we apply a two-step morphism i.e. **erosion and dilation.**The noise filtered image referred to as mask in the program is then sent for locating the centres.

# cv2.inRange function is used to filter out a particular color from the frame

# The result then undergoes morphosis i.e. erosion and dilation

# Resultant frame is returned as mask

def makeMask(hsv\_frame, color\_Range):

mask = cv2.inRange( hsv\_frame, color\_Range[0], color\_Range[1])

# Morphosis next ...

eroded = cv2.erode( mask, kernel, iterations=1)

dilated = cv2.dilate( eroded, kernel, iterations=1)

return dilated

# Contours on the mask are detected.. Only those lying in the previously set area

# range are filtered out and the centroid of the largest of these is drawn and returned

def drawCentroid(vid, color\_area, mask, showCentroid):

contour, \_ = cv2.findContours( mask, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

l=len(contour)

area = np.zeros(l)

# filtering contours on the basis of area rane specified globally

for i in range(l):

if

cv2.contourArea(contour[i])>color\_area[0] and cv2.contourArea(contour[i])

<color\_area[1]: area[i]="cv2.contourArea(contour[i])" else:="" a="sorted(" area,="" reverse="True)" <="" p=""></color\_area[1]:>

# bringing contours with largest valid area to the top

for i in range(l):

for j in range(1):

if area[i] == a[j]:

swap( contour, i, j)

if l > 0 :

# finding centroid using method of 'moments'

M = cv2.moments(contour[0])

if M['m00'] != 0:

cx = int(M['m10']/M['m00'])

cy = int(M['m01']/M['m00'])

center = (cx,cy)

if showCentroid:

cv2.circle( vid, center, 5, (0,0,255), -1)

return center

else:

# return error handling values

return (-1,-1)

• Due to noise captured by the webcam and vibrations in the hand, the centres keep vibrating around a mean position. On scaling up, these vibrations create a lot of problem with the accuracy of cursor position. To reduce the shakiness in cursor, we make use of differential position allocation for the cursor. We compare the new centre with the previous position of the cursor. If difference is less than 5 pixels, it is usually due to noise. Thus the new cursor position is inclined more towards the previous one. However, a larger difference in previous position and new centre is considered as voluntary movement and the new cursor position is set close to the new centre. For details, go through the setCursorPosition() function in the code.

'''

This function takes as input the center of yellow region (yc) and

the previous cursor position (pyp). The new cursor position is calculated

in such a way that the mean deviation for desired steady state is reduced.

def setCursorPos( yc, pyp):

yp = np.zeros(2)

if abs(yc[0]-pyp[0])<5 and abs(yc[1]-pyp[1])<5:

yp[0] = yc[0] + .7\*(pyp[0]-yc[0])

yp[1] = yc[1] + .7\*(pyp[1]-yc[1])

else:

yp[0] = yc[0] + .1\*(pyp[0]-yc[0])

yp[1] = yc[1] + .1\*(pyp[1]-yc[1])

return yp

Now the three centres are sent for deciding what action needs to be performed depending  
on their relative positions. This is done in the chooseAction() function in the code. Depending upon its output, the performAction() function carries out either of the following using the PyAutoGUI library:

* free cursor movement
* left click
* right click
* drag/select
* scroll up
* scroll down

# Depending upon the relative positions of the three centroids, this function chooses whether

# the user desires free movement of cursor, left click, right click or dragging

def chooseAction(yp, rc, bc):

out = np.array(['move', 'false'])

if rc[0]!=-1 and bc[0]!=-1:

if distance(yp,rc)<50 and distance(yp,bc)<50 and distance(rc,bc)<50 :

out[0] = 'drag'

out[1] = 'true'

return out

elif distance(rc,bc)<40:

out[0] = 'right'

return out

elif distance(yp,rc)<40:

out[0] = 'left'

return out

elif distance(yp,rc)>40 and rc[1]-bc[1]>120:

out[0] = 'down'

return out

elif bc[1]-rc[1]>110:

out[0] = 'up'

return out

else:

return out

else:

out[0] = -1

return out

def performAction( yp, rc, bc, action, drag, perform):

if perform:

cursor[0] = 4\*(yp[0]-110)

cursor[1] = 4\*(yp[1]-120)

if action == 'move':

if yp[0]>110 and yp[0]<590 and yp[1]>120 and yp[1]<390:

pyautogui.moveTo(cursor[0],cursor[1])

elif yp[0]<110 and yp[1]>120 and yp[1]<390:

pyautogui.moveTo( 8 , cursor[1])

elif yp[0]>590 and yp[1]>120 and yp[1]<390:

pyautogui.moveTo(1912, cursor[1])

elif yp[0]>110 and yp[0]<590 and yp[1]<120:

pyautogui.moveTo(cursor[0] , 8)

elif yp[0]>110 and yp[0]<590 and yp[1]>390:

pyautogui.moveTo(cursor[0] , 1072)

elif yp[0]<110 and yp[1]<120:

pyautogui.moveTo(8, 8)

elif yp[0]<110 and yp[1]>390:

pyautogui.moveTo(8, 1072)

elif yp[0]>590 and yp[1]>390:

pyautogui.moveTo(1912, 1072)

else:

pyautogui.moveTo(1912, 8)

elif action == 'left':

pyautogui.click(button = 'left')

elif action == 'right':

pyautogui.click(button = 'right')

time.sleep(0.3)

elif action == 'up':

pyautogui.scroll(5)

# time.sleep(0.3)

elif action == 'down':

pyautogui.scroll(-5)

# time.sleep(0.3)

elif action == 'drag' and drag == 'true':

global y\_pos

drag = 'false'

pyautogui.mouseDown()

while(1):

k = cv2.waitKey(10) & 0xFF

changeStatus(k)

\_, frameinv = cap.read()

# flip horizontaly to get mirror image in camera

frame = cv2.flip( frameinv, 1)

hsv = cv2.cvtColor( frame, cv2.COLOR\_BGR2HSV)

b\_mask = makeMask( hsv, blue\_range)

r\_mask = makeMask( hsv, red\_range)

y\_mask = makeMask( hsv, yellow\_range)

py\_pos = y\_pos

b\_cen = drawCentroid( frame, b\_area, b\_mask, showCentroid)

r\_cen = drawCentroid( frame, r\_area, r\_mask, showCentroid)

y\_cen = drawCentroid( frame, y\_area, y\_mask, showCentroid)

if py\_pos[0]!=-1 and y\_cen[0]!=-1:

y\_pos = setCursorPos(y\_cen, py\_pos)

performAction(y\_pos, r\_cen, b\_cen, 'move', drag, perform)

cv2.imshow('Frame', frame)

if distance(y\_pos,r\_cen)>60 or distance(y\_pos,b\_cen)>60:

break

pyautogui.mouseUp()

FINAL CODE

# All packages needed for the program are imported ahead

import cv2

import numpy as np

import pyautogui

import time

# Some global variables or others that need prior intialization are initalized here

# colour ranges for feeding to the inRange funtions

blue\_range = np.array([[88,78,20],[128,255,255]])

yellow\_range = np.array([[21,70,80],[61,255,255]])

red\_range = np.array([[158,85,72],[180 ,255,255]])

# Prior initialization of all centers for safety

b\_cen, y\_pos, r\_cen = [240,320],[240,320],[240,320]

cursor = [960,540]

# Area ranges for contours of different colours to be detected

r\_area = [100,1700]

b\_area = [100,1700]

y\_area = [100,1700]

# Rectangular kernal for eroding and dilating the mask for primary noise removal

kernel = np.ones((7,7),np.uint8)

# Status variables defined globally

perform = False

showCentroid = False

# 'nothing' function is useful when creating trackbars

# It is passed as last arguement in the cv2.createTrackbar() function

def nothing(x):

pass

# To bring to the top the contours with largest area in the specified range

# Used in drawContour()

def swap( array, i, j):

temp = array[i]

array[i] = array[j]

array[j] = temp

# Distance between two centroids

def distance( c1, c2):

distance = pow( pow(c1[0]-c2[0],2) + pow(c1[1]-c2[1],2) , 0.5)

return distance

# To toggle status of control variables

def changeStatus(key):

global perform

global showCentroid

global yellow\_range,red\_range,blue\_range

# toggle mouse simulation

if key == ord('p'):

perform = not perform

if perform:

print('Mouse simulation ON...')

else:

print('Mouse simulation OFF...')

# toggle display of centroids

elif key == ord('c'):

showCentroid = not showCentroid

if showCentroid:

print('Showing Centroids...')

else:

print('Not Showing Centroids...')

elif key == ord('r'):

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(' You have entered recalibration mode.')

print(' Use the trackbars to calibrate and press SPACE when done.')

print(' Press D to use the default settings')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

yellow\_range = calibrateColor('Yellow', yellow\_range)

red\_range = calibrateColor('Red', red\_range)

blue\_range = calibrateColor('Blue', blue\_range)

else:

pass

# cv2.inRange function is used to filter out a particular color from the frame

# The result then undergoes morphosis i.e. erosion and dilation

# Resultant frame is returned as mask

def makeMask(hsv\_frame, color\_Range):

mask = cv2.inRange( hsv\_frame, color\_Range[0], color\_Range[1])

# Morphosis next ...

eroded = cv2.erode( mask, kernel, iterations=1)

dilated = cv2.dilate( eroded, kernel, iterations=1)

return dilated

# Contours on the mask are detected.. Only those lying in the previously set area

# range are filtered out and the centroid of the largest of these is drawn and returned

def drawCentroid(vid, color\_area, mask, showCentroid):

contour,\_ = cv2.findContours( mask, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

l=len(contour)

area = np.zeros(l)

# filtering contours on the basis of area rane specified globally

for i in range(l):

if cv2.contourArea(contour[i])>color\_area[0] and cv2.contourArea(contour[i])<color\_area[1]:

area[i] = cv2.contourArea(contour[i])

else:

area[i] = 0

a = sorted( area, reverse=True)

# bringing contours with largest valid area to the top

for i in range(l):

for j in range(1):

if area[i] == a[j]:

swap( contour, i, j)

if l > 0 :

# finding centroid using method of 'moments'

M = cv2.moments(contour[0])

if M['m00'] != 0:

cx = int(M['m10']/M['m00'])

cy = int(M['m01']/M['m00'])

center = (cx,cy)

if showCentroid:

cv2.circle( vid, center, 5, (0,0,255), -1)

return center

else:

# return error handling values

return (-1,-1)

# This function helps in filtering the required colored objects from the background

def calibrateColor(color, def\_range):

global kernel

name = 'Calibrate '+ color

cv2.namedWindow(name)

cv2.createTrackbar('Hue', name, def\_range[0][0]+20, 180, nothing)

cv2.createTrackbar('Sat', name, def\_range[0][1] , 255, nothing)

cv2.createTrackbar('Val', name, def\_range[0][2] , 255, nothing)

while(1):

ret , frameinv = cap.read()

frame=cv2.flip(frameinv ,1)

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

hue = cv2.getTrackbarPos('Hue', name)

sat = cv2.getTrackbarPos('Sat', name)

val = cv2.getTrackbarPos('Val', name)

lower = np.array([hue-20,sat,val])

upper = np.array([hue+20,255,255])

mask = cv2.inRange(hsv, lower, upper)

eroded = cv2.erode( mask, kernel, iterations=1)

dilated = cv2.dilate( eroded, kernel, iterations=1)

cv2.imshow(name, dilated)

k = cv2.waitKey(5) & 0xFF

if k == ord(' '):

cv2.destroyWindow(name)

return np.array([[hue-20,sat,val],[hue+20,255,255]])

elif k == ord('d'):

cv2.destroyWindow(name)

return def\_range

'''

This function takes as input the center of yellow region (yc) and

the previous cursor position (pyp). The new cursor position is calculated

in such a way that the mean deviation for desired steady state is reduced.

'''

def setCursorPos( yc, pyp):

yp = np.zeros(2)

if abs(yc[0]-pyp[0])<5 and abs(yc[1]-pyp[1])<5:

yp[0] = yc[0] + .7\*(pyp[0]-yc[0])

yp[1] = yc[1] + .7\*(pyp[1]-yc[1])

else:

yp[0] = yc[0] + .1\*(pyp[0]-yc[0])

yp[1] = yc[1] + .1\*(pyp[1]-yc[1])

return yp

# Depending upon the relative positions of the three centroids, this function chooses whether

# the user desires free movement of cursor, left click, right click or dragging

def chooseAction(yp, rc, bc):

out = np.array(['move', 'false'])

if rc[0]!=-1 and bc[0]!=-1:

if distance(yp,rc)<50 and distance(yp,bc)<50 and distance(rc,bc)<50 :

out[0] = 'drag'

out[1] = 'true'

return out

elif distance(rc,bc)<40:

out[0] = 'left'

return out

elif distance(yp,rc)<40:

out[0] = 'right'

return out

elif distance(yp,rc)>40 and rc[1]-bc[1]>120:

out[0] = 'down'

return out

elif bc[1]-rc[1]>110:

out[0] = 'up'

return out

else:

return out

else:

out[0] = -1

return out

# Movement of cursor on screen, left click, right click,scroll up, scroll down

# and dragging actions are performed here based on value stored in 'action'.

def performAction( yp, rc, bc, action, drag, perform):

if perform:

cursor[0] = 4\*(yp[0]-110)

cursor[1] = 4\*(yp[1]-120)

if action == 'move':

if yp[0]>110 and yp[0]<590 and yp[1]>120 and yp[1]<390:

pyautogui.moveTo(cursor[0],cursor[1])

elif yp[0]<110 and yp[1]>120 and yp[1]<390:

pyautogui.moveTo( 8 , cursor[1])

elif yp[0]>590 and yp[1]>120 and yp[1]<390:

pyautogui.moveTo(1912, cursor[1])

elif yp[0]>110 and yp[0]<590 and yp[1]<120:

pyautogui.moveTo(cursor[0] , 8)

elif yp[0]>110 and yp[0]<590 and yp[1]>390:

pyautogui.moveTo(cursor[0] , 1072)

elif yp[0]<110 and yp[1]<120:

pyautogui.moveTo(8, 8)

elif yp[0]<110 and yp[1]>390:

pyautogui.moveTo(8, 1072)

elif yp[0]>590 and yp[1]>390:

pyautogui.moveTo(1912, 1072)

else:

pyautogui.moveTo(1912, 8)

elif action == 'left':

pyautogui.click(button = 'left')

elif action == 'right':

pyautogui.click(button = 'right')

time.sleep(0.3)

elif action == 'up':

pyautogui.scroll(5)

# time.sleep(0.3)

elif action == 'down':

pyautogui.scroll(-5)

# time.sleep(0.3)

elif action == 'drag' and drag == 'true':

global y\_pos

drag = 'false'

pyautogui.mouseDown()

while(1):

k = cv2.waitKey(10) & 0xFF

changeStatus(k)

\_, frameinv = cap.read()

# flip horizontaly to get mirror image in camera

frame = cv2.flip( frameinv, 1)

hsv = cv2.cvtColor( frame, cv2.COLOR\_BGR2HSV)

b\_mask = makeMask( hsv, blue\_range)

r\_mask = makeMask( hsv, red\_range)

y\_mask = makeMask( hsv, yellow\_range)

py\_pos = y\_pos

b\_cen = drawCentroid( frame, b\_area, b\_mask, showCentroid)

r\_cen = drawCentroid( frame, r\_area, r\_mask, showCentroid)

y\_cen = drawCentroid( frame, y\_area, y\_mask, showCentroid)

if py\_pos[0]!=-1 and y\_cen[0]!=-1:

y\_pos = setCursorPos(y\_cen, py\_pos)

performAction(y\_pos,r\_cen, b\_cen, 'move', drag, perform)

cv2.imshow('Frame', frame)

if distance(y\_pos,r\_cen)>60 or distance(y\_pos,b\_cen)>60 or distance(r\_cen,b\_cen)>60:

break

pyautogui.mouseUp()

cap = cv2.VideoCapture(0)

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(' You have entered calibration mode.')

print(' Use the trackbars to calibrate and press SPACE when done.')

print(' Press D to use the default settings.')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

yellow\_range = calibrateColor('Yellow', yellow\_range)

red\_range = calibrateColor('Red', red\_range)

blue\_range = calibrateColor('Blue', blue\_range)

print(' Calibration Successfull...')

cv2.namedWindow('Frame')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(' Press P to turn ON and OFF mouse simulation.')

print(' Press C to display the centroid of various colours.')

print(' Press R to recalibrate color ranges.')

print(' Press ESC to exit.')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

while(1):

k = cv2.waitKey(10) & 0xFF

changeStatus(k)

\_, frameinv = cap.read()

# flip horizontaly to get mirror image in camera

frame = cv2.flip( frameinv, 1)

hsv = cv2.cvtColor( frame, cv2.COLOR\_BGR2HSV)

b\_mask = makeMask( hsv, blue\_range)

r\_mask = makeMask( hsv, red\_range)

y\_mask = makeMask( hsv, yellow\_range)

py\_pos = y\_pos

b\_cen = drawCentroid( frame, b\_area, b\_mask, showCentroid)

r\_cen = drawCentroid( frame, r\_area, r\_mask, showCentroid)

y\_cen = drawCentroid( frame, y\_area, y\_mask, showCentroid)

if py\_pos[0]!=-1 and y\_cen[0]!=-1 and y\_pos[0]!=-1:

y\_pos = setCursorPos(y\_cen, py\_pos)

output = chooseAction(y\_pos, r\_cen, b\_cen)

if output[0]!=-1:

performAction(y\_pos, r\_cen, b\_cen, output[0], output[1], perform)

cv2.imshow('Frame', frame)

if k == 27:

break

cv2.destroyAllWindows()

**CHAPTER – 7**

**TESTING STAGES**

**7.1 TESTING**

**7.1.1 What is testing:-**

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

**7.1.2 Testing architecture:-**

REQUIREMENT

PLANNING

DESIGN

CODING

TESTING

MAINTANANCE

Fig7.1: Waterfall Diagram

**7.1.3 Testing techniques:-**

In this project, following testing techniques are used.

* Unit Testing
* Integration Testing
* System Testing

**Unit Testing:-**

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

**Integration Testing:-**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Sometime there is a mean to integrate & merge the unit program together would also be work without error. Integration testing can be done in two ways:

Bottom-up integration:- This testing begins with unit testing, followed by tests of progressively higher level combinations of units called modules or builds.

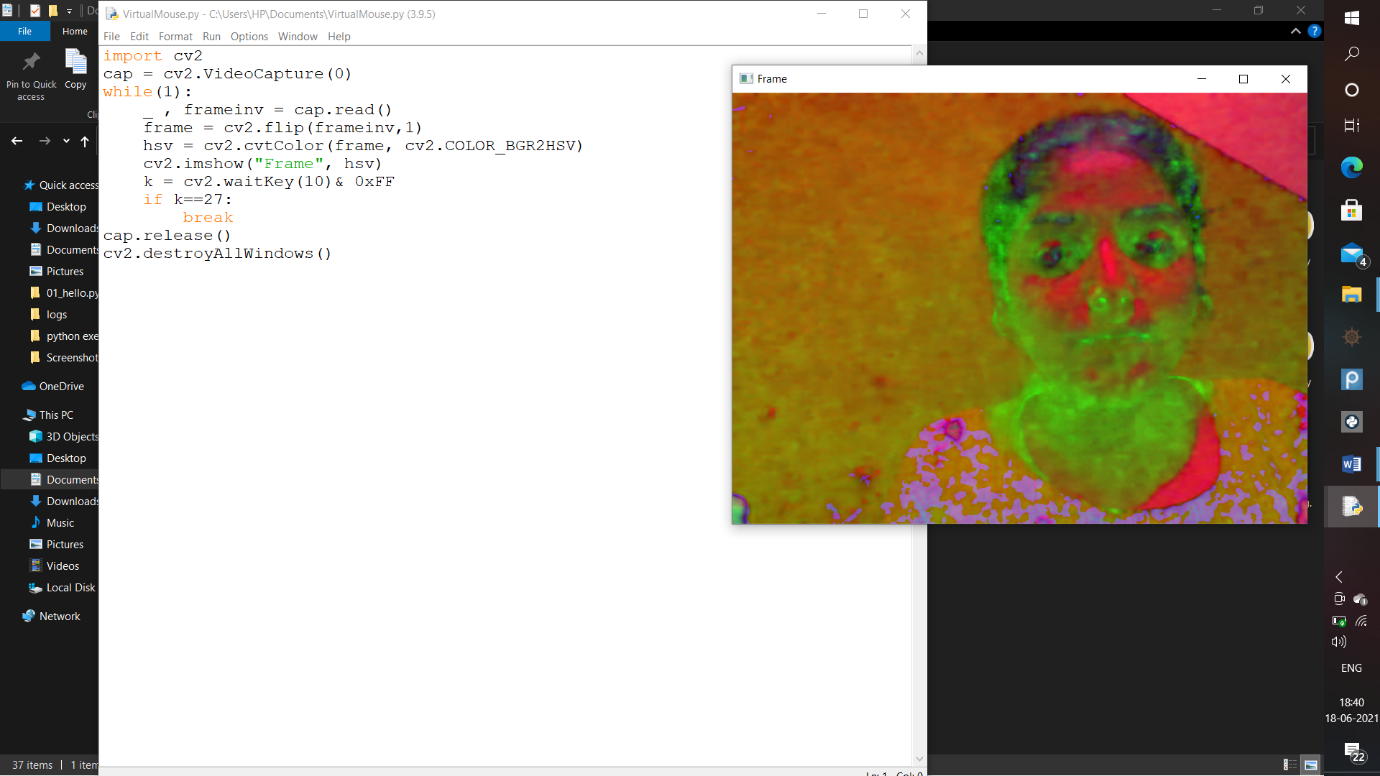
Top-down integration:- In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter.

**System Testing:-**

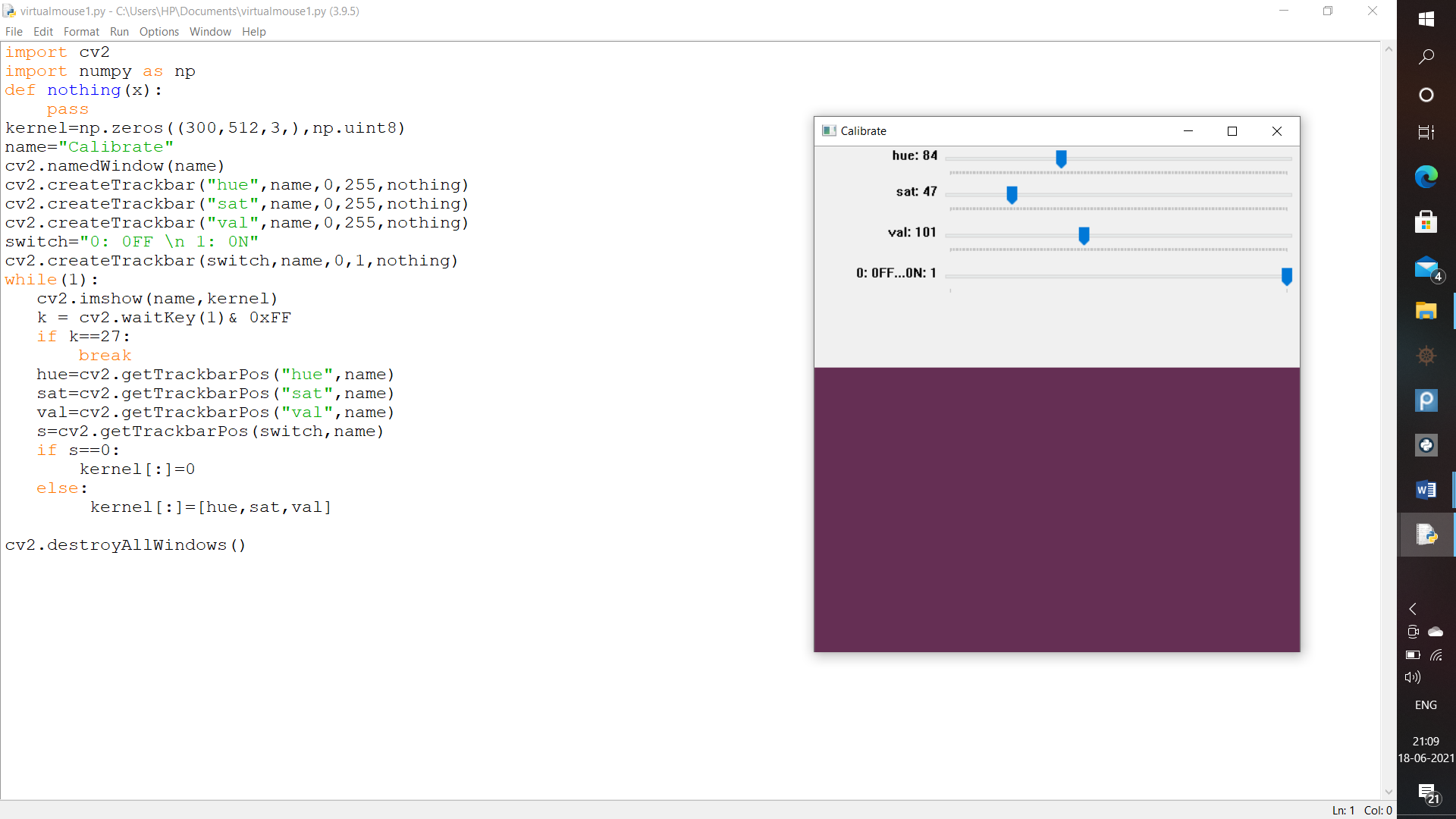
after completing unit and integration testing, system testing is perform we test complete program software along with its expected environment system testing defined as combination of software, hardware & other associated parts that work together. It is only the phase where both functional and nonfunctional requirement of system are tested.

**SNAPSHOTS**

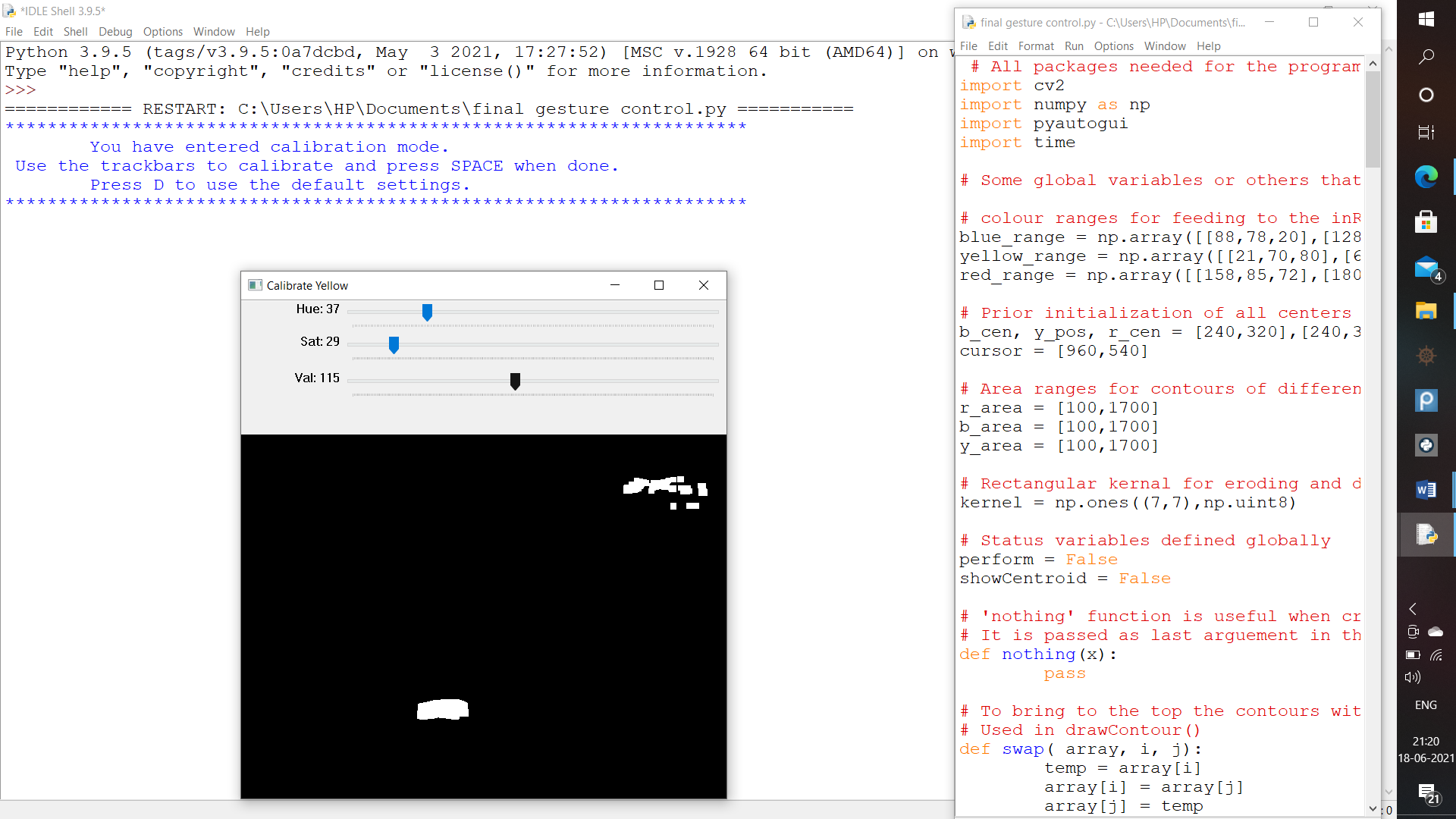
**HSB IMAGE**

****

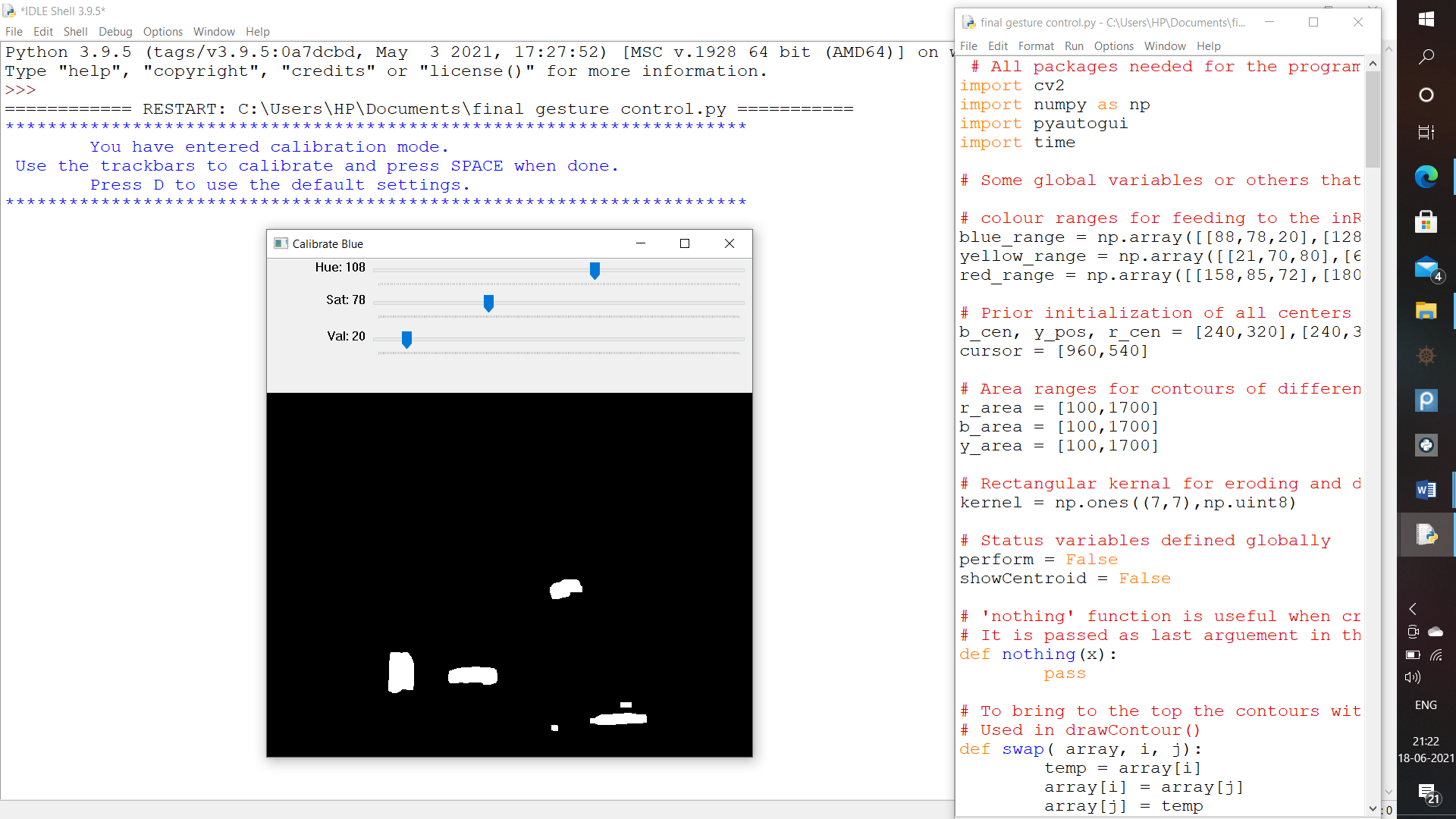
**CALIBRATION**



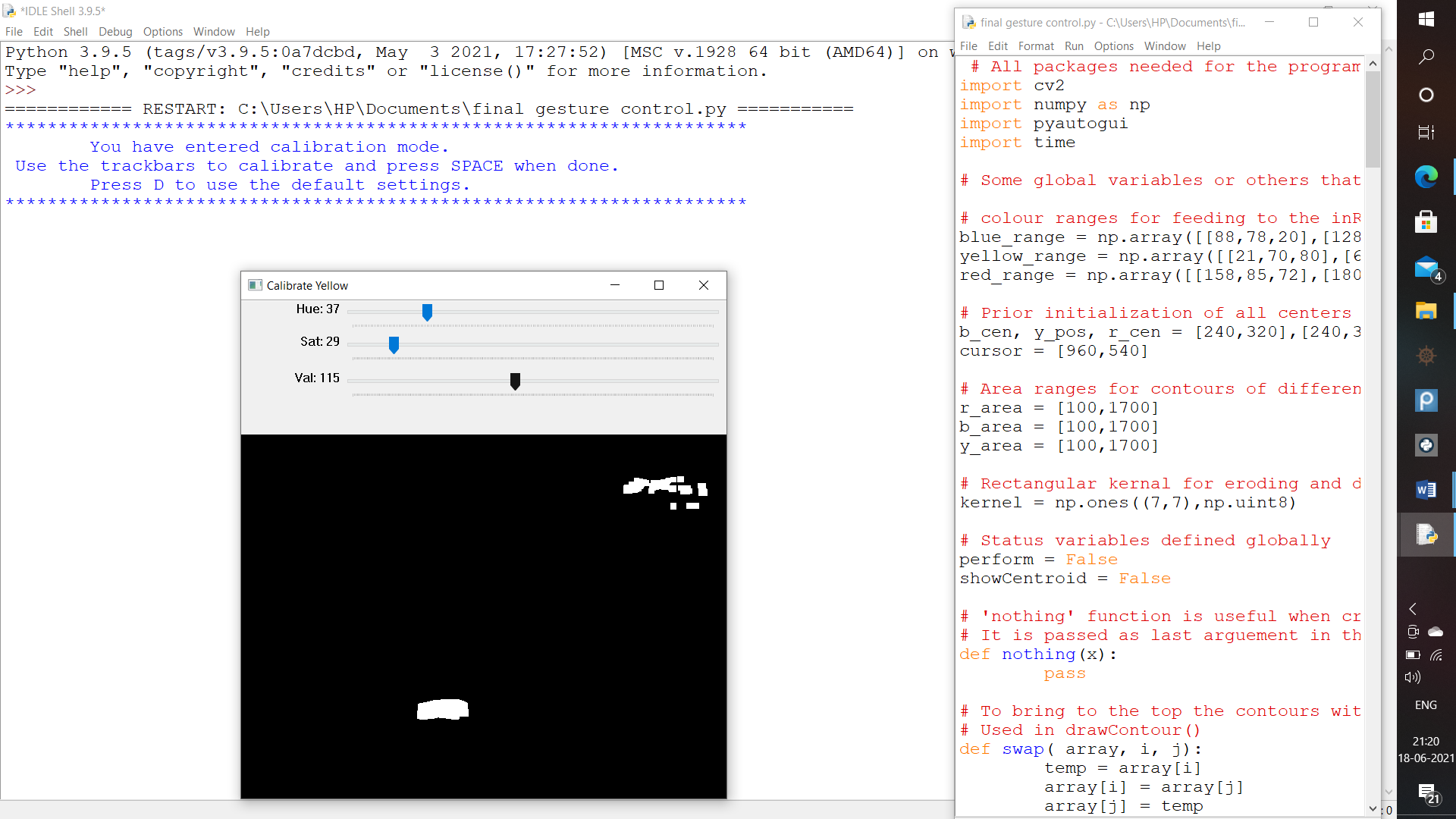
**RED COLOR CALIBRATION**



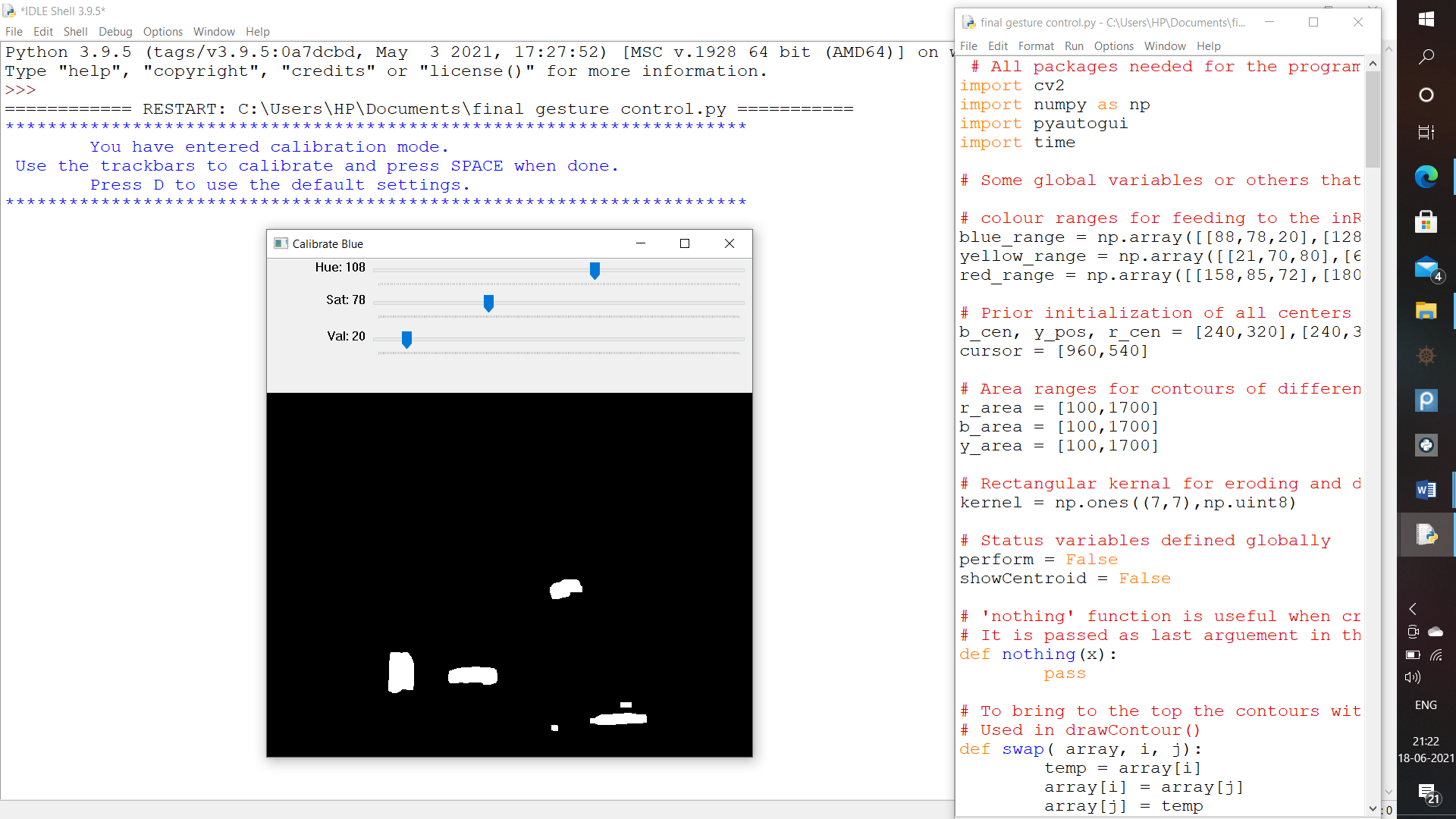
**BLUE COLOR CALIBRATION**



**YELLOW COLOR CALIBRATION**



**REMOVE NOISE**



**MID POINTS**



**SELECT ITEM**



**RIGHT CLICK**

**SCROLL**

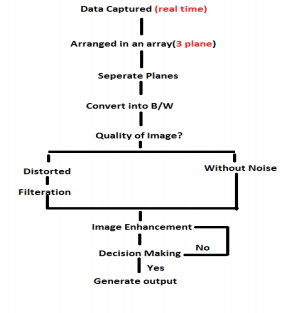




**PERFORMANCE ANALYSIS**



is only the representation of Image processing part in the firmware. Above diagram explains how single shot of the image processes throughout. Every image taken from real time camera has to go from the above mentioned processes. Every real time camera has its own frames per second. In a certain video every single frame has to go from this process.Figure 15 is the overall system flow. System architecture of this project is already mentioned in Figure 1. This is basically a software architecture which helps decide how to process the image and how to take the required decisions. The procedure from capturing image to performing demanded action can be seen from the design,



**CONCLUSION**

Gesture recognition gives the best interaction between human and machine. Gesture recognition is also important for developing alternative human computer interaction modalities. It enables human to interface with machine in a more natural way. Gesture recognition can be used for many applications like sign language recognition for deaf and dumb people, robot control etc. This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. Digital Canvas is an extension of our system which is gaining popularity among artists, by which the artist could create 2D or 3D images using the Virtual Mouse technology using the hand as brush and a Virtual Reality kit or a monitor as display set. This technology can be used to help patients who don’t have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person’s motions are tracked and interpreted as commands. The system architecture that has been proposed will completely change the way people would use the Computer system. Presently, the webcam, microphone and mouse are an integral part of the Computer system. This project will completely eliminate the necessity of mouse. Also this would lead to a new era of Human Computer Interaction (HCI) where no physical contact with the device is required. The use of object detection and image processing in MATLAB for the implementation of our proposed work proved to be practically successful and the movement of mouse cursor is achieved with a good precision accuracy. This technology can be used to help patients who don‟t have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person‟s motions are tracked and interpreted as commands. Most of the applications require additional hardware which is often very costly. The motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system. Various application programs can be written exclusively for this technology to create a wide range of applications with the minimum requirement of resources

**FUTURE SCOPE**

Every innovation,it is important factor to reduce the humans effort in that Virtual reality place important role. In future all technology is based on artificial intelligence and virtual reality.

• In future driverless cars will be implemented.

• Face detection is possible in low budget phones in future

. • 3D models(video games).

• Medical imagery during surgery without touching patients or any things.

In this paper, an object tracking based virtual mouse application has been developed and implemented using a webcam. The system has been implemented in JAVA environment. We implemented all mouse tasks such as left and right clicking, double clicking, and scrolling. In the future, we plan to add more features such as enlarging and shrinking windows, closing window, etc. This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. Furthermore a similar technology can be applied to create applications like a digital canvas which is gaining popularity among artists. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person’s motions are tracked and interpreted as commands. Most of the applications require additional hardware which is often very costly. Our motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system. Various application programs can be written exclusively for this technology to create a wide range of applications with the minimum requirement of resources

**REFERENCES**

[1] https:// www .stackoverflow.com

[2] <https://wikipedia.com>

[3] https://www.pythonpoint.com

[4] https://www.android.com.

[5] [Mouse Control using a Web Camera and Hand Gestures with Colour Tapes (ijste.org)](http://www.ijste.org/articles/IJSTEV2I11303.pdf)

[6] https://developer.android.com.

[7] [virtual-mouse-using-hand-gesture.pdf (ijmter.com)](https://ijmter.com/papers/volume-4/issue-3/virtual-mouse-using-hand-gesture.pdf)

[8] [Virtual Mouse using Hand Gesture and Color Detection (ijcaonline.org)](https://www.ijcaonline.org/research/volume128/number11/ghodichor-2015-ijca-906665.pdf)

[9] [IRJET-V5I4872.pdf](https://www.irjet.net/archives/V5/i4/IRJET-V5I4872.pdf)

[10] [Working With Files in Python – Real Python](https://realpython.com/working-with-files-in-python/)

[11] [Python 3 basics - GeeksforGeeks](https://www.geeksforgeeks.org/python-3-basics/?ref=lbp)

[12] [Gesture Controlled Mouse : 6 Steps (with Pictures) - Instructables](https://www.instructables.com/Gesture-Controlled-Mouse/)

[13] https://www.w3schools.com