**Mouse Control using Gesture recognition**

By

Krishnaja M - 20186028

Naveen B - 20186068

Mahateja A - 20186025

Project Submitted to

MSIT, IIIT HYDERABAD

Copyright:

Krishnaja M,2019

Naveen B,2019

Mahateja A,2019

Approved: Manasa A

Assistant Mentor, MSIT

IIIT Hyderabad

Date: 29-06-2019

**Statement of Confidentiality**

This document is submitted in the requirement for the degree of MSIT in IIIT Hyderabad. This is the product of our own labor except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

**Acknowledgments**

The usage of the libraries and individual acknowledgments are included within in the docstrings.

We would like to specifically thank the following people:

**Manasa(Assistant Mentor)**: For being a brilliant project guide, that has provided constant support and reassurance throughout the entire project.

**All the knowledgeable contributors**: Gone through the functionalities in OpenCV documentation where we found some of the free libraries that the project makes use of.

**Abstract**

In this growing modern world, everyone tries to reduce their work either at work or at home. Our project creates a lucrative way of using the present resources to gesture recognition. Using the open source C/C++ library of OpenCV we trained and developed the model which recognizes hand gestures that implement actions like tracking, left click, right click and drag N drop. Mouse control based on Gesture interaction takes advantage of continuity and dynamics of the user's movement of the hand to control the mouse movement. The flow in the images is recorded in separate frames and separately recognized. The recognition involves using techniques like thresholding and binary masking on the region of interest. The data set is obtained from the previously trained models which showed potential to improve. The functions are preassigned and can only be customized altogether by training the model for specific applications again.

**Team Members**

Krishnaja - 20186028

Naveen - 20186068

Mahateja - 20186025

**TABLE OF CONTENTS**

**1. Introduction**

**1.1 Goal**

**1.2 Methodology**

**1.3 Constraints**

**2. Project Prerequisites**

**3. Additional Code Libraries Used**

**4. System Overview**

**5. GUI Components**

**6. Conclusion**

**7. Future works**

**8. References**

1. **Introduction**

The gesture recognition for mouse control requires Machine learning package called Open Computer Vision to recognize and perform operations. The Open Computer Vision was developed in 1999 at Intel by Gary Bradsky. The package helps in recognizing the human face and parts like eyes, nose, etc. Apart from just recognizing this package helps in operations like removing red eyes and adjusting the brightness in certain areas, reshaping any imperfections, etc. Importing the package we first load the data set of haar cascade frontal face default and perform operations. We obtain the region of interest by setting the x,y and h parameters. Once the image with ROI is known we flip the image. Since the mouse control needs to be concurrent with the user.

* 1. **Goals of the project**

The aim of overtaking this project is to provide a lucrative way of using the resources to make sure the work of the user becomes simpler.

Artificial intelligence project along with computer vision technology provides myriad uses for recognizing hand gestures and inturn we use these gestures to handle mouse events in OpenCV

* 1. **Methodology**

Initially

* 1. **Constraints**

As with all projects of this type, the major constraint is time. There is only a limited amount of time to finish the project and therefore only a limited amount of functionality can be implemented. This is something that we tried to take into account by dividing the requirements into core features and extended features. The core features are to be seen as the minimum specification and the extended features are to be implemented if time permits.

1. **Project requirements**

Programming language - Python

AI/ML Libraries - OpenCV(Open computer vision)

Packages - wxPython Wrapper Class, Numpy, Pynput

1. **Functionalities**

By tracking the number of fingers opened the mouse operation are performed. Tracking operation is done whenever asingle finger is opened and right-click operation is done by pinching two fingers close.

1. **Coding and implementation**

1. **Algorithm/Tools Used**

**5.1 Morphology transformation:**

Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images. It needs two inputs, one is our original image, second one is called structuring element or kernel which decides the nature of operation. Two basic morphological operators are Erosion and Dilation. Then its variant forms like Opening, Closing, Gradient etc also comes into play.

**5.1.1 Erosion and Dilation:**

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion.

**5.1.2 Opening:**

Opening is just another name of erosion followed by dilation. It is useful in removing noise, as we explained above.

**5.1.3 Closing:**

Closing is reverse of Opening, Dilation followed by Erosion. It is useful in closing small holes inside the foreground objects, or small black points on the object.

1. **How different technologies are used**
2. **UML diagrams**
3. **Project Screenshots**
4. **Output**
5. **Conclusions**

The aim of the project is to utilize the preinstalled resources in our PC to produce a tool/software that provides a lucrative way for users to control the mouse operation at their very own fingertips.

The overview of the project involves the user giving a pre-modeled gesture that can be recognized and perform appropriate operation. All the features mentioned in the functional requirements work.

The only downfall is due to lack of time the modeled gesture accuracy is very low than expected. From our side, a considerable effort has been made to refine the model, to its utmost efficiency.

The software tool developed needs no external authorization license to work in other systems, i.e., our tool is platform independent and is open source,i.e., not for profit applications. To sum up there are many opportunities to develop and extend new features like shut down, sleep, etc.

1. **Future improvements**

As specified already functionalities like shutdown, sleep and drag N drop can be performed by improving the model to recognize additional functionalities.

1. **References**
2. [**http://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid699113.pdf**](http://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid699113.pdf)

**Github:**[**https://gogul09.github.io/software/hand-gesture-recognition-p1**](https://gogul09.github.io/software/hand-gesture-recognition-p1)

[**https://gogul09.github.io/software/hand-gesture-recognition-p2**](https://gogul09.github.io/software/hand-gesture-recognition-p1)

**Medium:**[**https://medium.com/@muehler.v/simple-hand-gesture-recognition-using-opencv-and-javascript-eb3d6ced28a0**](https://gogul09.github.io/software/hand-gesture-recognition-p1)