



GUI CONTROL WITH HEAD GESTURE

A PROJECT REPORT

Submitted by

REAJUL HAQUE REAYZ

ROLL No: 1208017

PRASHANTA KUMAR DEY

ROLL No: 1208019

in partial fulfillment for the award of the degree

of

BACHELOR OF SCIENCE (ENGG.)

in

COMPUTER SCIENCE AND ENGINEERING

COMILLA UNIVERSITY:: COMILLA-3506

AUGUST 2018

COMILLA UNIVERSITY:: COMILLA-3506

BONAFIDE CERTIFICATE

Certified that this project Report “**GUI CONTROL WITH HEAD GESTURE**” is the bona-fide work of “**REAJUL HAQUE REAYZ & PRASHANTA KUMAR DEY**” who carried out the project work under my supervision.

SIGNATURE

MAHMUDUL HASAN

**CHAIRMAN OF THE EXAM
COMMITTEE**

Department of Computer Science and
Engineering

Comilla University, Comilla

SIGNATURE

MAHMUDUL HASAN

SUPERVISOR

Assistant Professor

Department of Computer Science
and Engineering

Comilla University, Comilla

ABSTRACT

This system can be adapted accurately for detecting facial features. It is a free line up that allows a physically challenged user to control the pointer of the mouse on a processor with the help of head movements. The movements of head are liberated clicking with the help of mouse pointer abide over a spot on the screen. This is application which is helpful for the people who are disabled. The foremost spectators for this program are people who do not have committed control of a hand but who can only move their head.

In this project, we have used the Haar-like features to detect faces with OpenCV and implement those using python. Only the frontal faces can be detected. The face detection is done in three phases with the algorithm specified by Paul Viola and Michael J. Jones. We get some of non- faces which are classified as faces along with the faces. We tried to reduce those non-faces from the image and keep only faces. After face detection we control mouse movement with the coordinate of face return by the first phase. For mouse control we use pynput package of python. It helps to control the mouse cursor, also perform some activity, such as left/right/double clicking and changing mouse position.

This system is based on computer vision algorithm. Most vision algorithms have illumination issues. From the results, we can expect that if the vision algorithms can work in all environments then our system will work more efficiently. However, it is difficult to get stable results because of the variety of lighting and skin colors of human races.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to **MAHMUDUL HASAN**, our project supervisor for his patient guidance, valuable and constructive suggestions during the design and implementation of this project. His willingness to give his time so generously has been much appreciated.

Our grateful thanks are also extended to the faculties and supporting staffs at the department of Computer Science and Engineering for their help in offering us the resources in running the program. We would also like to extend our thanks to our friends for their opinions, useful critiques of this work and offering us the test data.

Finally, we wish to thank our family for their support and encouragement throughout our study and the Almighty God.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iii
	ACKNOWLEDGEMENT	iv
	LIST OF TABLE	vii
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	
	1.1 Background of the study	1
	1.2 Aims and objectives	2
	1.3 Overview of the system	2
	1.4 Limitation of the existing system	3
	1.5 Project schedule	3
	1.6 Chapter summary	3
2	LITERATURE REVIEW	
	2.1 Introduction	4
	2.2 History of face detection	4
	2.3 Face detection method	5
	2.3.1 Viola jones method	5
	2.3.2 Gabor feature method	5
	2.3.3 Constellation method	6
	2.4 Chapter summary	6
3	ANALYSIS AND DESIGN	
	3.1 Introduction	7
	3.2 Viola jones object detection framework	7
	3.2.1 Haar-like features	10

	3.2.2 Cascaded classifier	10
	3.3 Process of face detection	11
	3.4 Tools	12
	3.4.1 Python	12
	3.4.2 OpenCV	13
	3.5 System specification	14
	3.5.1 Hardware specification	14
	3.5.2 Software specification	14
	3.6 General overview/Flowchart	15
	3.7 The proposed procedure	16
	3.8 Chapter summary	16
4	IMPLEMENTATION AND RESULT DISCUSSION	
	4.1 Installation	17
	4.2 User interface of the system	17
	4.3 Face detection	18
	4.4 Controlling mouse	19
	4.5 Perform mouse operation	20
	4.5.1 Left click	20
	4.5.2 Right click	21
	4.5.3 Double click	22
	4.6 Result/Comparison	23
	4.7 Chapter summary	24
5	CONCLUSION	
	5.1 Conclusion	25
	5.2 Limitation of our system	26
	5.3 Future Work	26

LIST OF TABLES

<i>Table 1.1</i>	<i>Project schedule</i>	<i>3</i>
<i>Table 3.1</i>	<i>Hardware specification</i>	<i>14</i>
<i>Table 4.1</i>	<i>Accuracy rate</i>	<i>23</i>

LIST OF FIGURES

<i>Figure 3.1</i>	<i>Haar-like features with different sizes and orientation</i>	<i>8</i>
<i>Figure 3.2</i>	<i>How the Haar-like feature of figure 3-1 can be used to scale the eyes</i>	<i>8</i>
<i>Figure 3.3</i>	<i>Several classifiers combined to enhance face detection</i>	<i>10</i>
<i>Figure 3.4</i>	<i>Several figures represents a general feature of a human face</i>	<i>11</i>
<i>Figure 3.5</i>	<i>Combine figures represents a general feature of a human face</i>	<i>11</i>
<i>Figure 3.6</i>	<i>Combine features used to detect a face</i>	<i>12</i>
<i>Figure 3.7</i>	<i>Flowchart of the system</i>	<i>15</i>
<i>Figure 4.1</i>	<i>User interface of our system</i>	<i>17</i>
<i>Figure 4.2</i>	<i>Face detection by our system</i>	<i>18</i>
<i>Figure 4.3</i>	<i>Controlling mouse cursor in our system</i>	<i>19</i>
<i>Figure 4.4</i>	<i>Perform left click</i>	<i>20</i>
<i>Figure 4.5</i>	<i>Perform right click</i>	<i>21</i>
<i>Figure 4.6</i>	<i>Perform double click</i>	<i>22</i>
<i>Figure 4.7</i>	<i>Accuracy level</i>	<i>24</i>

LIST OF ABBREVIATIONS

Short Name	Abbreviations
CCTV	Closed Circuit Television
GUI	Graphical User Interface
NN	Neural Network
OpenCV	Open Source Computer Vision
OS	Operating System
PC	Personal Computer
PCA	Principle Component Analysis
PIP	Python Installation Package
RGB	RGB color model