



# **iJRASET**

International Journal For Research in  
Applied Science and Engineering Technology



---

# **INTERNATIONAL JOURNAL FOR RESEARCH**

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume: 10    Issue: VI    Month of publication: June 2022**

**DOI: <https://doi.org/10.22214/ijraset.2022.44478>**

**[www.ijraset.com](http://www.ijraset.com)**

**Call:  08813907089**

**E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)**

# Human-Eye Controlled Virtual Mouse

Mr. Dhanaraju<sup>1</sup>, Dr. Sreenivas Mekala<sup>2</sup>, A. Harsha Vardhan Rao<sup>3</sup>, CH. Pavan Kumar<sup>4</sup>, R. Lokesh<sup>5</sup>

Associate Professors, Dept of IT, Sreenidhi Institute of Science and Technology(A), Hyderabad.

B.Tech Students, Dept of IT, Sreenidhi Institute of Science and Technology(A), Hyderabad.

**Abstract:** *There's always room for advancement in the realm of computers, thanks to new technology in our digital era. Hands-free computing is popular nowadays because it caters to the needs of quadriplegics (people suffering from paralysis of all four limbs). We want to demonstrate a Human-Computer Interaction (HCI) system that is essential for amputees and others who have trouble using their hands. The system is a mouse-like eye-based interface that converts eye movements like blinking, staring, and squinting into mouse cursor actions. The software requirements for this technique include Python, OpenCV, NumPy, and a few more facial recognition tools, as well as a basic camera. In the creation of face detectors, the HOG (Histogram of Oriented Gradients) is often utilised. and hence the window technique, which combines this feature with a linear classifier. It is completely hands-free and does not require any additional hardware or sensors.*

## I. INTRODUCTION

Moving the pointer along with the screen using a computer mouse or by moving one's finger has become fairly common in today's technology. Every movement of the mouse or finger is detected and mapped to the movement of the pointer by the system. Because their arms are not functioning, certain people, known as "amputees," will be unable to use present technology to use the mouse. If the amputee's eyeball and facial features, as well as the direction in which their eye is staring, can be recorded, the movement of the facial features may be transferred to the cursor, allowing the amputee to move the cursor at whim. An 'eye-tracking mouse' is a gadget that tracks the user's eye movements. The project relies on mapping facial traits to the cursor to recognize and capture them in video. When the camera is opened, the application must extract all of the video's frames. Since the video's frame rate is typically around 30 frames per second, every frame will be processed in about 1/30th of a second. The application then goes through a series of steps to identify and map the characteristics of the video to the point. After the frame has been retrieved, the face areas must be identified. As a result, the frames will go through a set of image-processing routines to appropriately analyse the frame, allowing the algorithm to distinguish things like eyes, mouths, and noses. The Overview Our Project performs the following functions or actions which are as follows.

- Opening the Mouth
- Right Eye Wink
- Left Eye Wink
- Squinting Eyes
- Head Movements (Pitch and Yaw)

### A. Objective of the Project

The objective of our project is to make the work of 'amputees' (people who don't have their arms to be operational) easy. Amputees or quadriplegics can benefit from our project (people affected by paralysis of all four limbs) can use and operate the mouse using their facial features and actions of their eyes.

To deliver a user-friendly human-computer interaction project, this project will design a system that will just require a camera to employ human eyes and facial characteristics as a pointing device for the computer system. The following are the objectives:

- Eyes and face Detection
- Eye end points extraction
- Develop an algorithm to calculate the point gaze based on eye features extracted

- Develop a GUI that shows a result
- Develop a Calibration technique

## II. DESIGN OF THE SYSTEM

### A. System Design

The process of building a system's architecture, components, modules, interfaces, and data to meet specified criteria is known as system design. In a word, it is when systems theory is applied to product development. Object-oriented design and analysis processes are progressively gaining popularity as the most frequent methodology for designing computer systems.

### B. Architectural Design

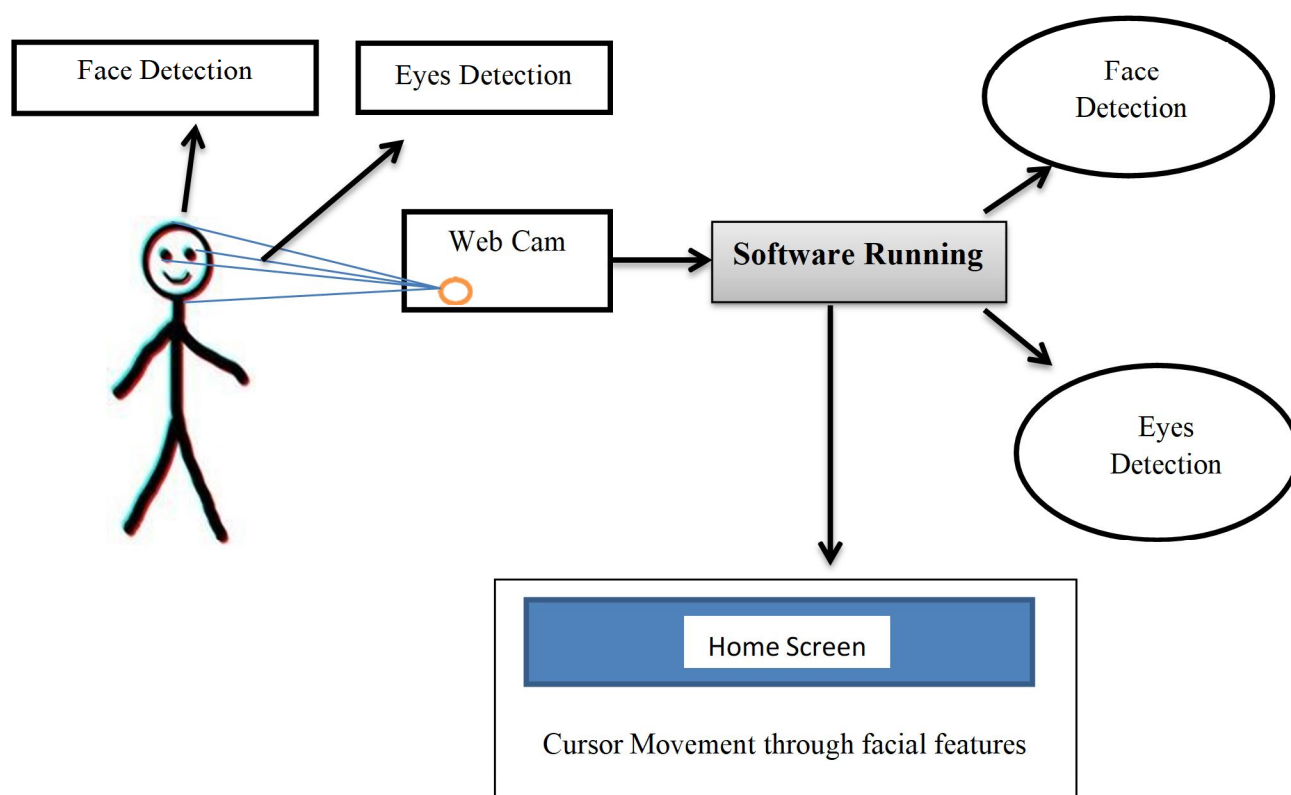


Figure 2.2. Architectural Design

## III. METHODOLOGY

### A. Implementation

- 1) *Language or Technology Used:* Python language is used to write the code. Python provides a wide variety of libraries for scientific and computational usage. Libraries such as hashlib, rsa
- 2) *Algorithms Implemented:* Deep learning models may currently be the most effective for face identification. Face detection, on the other hand, existed before deep learning. Previously, traditional feature descriptors and linear classifiers were a great way to recognize faces. HOG and Linear SVM, to be precise. The HOG (Histogram of Oriented Gradients) feature descriptor and a Linear SVM machine learning approach are used to identify faces. HOG is an easy-to-understand and useful feature description. It is frequently employed in the detection of objects, such as vehicles, dogs, and fruits, in addition to face detection. Because the local intensity is used to characterize the geometry of the item, HOG is reliable for object detection.



**Step 1:** HOG's primary concept is to divide a picture into small linked cells

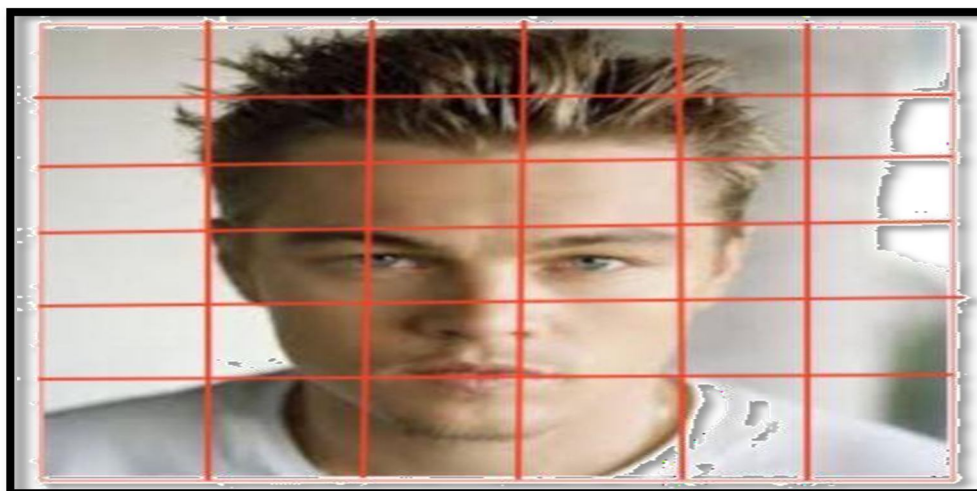


Figure 3.1. Dividing an image into cells

**Step 2:** Computes the histogram for each cell.



Figure 3.2. Cell to Histogram

**Step 3:** Bring all of the histograms together to create a feature vector, which creates a single histogram from all of the tiny histograms that are unique to each face.

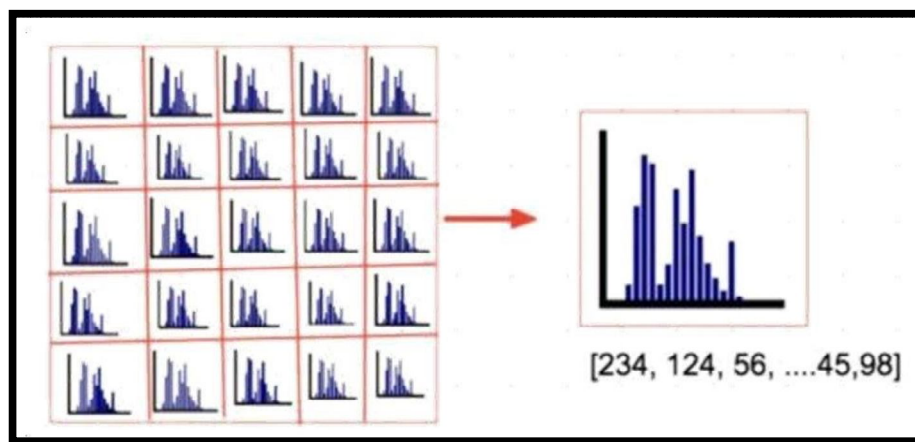


Figure 3.2. Histogram to feature vector

#### a) SVM Algorithm

Support Vector Machine, also known as Support Vector Classification, is a supervised and linear Machine Learning technique for dealing with classification issues. Support Vector Regression (SVR) is a subset of Support Vector Machines (SVM) that solves regression issues using the same ideas.

#### b) Implementing SVM in Python

With Scikit Learn's SVM package, the implementation is quite easy and basic.

##### - Importing Dataset

```
import pandas as pd
data = pd.read_csv("dataset.csv")
```

##### - Creating Training and Test Samples from the Dataset

```
from sklearn.model_selection import train_test_split
training_set, test_set = train_test_split(data, test_size = 0.2, random_state = 1)
```

##### - Classifying the Predictors and Determining the Target

```
X_train = training_set.iloc[:,0:2].values
Y_train = training_set.iloc[:,2].values
X_test = test_set.iloc[:,0:2].values
Y_test = test_set.iloc[:,2].values
```

### 1. Setting Up Support Fitting using the training data using VectorMachine

```
from sklearn.svm import SVC
```

```
classifier = SVC(kernel='rbf', random_state = 1)
```

```
classifier.fit(X_train,Y_train)
```

### 2. Predicting classes for a test set

```
Y_pred = classifier.predict (X_test)
```

### 3. Attaching the predictions to the test set for comparing

```
test_set["Predictions"] = Y_pred
```

### 4. Calculating the accuracy of the predictions

```
from sklearn.metrics import confusion_matrix
```

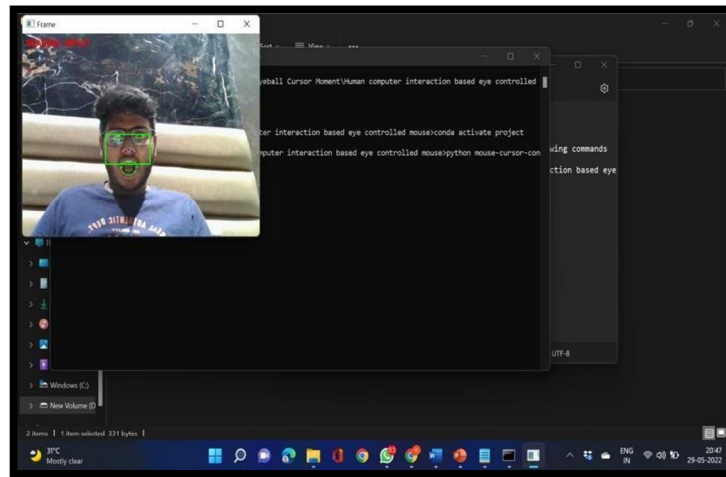
```
cm = confusion_matrix(Y_test,Y_pred)
```

```
accuracy = float(cm.diagonal().sum())/len(Y_test)
```

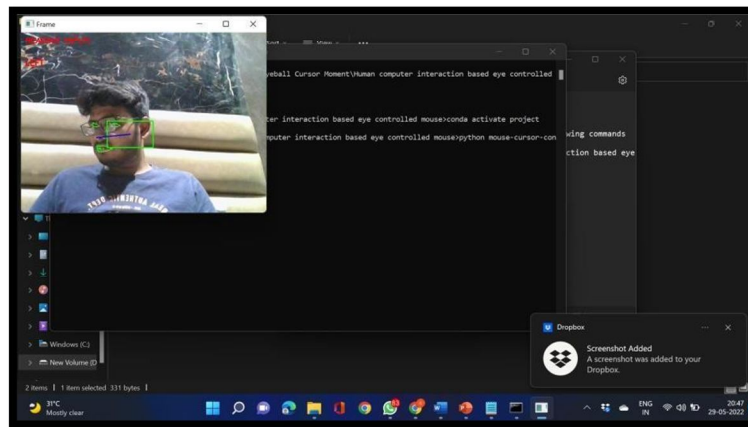
```
print("Accuracy Of SVM For The Given Dataset : ", accuracy)
```

Output: Accuracy Of SVM For The Given Dataset : 0.875

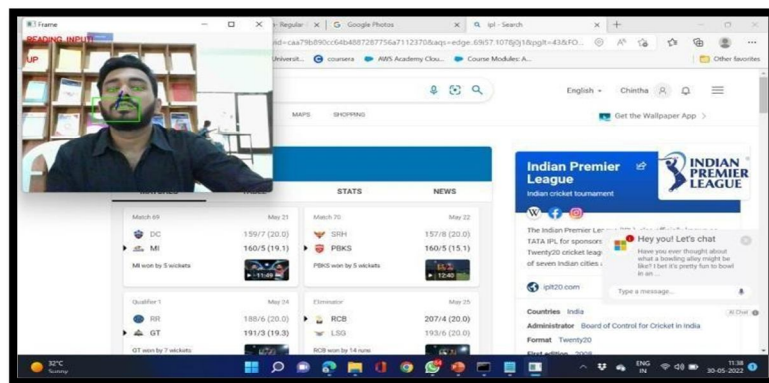
#### IV. RESULTS AND ANALYSIS



To move the pointer in the desired direction, the user must point his nose towards the left and right of a rectangle. The pointer moves quicker the more he moves away from the rectangle.

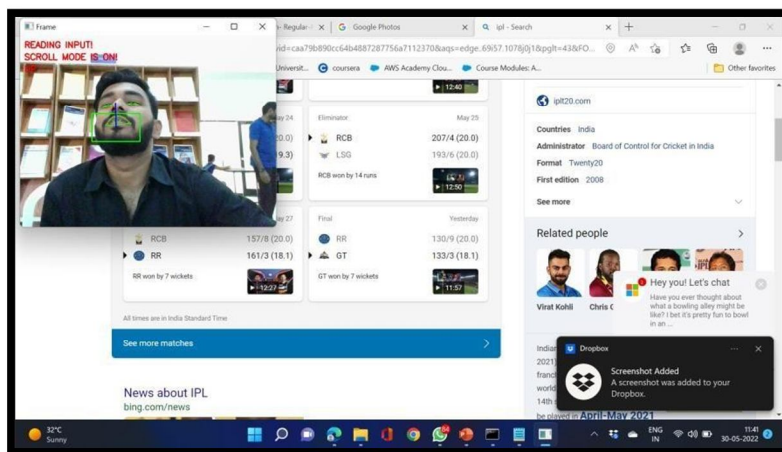


To move the pointer in the desired direction, the user must point his nose on the top and down of a rectangle. The pointer moves quicker the more he moves away from the rectangle.





The mouse can be scrolled upwards or downwards by the user. He must squint his eyes so that both of their aspect ratios are less than the necessary value. Instead of moving the pointer, the mouse performs a scrolling function if the user nose is outside the rectangle. He may scroll higher by moving his nose above the rectangle or downwards by moving his nose below the rectangle.



## V. CONCLUSION AND FUTURE SCOPE

The works may be enhanced to boost the system's speed by using better-trained models. Furthermore, the computer may be made more dynamic by making the pointer position change proportionate to the degree of rotation of the user's head, allowing the user to choose how quickly the cursor position changes. Given that the range of values is a function of the aspect ratios, which are typically minuscule, future research might focus on enhancing the ratio's precision. As a result, it's possible that some adjustments to the formulae for the aspect ratios used will be required to increase the algorithm's detection accuracy. Certain image processing techniques may be used before the model recognises the face and its attributes in order to speed up the face recognition process.

Many persons who are unable to use a normal computer mouse or keyboard due to hand or arm limitations may be able to benefit from a multimodal system, which allows them to manage a computer without utilizing a standard mouse or keyboard in the future. Head motions are used to move the pointer across the screen, while speech is used to give control commands.

## REFERENCES

- [1] Alex Poole and Linden J. Ball, "Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects," in Encyclopedia of Human Computer Interaction (30 December 2005) Key: citeulike:3431568, 2006, pp. 211-219.
- [2] D. H. Yoo, J. H. Kim, B. R. Lee, and M. J. Chung, "Non-contact Eye Gaze Tracking System by Mapping of Corneal Reflections," in Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR02), 2002, pp. 94-99.
- [3] Rafael Barea, Luciano Boquete, Manuel Mazo, and Elena Lpez, "System for assisted mobility using eye movements based on electrooculography," IEEE TRANSACTIONS ON NEURAL SYSTEMS
- [4] H. Singh and J. Singh, "A Review on Electrooculography," International Journal of Advanced Engineering Technology, vol. III, no. IV, 2012.
- [5] P Ballard and George C. Stockman, "Computer operation via face orientation," in Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on, 1992, pp. 407-410.
- [6] T. Horprasert, Y. Yacoob, and L.S. Davis, "Computing 3-D head orientation from a monocular image sequence," in Second International Conference on Automatic Face and Gesture Recognition, 1996, pp. 242- 247.





10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)