Impact Factor (SJIF): 5.301

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 8, Issue 2, February-2021

Automating Youtube using blink detection and gesture control

NIDHI M
Manipal Institute of Technology, Manipal, India
nidhim214@gmail.com

Abstract

Machine learning is a process of studying computer algorithms and training the computer system to improve automatically through experience i.e feeding new sets of data to the machine. This technology is being widely used all throughout the world and is becoming increasingly pertinent considering the massive advancement in the field of Computer Science especially Artificial Intelligence also increasing our dependence on such technology as well as the impact on society. With the breakthrough of Human Computer Interaction we have witnessed a huge leap in mankind interacting with and integrating technology in everyday lives to ease and increase the quality of living. We see how every aspect of human life is being automated thus reducing the efforts required to complete a given task and yet obtaining the exact same result. This project aims to automate a widely known and used software - Youtube. This paper mainly uses two applications of Human Computer Interaction namely eye blink detection and hand gesture control to automate Youtube using a simple Web Camera.

Keywords: automation of Youtube, Human Computer Interaction, blink detection, gesture control, limitations.

1. Introduction

The past few years have been revolutionary in the field of Artificial Intelligence and Machine Learning. We have witnessed several breakthroughs such as imitation learning,assistive and medical tech, automatic translation, recognition of emotions and more. We have been spending an incredible amount of our time on our electronic devices and thus they have penetrated to become a very important aspect of human life. We've let machines control whom we communicate with and how we socialise in a community. The internet has grown to become the largest community mankind has ever witnessed and has been growing ever since letting people explore, learn, and innovate. This innovation has led to special techniques that we use today that make human life simpler. Smart home automation is an example of how humans have successfully managed to integrate technology with everyday activities to significantly ease our lives. Humans have figured out various techniques to integrate and interact with machines in an easy and innovative way leaving behind our traditional methods of either having to walk up to a switch to turn off lights! This leads us to voice recognition systems in Apple devices which turned out to be one of the most innovative solutions of the time. The emergence of Human Computer Interaction can be considered as a revolution in the field of technology allowing easy, user friendly interaction and highly responsive communication between any user and a computer device. HCI has grown from early computers such as ENIAC to increase computing power to Ubiquitous Computing - a sensor based computing detecting and automating every aspect of human life. With such a heavy dependence and usage of this technology the importance and popularity of AI is only destined for success. People would aim to interact with the surrounding using basic actions such as recognising a person's voice or gesture or the touch of a person or simply through a blink. This project aims to take advantage of the same human tendency to either control their device either with a gesture or even as miniscule as that of a blink of an eye!

This project uses blink detection and hand gesture detection to control specifications such as playback speed, playing or pausing the video or even muting and unmuting the video. This is made possible with a simple web camera which is available on all computer devices making this project highly applicable. This enables us to control a software in a much more natural and hands-free way without having to go through the hassle of using a mouse or even a keyboard.

2. Technical Overview

The flow chart represented below gives a brief overview regarding the flow of the application.

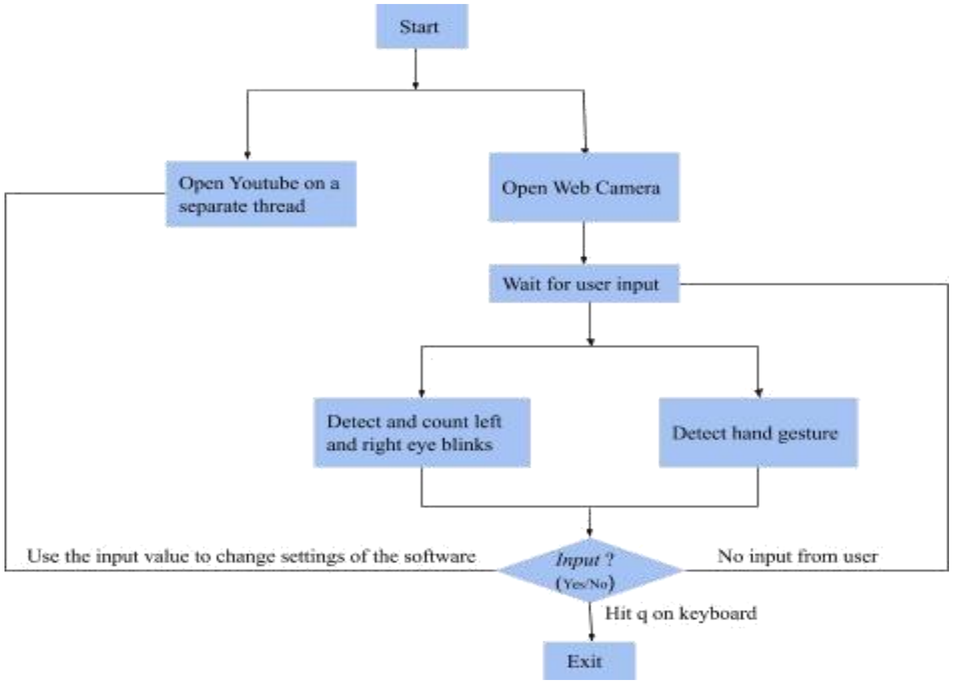

Start
Open Youtube on a separate thread
Open Web Camera
Wait for user input
Detect and count left and right eye blinks
Detect hand gesture
Use the input value to change settings of the software
Input ? (Yes/No)
No input from user
Hit q on keyboard
Exit

Fig 1 System flow chart

As seen from the above flowchart, initially a thread is created for Youtube software launch and handling all related settings and the rest of the code runs on the main thread. On the main thread the web camera is given access to accept user input. Here the user input is divided into two categories - a blink or a gesture . When an input is received in the form of a left blink or a right blink, this data is used to control the playback speed of the Youtube video playing. A single left blink will reduce the speed whereas a single right blink will increase the speed accordingly. If the user input is a hand gesture this will automatically either play/ pause/ mute the video based on the respective hand gesture. At any point the user can choose to exit the detection process by simply pressing the ‘q’ key on the keyboard and the system can no longer detect a blink or a gesture. The script will need to be run again to enable the same.

3. Eye blink detection

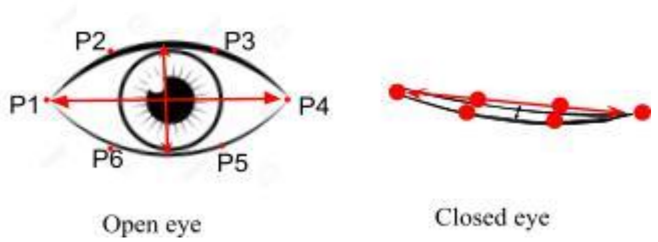


Fig 2 Eye landmarks

In the process of detecting a blink we focus on 6 coordinates namely P1, P2, P3, P4, P5, P6 as represented in the figure below.

We then derive an equation that represented the eye aspect ratio (EAR) given by:

$$\text{EAR}= \frac{|P2-P6|+|P3-P5|}{2|P1-P4|}$$

From the figure we observe that the vertical distance between the eye goes close to zero when the eye is closed. We take advantage of this property to detect a blink. Also we've made use of a pre trained face detector that is defined in the dlib library so as to detect the eye hull. This pre face detector marks 68 points on the face corresponding to each feature of the face. We then extract the landmarks of the left and right eye and use these values to determine the shape of a human eye. We use a built in function called shape defined within the imutils library and provide the value of landmarks as indices to the function to extract the shape of each eye.

A threshold value of 0.3 corresponds to an open eye and when an eye blink is detected the value of EAR drops to approximately zero.

If the value of EAR remains less that 0.3 for more than 5 consecutive frames then it will be considered as a blink. We use two variables left_blinks and right_blinks to store the same values. Both these variables are global so as to be able to access the blink value in the parallel thread. When the left eye blinks the playback speed of the youtube video is decreased and on the right eye blink the speed will increase accordingly.

The figure given below gives a simplistic overview of the entire process explained above.

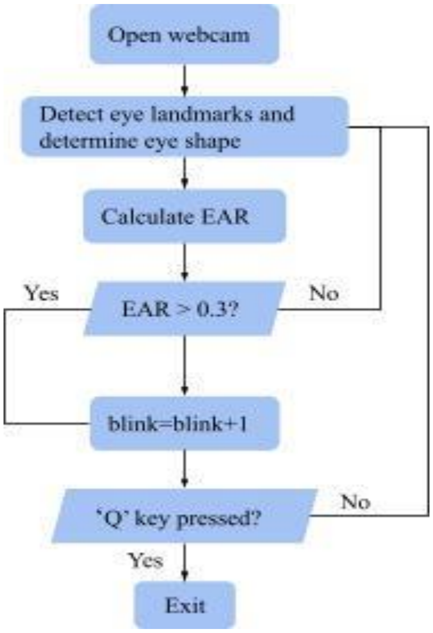


Fig 3 Process of blink detection

4. Gesture detection

This application uses a simplistic approach towards hand gesture detection. The basic functionality is to recognize if there is a hand within the region of interest and to detect the gesture being detected. In this project we place emphasis on three gestures- each having its own functionality.

4.1 Convex hull method

Firstly we define a region of interest - a smaller area part of the entire frame which is dedicated to detect hand gestures. Further the frame is converted to a HSV scale so as to ease the detection process by eliminating any noise and thus enhancing the accuracy of detection.

We define an upper and lower limit on color which represents the skin color which can be modified accordingly. Thus the frame focuses mainly on the color i.e skin color bound by these limits. Now that we have our image in HSV we blur the image and find the largest contour so as to avoid smaller contoured areas which might be present due to noise. The largest contoured area will be occupied by the hand.

Now we use the convex hull method which extracts the outline of the hand and identifies the convexity defects. Fig 4, 5, 6 shows the outline in the form on a green line which represents the convex hull for various gestures. The convexity defect is a cavity observed in the image (as shown in Fig 6). Any deviation of the object from our convex hull can be considered as a defect. For example in a hand we have 5 convex points and 4 defects between adjacent fingers. Thus using this functionality we can identify the number of fingers the user is showing by counting the defects.

Now we define a variable called area ratio to detect differences between gestures with the same number of convexity defects. The area ratio is simply defined as the area that is covered by the convex hull relative to the area of contour.

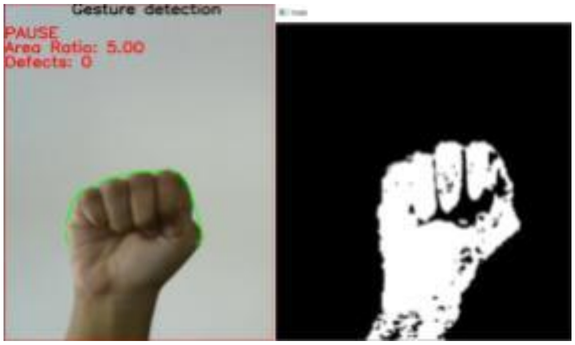
$$\text{Area ratio=} \frac{Area\ of\ hull - Area\ of\ contour}{Area\ of\ contour} * 100$$

We use this value to identify how much area is occupied by each gesture corresponding to the same number of defects and based on the area covered we can differentiate between different gestures. For example, the number three and the gesture “OK” have the same number of defects. But the area ratio of “OK” lies within a lesser range than the gesture showing number 3. Thus depending on the area we can detect either of the gestures.

4.2 Gesture Control

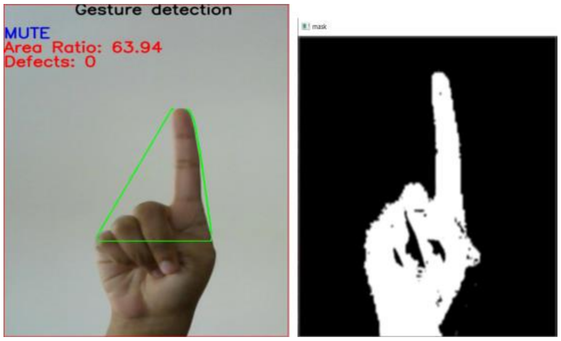
As mentioned above we use three main gestures to control three aspects of the Youtube aspect - Play , Pause , Mute.

Initially when the frame is empty this corresponds to an area ratio of above 1500 thus indicating no gesture by the user. Now if the number of defects is equal to zero we define two gestures - pause and mute. A closed fist corresponds to a pause and a gesture showing the index finger corresponds to mute. A closed fist is associated with a smaller area ratio and a higher value is associated with a mute signal.


Gesture detection

PAUSE
Area Ratio: 5.00
Defects: 0

Fig 4 Gesture for pause

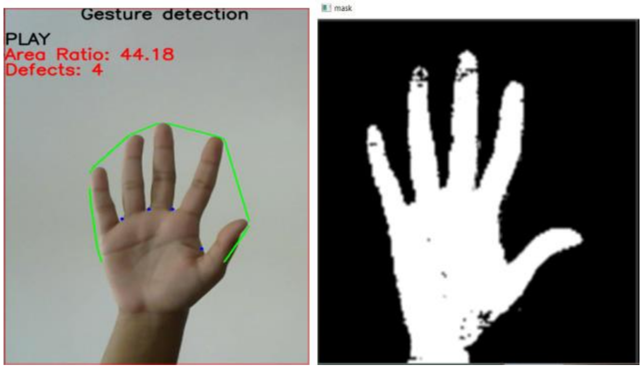

Gesture detection

MUTE
Area Ratio: 63.94
Defects: 0

Fig 5 Gesture for mute

From the above figures we can compare the area ratios for both the gestures. The pause has a ratio of around 5.00 whereas the mute has an area ratio of 63.94 which is much higher for the same value of defects in both the cases. Thus we successfully detected two gestures for different functionalities.

For the number of defects equal to 4 we define our last gesture which corresponds to playing a paused video or unmuting a muted video. We set a flag which indicates whether the video is muted or not. For every mute gesture we set the flag to true. If the number of defects is equal to 4 and the flag is set to true then the hand gesture unmutes the video and sets the flag to false else when the flag is false the gesture will play a paused video.


Gesture detection

PLAY
Area Ratio: 44.18
Defects: 4

Fig 6 Gesture for play/unmute

The blue points marked in the figure above are the convexity defects due to cavities between adjacent fingers which enables us to distinguish between the fingers.

5. Integration into Youtube

To integrate all the above functionalities into Youtube we use selenium which allows us to automate web browser interaction from Python. We create an instance of the Web Driver and thus use this instance to interact with the automated browser. The variables of blink and gesture control are incorporated into Youtube by javascript which is then executed by the in built function provided by the driver to run JS code.
Functionalities:

- For every left blink the speed is decreased by 0.25x and for every right blink the speed is increased by 0.25x.
- A gesture of the five fingers up with play a paused video or unmute a muted video
- A gesture of a closed fist will pause a running video
- A gesture of one finger up to mute a video accordingly.

6. Future Scope

This project was designed to ease human tasks in this case either using a mouse or a keyboard for most features associated with operating Youtube. We use basic techniques to implement the automation that is defined within the python library. The future of this project holds unlimited possibilities. The limitations of this project is bound by noise in a video stream or incorrect facial landmarks or fast changes in viewing angle which could report a blink even though it hasn't occurred in reality. A small cavity within the region of interest might show an incorrect value of convexity defect and this might deviate from the expected result. Thus we can use more sophisticated methods to implement the blink detection or gesture control which will result in increased accuracy of detection of the user input. The noise due to external factors can be reduced significantly to refine the code. We could use an existing dataset of gestures and then use a prediction model to predict the outcome of the new gesture provided as an input to the frame. The blink detection can be significantly improved computing the eye aspect ratio for the N-th frame along with EAR for N-6 and N+6 frames and then concatenating these eye aspect ratios to form a 13D feature vector or by training a Support Vector Machine as explained by Soukupová, Tereza and J. Cech [1].

We can improve hand gesture methods by using deep learning in neural networks. This can be obtained by performing finger segmentation and normalization of segmented finger image using CNN classifier [2].

6.1 Additional features

We can integrate swipe to control gestures using a higher level motion detection technique. We can swipe left or right in air to control forwarding and backwarding the video by the set time frame. We can also use swipe gestures to control the volume of the video.

7. Conclusion

It is very important for software developers to create projects with the goal as to not affect or detriment human knowledge or behaviour or effect the working of mankind in a negative manner. Any new idea has to be implemented with the sole purpose of nurturing and easing the work of mankind thus providing more room for research and breakthroughs.

The aim of Human Computer Interaction is to provide functional systems with efficient and effective usability. We try to understand how people use a well known software such as Youtube and thus automate it to reduce time involved in physical human interaction with the system and thus achieve the same results with fewer efforts. The project has much room for improvement and progress but it effectively implements the features to automate Youtube which was the original goal for the project. This paper provides an understanding of all the methods used of blink detection and gesture detection and its work flow using a simplistic approach and also yields a better solution for increasing accuracy and efficiency.

References

[1] Soukupová, Tereza and J. Cech. “Real-Time Eye Blink Detection using Facial Landmarks.” (2016)

[2] Neethu, P.S., Suguna, R. & Sathish, D. An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. *Soft Comput* 24, 15239–15248 (2020)