**SDP Project:-Media Control by Hand Gesture Recognition**

SDP Guide: Prof. Mrs. Shilpa Lambor

Shreyas Mendhekar

Department of Information Technology and MCA (SY)

***Abstract* —** *This project is mainly based on detecting the different hand positions (Hand gestures).After detecting that particular hand position set some media player controlling features like pause the video, increase or decrease the volume,fast-forward etc.*

***Keywords:*** *open CV, Convex hull,Conveixity*

*Defect,Threshhold,Gaussian blur,spyder console ,contour,cosine rule.*

# **INTRODUCTION**

Gestures are the movement of any body part used to convey the meaningful information.

Communication through gestures has been widely used by humans to express their thoughts and feelings. Human gesture is a mode of non - verbal interaction. Gestures recognition is the process of identifying gestures performed by human. Gestures has been classified in two categories static and dynamic. Static gestures refer to still body posture and dynamic refers to movement of body part. Gestures can be performed with any body part like head, face, arms, hands, etc.

The hand gestures are mainly used for the purpose of controlling the devices from a distance without the help of any intermediates like the remote. This is a direct interaction between the devices and the users. In general, the framework for hand gesture recognition includes taking in the input i.e. hand image, hand detection, preprocessing i.e. removing unwanted features, segmentation, feature extraction, recognition (gesture dictionary) and execution (commands).

# **LITERATURE REVIEW**

* Amit Gupta, et. Al [4]: hand gesture recognition technique which utilizes an FPGA based smart camera for gesture analysis. The experiment is performed using shape-based features. For performing the experiment images of 25 different persons is captured and is found that system is able to recognize 10 different gestures with accuracy 94.40%
* Jos´e Manuel Palacios[7]: developed hand gesture recognition technique using RGB-D sensors taking the advantage of depth information to remove the problems caused by lightning conditions and cluttered background. The proposed methodology includes four basic steps - Hand segmentation, Feature extraction, Static gesture classification and Dynamic gesture classification. For hand segmentation skin color segmentation and background subtraction is used. For static gesture recognition fingertip detection is used and fingertip is detected using maximum curvature and convexity defects. For dynamic, Euclidean distance and direction is used. The experiment is performed on dataset consisting of 90 images of 9 different persons with 10 dynamic and 6 static gestures and achieves Precision- 92.1% and Recall-83.3%.
* Chetan Dhule, et. Al[9]: proposed a vision based hand gesture recognition technique for human computer interaction. They proposed the method keeping in view that most the earlier methods are based on gesture recognition algorithms that require ANN training which is very time consuming and is not much accurate. So by using color detection techniques they develop real time application to restrict the mouse’s motion in windows by detecting change in pixel value of RGB colors and which is possible without ANN training
* Chang-Yi Kao. et. Al[3] : hand gesture recognition technique based on path of hand motion by using HMM as a classifier. Eight different kinds of gestures have been developed using either single hand or both hands

# **IMPLEMENTATION DETAILS**

**Category of the project** – Image Processing

**Tools / Platform** :–

* Anaconda Navigator.
* Spyder Console

**Front end-Backend Software Requirement –**

* Language used: Python 3.5.
* Libraries:- Open CV, numpy.
* Hardware:- Laptop’s Camera.

**Open CV installation :-**

In a command prompt console or you should type (a or b):

a) pip install file\_path/opencv\_python‑3.4.2‑cp36‑cp36m‑win\_amd64.whl

b) pip install file\_path/opencv\_python‑3.4.2+contrib‑cp36‑cp36m‑win\_amd64.whl

# **Methodology/working**

Method to detect hand gestures:-

1. **Access Laptop’s Camera**(By using the open cv function)
2. **Capture frames and convert to grayscale:-**

Our region of interest, is the the hand region. Images of the hand are captured and converted to grayscale. We convert an image from RGB to grayscale and then to binary in order to find the ROI i.e. the portion of the image we are further interested for image processing. By doing this our decision becomes binary: "yes the pixel is of interest" or "no the pixel is not of interest"

1. **Gaussian Blur:-**

Gaussian Blurring is used on the original image. We blur the image for smoothing and to reduce noise and details from the image. We are not interested in the details of the image but in the shape of the object to track. By blurring, we create smooth transition from one color to another and reduce the edge content. We use thresholding for image segmentation, to create binary images from grayscale images.

1. **Thresholding :-**

* In thresholding, each pixel value is compared with the threshold value. If the pixel value is smaller than the threshold, it is set to 0, otherwise, it is set to a maximum value (generally 255).
* used for separating an object considered as a foreground from its background.
* Otsu's Binarization method is used. In this method, OpenCV automatically calculates/approximates the threshold value of a bimodal image from its image histogram. But for optimal results, we may need a clear background in front of the webcam which sometimes may not be possible.

1. **Draw Contours :-**

When we join all the points on the boundary of an object, we get a contour. Typically, a specific contour area is related to the boundary pixels, having similar color and intensity. Whenever the intensity or color changes greatly, then almost always we get a new contour area starting from there.

* Why we use grey scale threshholding in contour detection?

For the proper detection of contours, we need to convert the image to a single-channel color format (like grayscale), and then apply binary thresholding. Applying binary thresholding makes the objects completely black and white. The objects of interest, and their border will be completely white having the same color intensity. This is actually required for the contour detection algorithm to work properly. It will detect the borders of the objects from the white pixels (and of course, similar intensity as well because every white pixel will have a value of 255). Note: the black pixels with the value of 0 will be perceived as background and ignored.

1. **Convex hull and Convexity defect:-**

The convex hull of a set of points in the euclidean space is the smallest convex set that contains all the set of given points. For example, when this set of points is a bounded subset of the plane, the convex hull can be visualized as the shape formed by a rubber band stretched around this set of points. Convex hull is drawn around the contour of the hand, such that all contour points are within the convex hull. This makes an envelope around the hand contour.

We now find the convex points and the defect points. The convex points are generally, the tip of the fingers. But there are other convex point too. So, we find convexity defects, which is the deepest point of deviation on the contour. By this we can find the number of fingers extended and then we can perform different functions according to the number of fingers extended.

1. **Hand gesture detection:-**

Now, we have to detect the different hand gestures, up tile 6th step we have our convex hull and contour.

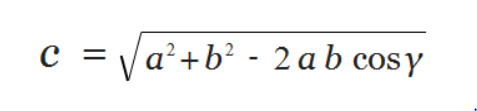
To, detect the hand gestures we have to use the cosine rule of geometry.because the output of the convexity defect function for different hand position is an array of

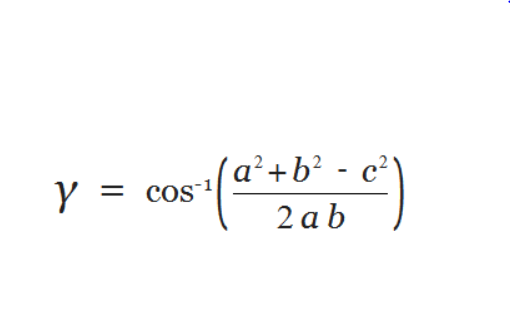
* **start point**
* **end point**
* **farthest point**
* **approximate distance to farthest point**

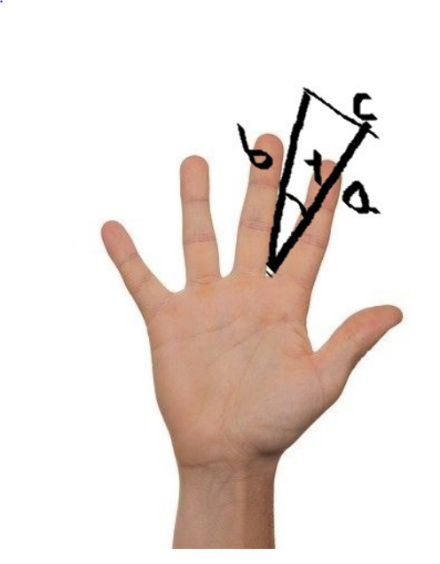
By, this point we can easily derive Sides: *a,b,c*(see CODE) and from cosine theorem we can also derive *gamma* or *angle between two finger.*As you read earlier, if gamma is less than 90 degree we treated it as a finger. After knowing gamma we just draw circle with radius 4 in **approximate distance to farthest point.**And after we just simple put text in images we represent finger counts (cnt).

Cosine rule:-

If we know the vales of a, b and angle b/w them then we can find value c & vise versa.







1. **Feature assigned to each gesture:-**

For assigning the feature to each gesture we used the pyautogui library.

Media contol features:-

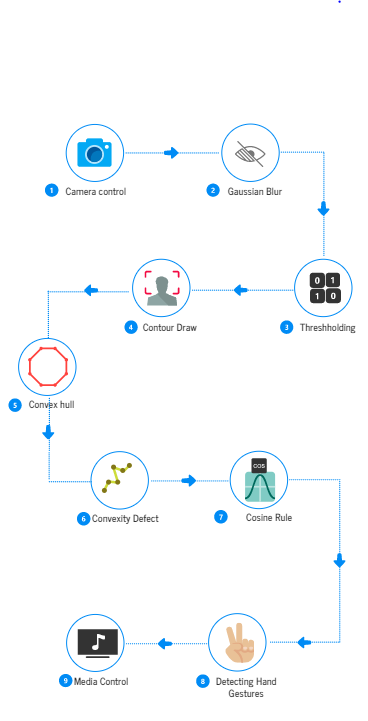
For count 2 - “Increase Volume”

For count 3 - “Decrease Volume”

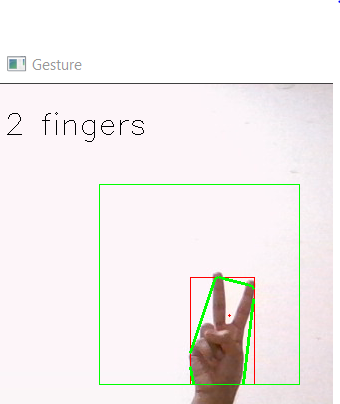
For count 4 - “Fast Forword”

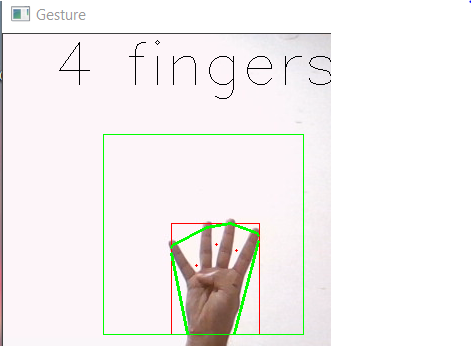
For count 5 - “Stop/pause”

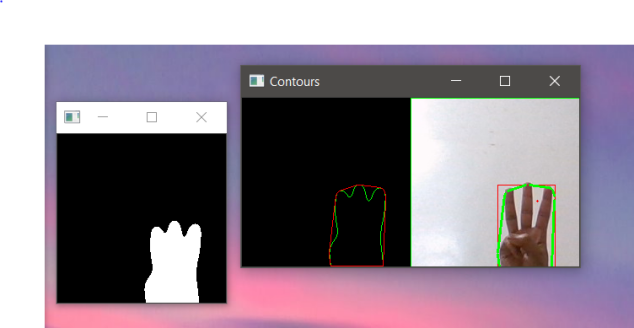
* ***Workflow Diagram:***



# **Application Output**







# **Future Scope**

We can use different techniques to detect different multiple hand gestures.We can modify this model and create the widget for any os ( so that we can easily control the different features of computer , laptop by just simple hand gestures).ANN(Artificial Neural Network ),supervised machine learning and 3d modelling are the latest technology could make this gestures detection more precise and featuring.

# **Conclusion**

Hand Gesture Recognition has number of applications like human computer interactions, robotics, sign language recognition, etc. Lots of work has been done hand gesture recognition field. We have discussed the basic methodology and various techniques of hand gesture recognition for media control. In the future, hand gesture can be combined with other gestures like body posture, face recognition, etc. for better results.

# **Acknowledgement**

Each and every one of us is overwhelmingly thankful to all the teachers and mentors who helped us create this project. Not only the guidance, but the overall learning experience from the professors helped us learn and understand a lot of things.

Our guide Prof. Shilpa Lambor Madam helped us through the thick and thins of our project, her opinions and guidance was the driving force to all of us. Our mentor Prof Prof. Shilpa Lambor madam’s suggestions and useful technical insights help us solve our problems whenever we got stuck, madam’s expertise in the field helped us learn a lot new things and bought us to successful completion of our project.

# **References**

1. Jing Lin, Yingchun Ding, “A temporal hand gesture recognition system based on hog and motion trajectory”, Optik 124 pp.6795– 6798, 2013.
2. Z. Ren, J. Yuan, J. Meng, and Z. Zhang, “Robust part-based hand gesture recognition using Kinect sensor,” IEEE Trans. Multimedia, vol. 15, no. 5, pp. 1110–1120, Aug. 2013.
3. Jos´e Manuel Palacios, Carlos Sagues , Eduardo Montijano and Sergio Llorente , “HumanComputer Interaction Based on Hand Gestures Using RGB-D Sensors” , Sensors 2013, vol.13, pp. 11842-11860, 2013.
4. <https://medium.com/analytics-vidhya/hand-detection-and-finger-counting-using-opencv-python-5b594704eb08>
5. [https://www.geeksforgeeks.org/opencv-python-tutorial/#:~:text=OpenCV%20is%20a%20huge%20open,the%20handwriting%20of%20a%20human.](https://www.geeksforgeeks.org/opencv-python-tutorial/" \l ":~:text=OpenCV%20is%20a%20huge%20open,the%20handwriting%20of%20a%20human.)