International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 11, November-2022

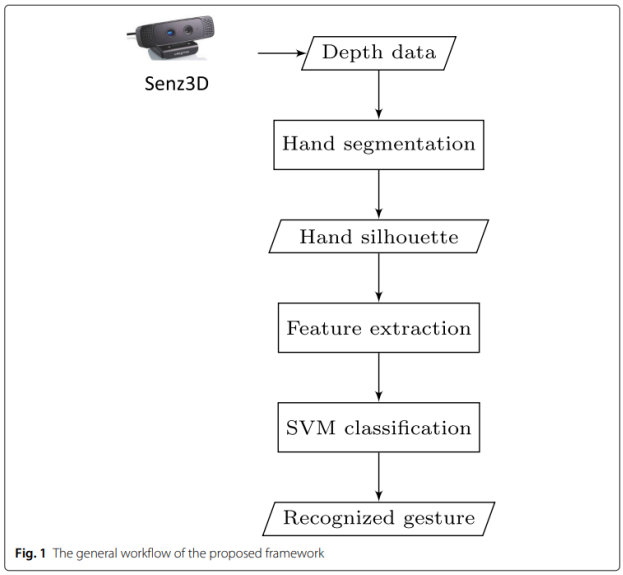


Fig. 1 The general workflow of the proposed framework

4. Gesture Controlled Media Player using TinyYoloV3 [4]
Author: Abhilash Dayanandan, Akshay Chakkungal, Anooj Kommeri, Deepak Koppuliparam bil, Dr. Prashant Nitnaware (2020)

YOLO is a fast and accurate algorithm. This algorithm can be used for gesture recognition and for training model by bringing balance between speed and accuracy. The goal of the proposed project is to develop a gesture-controlled media player that will enable us to use our hands to manage the computer-played movie. In this paper, instead relying just on image processing and machine learning, the YOLO object identification model can recognize a variety of hand gesture combinations more accurately thanks to deep learning and neural networks. There is a tradeoff between speed and accuracy which can be controlled. If higher accuracy is required, speed decreases and vice versa.

5. Static Hand Gesture Recognition Based on Convolutional Neural Networks [5]
Author: Raimundo F.Pinto, Carlos D. B. Borges, Antônio M. A. Almeida,and Iális C. Paula (2019)

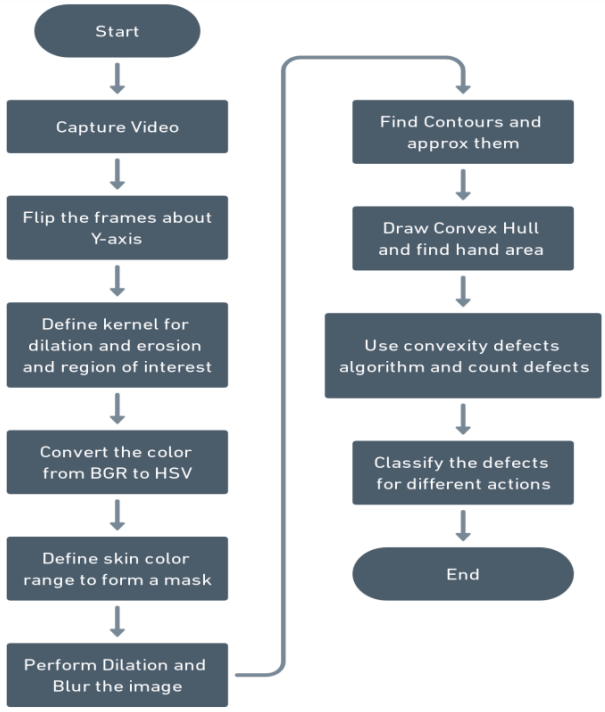
In this paper, a convolutional neural network-based technique for gesture recognition is proposed. In order to improve feature extraction, the approach applies morphological filters, contour generation, polygonal approximation, and segmentation during preprocessing. The images are used to train a CNN and assess the performance of the technique with cross validation. Finally, the validation results are analysed. The proposed method results were superior to other methods that use the same method of classification of gestures, the success rate of 96.83% shows the robustness of the presented methodology.

6. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition [6]
Author: Ali Moin, Andy Zhou, Abbas Rahimi, Alisha Menon, Simone Benatti, George Alexandrov, & Rabaey, J. M. (2021)

This paper describes the use of wearable devices that monitor muscle activity based on surface electromyography could be of use in the development of hand gesture recognition using ML. The system is implemented using a hyperdimensional computer technique with neural inspiration that is locally implemented for real-time gesture classification, model training, and updating under a variety of scenarios like changing arm postures and sensor replacement. When trained with a single trial for each gesture, the system can categorise 13 hand gestures for two participants with an accuracy of 97.12%. A drawback identified could be that the local processing cannot offer training and updating of the machine-learning model during use, resulting in suboptimal performance under practical conditions.

III. PROPOSED ARCHITECTURE

In the proposed architecture, the real-time hand gesture input from the user through the integrated webcam is used by the hand gesture recognition system on controlling media player, and it works when the user's inputted gesture matches one that can control the media player. The model offers the fundamental controls for the media player, including play, pause, volume adjustment. To develop the application, we employ a variety of Python tools and modules, including PyAutoGUI, OpenCV and subprocess. The architecture diagram of a proposed model is,



We have used PyAutoGUI to control the keys, map gestures to keys. i.e., to automatically press the keys according to conditions. The image frames are obtained from the video feed and converted from bgr image to hsv image using library function called, cvtcolor bgr2hsv. Crop image is used to divide the result window into gettrackbarpos gets the hsv values from trackbar. resize is used to make the screen small to show multiple tabs.

The mask is used ‘inRange ’where only the values within the range are displayed after applying the mask. The image

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 11, November-2022

with the mask is filtered using the bitwise image. It's better to have foreground as white and background as black in order to identify contours. hence, we use "not" operator to invert the background and foreground of the image.

Contours

Contours are defined as the line joining all the points along the boundary of an image that are having the same intensity. Contours come handy in shape analysis, finding the size of the object of interest, and object detection.

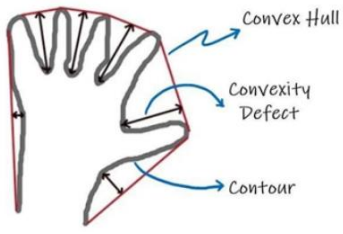
The contour of the hand in the region of interest is the set of points which correspond to the extremities of the human hand, which in turn define the hand's boundaries. The contour is then analyzed for the gesture and also approximated into a polygon. Canny edge detection method is opted to calculate the contour.

Hand Contour with OpenCV

Contours are found out using the 'findContours 'function. Using this we find out the convexity defect and convexity hull. The contour of the hand is a series of points which are the boundary pixels of the hand area. saved as a list. Find the contour with the largest area; if there is no contour, use try. The maximum contour arc length is determined, and the approxpolydp is then utilized to roughly determine the polygon (in this case, our hand) with the desired precision (which is epsilon). Now the red color line, which represents the contour, is used to create the convex hull. Draw the contour line around the hand in green and the convex hull in green on the figure.

Convexity Defects

Convexity defects are found out from the convexity hull, it is the farthest points from the convex points, i.e. if the finger tips are convex points, then the trough between fingers are the convexity defects. and these defects are counted. The angle between the fingers is found out using the cosine rule, so we understand the difference between index, middle, ring, little and thumb fingers.



Finding defects using Cosine Theorem

After detecting the convexity defects, we will get to know the number of fingers shown by the user then we can bind the count of convexity defect with the key to control the media.

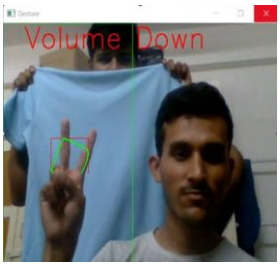
- Volume Up - 0 defect
- Volume Down - 1 defect
- Volume Mute - 2 defects
- Play/Pause – 3 defects
- Stop – 4 defects

IMPLEMENTATION RESULTS

The model is restricted to a plain background so that the model can detect the color of the skin based on the HSV values. After detecting the convexity defects, we will get to know the number of fingers shown by the user then we can bind the count of convexity defect with the key to control the media.

Video Capture:

A real time video is captured using OpenCV library and camera. The process involves selecting a rectangular portion of the captured image which is then filtered by a mask to extract the gray scaled binary image of the hand. For accurate tracking the user's hand the hue and saturation range should be selected appropriately according to the user. Tracking can be done by creating a filter of the hue and saturation range and applying it to the input image through bitwise-and operation.



Captured Image

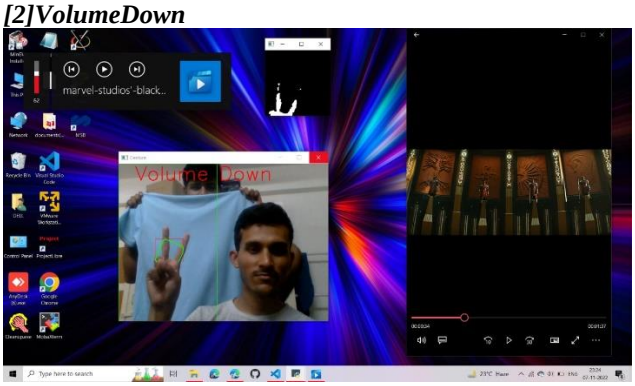
Masked Image

Resultant Outputs:

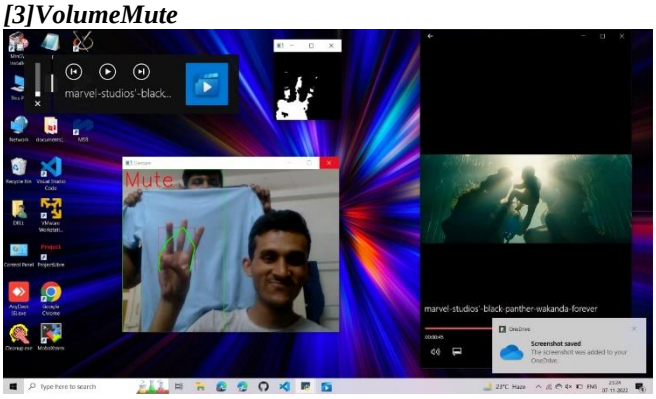
[1]VolumeUp

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 11, November-2022

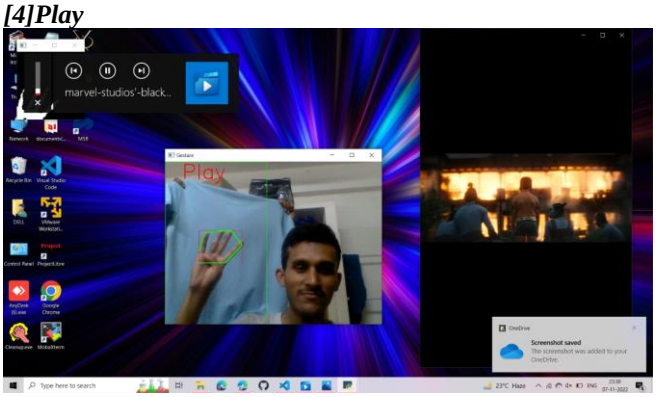
[2]VolumeDown



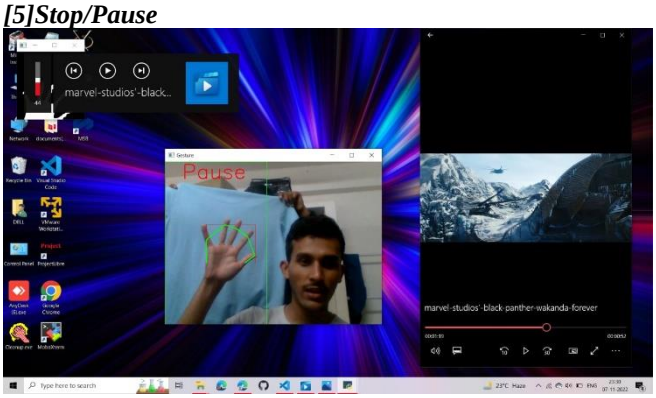
[3]VolumeMute



[4]Play



[5]Stop/Pause



The above-mentioned examples demonstrate how the created system functions well against a plain background. However, this also serves as a drawback. In order for the model to determine the color of the skin using the predetermined HSV values, a plain background is necessary. Similar-colored backgrounds might cause false detections, rendering the technique worthless. In the future, hand detection can be enhanced by using a stronger and more reliable technique. Such a system would depend on the other characteristics of a human hand in addition to the skin tone.

The finger detection mechanism is also affected if the hand is not properly detected. The contours and convexity defects are expected based on the shape of a human hand. If the hand detection is imperfect, it creates a domino effect, making the following steps unusable. A good hand detection mechanism ensures that the finger counting mechanism works correctly.

Following are some key observations and results of our project:

- The lower and upper bound of colour for the hand is critical in detecting the hand.
- Extracting contours was easier in the binary image which was obtained by binary thresholding.
- Sometimes the application fails to detect the hand gesture due to poor image quality.
- The implementation works well with plain backgrounds.

CONCLUSION

In this project, we performed hand and gesture recognition in python using OpenCV. Using the recognized gestures, we controlled various UI elements in the screen, hence making it easier for users to interact with the system. Hand recognition is based on the lower and upper bound of color, resulting in a binary threshold image. Contours and convexity defects are calculated, based on which the number of fingers is found out, which becomes the gesture. This is then used to control UI elements. This implementation works well with a suitable background. In the future, we can improve upon many aspects of the project. Some of them are:

- ➢ Adaptive thresholding to properly segment hand and background
- ➢ Removing the dependency on background colour to identify the hand
- ➢ Adding more UI control with the recognized gestures specific to a particular application
- ➢ Improving the gesture recognition, possibly by using machine and deep learning

REFERENCES

[1] Oudah M, Al-Naji A, Chahl J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. J Imaging. 2020 Jul 23;6(8):73. doi: 10.3390/jimaging6080073. PMID: 34460688; PMCID: PMC8321080.

[2] Karapinar Senturk, Z., & Bakay, M. S. (2021). Machine Learning Based Hand Gesture Recognition via EMG Data. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 10(2). https://doi.org/10.14201/ADCAIJ2021102123136

[3] Bakheet, S., Al-Hamadi, A. Robust hand gesture recognition using multiple shape-oriented visual cues. *J Image Video Proc.* **2021**, 26 (2021). https://doi.org/10.1186/s13640-021-00567-1

[4] Abhilash Dayanandan, Akshay Chakkungal, Anooj Kommeri, Deepak Kopuliparam bil, Dr. Prashant Nitnaware (May 2020). Gesture

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 11, November-2022

Controlled Media Player using TinyYoloV3. IRJET: International Research Journal of Engineering and Technology, 07(05). https://www.irjet.net/archives/V7/i5/IRJET-V7I5573.pdf

[5] Raimundo F. Pinto, Carlos D. B. Borges, Antônio M. A. Almeida, Iális C. Paula, "Static Hand Gesture Recognition Based on Convolutional Neural Networks", Journal of Electrical and Computer Engineering, vol. 2019, Article ID 4167890, 12 pages, 2019. https://doi.org/10.1155/2019/4167890

[6] Moin, A., Zhou, A., Rahimi, A., Menon, A., Benatti, S., Alexandrov, G., Tamakloe, S., Ting, J., Yamamoto, N., Khan, Y., Burghardt, F., Benini, L., Arias, A.C., & Rabaey, J.M. (2020). A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. Nature Electronics, 4, 54-63.

[7] Rubin Bose, S. and Sathiesh Kumar, V. 'In-situ Identification and Recognition of Multi-hand Gestures Using Optimized Deep Residual Network'. 1 Jan. 2021 : 6983 – 6997.

[8] Verdadero, M. S., Martinez-Ojeda, C. O., & Cruz, J. C. D. (2018). Hand Gesture Recognition System as an Alternative Interface for Remote Controlled Home Appliances. In 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-5). IEEE.

[9] https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/

[10] https://learnopencv.com/contour-detection-using-opencv-python-c/

[11] https://learnopencv.com/convex-hull-using-opencv-in-python-and-c/

[12] https://theailearner.com/2020/11/09/convexity-defects-opencv/