Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 07, July-2021

Implementation of Image Processing Tools for Real-Time Applications

Dr. Bapurao Bandgar
JSPM's Rajarshi Shahu College of Engineering
Dept. of MCA
Tathawade, Pune-33, India

Abstract: **Now a day the image processing is the one of the challenging task. There are number of the models present for the implementation of the same. Here I tried to implement the open source tools OpenCV and MediaPipe for the real-time applications of the image processing. The impletion is applied for the finger count with object detection. The result were obtained the with Good accuracy for the finger count.**

Index Term - OpenCV, Media Pipe, image Detection.

I.INTRODUCTION

The image processing has very much usage in the real-time applications. There are different models available for the image processing including Deep learning algorithms. Also there are different open source tools [2,3] available for the implementation of real time applications. The different functionality such as image processing, Video Analysis, 2D Feature Frame works, Object detection etc. [3] of the tools can be applied to the real time application.

With the latest advances in information and media technology[3,4], human computer interaction (HCI) systems that involve hand processing tasks such as finger detection and finger gesture recognition.

Finger detection is an interesting topic to explore in image processing, especially when it is applied in human-computer interaction. In this work we can detect the number of fingers in the video captured by a laptop camera.

Basically, the primary task is to detect hand within the video frame. This is the most challenging part. The proposed way is to use Background Subtraction and HSV Segmentation together to create a mask. After the hand is segmented, we'll detect the amount of fingers raised. There are 2 proposed methods. The first is to seek out the most important contour within the image which is assumed to be the hand. Then, we will find the convex hull and convexity defects which are most probably the space between fingers. This is a manual way of finding the amount of fingers. The second way is to use a convolutional neural network with the mask as input to work out the amount of fingers.

Here we focused on the how a system could detect, recognize and interpret the finger gesture recognition through computer vision with the challenging factors which variability in pose, orientation, location and scale. To perform the different types of gestures such as numbers and sign languages need to be created in this system.

II. TOOLS USED

During this work various tools have been used. Python may be a general-purpose, versatile, and powerful programing language. It's an excellent mother tongue because it's concise and straightforward to read. Whatever you want to do, Python can do it. From web development to machine learning to data science, Python is that the language for us. PyCharm is the best IDE. With PyCharm, we can access the command line, connect to a database, create a virtual environment, and manage your code.

OpenCV provides a real-time optimized Computer Vision library, tools, and hardware. It also supports model execution for Machine Learning (ML) and AI.
We look for the regions in images which have maximum variation when moved (by a small amount) in all regions around it. So finding these image features is named Feature Detection. So during this module, we are looking to different algorithms in OpenCV to seek out features, describe them, match them etc.

MediaPipe is a cross-platform framework for building multimodal applied machine learning pipelines. The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms.

III. RESULTS AND DISCUSSIONS

During this work we have used the Python language for the coding along with different modules from the OepenCV and the Media Pipeline. For the detection of the real-time image and the marking the landmark for the identification of the object.

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together: A palm detection model that operates on the complete image and returns an oriented hand bounding box. A hand landmark model that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand key points. This strategy is analogous thereto employed in our MediaPipe Face Mesh solution, which uses a face detector along side a face landmark model.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 07, July-2021

Providing the accurately cropped hand image to the hand landmark model drastically reduces the necessity for data augmentation (e.g. rotations, translation and scale) and instead allows the network to dedicate most of its capacity towards coordinate prediction accuracy. In addition, in our pipeline the crops also can be generated supported the hand landmarks identified within the previous frame, and only the landmark model could no longer identify hand presence is palm detection invoked to relocalize the hand.

The pipeline is implemented as a MediaPipe graph that uses a hand landmark tracking subgraph from the hand landmark module, and renders employing a dedicated hand renderer subgraph. The hand landmark tracking subgraph internally uses a hand landmark subgraph from an equivalent module and a palm detection subgraph from the palm detection module.

In this work we can detect the number of fingers in the video captured by a laptop camera. Basically, the primary task is to detect hand within the video frame. This is the most challenging part. The proposed way is to use Background Subtraction and HSV Segmentation together to create a mask. After the hand is segmented, we'll detect the amount of fingers raised. There are 2 proposed methods. The first is to seek out the most important contour within the image which is assumed to be the hand. Then, we'll find the convex hull and convexity defects which are most likely the space between fingers. This is a manual way of finding the amount of fingers. The second way is to use a convolutional neural network with the mask as input to work out the amount of fingers.

The proposed system consists mainly of three phases: the primary phase (i.e., pre-processing), subsequent phase (i.e., feature extraction) and therefore the final phase (i.e., classification)

This deals with the problem of developing a vision-based static finger gesture recognition algorithm to recognize the following six static finger gestures: 1 finger, 2 finger, 3 finger,4 finger,5finger.

These gestures are chosen because they are commonly used to communicate and can thus be used in various applications, such as, a virtual mouse that can perform six tasks (Open, Close, Cut, Paste, Maximize, Minimize) for a given application.

The models of finger gestures were built by considering gesture differentiation and human tendency, and human skin colors were used for finger segmentation.

The obtained results are shown in the fig.1,2,3,4. The figure 1,3 are the capture images through the Web Camera and the figure 2,4 are the output images. These show the detection of the gesture and the figure count.

Fig. 1 Capture real-time Hand image

Fig. 2. Output Hand image with Finger count

Fig. 3 Capture real-time Hand image

Fig. 4. Output Hand image with Finger count

CONCLUSION

As a conclusion based on the result of the works, it can be seen that developing the hand gesture recognition using Python and OpenCV can be implemented by applying the theories of hand segmentation and therefore the hand detection system which use the Haar-cascade classifier.

To summarize it, this work accomplished several objectives: 1. manage to establish a complete system for detecting, recognizing and interpreting hand gesture recognition through computer vision using Python, OpenCV, and Mediapipe and

2. able to create the numbers and sign languages of hand gesture shown in the system. The obtained results are better in accuracy.

ACKNOWLEDGE

Special thanks our beloved Director, Dr. R. K. Jain for his continuous encourage and support. I also take an opportunity to thank my colleagues for their support.

FUTURE SCOPE

For the future recommendation, this system will include the execution of additional gestures that will allow any users with different skin color and size of palm to perform more functions easily. The current system only uses the right hand with specific area in ROI to perform gestures. Therefore, the desired enhancement of the technique may possibly using both of user hands to perform different signs with computer operations. Additionally, background subtraction algorithms can be used for more effective performance. Implementing the Graphical User Interface (GUI) will be done in the future where the users know how to translate the gesture from its meaning to the sign or number and vice versa.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 07, July-2021

REFERENCE

[1] https://www.theseus.fi/bitstream/handle/100/24/140208/aalamaki_thesis.pdf
[2] https://developpaper.com/python-opencv implementation-of-rotating-text-correction/
[3] https://data-flair.training/blogs/computer-vision-project-ideas/
[4] https://data-flair.training/blogs/opencv-python-tutorial/
[5] https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
[6] https://www.javatpoint.com/opencv
[7] https://www.hackster.io/mjrobot/real-time-face-recognition-an-end-to-end- project-a10826