Rock Paper Scissors

-Real Time Fingers Detection

Open Source Software Development Lab project presentation
by Boxuan Zhang and Jing Yang

Introduction

In this project, an algorithm is implemented to detect the fingers in front of the camera in real time. This project is built on the OpenCV library and sklearn library.

Project Overview

Set Up Background

- Create an Detecting Area (Contribute a region to detect hand moving)
- When camera starts to record, calculate a running average background value for 35 frames by accumulateweighted mathord from OpenCV. Screen shows detail information when process calculating background.
- Once average value is established, the object hand is ready to enter the detector, screen display message to tell users.
- The motion of the hand is recorded by the camera for future analysis.

Image:https://www.pngkey.com/detail/u2q8q8q8i1a9q8w7_boy-student-cartoon/

Segmentation

- Use thresholding method from OpenCV to grab the hand segment in the detecting area, when camera detect there’s motion in detecting area, the difference of the background and new frame is recorded.
- The foreground represent the hand which is set to be white in color and background is black for the calculation of the contour of the object hand. Thresholding requires a binary color input and also lead to a binary color output. The thresholding applys to every new frame during the hand motion.

Fingers Counting with Convex Hull

- Convex Hull method from OpenCV is used to draw a convex hull around the hand.
- Contours method from OpenCV is used to calculate the contours of the hand.
- The center point of the hand can be calculated against the angle of the outer points for fingers counting.

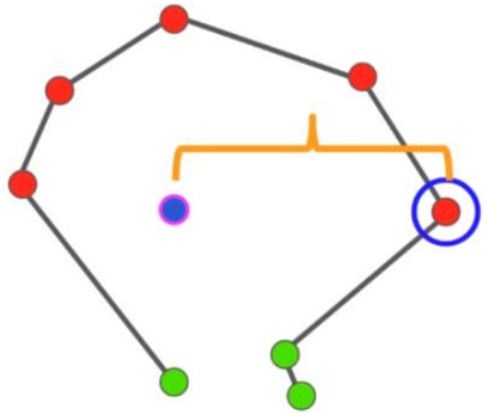
image: https://i.stack.imgur.com/vZbis.png

Fingers Counting with Convex Hull

- When the background and hand segmentation have been set up, the cv2.convexHull method can help to draws a polygon by connecting points on the hull of the hand segmentation in a frame.

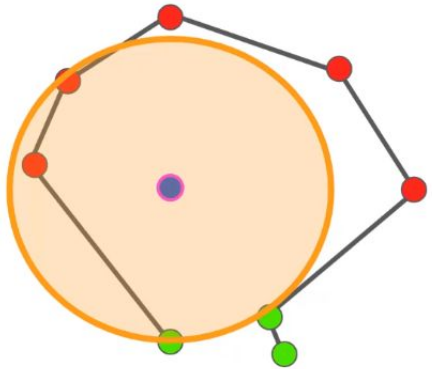
Image: https://docs.opencv.org/3.4/d7/d1d/tutorial_hull.html

- First, find the top, bottom, leftmost, and rightmost points on the convex Hull.
- Then, calculate the center of the hand based on the four points found in the previous step.
- Next, calculate the euclidean distance between the point on the hull and the center by sklearn pairwise method.



https://medium.com/@estevestoni/segmenta%C3%A7%C3%A3o-com-envolt%C3%B3ria-convexa-e-opencv-118ef7138238

- Use the ratio times the distance as the radius to create a circle. The circle is centered at the center of the hand and r = C*distance.
- Any points outside of this circle are far away enough from the bottom, should be finger points.



https://medium.com/@estevestoni/segmenta%C3%A7%C3%A3o-com-envolt%C3%B3ria-convexa-e-opencv-118ef7138238

Demo Display

https://github.com/psu-os-rps/Rock-Paper-and-Scissors/tree/master/example

Lesson Learned

- By using the erosion method and dilation method from OpenCV, the detection of fingers from a complicated background can be improved due to the better segmentation.

Limitations and future Plan

- It’s relatively stable to detect one, two, three and five fingers, but four fingers is still difficult to detect. Should build a automatic detector to perform a better algorithm when people with different size of hands.
- If the fingers are not holding up perfectly face the camera, the algorithm might not be counting.

Reference

- Finger Detection using OpenCV and Python From Izane
- hand-gesture-recognition using OpenCV and Python From Aakash Jhawar
- COVID-19 Face Mask Detector with OpenCV Keras TensorFlow and Deep Learning
- COVID-19 Face Mask Detector with OpenCV Dataset
- OpenCV Documentation

Thanks for watching!