Final assignment – Embedded Vision

Counting finger with camera Raspberry Pi

Hong Trinh - 438443

2022

# Introduction

The goal of this assignment is detecting the number of fingers which is in front of the camera with raspberry pi. Then the result of counting should be shown in the 7 segments.

# Solution

For detecting the hand, I use the library mediapipe with the hand landmark model to perform the keypoint localization of 3D hand-knuckle coordinates inside the detected hand regions.

Text

Description automatically generated with medium confidence

Figure 1: Hand landmarks

To count the fingers, we detect by comparing the position of keypoint on each finger. For example: with the index finger, if the position of point 8 is lower than the point 6, that means index finger is down. With the middle finger, if the position of point 12 is lower than the point 10, the finger is down. The same case is applied to ring finger and pinky finger. For the thump, it is compared in horizon direction. If the point 4 is on the right of point 2, thump is close.

Diagram

Description automatically generated

Figure 2: Flow chart

Import essential library:

Text

Description automatically generated

First, we create a variable for the mediapipe hands class. The static\_image\_mode parameter set to false for the input image as a video stream. Otherwise, if it sets to true, hand detection runs on every input image, ideal for images. Max\_num\_hands parameter is the maximum number if hands to detect (default is 2). Min\_detection\_confidence parameter is the minimum confidence value from the hand detection model for the detection to be considered successful ([0.0, 1.0])

Graphical user interface

Description automatically generated with medium confidence

The output of the function is MULTI\_HAND\_LANDMARKS, MULTI\_HAND\_WORLD\_LANDMARKS, MULTI\_HANDNESS.

The function detectHandsLandmarks detect the hand and draw the landmarks on hand image. Multi\_hand\_landmarks are a collection of tracked hands. Each hand is represented as a list of 21 hand landmarks which is composed of x,y,z.

Text

Description automatically generated

The function countFingers to count the fingers up of both hands. The dictionary variable ‘count’ will store the number of fingers of each hand. We initializes a list of finger tip and a dictionary store the status of each finger (open or close)

A screenshot of a computer

Description automatically generated with medium confidence

Using the loop to detect the hand. Multi\_handedness is a collection of handedness of the tracked hand such as left hand, right hand and score. A loop to iterate each finger in order to compare the position of tip and pip landmarks. As mentioned above, except the thump, if the tip point is lower than the pip point, finger is close and reverse. In this case, the y-axis starts from above and go down, then the finger is up when the y-coordinates of the tip lower pip landmarks. The same approach for thumps but compares the x-coordinates. Store the count value into the dictionary.

Text

Description automatically generated

Show the result on the screen:

Text

Description automatically generated



Using VideoCapture function to read the video object from camera. If the camera is open and can read the video, the detectHandsLandmarks is called to detect and draw hand landmarks. If the hand landmarks are drawn, the function countFingers is called and the results will be stored. Show the result in the frame by imshow().

Text

Description automatically generated

A picture containing treemap chart

Description automatically generated

Figure : Raspberry pi GPIO

The pin map for 7 segments connects with raspberry pi: a-40, b-37, c-35, d-33, e-31, f-29, g-23 and gnd-39. Set the pin output for each segment. Show the number on 7 segments depend on the value of count.

Text

Description automatically generated with medium confidence

Text

Description automatically generated

# Reference

[Hands - mediapipe (google.github.io)](https://google.github.io/mediapipe/solutions/hands.html#min_tracking_confidence)