

Faculty of Engineering

Computer Department

Image Processing Project

“Sign Recognition”

Team 2

Team Members:

1- Ahmed Salah

2- Khaled Sabry

3- Mahmoud Yossri

4-Mohamed Abd-Elaziz

Project overview :

The project is intended to identify on your hand gesture to write characters using image processing and or for future development changed to make orders to your computer to make something.

Inputs and Outputs:

Input: Images from camera.

To open/close windows press following keys:

f => Frame window

m => Mask window

t => Threshold window

h => Hue window

s => Saturation window

v => Value window

Output: letters extracted from image.

Development Process:

Currently, the app is developed on the computer for faster debugging and the photo will be run on different filters.

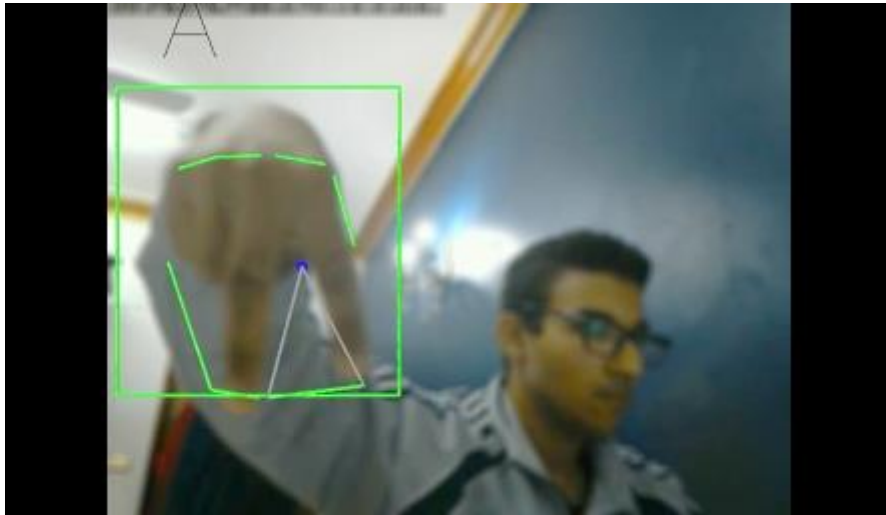
Open Sources :

- OpenCV
- NumPy

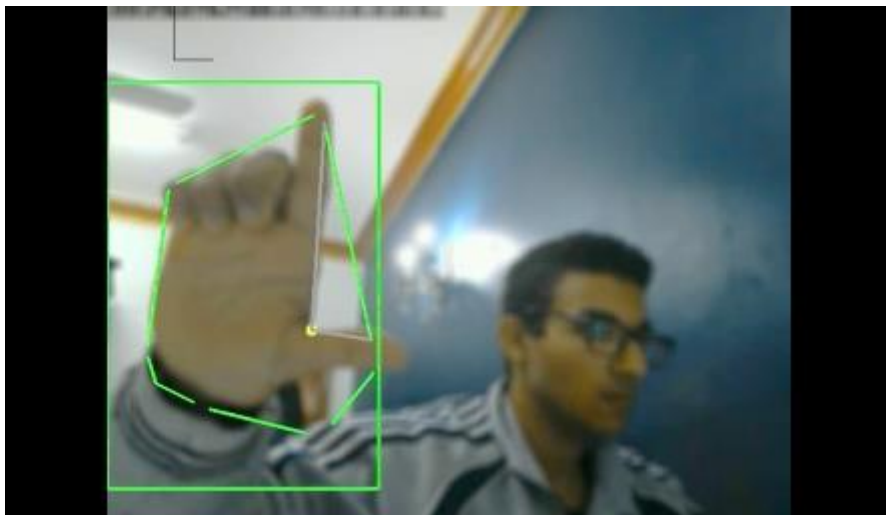
Identification:

We assigned 5 characters to our program to identify: "A, L, M, V, W"

A



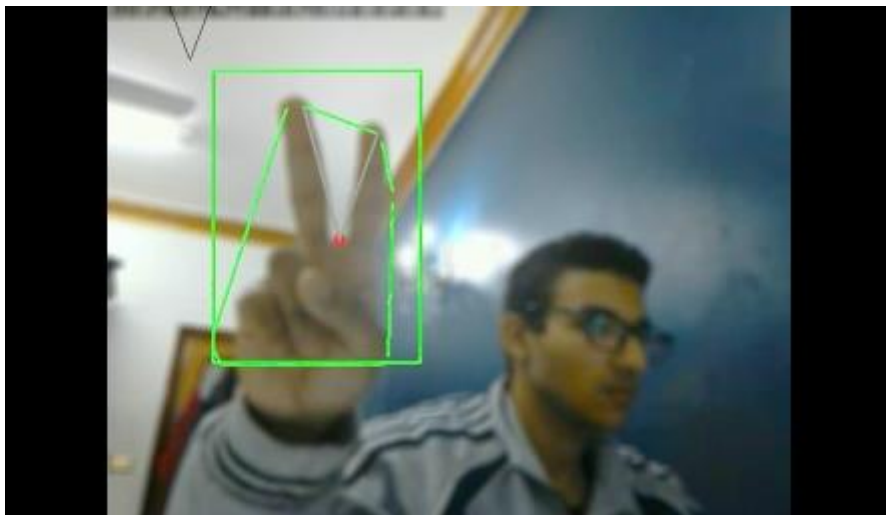
L



M



V



W



Work Division:

We all searched for masking the hand out of background and detecting characters from hand and also for better results.

Tries which guided us to solution:

we categorized our problem to recognize the hand into two categories:

- mask the hand out of background.
- detect the characters from hand.

- Mask Hand Out Of Background:

Try No.1: try to predict hand color by static values.

Problem: the hue channel was okay but the value and saturation channel was changing from position to another according to the light incident on the hand.

Try No.2: take the sample color from hand by first take the background without hand and after you take a sample you put your hand on the screen.

Problem: it was worse than previous tries because if you moved your body the image will detect your body also so you needed to take a sample from time to time and also you refused this idea.

Try No.3: take a sample from hand color by making a big rectangle to take the sample by averaging the hue, value and saturation channels from the whole rectangle.

Problem: didn't make good sample because of the we take it on the whole area.

Try No.4: try to fix the previous try by taking a sample from hand color by making multiple rectangles to take the sample by averaging the hue, value and saturation channels from the multiple rectangles and take the most common.

Problem: didn't make good sample because of some samples where very different than others.

Try No.5: we tried to make change to previous try by not taking the average but sort the values and take the lowest and highest value from each rectangle and then take the average among them all.

Problem: some areas now were having different hue and value so the mask excluded a lot of areas from the hand.

Try No.6: we then tried to make slight change to the previous try by not taking the lowest and highest value from each rectangle and then take the average among them all but we made multiple masks from the lowest and highest value from each rectangle and then add all these masks together.

Problem: results also were very bad a lot of background noise

Try No.7: we then tried to make slight change again to the previous try by not taking lowest and highest value from each rectangle but taking the average of the rectangle.

Problem: results also were better than previous but there were also a lot of background noise.

Try No.8 : try to get a good quality camera to apply it on the previous try as we thought the problem from the laptop camera.

Problem: found that the problem was not in the camera but the problem the idea itself.

Try No.9 : tried to make a lot of smoothing (blur, gaussian and median) filters to theses previous tries.

Problem: results were better than before but still we couldn't extract the hand from background.

Try No.10: we try to detect the difference from current frame and the previous frame.

Problem: we sometimes were able to extract the hand but you need to keep you hand moving a lot.

Try No.11: we then figure it out what about we keep the previous ten frames and compare it with the current frame.

Problem: fixed the problem of moving your hand fast but also you need to move you hand also it is bad to save ten frames in memory.

Try No.12: we then looked for the histogram what if we sum the previous ten frames histogram and then make a condition if the sum was less than specific number then your hand didn't move then keep the previous frame as if it is the current frame.

Problem: this try failed also because of the noise or if you moved your hand fast it will detect a lot of frames which will be kept as the sum and then your hand will disappear.

Try No.13: we then looked for the histogram of mask current frame minus previous frame then if the histogram was at specific range then the hand moved.

Problem: when we applied the contour, the results were very bad because the mask image were having a very low white pixels so the contour took a lot of contours.

Try No.14: we tried then to fix the previous try by taking the last five images instead of the current and previous only.

Problem: the results were very different if the fifth image was having a huge move then it will result in bad detection.

Try No.15: so, we returned the idea of taking “the histogram of mask current frame minus previous frame then if the histogram was at specific range then the hand moved” but the change is to make a huge dilation to make these small pixels become huge and then move the sample rectangles from first tries to this position.

Problem: the detection was good but the sampling rectangles were very bad because we couldn't know a which distance the hand is and what is the size of the hand.

Try No.16: [**Solution**] the solution also was a try what if we make the huge dilation to the subtract of current frame with the previous frame and then apply the contour to make a mask then make the histogram from the contour mask.

Problem: some background become bigger than hand which then take the contour from the hand.

Try No.17: [**Solution**] we tried once more by suppress whole background and show only the hand region which gave to us more pleasing results as it removes all background instead of hand area

Problem: if the hand area - not the whole background - has background with the same hand color and illumination the

program will see the hand and its background as the hand absolutely there is a problem with it because always there is another try to make.

- Detect the characters from hand:

Try No.18 : using Harris algorithm implemented in OpenCV.

Problem: you refused this try even if we use it to remove the head.

Try No.19 : by using the contour method which will get the convexity from contour.

Problem: detect a lot of contours not just the hand.

Try No.20 : we make a difference to the previous try by taking the maximum contour area.

Problem: we were able to detect the hand contour and also removing the face but when we detect the convexity we were having a lot of them.

Try No.21 :**[Solution]** we build up on the previous try and put conditions to only accept specific number of convexities and with specific angles

Problem: needs more condition to make it perfect but there is always a another try.

Performance: -

All code for all tries were very fast and we didn't face any problem with performance.

Conclusion: -

- Points of strength

- 1) detect hand dynamically.
- 2) detect hand under any illumination or position.
- 3) recognize hand signs effectively.
- 4) no need to put static color to segment hand.

- Points of weakness

- 1) if the hand area - not the whole background - has background with the same hand color and illumination the program will see the hand and its background as the hand.
- 2) you need your body to be static so the program can detects the dynamic movement of the hand.

References: -

- [1] OpenCV Documentation: <https://goo.gl/Epmiwd>
- [2] NumPy Documentation: <https://goo.gl/P4CJYk>