

COMS W4735 Assignment 1

Wentao Jiang (wj2227)

1. Pictures were captured with cell phone (LG Flex) at 960*1280 resolution, with white wall as background under white bulb light. To avoid further problems, skin under wrist was covered by clothes.

Hardware: back camera on cell phone (LG Flex, Android 4.4)

OS of processing: Windows 8.1

Language and packages: Matlab

Images are stored in .jpg form.

.m files: hw1.m (main function) / verify_gesture.m / generate_skinmap.m / grayworld.m

2. Data reduction step

For this visual combination lock, two gestures are required to pass the test, although it is capable of further extend to more gestures. The two required gestures include: (1) hand with closed fist at the center of image, followed by (2) a flat palm at the center of image with fingers outwardly splayed (i.e., the gesture for "five!"). The detection of skin was done by simple color segmentation under $YCbCr$ space [1]. The assumption made is that pixels with $77 \leq C_b \leq 127$ and $133 \leq C_r \leq 173$ is considered to be skin. This method works fairly well, although it is possible that the result be affected by the conditions such as lighting and shading. For each image, pixels are binarized based on the color, and will be parsed in step 3.

Picture list: 2fingers/ fist_1/ fist_2/ fist_3/ fist_4/ palm_1/ palm_2/ palm_3/ palm_4/ palm_5/ palm_6/ unknown_1/ unknown_2/ unknown_3/ unknown_4

All pictures are in .jpg form.



3. The grammar for the system is that the two required gesture must be shown in order: (1) hand with closed fist at the center of image, followed by (2) a flat palm at the center of image with fingers outwardly splayed (i.e., the gesture for "five!"). The "center of image" is defined as cases when the centroid of the hand in the image is within the middle 1/3 both horizontally and

vertically, as the green region shown in figure below. Similarly, although no grammar involves concept of corners, it is trivial to extend by verifying if the centroid of hand in the image is within the corner areas.

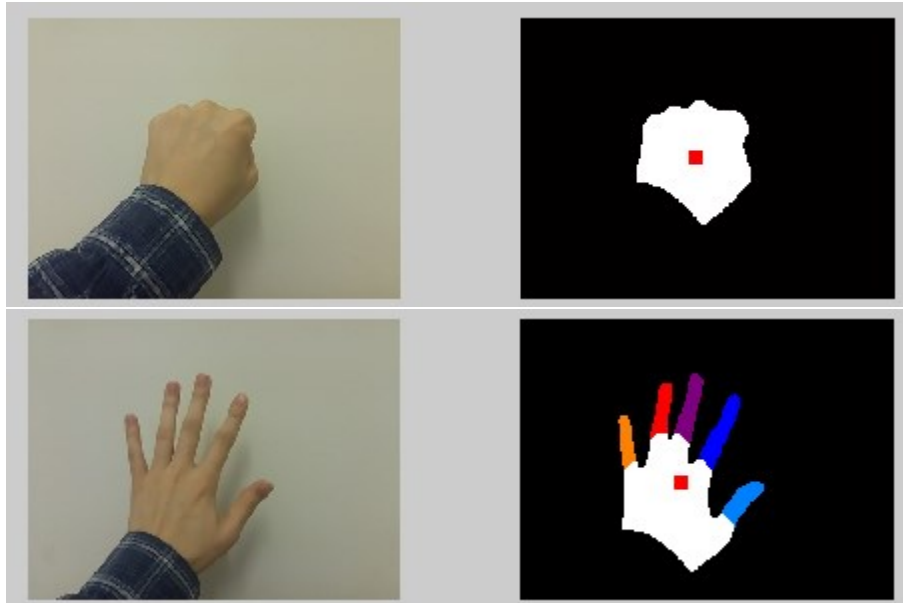


For each gesture detected, the system will count the number of splayed fingers, and make judgment based on the counting. If there is skin (supposed to be hand) detected but no splayed fingers, it is considered to be closed fist. If there is skin detected and there are five fingers attached to it, the image is considered to be a flat palm with fingers splayed. Each finger detected is labeled with a different color on the binary image. Other (some of which peculiar) gestures are used as interference in order to test the system.

The method used to detect and count finger is explained as below[2]. First, the centroid of detected skin area is found from the binary image (here the assumption that there is one object is made). Then by looping through the boundary points, the distances from each boundary point to the centroid is calculated, and the minimum is found. The minimum distance and centroid are used to generate a shape similar to the palm area by dilation after erosion from the centroid. So after subtracting the generated palm from the original hand, fingers are supposed to be left and disconnected from each other, in which case it is easy to count the number of connected areas in the binary image after operations. Note that when there is no splayed fingers, the binary image will be all zeros, which indicates that nothing left after subtracting the palm.

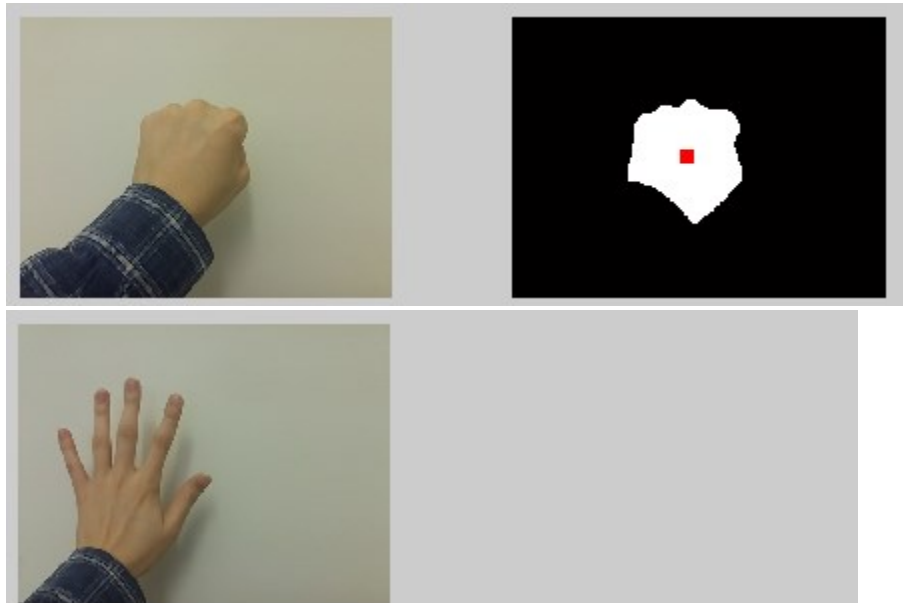
For the sequences below the grammar was run. (1) to (7) are cases that succeeded.

- (1) fist_1.jpg palm_1.jpg
result: true positive.



```
>> hw1
1. Place your hand with a closed fist in the center of the first image:
Fist detected.
2. Place your hand as a flat palm in the center with fingers outwardly splayed:
System unlocked!
```

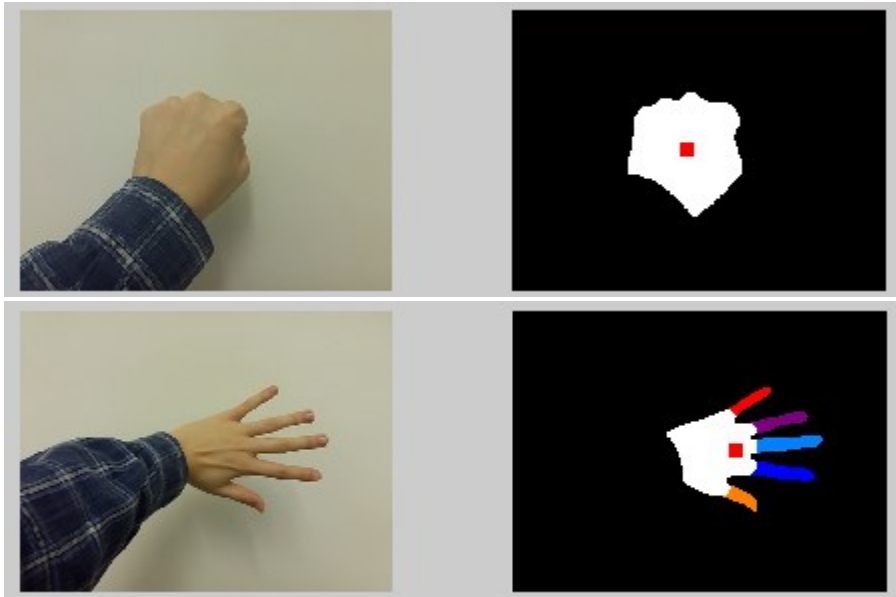
- (2) fist_1.jpg palm_2.jpg
result: true negative. The palm is not in the center.



```
1. Place your hand with a closed fist in the center of the first image:
Fist detected.
2. Place your hand as a flat palm in the center with fingers outwardly splayed:
Not at center. Please try again
```

(3) fist_1.jpg palm_3.jpg

result: true positive. The system can work with any orientation of the gestures as long as the palm plane is facing to the camera.



1. Place your hand with a closed fist in the center of the first image:

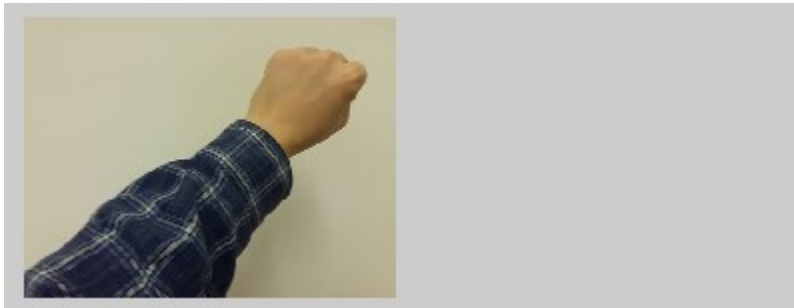
Fist detected.

2. Place your hand as a flat palm in the center with fingers outwardly splayed:

System unlocked!

(4) fist_2.jpg palm_1.jpg

result: true negative

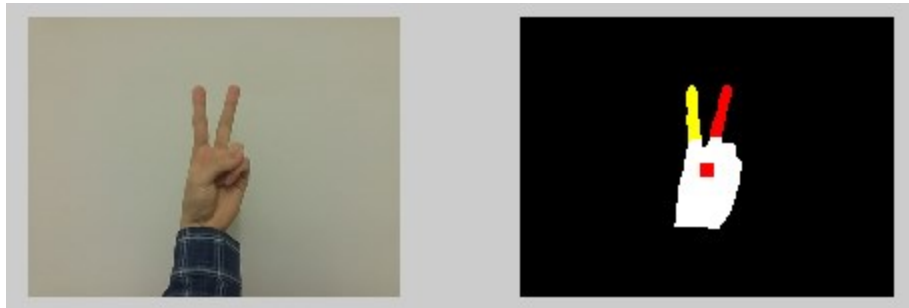


1. Place your hand with a closed fist in the center of the first image:

Not at center. Please try again

(5) 2fingers.jpg, palm_1.jpg

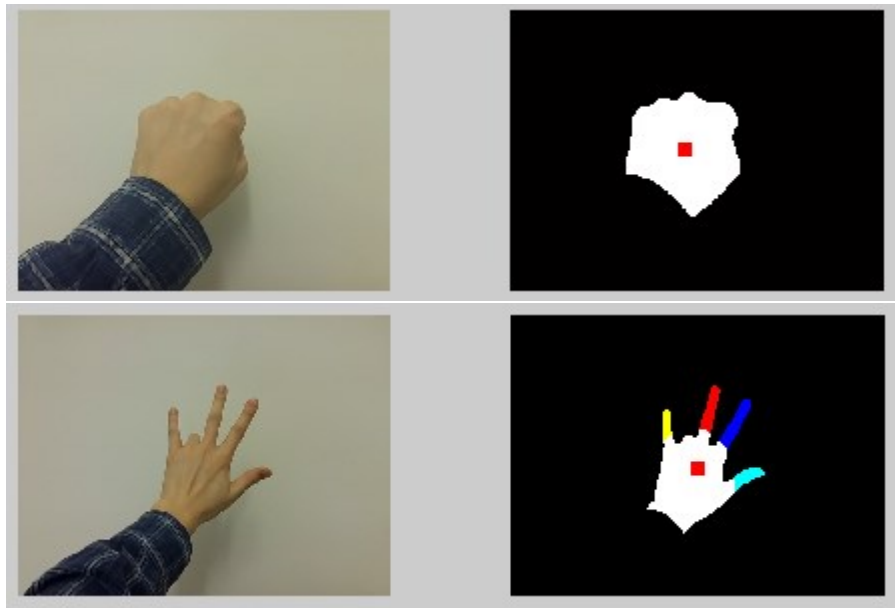
Result: true negative. The program caught the two splayed fingers for the frame that is supposed to be fist.



1. Place your hand with a closed fist in the center of the first image:
Not closed fist. Please try again

(6) fist_1.jpg, unknow_3.jpg

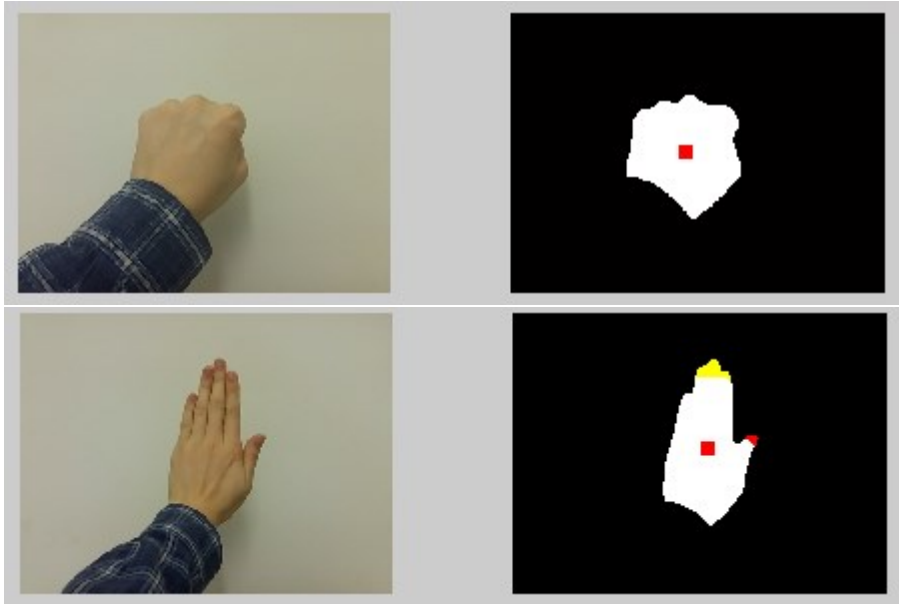
result: true negative



1. Place your hand with a closed fist in the center of the first image:
Fist detected.
2. Place your hand as a flat palm in the center with fingers outwardly splayed:
4 splayed fingers detected. Please try again

(7) fist_1.jpg, palm_4.jpg

result: true negative. Since the fingers is not splayed, it is considered a negative case. The number of fingers was counted to be 2, although it may not represent the real situation.



1. Place your hand with a closed fist in the center of the first image:

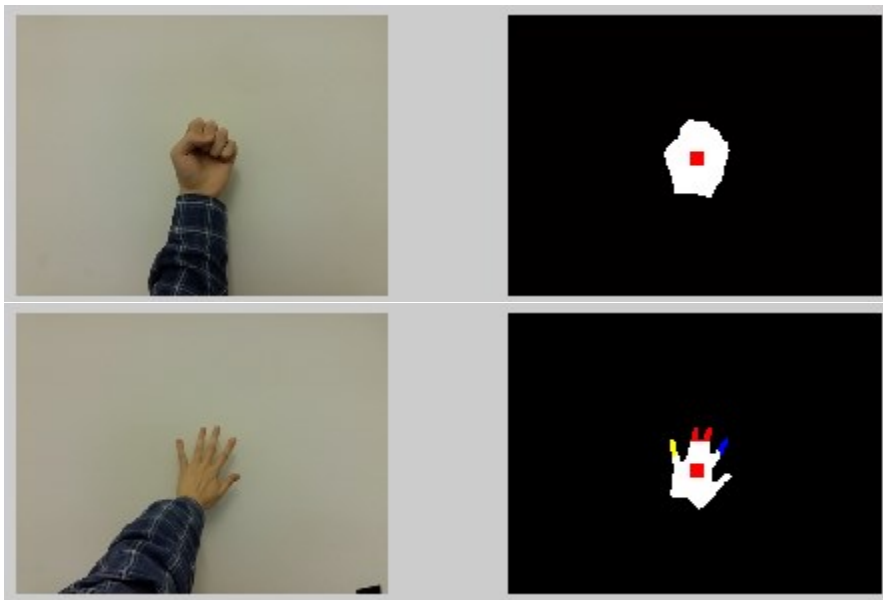
Fist detected.

2. Place your hand as a flat palm in the center with fingers outwardly splayed:

2 splayed fingers detected. Please try again

(8) fist_3.jpg, palm_5.jpg

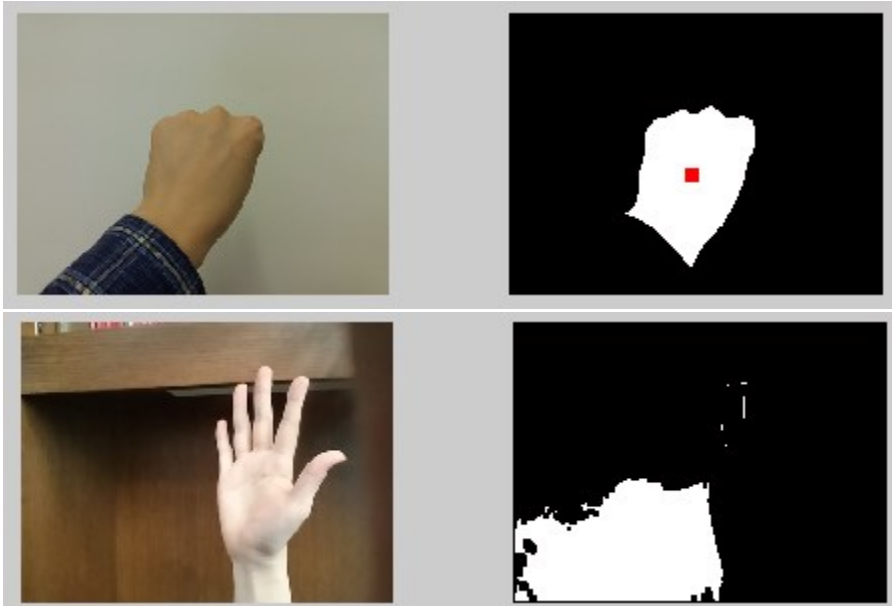
result: false negative. It is supposed to pass the test but didn't, because the palm was farther from the camera and smaller, and the process of counting finger didn't catch all fingers. So one of the limitation is the distance of hand from the camera.



```
1. Place your hand with a closed fist in the center of the first image:  
Fist detected.  
2. Place your hand as a flat palm in the center with fingers outwardly splayed:  
3 splayed fingers detected. Please try again
```

(9) fist_4.jpg, palm_6.jpg

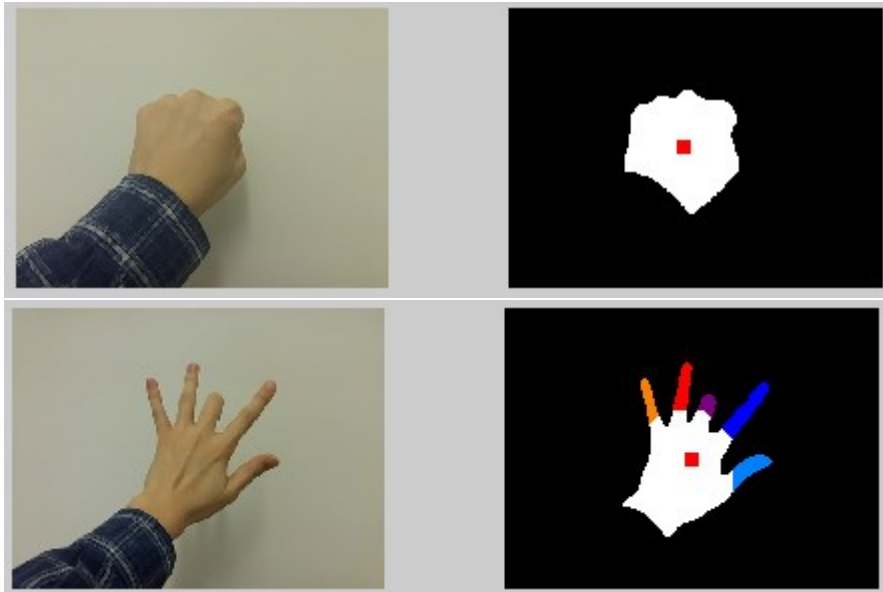
result: false negative. This case the background was changed. While part of the background was detected as skin, the hand was conversely not. So the lighting and background may be potential source of error.



```
1. Place your hand with a closed fist in the center of the first image:  
Fist detected.  
2. Place your hand as a flat palm in the center with fingers outwardly splayed:  
Not at center. Please try again
```

(10) fist_1.jpg, unknown_2.jpg

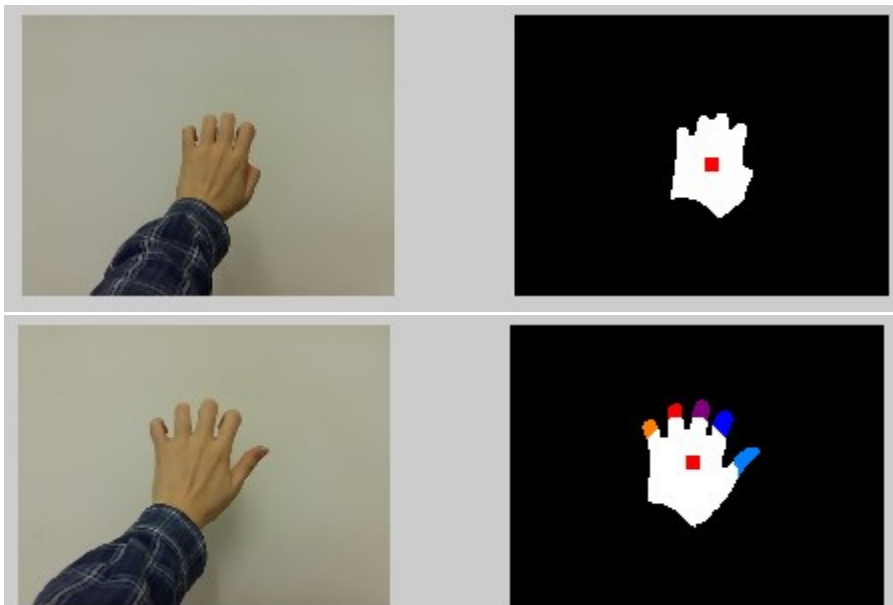
result: false positive. A weird gesture was used with one finger bent for the palm test and passed by false. It indicates that the system is not sensitive to the change of length of fingers and could make mistakes for these cases.



1. Place your hand with a closed fist in the center of the first image:
Fist detected.
2. Place your hand as a flat palm in the center with fingers outwardly splayed:
System unlocked!

(11) unknow_4.jpg, unknown_1.jpg

result: false positive. When the fingers are bent and close, it may be taken as a closed fist, which is actually not. And again, the system is not highly sensitive to the length of fingers, which can result in false positive.



1. Place your hand with a closed fist in the center of the first image:
Fist detected.
2. Place your hand as a flat palm in the center with fingers outwardly splayed:
System unlocked!

Case (8) and (9) are false negatives, and case (10) and (11) are false positives.

4. For this part, figures 0 to 5 are represented by number of splayed fingers for different hand gestures. The number is concatenated and compared with a number set in the program beforehand. A gesture of 'O' means reset the system and clear previous gesture inputs. The detection of 'O' is based on the recalculation of area in binary image after filling holes and Euler number. If the area changes and the Euler number of not-filled binary image is 0, there exist a hole in the picture. Images used are listed below:

0.jpg/ 1.jpg/ 2.jpg/ 3.jpg/ 4.jpg/ 5.jpg/ O.jpg

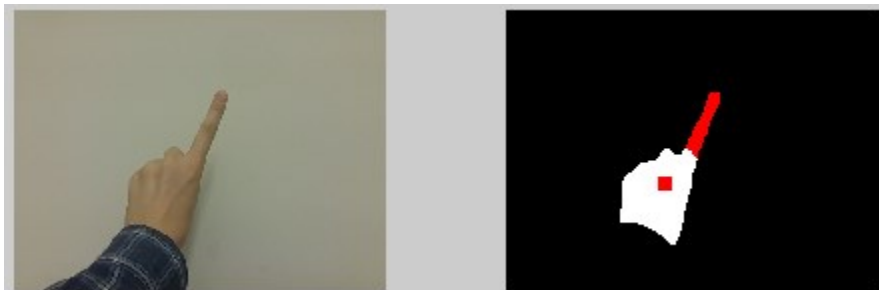
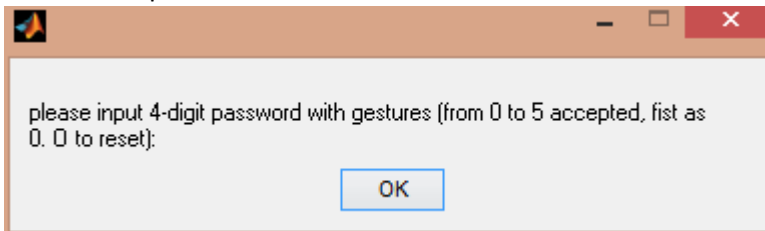
.m files: hw1.m_p4 (main function) / verify_gesture_p4.m / generate_skinmap.m / grayworld.m

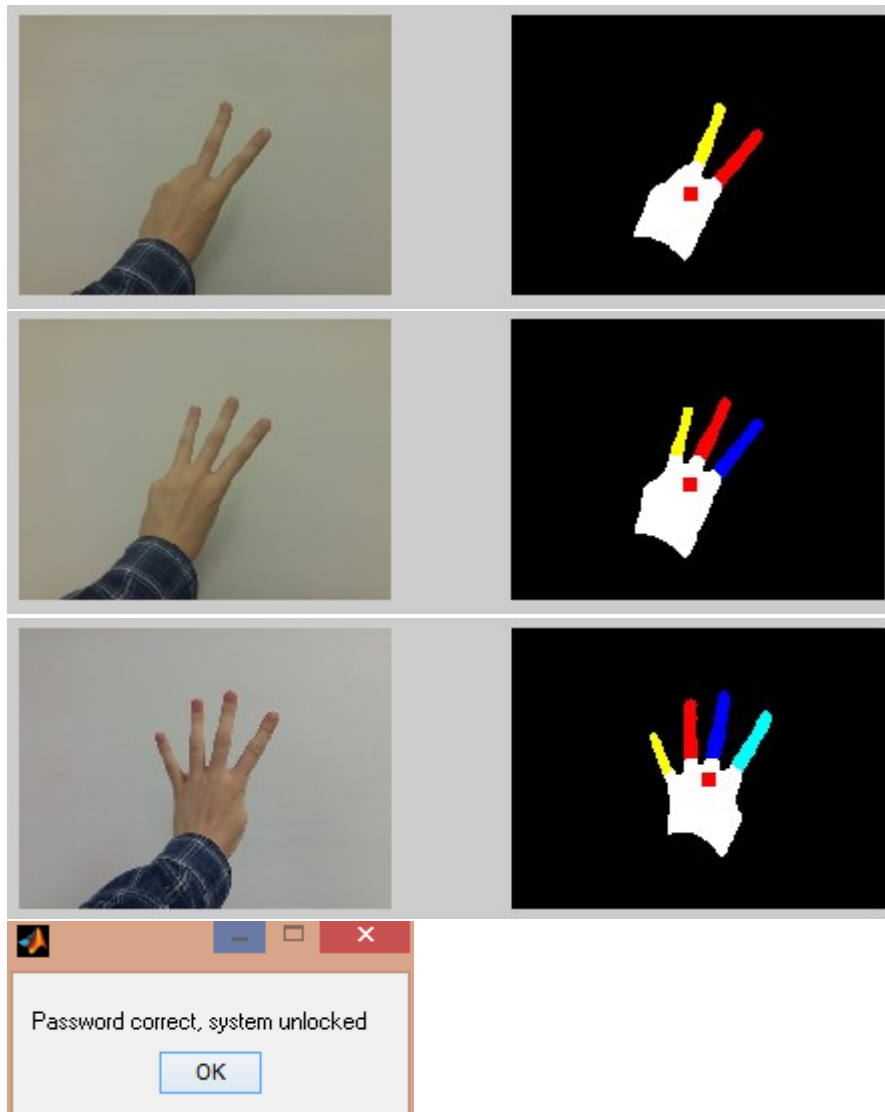


The password in program is set to be 1234. Test cases are as below:

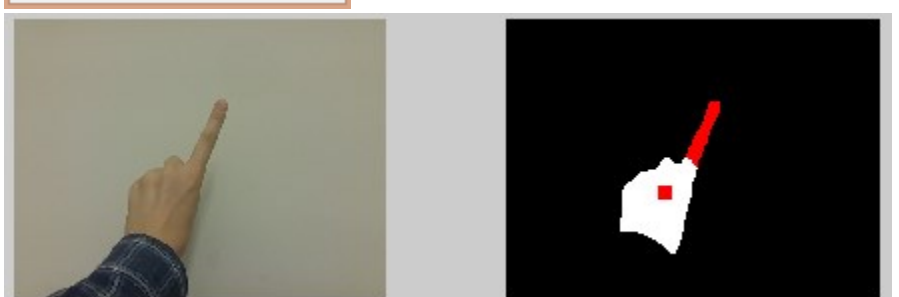
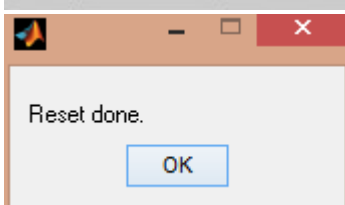
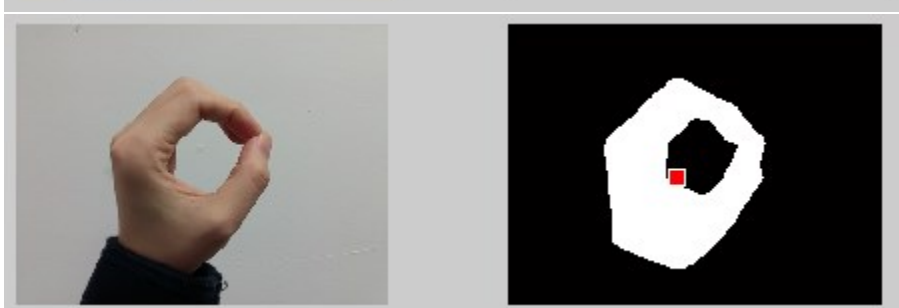
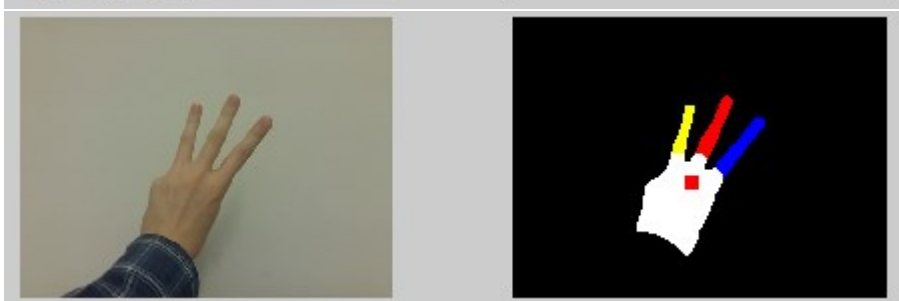
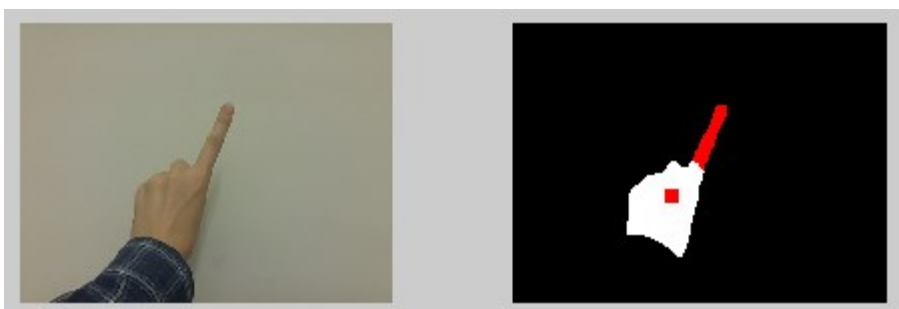
- (1) Input sequence: 1.jpg, 2.jpg, 3.jpg, 4.jpg

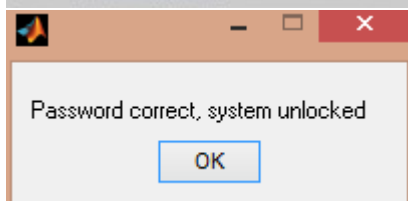
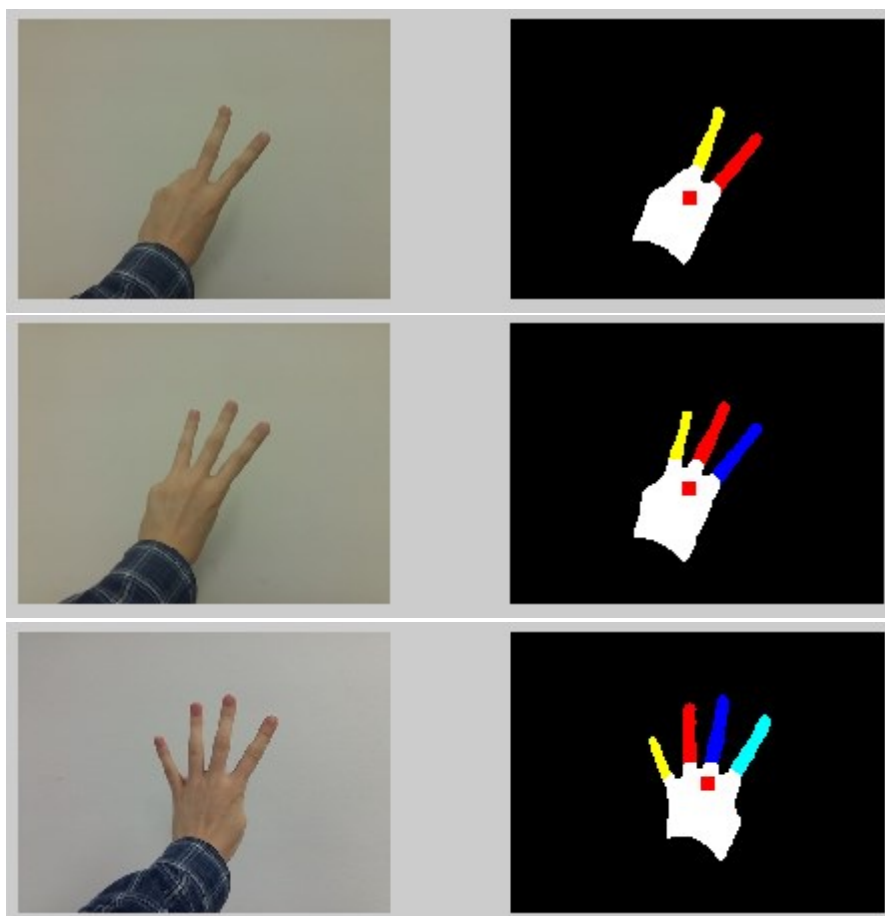
Result: true positive.



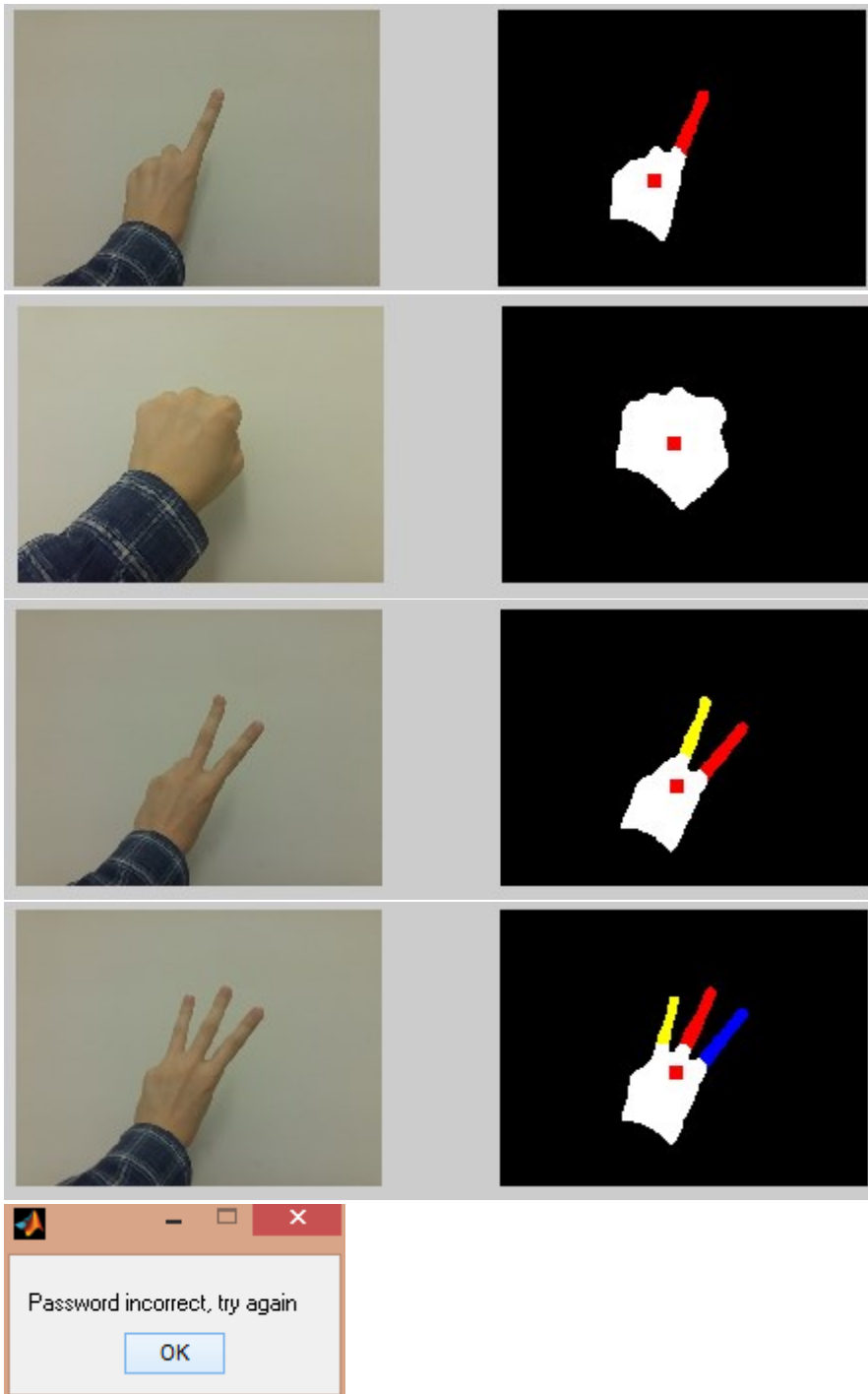


- (2) Input sequence: 1.jpg, 3.jpg, O.jpg, 1.jpg, 2.jpg, 3.jpg, 4.jpg
Result: true positive. The system is reset after seeing 'O'.





(3) Input sequence: 1.jpg, 0.jpg, 2.jpg, 3.jpg
Result: true negative



The vulnerabilities that can lead to errors and mismatches are the same as part 3, such as lighting variation, length of fingers, and distance of hand from the camera.

5. Video based detection of gestures are included in files: hw1_real.m / verify_gesture_real.m

References:

- [1] Skin detection packet used from <http://www.mathworks.com/matlabcentral/fileexchange/28565-skin-detection>.
- [2] Idea partially from <http://codetocreate.blogspot.in/2014/07/how-to-detect-fingers.html> and code were done by self.