

Assignment #1

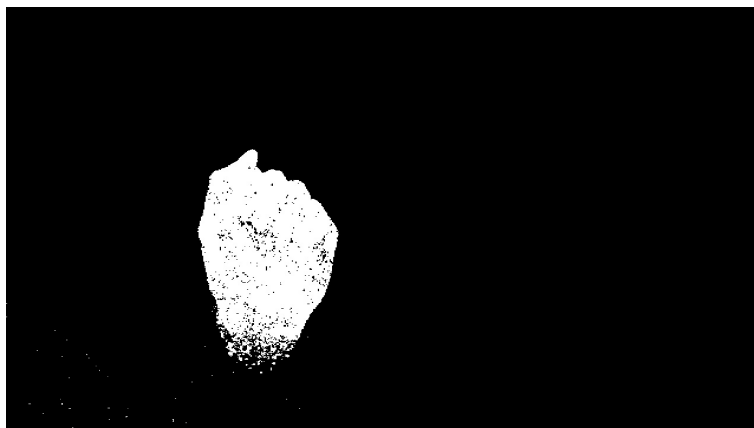
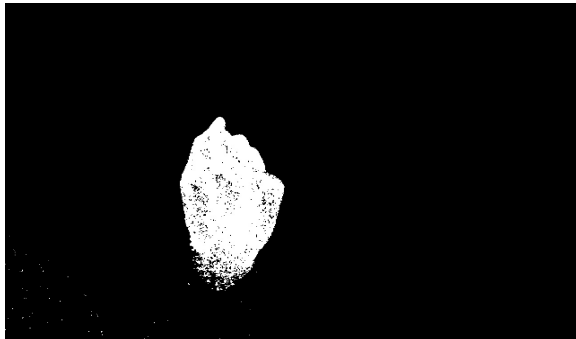
Part 1:

I elected to store images using the jpeg format (see Part 2 for evidence that I successfully did so).

Part 2:

I used a black box solution that implemented an `isSkin()` function. The solution had too many false negatives (Asian skin is oftentimes more difficult to detect) so I went in and lowered the threshold values for hue and saturation to better adapt for my case. The benefit is it recognizes my skin better, the drawback is it is more prone to false positives and extra caution must be taken with ensuring the background noise isn't distracting. Here are some images of what the camera sees in `isSkin()` mode.

The hand detection solution was simple but effective. It relies on set thresholds for hue and saturation to make a judgment.



Part 3:

I intended to work on this, but alas I slept through my alarm clocks... Here's the approach I was planning to use however (pseudocode shown in documentation)

Step 1) Calculate the area. This was implemented and was simply done by counting the number of

white pixels. Although there are a few white pixels that aren't skin, they are so sparse that it will unlikely affect results.

Step 2) Calculate the center of mass. This allows a location to be determined from the poses. This has a limitation though in that it's ineffective for gestures using multiple different body parts (center of mass would have to be isolated into a number of blobs in that case).

Step 3) Calculate perimeter, then analyze the area/perimeter ratio.

Next time I will set stronger alarm clocks.

Part 4:

I intended to implement live video, and in fact I did succeed in that. After 6 seconds a picture is taken from the pose, the user receives feedback on the pose, and the user can then choose whether to retry the password again. The three images above only required one program run.

Code:

```
import cv, time
import handdetect

def getGesture(gesture, nth):
    global startTime
    if nth == 1:
        while time.time() - startTime < 6:
            frame = cv.QueryFrame(capture)
            cv.ShowImage('w1', frame)
            cv.SaveImage(gesture + ".jpg", frame)
    else:
        while time.time() - startTime < 6*nth + 2*(nth-1):
            frame = cv.QueryFrame(capture)
            if (time.time() - startTime > 6*(nth-1) + 2*(nth-1)):
                cv.ShowImage('w1', frame)
            cv.SaveImage(gesture + ".jpg", frame)
        validatePassword(gesture, frame)

def validatePassword(gesture, frame):
    frame = getSkin(gesture, frame)
    mass = getMass(frame)
    center = getCenterOfMass(frame)
    xLength = getXLength(frame)
    yLength = getYLength(frame)

def getSkin(gesture, frame):
    #handdetect came from black box solution via
    https://raw.githubusercontent.com/thisismyrobot/gnomecam/master/hand-detect.py
    frame = handdetect.get_hands(frame)
    cv.SaveImage(gesture + ".jpg", frame)
    return frame

def getMass(frame):
    return cv.CountNonZero(frame)

def getCenterOfMass(frame):
    pass
def getYLength(frame):
    pass
def getXLength(frame):
    pass

if __name__ == '__main__':
```

```
#configure frame and set timer
cv.NamedWindow("w1", cv.CV_WINDOW_AUTOSIZE)
capture = cv.CaptureFromCAM(0)
startTime = time.time()

getGesture("fist1", 1)
getGesture("palm", 2)
getGesture("fist2", 3)
```