CS370 Lab 4 Part 2 Writeup

Lucas Hawk, Patrick Palad, Michael Dilla

1 PiSurveillence

The following are three captures from the provided PiSurveillence program. The camera was set up in Alden 101 during a CS112 lab. Disregard the upsidedowness.

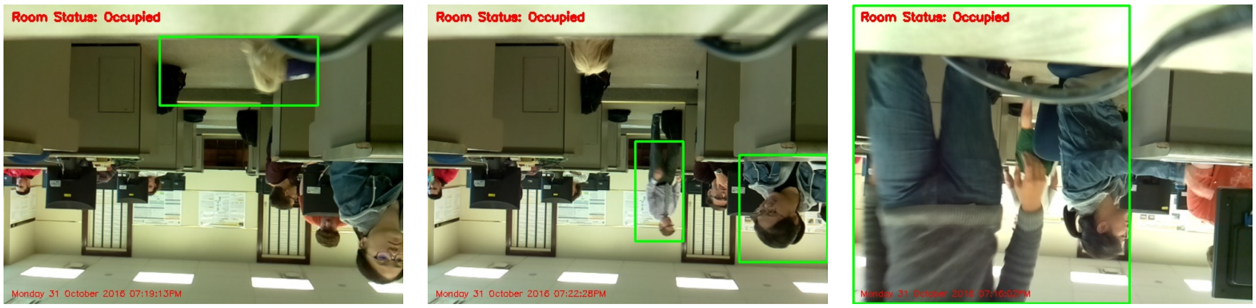
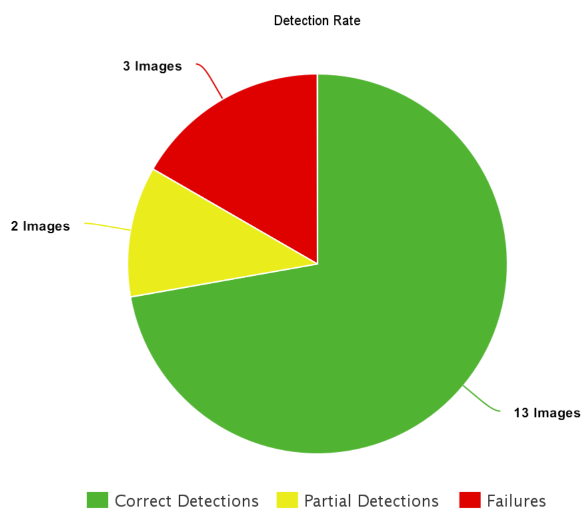


Figure 1: Three Captures from PiSurveillence.py

2 BodyDetect Experiments

For our experiment, we set up our Pi in Alden Hall. It was placed near the doors to room 101 and faced down the hallway between 6:00PM and 7:00PM. During this time period, the Pi detected a person and uploaded a picture to dropbox a total of 18 times. Considering the time for which it was set up, we believe this is a resonable number of detections over an hour. Of the 18 detections, all subjects were correctly identified 13 times. In 3 detections, only some of the subjects were properly detected (e.g. only found one person when two are present), or it correctly identified all subjects and more (e.g. found three people when two are present). In the remaining 2, the program failed to properly draw a bounding box over any of the subjects. With these 2 failures, though, they seem to be due to how close the subjects were to the camera — the subjects either weren't fully visible in the frame, or they took up a large portion of the frame.



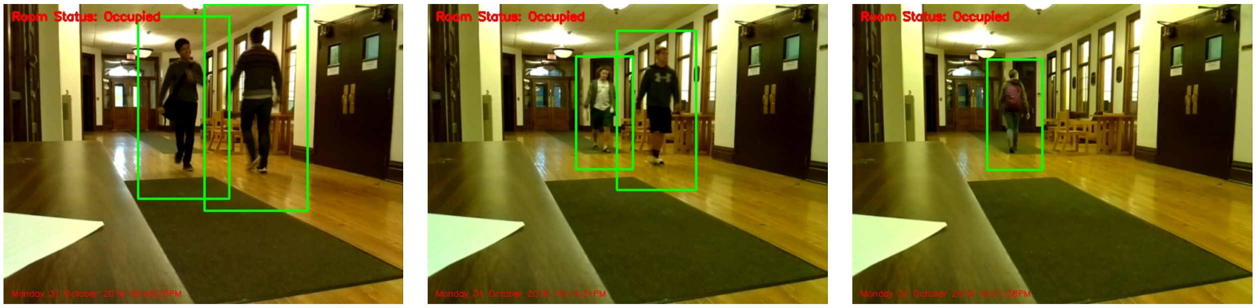
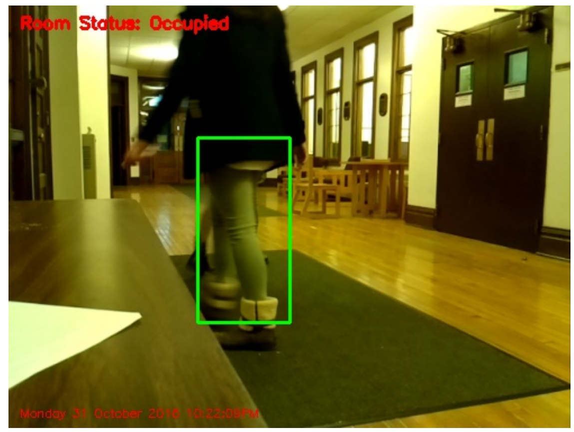


Figure 2: Three Examples of Successful Detection

(a) Partial Detection



(b) Failed Detection

Figure 3: Partial and Failed Detection

3 Bayesian Network

Factors: Whether or not the Pi or camera has been tampered with The time of day Whether or not a person has been detected Whether or not something has been stolen

CPT Tables: Time of Day: P(D) = .50, P(N) = .50

Tampering: P(T) = .10, P(F) = .9

Person Detected: P(T—Day and tampered) = .2 P(T—Day and not tampered) = .95 P(T—Night and tampered) = .05 P(T—Night and not tampered) = .15

Something Stolen: P(Theif—Person detected, Day and tampered) = .75 P(Theif—Person detected, Day and not tampered) = .05 P(Theif—Person detected, Night and tampered) = .99 P(Theif—Person detected, Day and tampered) = .8 P(Theif—Person not detected, Day and tampered) = .5 P(Theif—Person not detected, Day and not tampered) = .01 P(Theif—Person not detected, night and tampered) = .75 P(Theif—Person not detected, night and not tampered) = .15

Node Justification: Time of day: Thief is more likely to happen during night and can also have an effect on the amount of people there to work in the garden.

Tampering: If the Pi or camera is tampered with it is more likely that something is stolen. However, it could have been tampered by natural factors (ex. the wind blowing it over)

Person Detected: If it is night and the equipment was tampered with, it is less likely that there will be a person detected because there will be less people in the garden and it will be harder for the program to detect people. Also, it will be harder to detect people when the equipment has been tampered with.

Something is Stolen: It is more likely for theft to happen at night especially if the equipment has been tampered with and it manages to detect a person. If it is day and the Pi detects a person it is likely that the person is there to work in the garden. Even if the Pi was been tampered with, although it is slightly more likely that something is stolen, it is still not likely that something was stolen.

4 Reflection

1) Description of our addition: For our detection we used the default HOG people detection in openCV. We decided to use this method because we thought that full body detection would work better in a practical setting than face detection. This method also had substantial documentation, which would be helpful in our implementation.

2) Analysis of experiments, trend, assumptions, what worked and what didn't For our expirement we surveyed commmon area/ hallway in Alden hall from around 6 PM to 7PM. Our camera had a full view of the hallway and door facing north main street during this time. A general trend that we saw in our expirement was that our detection algorithm would successfully pick up people moving in the hallway at a medium distance. In general this was the area outside of lab 109, and the waiting area with the tables. There were some problems when there were people up closer to the camera; between 2-4 feet from it. What would happen in this range is that it would often recognize the lower part of someones body as a whole person. This led to a couple cases where the bounding box was drawn only around the persons legs. In addition, when there are multiple people in the frame, it does not always detect both people. More often than not it will only detect the person closest to the camera. It could successfully detect multiple people in the same frame if they were far enough apart from each other. For our expirements we spent a good amount of time working on fine tuning our parameters to the HOG function. Specifically, we worked on balancing the time tradeoff of the windstride parameter. A higher windstride meant faster execution time, but overall significantly less detection capability. We tried values of 64, 32, 16, 8, and 4. We found that while 64, 32, and 16 ran moderately fast, they were completely unable to detect any meaningful movement. A windstride of 4 had such a long startup time that the process often froze before it started. Ultimately we found that a windstride of 8 was a good balance. Lastly, we also decided to get rid of the live preview of the camera when testing, as we found that it was responsible for many of the hangups we had in our testing.

3) Bayesian stuff

4) Hardware problems Our process had minimal hardware problems. In the beginning we had difficulty setting up openCV on the raspberry pi. This was due to an error when creating the make file. After resolving that error, we were successfully able to create the make file by running the Pi overnight. Once we had OpenCV we had a quick problem where we forgot to download some dependencies, and later on had a similar problem. Overall most of our roadblocks in this lab were due to software problems, rather than hardware problems.