CS4053: Computer Vision – Lab 2

Matt Donnelly – 11350561

Abstract

For this lab we were asked to develop a program to monitor a set of postboxes and determine if there is post in each one of the boxes. The program is required to ignore frame if it identifies that there are any moving objects in it and use edge detection to determine if there is post in the boxes.

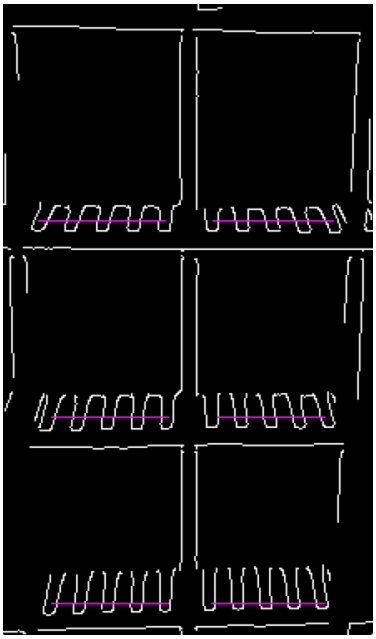
Solution

To start my solution to this assignment I began by implementing motion detection in frames. To achieve this I wrote a function that gets applied to each frame called containsMotion. The function works by simply calculating absolute difference between the current frame and the last know still frame which is stored in a variable. It then binary thresholds the result of this and and determines the percentage of non-zero values in the image. If they take up more 10% of the image, then we can assume there is motion occurring in the image so the function returns true. Any time it returns true, I assign the current frame to the variable storing the last know still frame which works because it will be continually updating until this variable until it sees two still frames in a row – in which case the previous frame will be still. Using this solution I'm able to eliminate frames containing motion from further processing.

A frame detected as containing motion after being compared with last know still frame

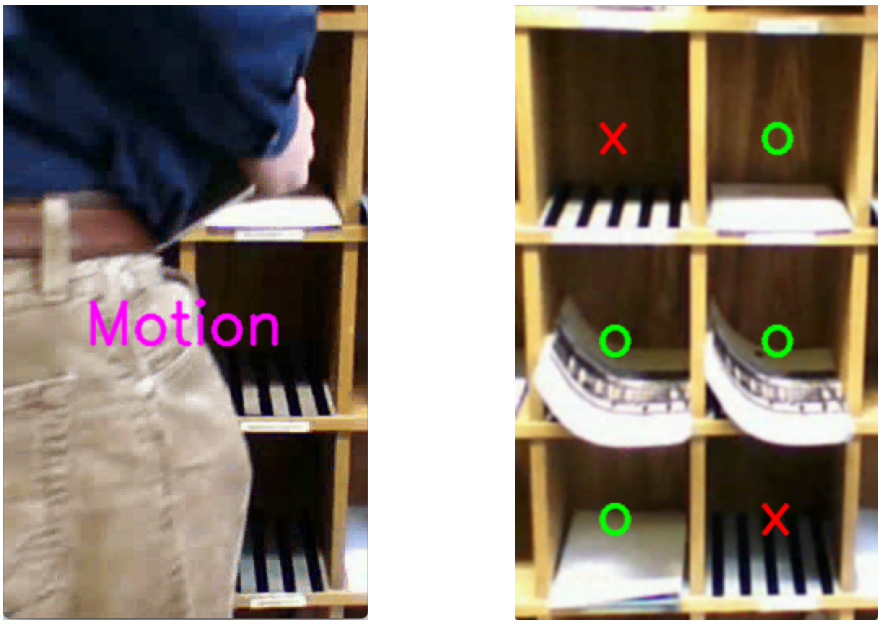
Following this, I began working on detecting which boxes contain posts by counting the number of visible edges of black lines in each box. To do this I first create an edge image by performing Canny edge detection on the frame. The image is blurred slightly beforehand to eliminate small inaccuracies. Using this edge image I then check a section of a row of pixels intersecting with black lines in each post box to count the number of visible lines. The locations to check for each post box

are hard coded for simplicity as we can assume the camera wont be moving. For each row, the program iterates across each pixel and any time it encounters a white pixel it increments a counter storing the number of lines. Originally, I decided to only count a white pixel as a line if it extended upwards by at least 10 more pixels, however I found that if I increased the threshold values of the Canny edge detection this wasn't necessary. Using this information I'm able to determine if a postbox is empty if there are at least 6 visible edges, otherwise it must contain post. I chose 6 instead of 8 in case of some the lines became obscured by a shadow.



Visualisation of the rows of pixels being checked for lines in each postbox

Now, knowing which postboxes are empty or contain post and which frames contain motion, I annotate the video frames appropriately as show below.



Video being annotated to indicate when there's motion and which boxes contain post

Results

In my results I define a true positive as when my program correctly determines a post box contains post and a true negative as when it correctly determines a postbox is empty. I disregard any frames containing motion.

Total Samples = 43 Motionless Frames * 6 Postboxes = 258

True Positives = 116
False Positives = 0
True Negatives = 142
False Negatives = 0

Precision = 116 / (116 + 0) = 1

Recall = 116 / (116 + 0) = 1

Accuracy = (116 + 142) / 258 = 1

Specificity = 142 / (0 + 142)

$F(1) = (1 + 1^2) \times ((1 \times 1) / ((1 \times 1) + 1)$
$= (2(1)) / 2$
$= 2 / 2$
$= 1$