Computer Vision - Assignment 2

Michael Gallagher - 10325071

November 20, 2013

Abstract

Assignment to perform analysis on images of sets of glue bottles and determine if:

1. There is a label on the bottle
2. The label has appear correctly

Figure 1: Sample image to process

Method

1 Isolating relevant parts of the image

Figure 2: Image after slicing

I initially divide the image into 5 as I want to deal with each glue bottle individually to reduce interference. I then cut the image in half so that I am only left with the part of the bottle with the label.

2 Removing the background

Figure 3: Image after slicing

While this is largely unnecessary with Canny, it was useful to remove the background while experimenting with alterative methods of isolating the label. To remove the background I simply apply adaptive thresholding.

```
65          cvtColor(section, grey_scale, CV_BGR2GRAY);
66          adaptiveThreshold(grey_scale, thresholded, 255, ADAPTIVE_THRESH_MEA
67          just_bottle = thresholdANDrgb(thresholded, section);
```

3 Get edges

Figure 4: Edges after using Canny

Canny is then used to obtain the edges. I found a large range of parameters worked for Canny but this following were the most consistent:

```
69          // get edges with canny
70          cvtColor(just_bottle, grey_bottle, CV_BGR2GRAY);
71          Canny(grey_bottle, canny, 100, 200);
```

4 Remove bottle outline

Figure 5: Bottle outline removed

Recursively remove bottle outline.

```
27 void delete_white_border(Mat edge_image, int row, int col){
28      if (row < 2 || col < 2 || row > edge_image.rows-2 || col > edge_image.r
29      edge_image.at<uchar>(row, col) = 0;
30      for (int i = row-1; i < row+2; i++){
```

```
31          for (int j = col-1; j < col+2; j++){
32              if (edge_image.at<uchar>(i, j)){
33                  delete_white_border(edge_image, i, j);
34              }
35          }
36      }
37  }
```

5 Find label rectangle

Figure 6: Lines detected

5.1 Perform Hough transform

I found the best results were with a standard HoughLines transform.

```
84          vector<Vec2f> lines;
85          HoughLines(canny, lines, 1, CV_PI/80.0, 50);
```

5.2 Find hough lines that match desired angle

I filter out lines that are not either horizontal or vertical $\pm 3°$. I do this by extracting the points from each of the lines and working out the angles manually. I then detect for at least 1 horizontal line and 1 vertical line, and if I found those then I determine that the glue bottle has a label. This is not perfect but it offers the best results.

```
bool found_vert = false;
bool found_horiz = false;
for( size_t i = 0; i < lines.size(); i++ )
{
    float rho = lines[i][0];
    float theta = lines[i][1];
    double a = cos(theta), b = sin(theta);
```

```
    double x0 = a*rho, y0 = b*rho;
    Point pt1(cvRound(x0 + 1000*(-b)), cvRound(y0 + 1000*(a)));
    Point pt2(cvRound(x0 - 1000*(-b)), cvRound(y0 - 1000*(a)));

    double Angle = atan2(pt2.y - pt1.y,pt2.x - pt1.x) * 180.0 / CV_PI;
    if(Angle<0) Angle=Angle+360;
    if (plusOrMinus(Angle, 270.0, 3.0) ||
                plusOrMinus(Angle, 90.0, 3.0)) {
        found_vert = true;
        line( section, pt1, pt2, Scalar(0,0,255), 1, 8 );
    } else if (plusOrMinus(Angle, 0.0, 3.0) ||
                plusOrMinus(Angle, 180.0, 3.0)) {
        found_horiz = true;
        line( section, pt1, pt2, Scalar(0,0,255), 1, 8 );
    }
}
```

Metrics

For the task of identifying if the label is placed correctly:

- True positives: 11
- False positives: 0
- True negatives: 17
- False negatives: 2
- *Recall*: 84%
- *Precision*: 100%
- *Accuracy*: 93%
- *Specificity*: 100%
- *Specificity*: 100%
- F_1: 93%

Attempted techniques

- *Use k-means to just get the background, label and bottle*: Did not work when label was white or with strong shadow.
- *Otsu thresholding*: illegible results
- *Sobel edge detection*: Poor results
- *Caplacian*: Too noisy, even after blur
- *Probabilistic hough transform*: Too much interference with label text