Household Object Recognition

Sai Ma (u5224340)[1]*, Qiao Zhao (u5306220)[2]*

Abstract
This paper illustrates a visual household object recognition method based on *vector space model* (VSM) and *edge color histogram* (ECH). This training method is divided into two processes. The first method is transforming training images to vector space model based on *scale invariant feature transform* (SIFT) features, and optimizing models by *term frequency-inverse document frequency* (tf-idf) method. Then, we transform each image to a VSM, which contains SIFT feature information. The second method is based on edge color histogram. We use the edges of object *hue saturation value* (HSV) color space, and transform each image to a color edge color histogram. During the object recognition process, we combine distances of vector space model and edge color histogram to optimize detection result. The experimental result shows that this household object detection method performs well.

Keywords
SIFT; Vector Space Model; TF–IDF; Edge Color Histogram

[1] *Master of Computing, Research School of Computer Science*
[2] *Master of Computing, Research School of Computer Science*

Contents

Introduction

With the development of camera technique and population of smart phone, camera plays a increasingly significant role in people's lives. For instance, people always prefer to take a photo when finding something novel. Meanwhile, computer vision is able to be used to recognize new object, and to provide relevant information. The aim of this project is to develop a robust and effective household object visual detection system. Initially, we are provided with $50 \times 4 = 200$ images of 50 household objects. For each category, we use 3 images as training images, and the rest one as the query image to test our system. Thus, when recognizing an image, this program is supposed to recognize the object in the query image and display its belonged class.

In this system, we use two approaches to calculate distances between query image and training image, which are vector space model method, and histogram of object edge. In the vector space model method, we extract SIFT features from training images, and perform *k-means* algorithm to build visual word corpus. Then, the term frequency and inverse document frequency of each visual word are built. With the above results, it is possible to build the vector space model of each image. In the meantime, we extract edge color histogram of each image.

In order to optimize these methods, we assign different weights to VSM and ECH similarities based on quantity of detected SIFT features and edge color histogram. More specifically, higher weight will be assigned to VSM distance if more SIFT features are detected.

1. Related Work

Object recognition has been applied in many areas. For building recognition, it performs well in Zubud-zurich building database, and has recognition accuracy more than 90% [1]. Moreover, SIFT matching method is also another effective

approach to perform object recognition [2] [3]. Meanwhile, information retrieval method has been applied in object recognition, and reached remarkable achievement [7].

2. Methodology

Household object recognition is a image matching problem, and its main processes are:

1. Detect object in image.
2. Object feature Extract.
3. Transform feature vector.
4. Train classifier.
5. Perform classification.

In the term project data, all images only contain one object with empty background. This condition decrease the difficult of object recognition because there is no need to detect object from image. Therefore, we could extract object feature from image, and perform recognizing directly.

2.1 Vector Space Model

The object features describe the properties of this object, and can be used build a suitable object matching. In order to describe the keypoints of household object, we apply the *Scale Invariant Feature Transform* (SIFT) descriptor [2] [3]. The SIFT descriptor has the following features::

1. Determine optimal scale (by maximizing DoG in scale and in space).
2. Find local orientation as the dominant gradient direction.
3. Use this scale and orientation to make all further computations invariant to scale and rotation.
4. Compute gradient orientation histograms of several small windows (to produce a 128-D vector for each SIFT feature).

Thus, we apply SIFT descriptor to describe keypoints from images. The following are processes on SIFT feature extraction [4]:

1. Incrementally apply Gaussian blur on the original image to create a scale space.
2. Find the difference between adjacent Gaussian images in scale space.
3. Locate keypoints which are pixels in difference images, and its value are larger than or smaller than all 26 neighbors.
4. The gradient of pixels around each keypoint is determined in Gaussian scale at which it was found.
5. Sixteen hisograms are created using the gradients, with 8 orientations, produces 128-D feature vectors.

When we extract SIFT features from training images, we notice that different images have the different quantities of features. In other words, a image is represented to a set of SIFT features. In order to make our recognition result robust and efficiently, we perform vector space model method to transform features to a set of visual words, and build a corpus to assist in object recognition. Hence, we perform *k-means* clustering algorithm to obtain cluster's centroids, and apply these centroids as visual words [5]. Algorithm **??** is how to perform k-means algorithm to construct visual word corpus.

In the Algorithm **??**, the $dist(a,b)$ is the Euler distance between two vector a and b. Note that, the k-value in k-means is unknown. Hence, we should perform a modification k-value method to get optimal k-value in term-project data set. [1]

2.1.1 tf-idf Computing

We assume that each feature is independent. So, we could use the visual word histogram (term frequency) to represent a image. Moreover, we use a visual word inverse document frequency to ignore the *common* visual word. The tf-idf method is a natural language processing algorithm, which aims to transform image to a vector space model and derive similarity between different documents [6]. In the Computer Vision, we can perform same approach to achieve similarities between different images [7].

In tf-idf method, it transforms an image j, which contains a set of visual words w, we use vector space model as Equation 1 to describe a image.

$$Ij = \{w_{1,j}, w_{2,j}, ..., w_{n,j}\} \tag{1}$$

where n is the size of visual word corpus, and $w_{i,j} = w_{tf,i,j} \times w_{idf,i}$, $w_{tf,i,j}$ is the histogram of visual word i in image j, which is described in Equation 2.

$$w_{tf,i,j} = \frac{n_{i,j}}{n_j} \tag{2}$$

where $n_{i,j}$ is the number of visual word i in image j, and n_j is the number of SIFT features of image j. In the recognition, $w_{tf,i,j}$ is used to describe the weight of visual word i in the image j. Then, we use $w_{idf,i}$ to describe the weight of word i in the all training images.

$$w_{idf,i} = log[\frac{N}{n_i}] \tag{3}$$

where N is number of training image, and n_i is the quantity of training images which contain visual word i. According to Equation 3, it is easy to notice that when a visual word i contributes nothing in object recognition appear in every training image, which cannot make contribute to detect object, and $w_{idf,i} = log[N/N]$.

The entire process on transforming training images to VSM is described in Figure 1.

Algorithm 1, Algorithm 2 and Algorithm 3 are the main processes on how to perform tf-idf algorithm, and transform a image to a vector space model.

Algorithm 1 Build Vector Space Model

Require: $k, \{I_1, ...I_n\}$
Ensure: $\{VSM_1, ...VSM_n\}$
function TF-IDF($k, \{I_1, ...I_n\}$)
  for I_j in $\{I_1, ...I_n\}$ **do**
    $TotalSIFT \leftarrow getSIFT(Image_j)$
  end for
  $\{word_i, ...word_k\} \leftarrow$ K-MEANS($k, TotalSIFT$)
  $\{w_{tf,1}, ...w_{tf,n}\} \leftarrow$TF($\{I_1, ...I_n\}$, $\{word_i, ...word_k\}$)
  $\{w_{idf,1}, ...w_{idf,k}\} \leftarrow$ IDF($\{word_i, ...word_k\}$, $\{w_{tf,1}, ...w_{tf,n}\}$)
  for $image_j$ in $\{I_1, ...I_n\}$ **do**
    for $word_i$ in $\{w_{tf,1,j}, ...w_{tf,1,k}\}$ **do**
      $VSM_{i,j} = w_{tf,i,j} \times w_{idf,i}$
    end for
  end for
  return $\{VSM_1, ...VSM_n\}$
end function

Algorithm 2 TF

Require: $\{Image_1, ...Image_n\}$, $\{word_i, ...word_k\}$
Ensure: $\{w_{tf,1}, ...w_{tf,k}\}$
function TF-IDF($\{Image_1, ...Image_n\}, \{word_i, ...word_k\}$)
  for $Image_j$ in $\{Image_1, ...Image_n\}$ **do**
    for f in $Image_j$ **do**
      $i = argmin(dist(f, word_1, ...))$
      $term_i = term_i + 1$
    end for
    for $term_i$ in $Image_j$ **do**
      $w_{tf,i,j} = term_i / n_j$
    end for
  end for
  return $\{w_{tf,1}, ...w_{tf,k}\}$
end function

Algorithm 3 IDF

Require: $\{word_i, ...word_k\}$, $\{w_{tf,1}, ...w_{tf,n}\}$
Ensure: $\{w_{idf,1}, ...w_{idf,k}\}$
function TF-IDF($\{word_i, ...word_k\}, \{w_{tf,1}, ...w_{tf,n}\}$)
  for $word_i$ in $\{word_i, ...word_k\}$ **do**
    for $image_j$ in $\{I_1, ...I_n\}$ **do**
      if $w_{tf,i,j} > 0$ **then**
        $df_i = df_i + 1$
      end if
    end for
    $w_{idf,i} = log[\frac{N}{df_i}]$
  end for
  return $\{w_{idf,1}, ...w_{idf,k}\}$
end function

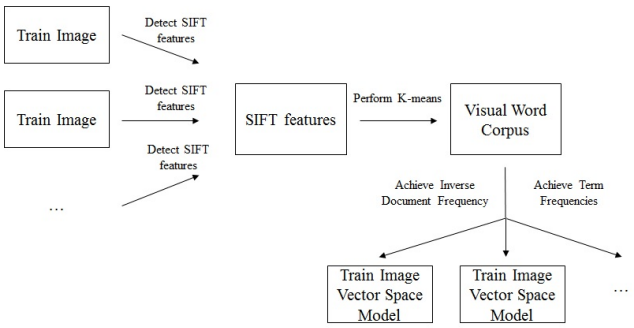

Figure 1. Transform Images to Vector Space Model

After performing these algorithms, each image can be transformed to a vector, and vector distance method to obtain the similarity between different images.

2.2 Edge Color Histogram

In the previous object detection method, VSM, we did not use the information on colors and edges of the object. In this section, we define a method using both of the information improve our object recognition. The following two steps are used to add color information HSV [8], and compute similarity between two images based on edges and colors.

We use the edge and color features as representation of each image. Since HSV model is similar to human vision, we select HSV space as our color model. However, when RGB values of the image are similar, singular value may occur in HSV space. Thus, to overcome this drawback, we add another 32 level G channel based on V channel. Besides, we adopt 12 level H channel, 2 level S channel and 3 level V channel based several experiments. And these levels are appropriate for representing the color feature of a image and do not slow down the computing too much.

The colors in the image plays a significant role in discriminating between household objects and hence the colors are normalized to get a better color representation. Also, the values are whose color value is lower than ten percent of the maximum are set to zero to eliminate noise. To obtain the color distribution in space, we project every pixel in both horizontal and vertical direction to derive the color histogram in these two directions. Finally, each image is represented as $B = \{B_H, B_V\}$.

2.2.1 ECH Computing

As we previously demonstrates, in order to represent an image in edge and color space, we should perform the following steps:

1. Represent the image in HSV space.
2. Set values of all pixels whose color values are lower than ten percent of the maximum to zero to eliminate the noise.

[1] We randomly choose test image and train images, and repeat 200 times. Then we achieve 1,150 is the best choice in this data set.

Figure 2. Edges of an Image

3. Transfer H channel to 12 levels, S channel to 2 levels, V channel to 3 levels and G channel to 32 levels based on V channel.
4. Get horizontal and vertical edges of the image.
5. Count the number of pixels within every color level in each channel which are also edge pixels , and divide it with the total number of edge pixels in the specific direction to get the edge color histogram B.
6. Normalize the histogram B.

Figure 2 shows edges which we detected. After performing the above processing, an edge color histogram is acquired. Algorithm 4 shows how to perform this method.

Algorithm 4 Edge Color Histogram

Require: $Image$
Ensure: $\{hist_H, hist_S, hist_V, hist_G\}$
function ECH($Image$)
  h,s,v are H, S, V value of Image
  for c in h,s,v **do**
    for p in c **do**
      if $p < 0.1 * max(c)$ **then**
        $p = 0$
      end if
    end for
  end for
  $g = ceil(v*32), h = ceil(h*12)$
  $s = ceil(s*2), v = ceil(v*3)$
  $Edge_H, Edge_V = edge(image)$
  for e in $Edge_H, Edge_V$ **do**
    for c in h,s,v,g **do**
      for $level$ in c **do**
        $n = 0$
        for p in c **do**
          if p in edge && p in level **then**
            $n = n+1$
          end if
        end for
        $hist_c = n/sum(e)$
      end for
    end for
  end for
  $hist = hist/norm(hist)$
  return $\{hist_H, hist_S, hist_V, hist_G\}$
end function

3. Image Similarity

After we perform VSM and ECH approaches, each image can be transformed to two matrices which are SIFT features and edge color histogram respectively. The next task is to compute distances between these two matrices, and derive the similarity between two images.

3.1 Similarity Based on VSM

Cosine similarity is an effective measure of similarity between two vectors, and it aims to calculate the cosine of the angle between two vectors. It is also a common approach in NLP to compare two document similarities [9]. Thus, we use this method to compute VSM distances. The details on computing distance between two VSMs are in Equation 4:

$$cosine_{A,B} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

Note that, in cosine similarity, the higher the similarity is (range of cosine similarity is from 0 to 1), the more similar the two vectors are. However, considering the method introduced in section 3.2, we use one minus cosine similarity between two images as distance between two images.

$$sim_{SVM(A,B)} = 1 - cosine_{A,B} \tag{5}$$

3.2 Similarity Based on ECH

After the computation of ECH of training images and test images, we calculate the similarity between two images I_{test} and I_{train} as follows [10]:

1. Set the weight of horizontal edges W_H and vertical edges W_V where $W_H + W_V = 1$ and normally, we set $W_H = 0.4$ since vertical edges are more important for discriminating most of the household objects.

2. The distance of two directions between two images is as the following:

$$d_H = 1 - min(\frac{B_{testH} - B_{trainH}}{sum(B_{testH})}) \\ d_V = 1 - min(\frac{B_{testV} - B_{trainV}}{sum(B_{testV})}) \tag{6}$$

3. Then we combine horizontal distance and vertical distance together by Equation 7.

$$sim_{ECH} = \frac{w_H d_H + w_V d_V}{w_H + w_V} \tag{7}$$

where $w_H + w_V = 1$. Hence, the closer the distance is to 1, the more similar these two images are.

Algorithm 5 illustrates how to compute ECH similarity.

Algorithm 5 Similarity Based on Edge Color Histogram

Require: $\{W_H, hist_{train}, hist_{test}\}$
Ensure: $similarity$
function SIM($\{W_H, hist_{train}, hist_{test}\}$)
$W_V = 1 - W_H$
$sim_H, sim_V = 1 - min(hist_{train} - hist_{test})/sum(hist_{test})$
$similarity = (W_H * sim_H + W_V * sim_V)/(W_H + W_V)$
return $similarity$
end function

3.3 Combing VSM and ECH

However, these two methods have their weaknesses. For the VSM method, it only focuses on keypoints which are in grey scale, but contain no edge and color information. Meanwhile, the ECH method only takes edges and colors into account, but does not takes care of object shape. Therefore, we require a method which have these two method advantages.

After we get similarities sim_{VSM} and sim_{ECH} between images from ECH and VSM respectively, we calculate the total similarity as follows:

$$sim = \frac{\alpha sim_{VSM} + \beta sim_{ECH}}{\alpha + \beta} \tag{8}$$

In Equation 8, we have the relationship between α and β as Equation 9.

$$\frac{\alpha}{\beta} = \frac{c \times n}{w \times h} \tag{9}$$

where $\alpha + \beta = 1$. The argument c denotes the coefficient decided by the experiment, n is the number of interest points output by SIFT descriptor and w, h are the width and height of the image respectively.

In this method, we calculate the similarity between two objects with SIFT features and color histogram, and it can make search result more effective. Figure 3 shows the main process on combining two similarity method to improve object recognition.

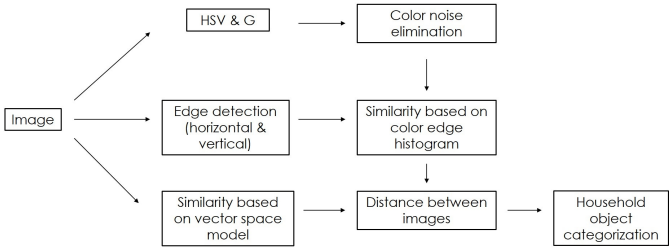

Figure 3. Achieve Image Total Similarity

4. Experiment and Result

In this term project data-set, we have 200 images in 50 classes. In order to verify the method performance, we randomly select one image from each image class, and use the rest as training images. Then perform classification method based on VSM, ECH and combined method in our previous sections.

In these three methods, we select the training image i which is most similar to the testing image, and categorize this testing image j to the class of image i. After labeling all testing images, we compare the labeled classes with the original classes to visualize the correctness.

4.1 Vector Space Model Method Result

In this part, we perform SVM classification. In this experiment, we repeat and compare the correctness at different operation time. Figure 4 is the correctness changes during 20 experiments.

From this figure, it is easy to be observed that even with the same k-value, the VSM method classification result is not stable. The reason is because we use k-means clustering algorithm, and its result is based on initial random sampling [11]. Moreover, the value k is hard to select in different training image sets, and hence, we need to repeat this procedure to get an optimal value. Therefore, it makes the classification result unstable. In addition, k-means clustering may causes running time problem. In Table 1, it shows the running time with different k values.

Each experiment will take more than 300 seconds.

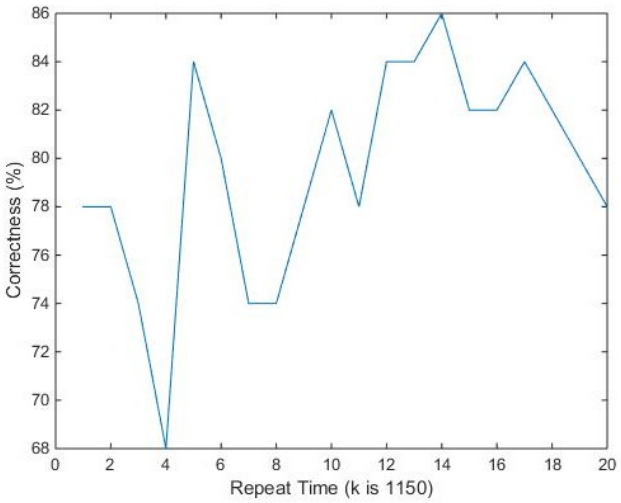

Figure 4. VSM Method Classification Correctness

Table 1. VSM Opeartion Time

k-value	Training Time	Test Time
900	45.33s	246.15s
1,000	46.64s	253.76
1,100	48.83s	271.94
1,200	53.00s	277.22

4.2 Edge Color Histogram Method Result

In this part, we perform ECH classification, and compared the correctness for different weight of horizontal edges. The result is as in Figure 5.

And then, we ran this process for 100 times to get a better statistics. The statistics is as in Figure 6.

Finally, we obtain the mean of the statistics and the result is as in Figure 7.

As can be seen, when the horizontal weight is between 0.3 and 0.5, the best correctness can be observed. Hence, we set the horizontal weight as 0.4 in ECH method, and achieve 94% correctness in optimal condition.

4.3 Combine Method Result

In this part, we perform combine method to calculate the total similarity, after which we compare the correctness of VSM and ECH method results. The result is shown in Figure 8.

Considering VSM and ECH methods result, we found that combined method performed better than separate methods. For example, in the time 1, the VSM correctness is 78% while ECH is 62%, which does not perform well in this data set. However, after we combine these two similarities, the correctness changes to 94%, which is much better than previous results.

In addition, Figure 10 shows object recognition result during demo.

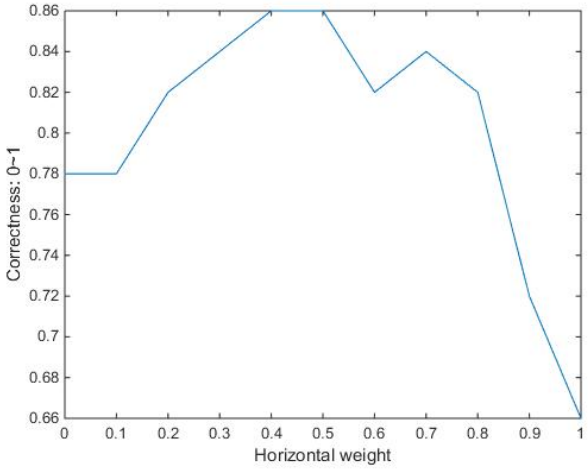

Figure 5. ECH Method Classification Correctness Sample

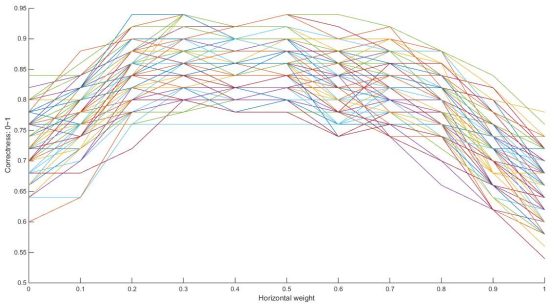

Figure 6. ECH Method Classification Correctness After Repeat 100 times

5. Conclusion

In this paper, we introduce three methods to perform household recognition. For the first method, we extract SIFT feature from images, and perform tf-idf algorithm to build a vector space model of each image. Then,cosine similarity between these VSMs is computed. After that, we use object edge HSV values to build edge color histogram. The last method combines the above two similarities to perform the classification.

These three method all work well in this term project data set. However, the VSM and ECH methods have their weaknesses. In VSM, it performs k-means clustering algorithm to get an effective classification. In ECH, it only contains the edge color information, which ignore the keypoints in objects. But by combining these two similarities with different weights, it achieves a better result than original methods, although it does not overcome the running time problem.

6. Appendix

6.1 Learning Outcome

After finishing this term project, we grasped the knowledge on transforming a image to a vector space model, applying edge color histogram of an object, and various strategies to compute

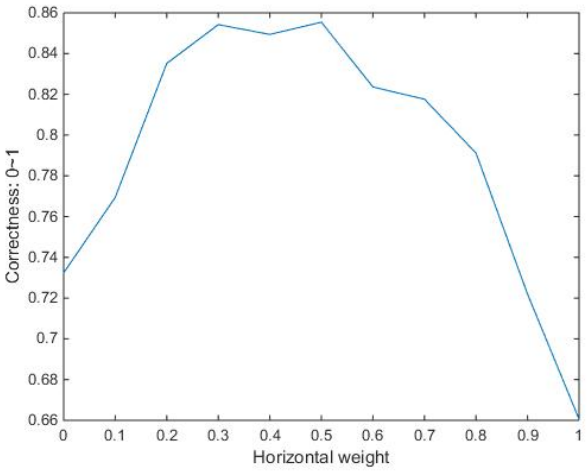

Figure 7. ECH Method Classification Mean Correctness

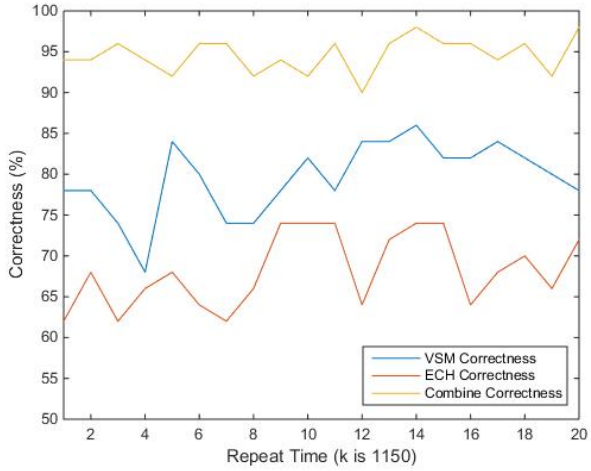

Figure 8. Combine Method, VSM Method and ECH Method Result

vector inner distances. Moreover, we learnt the process on executing experiments in limited data as well as adjusting parameters in algorithms. We also learnt the fundamental knowledge on performing object recognition. Finally, we applied variety of algorithms in opencv-python, and used Matlab engine in python.

6.2 How to Run Source Code

Please read README file in source code folder.

References

[1] Hao Shao, Tomáš Svoboda, and Luc Van Gool. Zubud-zurich buildings database for image based recognition. *Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep*, 260, 2003.

[2] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

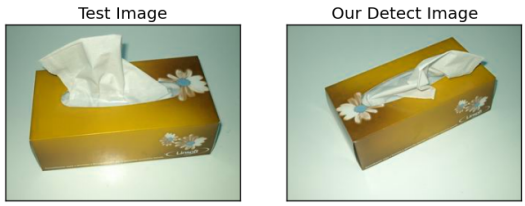

Figure 9. Object Recognition Sample - 1

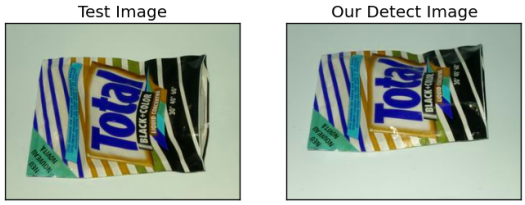

Figure 10. Object Recognition Sample - 2

[3] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.

[4] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[5] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[6] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

[7] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[8] Scott E Umbaugh. *Digital image processing and analysis: human and computer vision applications with CVIPtools*. CRC press, 2010.

[9] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[10] Seong-O Shim and Tae-Sun Choi. Edge color histogram for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages 957–960. IEEE, 2002.

[11] Eric Nowak, Frédéric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, pages 490–503. Springer, 2006.