Extracting Product Information Using Logo Recognition

Kinjal Kotadia
Department of Computer Science
University of Houston

Sagar Limaye
Department of Computer Science
University of Houston

Chien Chen
Department of Computer Science
University of Houston

Gaurav Roy
Department of Electrical and Computer Engineering
University of Houston

Raj Shah
Department of Electrical and Computer Engineering
University of Houston

***Abstract*—This paper proposes a logo identification algorithm using SIFT (Scale Invariant Feature Transform) to recognize logos in an image, which can serve as a helpful tool to extract information about the product from a database or the internet. SIFT is a robust algorithm which works very well under illumination or rotation or scale changes.**

***Keywords*—SIFT; keypoints; descriptors; localized;**

I. Introduction

A very useful application of computer vision algorithms is to automate the process of object detection and recognition. We have, in our project extended that idea to the field of brand logos. Every brand logo has a specific design and a specific color distribution that is unique to the logo. We decided to use these attributes to classify or cluster the logos depending on the kind of feature. The different logos used in our project are Coca-Cola, Pepsi, Starbucks, Sprite, Heineken, Fanta and Redbull. The next section explains about the two features considered and talks about their level of robustness. The third section talks in detail about SIFT features, how it works, and how it is implemented. In the fourth section, observations from 69 test images have been tabulated. Based on the observations, the accuracy of the algorithm is determined.

II. Approach

A. Color Histograms

Our first approach was using Color Histograms which was proposed as an initial identification mechanism in other papers. For example, Daniela Hall *et al* [1] proposes two modules for logo identification and detection respectively. For logo identification, a feature vector of eight scale normalized receptive fields is used. For logo detection, a two-dimensional histogram of pixel chrominance is determined. Although this proposal has a decent accuracy, there were many false positives that the system detects. So, color histograms, by itself is not a definitive feature because it is prone to false detection due to illumination changes. Furthermore, we found that histograms of different logos got grouped together due to background histogram matching.

B. Scale Invariant Feature Transform

Since Color Histograms was not efficient, Scale Invariant Feature Transform was considered. SIFT is a much more robust method of identifying abstract shapes because it uses localized features. It generates key point descriptors after obtaining maxima and minima from LoG computations of consecutive scale and octave values [2]. After that, it computes localized gradients and converts it into a 1-dimensional vector. The descriptor therefore, is scale and orientation invariant in addition to being illumination invariant.

III. Methodology

Logo detection and identification is expected to be a robust application because images of logos (or objects) change depending on the distance and angle at which the picture is taken. So, for robust detection localized features are better suited than globalized features. Pixel intensity values cannot be used as features because they will change from image to image.

SIFT features correspond to “parts” of images, at a more holistic level than intensity values. They are not sensitive to changes in image resolution, scale, rotation, changes in illumination (e.g., position of lights). To model logos that are abstract in nature, we have used SIFT features. In the following subsections, an explanation of SIFT working is shown in logo detection [2].

A. *Creating Scale Space*

The scale space is designed by varying the image size and blurring the image with different values. Different image sizes are called octaves and the amount of blurring on each image is determined by a factor σ. Each octave’s image size is half the previous one. For SIFT algorithm 4 octaves and 5 blur levels are ideal.

B. *Laplacian of Gaussian*

The Laplacian of Gaussian is computed by finding out the difference of Gaussians or the difference in the various blurring levels obtained in the earlier subsection. The original equation for Gaussian blurring is given as

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

Where x and y are the pixel dimensions and σ is the blur factor. The G (x, y, σ) is calculated by subtracting the different blurred images of a single octave. The scale-invariant Laplacian of Gaussian is given as $\sigma^2\nabla^2 G$.

C. *Finding Keypoints*

Keypoints are located by detecting maxima and minima in the DOG images obtained from previous step. This is done by iterating through each pixel and checking its neighbors for maxima and minima. The check is done within the current image, and the one above and below it. This way, a total of 26 checks are made. The pixel is marked as a keypoint if it is the greatest or least of all 26 neighbors.

D. *Keypoint Orientations*

After ensuring scale-invariance, SIFT also addresses the changes due to orientation. This is done by collecting gradient magnitudes and directions around each keypoint. Then the most prominent orientation(s) is assigned to the keypoint. This ensures rotation-invariance. The magnitude and orientation are calculated using the following formulae

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

where,

m(x, y) is the magnitude of the gradient,

Θ(x, y) is the orientation of the gradient.

The histogram created is broken down into bins of 10 degrees and the magnitude is assigned to each bin.

E. *Feature Generation*

The keypoints are uniquely identified using feature vectors that has 128 values each. It consists of a 16x16 window around the keypoint which is further broken down into sixteen 4x4 windows. Within each 4x4 window, gradient magnitudes and orientations are calculated. These orientations are put into an 8-bin histogram. The amount added also depends on the distance from the keypoint. So, gradients that are far away from the keypoint will add smaller values to the histogram. This is done using a Gaussian weighting function.

F. Brute Force Matching

After the feature descriptors are generated for each keypoint, they are matched to the keypoints of the test image using brute force method. The Brute force matcher takes the descriptor of one feature in first set and matches it with all other features in second set using a distance calculation [3]. For SIFT features, L2 norm is used which computes Euclidean distance as a distance metric to compare the two keypoint features.

G. Thresholding

To classify a test image as containing a particular logo, a threshold value, T_r of the distance metric was empirically set. All the matches of logos below T_r are considered for further processing, and those above T_r are rejected. To reduce the number of false positives another level of constraint is added. If the test image's keypoints match with more than those of one logo, the logo with the maximum number of matches are considered. Furthermore, if two or more logos have the same maximum number of descriptors matched, the one with the lowest distance metric value is selected. For our application, the T_r value is set to 200.

IV. OBSERVATIONS

One of the main advantages of SIFT over other algorithms is that you don't need a large dataset to train your **classifier**. In our case, just one image per logo was enough to obtain effective keypoints. However, the ideal logo image should be carefully chosen to make sure that keypoints obtained are sufficient in identifying the logos in other images. Figure 1. shows a few ideal images used.

Figure 1. Ideal logos

Keypoints are then located on the ideal images as shown below in Figure 2.

Figure 2. Keypoints on Starbucks and Coca-Cola

Finally, using the brute-force matching technique the keypoints obtained from the ideal images are matched with the keypoints of the test images as shown in Figure 3 and the products are identified.

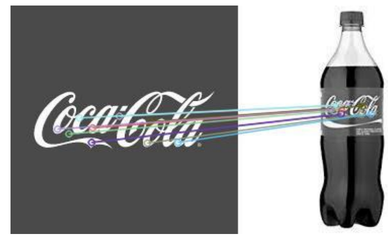

Figure 3. Keypoint matching

As we can see in the following images, SIFT also does a good job with rotated and scaled images.

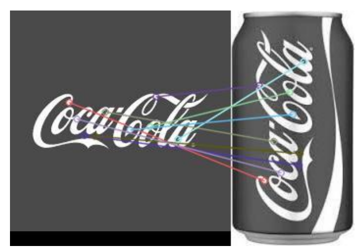

Figure 4. Rotation Invariance

Figure 5. Scale Invariance

However, the keypoints of different logos were also matching with the test image as shown in the following pictures.

Figure 6. Mismatched Keypoints

In order to avoid wrong identification, different constraints described in the previous section were used. Some final results of our observations can be found in the figures Figure7 and Figure 8 below. To access the rest, please open the directory – "Results".

Figure 7. Positive identification

Figure 8. False identification

V. Results

As mentioned before, our dataset consists of 7 different ideal logo images (one image per logo) and 69 test images. Based on our observations, we recorded 55 positive identifications and 14 false detections. The following table shows a part of our observations. The entire table is added in "Results" directory.

Table 1. Results

Image Number	Ground truth	Detection
0	Coca-Cola	Yes
1	Coca-Cola	Yes
2	Starbucks	Yes
3	Sprite	Yes
4	Fanta	Yes
5	Coca-Cola	Yes
6	Sprite	Yes

The accuracy is calculated as,

$$Accuracy = \frac{Number\ of\ positive\ identifications}{Total\ number\ of\ test\ images}$$

So, the accuracy of our algorithm is approximately 80%.

VI. Discussion

The logo identification using SIFT in our algorithm is effective in recognizing the logos. In addition to being invariant to rotation, illumination and scale it also works with skew transformations and excessive background detail. However, it has a few limitations. Since the result only shows single detection, it fails when there are multiple logos in the image. Moreover, there is always a risk of false detections if there are a lot of keypoints.

VII. References

[1] Machine Vision And Applications Manuscript No. "Brand Identification Using Gaussian Derivative Histograms." (n.d.): n. pag. Web.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Sinha, Utkarsh. "Introduction." *AI Shack - Tutorials for OpenCV, Computer Vision, Deep Learning, Image Processing, Neural Networks and Artificial Intelligence*. N.p., n.d. Web. 04 May 2017.

[3] *Differences between L1 and L2 as Loss Function and Regularization*. N.p., n.d. Web. 04 May 2017