Logo Recognition Technique using Sift Descriptor, Surf Descriptor and Hog Descriptor

Chinmoy Biswas
Jadavpur University
Kolkata
India

Joydeep Mukherjee
Jadavpur University
Kolkata
India

ABSTRACT

Logos sometimes also known as trademark have high importance in today's marketing world. Logo or trademark is of high importance because it carries the goodwill of the company and the product. Logo matching and recognition is important to discover either improper or unauthorized use of logos. Query images may come with different types of scale, rotation, affine distortion, illumination noise, highly occluded noise. Sift descriptor, surf descriptor and hog descriptor are very good features to use among the existing techniques to recognize the logo images from such difficulties more accurately.

General Terms

Logo Recognition, invariant to scale, rotation, invariant to illumination noise, occluded objects .

Keywords

KeyPoint localization, KeyPoint descriptor, Interest Point Descriptor

1. INTRODUCTION

Logos of different formats and styles exist in the database. To recognize the different types of logo images of different logo features can be used for extracting the feature values. From the training and testing logo images similar set of features are extracted and matching algorithm is used to match the similarity between training and testing logo images.

In this study, experimentations of logo images are done with three existing technique, which is Scale invariant feature transform (SIFT), Speeded up robust feature (SURF) and Histogram oriented Gradient (HOG). For extracting the shape based feature vectors of testing and training logo images scale invariant feature transform (SIFT), Speeded up robust feature (SURF) and Histogram Oriented Gradient (HOG) is used. Finally, Manhattan Distance classifier is used to measure the similarity between the sample images of training logo images and testing logo images of the classes.

2. METHODOLOGY

For experimentations of logo recognition Logo dataset from University of Maryland [19], google images and flickrlogos are used which includes 110 logo images in the TIFF format and resize to 150×150 pixel. Convert images into Binary image and into double.

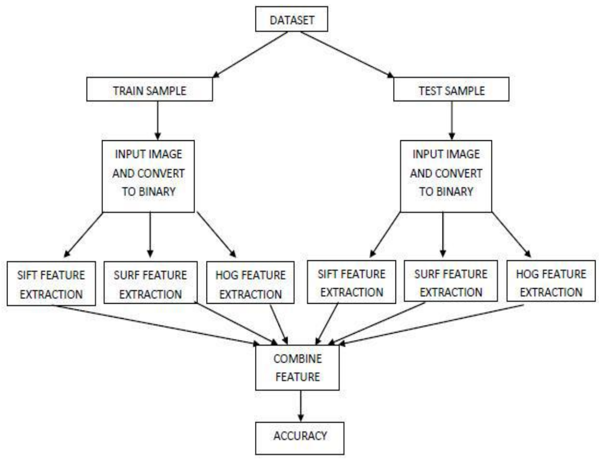

Figure 1a. Flow Chart

2.1. Scale Invariant Feature Transform

SIFT Keypoints are invariant to scale, rotation, invariant to illumination differences and partially occluded objects.

As David G. Lowe described SIFT [12] consists of four major stages:

(1) Scale-space peak selection

(2) KeyPoint localization

(3) Orientation assignment

(4) KeyPoint descriptor

First of all scale space have to be generated. To generate scale spaces apply Gaussian over sample logo image with varying σ value. Then down sample it and find Laplacian of Gaussian. To efficiently locate the key point location in scale space difference of Gaussian $D(x,y,\sigma)$(DOG), is calculated from differences of two nearby scale separated by a multiplicative factor. A point is decided as a local minimum or maximum with respect to neighborhood points. The scale of the KeyPoint is used to select the Gaussian smoothed image, with the closest scale, so that all computations are performed in a scale-invariant manner. For each image samples, the gradient magnitude and orientation are calculated. An orientation histogram is formed from the gradient orientation of the sample points within a region around the KeyPoint. Then a distinctive descriptor is achieved for each KeyPoint. The image gradient and magnitudes and orientations, relative to major orientation of the KeyPoint are sampled within a 16×16 region. Each sample is weighted by its gradient magnitude and by a Gaussian circular window centered at the local maximum. These samples are then accumulated into orientation histogram summarizing the contents into 4×4 sub region. Each histogram has 8 bins covering 360 degree of orientation. Therefore having a 4×4×8=128 feature vector is generated for each KeyPoint.

2.2. Histogram Oriented Gradient

Histogram of Oriented Gradients (HOG) [17] is inspired on Scale-Invariant Feature Transform (SIFT) descriptors proposed by [15]. To compose HOG, the cell histograms of each pixel within the cell cast a weighted vote. In this work the histogram channels are calculated over rectangular cells by the computation of unsigned gradient. The cells overlap half of their area, meaning that each cell contributes more than once to the final feature vector. In order to account for changes in illumination and contrast, the gradient strengths were locally normalized, i.e. normalized over each cell. The nine histograms with nine bins were then concatenated to make a 1x81 dimensional feature vector.

2.3. Speeded up Robust Feature

The major steps of SURF are,

Fast-Hessian Detector

Constructing the Scale-Space

Accurate Interest Point Localization

Interest Point Descriptor

Orientation assignment

Descriptor Components

Much of the performance increase in SURF is due to Integral Image [8]. The integral image $I\Sigma(x,y)$ of an image $I(x, y)$ represents the sum of all pixels in $I(x,y)$ of a rectangular region formed by (0,0) and (x,y). Now calculate the Hessian matrix, as function of both space and scale. Then calculate Hessian determinant using the approximated Gaussians. The task of localizing the scale and rotation invariant interest points in the image can be divided into three steps. First determine threshold value for the responses such that all values below the predetermined threshold are removed. Then, find a set of candidate points. To do this each pixel in the scale-space is compared to its 26 neighbors, comprised of the 8 points in the native scale and the 9 in each of the scales above and below. Once interest points have been localized both in space and scale, the next steps are:

(1) KeyPoint descriptor

(2) Orientation assignment

The SURF descriptor describes how the pixel intensities are distributed within a scale dependent neighborhood of each interest point detected by the Fast-Hessian. This approach is similar to that of SIFT but integral images used in conjunction with filters known as Haar wavelets are used in order to increase robustness and decrease computation time. In order to achieve invariance to image rotation each detected interest point is assigned a reproducible orientation.

- The image is convoluted with two first-order Haar wavelets.

- The filter responses at certain sampling points around the KeyPoint are represented as a vector in a two-dimensional space.

- A rotating window of 60° is used to sum up all vectors within its range, and the longest resulting vector determines the orientation.

Now the present worker has Interest Point descriptor vector of length 1x64 for each Interest Point. For a image the present worker have got 20-120 Interest Points. Suppose for image no. of Interest Points are 80. So a 2D matrix D1 of size 64x80 is generated.

AB=norm(D1(x,:));

D3(1,x)=AB; Where x=64;

And a scalar value from Orientation assignment. Combining these two features the present worker has got a vector of length 1X65 for each Image sample.

3. Training Phase

In the experiment total 10 classes has taken and to form training set total training set 5 images is used to form training set per class. First of all the images are resized to 150x150 pixels ,converted binary image and SIFT descriptor, SURF descriptor and HOG descriptors are calculated for first 5 image belongs to a particular class where feature vector's size is 128+65+81=274.

fv_t(SA, : ,CL)=F1;

F1 is feature vector of length 1X274.SA represents training sample no and CL represents class no. and : is for feature vector .

fv_t1=fv_t(:,:,1);

fv_t1 contains all 5 feature vectors of 5 training images of a class1.fv_ti (i=1 to 10)contains all 5 feature vectors of 5 training images of a class. Now size of fv_t1=5x274.So mean value, mn_t1 is calculated for fv_t1.Similiarly mean value is calculated for all classes, mn_ti=mean (fv_ti); And those mean are the feature vector of size 1×274.

mn_t1=mean(fv_t1);

size of(mn_t1) is 1x274 .

4. Testing Phase

For testing 6 images are chosen per class out of eleven images and feature vector is extracted for each 6 image belongs to ten different classes. The feature are then stored into a different variable. This feature vector can be referred as fv_sij - Where i= Test image number and j= Class number.

5. Classification Phase

After feature vector (vector size 274) extraction of each testing images, this vector are then subtracted from mean of different class (Feature vector of size 274) using Manhattan distance. Using Manhattan distance, difference of two feature vector of size 274 is finally converted into scalar. Each difference is stored in different variable.

mn_ti= Mean value of ith class (Feature vector of size 274) The differences and variables are shown below:

dijk = Difference of ith testing image of jth Testing class with mean of kth Training class

In this phase 10 different classes are taken and 60 images present per class out of which 5 images are used for training and 6 images is used for testing. Using the proposed approach described in this paper 274 element feature vector is calculated for each training sample of each class. After that mean of all 5 training samples are calculated for all 10 class. After that 274 element feature vector is calculated for each 6 samples of each 10 class. Then difference is calculated using the equation,

dijk=sum(abs(fv_sij - mn_tk));

and some example of calculations is given below for first sample of class 1 with all ten classes:

First sample of first class with mean of the first class:

For j= 1 to 6 test class

For i= 1to 6 test sample

For k= 1 to 10 mean of train class

dijk=(abs value(1st element of 274 element vector of ith test sample of jth class- 1st element of 274 element vector of mean of class k)+abs value(2nd element of 274 element vector of ith test sample of jth class- 2nd element of 274 element vector of mean of class kth)+……………………………………………………….+abs value (273th element of 274 element vector of ith test sample of jth class - 273th element of 274 element vector of mean of train kth class)+abs value(274^{th} element of 274 element vector of ith test sample of jth class - 274th element of 274 vector of mean of train class kth))

After calculation of difference between testing image and mean of training image (belongs to different class), it is to be identified that all testing image of a particular class belongs to that particular class or not and it is the responsibility of the identifier to correctly identify an image whether an image belongs to the proper class or not.

Now to calculate the accuracy individually for every class we have to check the following code:

```
count_1=0;
if min([d111, d112, d113, d114, d115, d116, d117, d118, d119,
d1110]) == d111, count_1 = count_1+1;
```

It identifies that first testing sample is a member of first class and in that case value of counter count_1 is incremented by 1. Same process calculated for all testing sample of first class and if the minimum value then the counter is incremented by 1. This count_1 is divided with number of testing sample for class 1 which is 6. After that this division is multiplied with 100 to calculate the percentage accuracy for class 1 and stored in variable acc(1). Same process is applied for all testing samples for all classes and the accuracy is calculated.

After calculation of percentage accuracy of different classes, mean of all accuracy are calculated and stored in a variable acc_comb.

Class1	d111	d112	d113	d114	d115
Sample1	34.64	44.05	64.58	58.22	44.63
Class1	d116	d117	d118	d119	d1110
Sample1	41.59	35.33	55.55	41.77	50.88

Table 1a. <u>Testing Sample1 of Class1 identified</u>

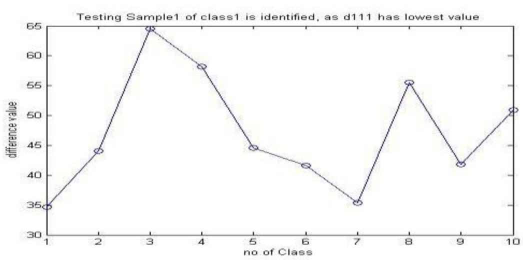

Figure 1b. <u>testing sample 1 of class1 is identified as d111 has lowest value</u>

This tabulation shows that value of d111, d211, d311, d511, d611 has the least value in the respective table. This identify that scalar value of the difference between first testing image of first class and mean of first class(from training set) is minimum. So the first testing image can be identified as a part of first class, because in this classification process the minimum value of the difference is taken, and if a particular testing image has minimum difference with a particular class then it signifies that the image belongs to that particular class.

6. EXPERIMENTAL RESULT

So overall accuracy for all classes using the Manhattan distance is 96.66% . It has been checked that different feature extraction process and there combination gives different accuracy for all classes .But in all cases classifier is Manhattan distance.

<u>Process</u>	**<u>Percentage Accuracy</u>**
Only SIFT	56.66%
Only HOG	76.66%
Only SURF	81.66%
SIFT with HOG	73.33%
SIFT with SURF	83.33%
SURF with HOG	85.00%
SIFT , HOG and SURF	96.66%

Table 1b. <u>Percentage Accuracy with different method</u>

Figure 1c. <u>Testing samples</u>

7. CONCLUSIONS AND FUTURE SCOPE

From experiment with different logos, it has been seen that this code is not very much invariant to illumination. It can tolerate little bit illumination noise. In future the code could be improved by making it more illumination invariant.

8. REFERENCES

[1] Sina Hassanzadeh, Hossein Pourghassem, “A Fast Logo Recognition Algorithm in Noisy Document Images, International Conference on Intelligent Computation and Bio-Medical Instrumentation,2011pp.64-67.

[2] Aya Soffer and Hanan Samet, “Using Negative Shape Features for Logo Similarity Matching, ”4th International Conference on Pattern Recognition, Brisbane, Qld., vol: 1, pp. 571-573, 1998.

[3] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: CVPR. Volume 2. (2003) 257 – 263

[4] Jan Neuman, Hanan Samet, and Aya Soffer, “Integration of local and Global Shape Analysis for Logo classification”, 4th International Workshop on Visual Form, Capri,Italy, pp. 769-778, 2001.

[5] Guangyu Zhu and David Doermann, “Automatic Document Logo Detection”, 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, pp.864-868, 2007.

[6] Liangfu Xia, Feihu Qi, Qianhao Zhou, “A Learning-based Logo Recognition Algorithm Using SIFT and Efficient Correspondence Matching”, International Conference on Information and Automation, Changsha, Hunan, pp. 1767-1772, 2008.

[7] Wenju Li and Ling Li, “A Novel Approach for Vehicle-logo Location Based on Edge Detection and Morphological Filter”, 2nd International Symposium on Electronic Commerce and Security, Nanchang, vol: 1, pp. 343-345, 2009.

[8] Nabeel Younus Khan,Brendan McCane,Geoff Wyvill SIFT and SURF Performance Evaluation Against Various Image Deformations on Benchmark Dataset, International Conference on Digital Image Computing: Techniques and Applications,2011 ,PP.501-506.

[9] Syed Yasser Arafat, Syed Afaq Husain, Iftikhar Azim Niaz and Muhammad Saleem, “Logo Detection and Recognition in Video Stream”, 5th International Conference on Digital Information Management, Thunder Bay, Canada, pp. 163 – 168, 2010.

[10] Liangfu Xia, Feihu Qi, Qianhao Zhou,“A Learning-based Logo Recognition Algorithm Using SIFT and Efficient Correspondence Matching “,IEEE International Conference on Information and Automation,2008,PP-1767-1773

[11] Dalal, N. and Triggs, B., Histograms of oriented gradients for human detection, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 886893

[12] Stefan Romberg,” From Local Features To Local Regions”,PP.-841-845

[13] Sina Hassanzadeh and Hossein Pourghassem, “A Fast Logo Recognition Algorithm in Noisy Document Images”,International Conference on Intelligent Computation and Bio-Medical Instrumentation, Wuhan, Hubei, pp. 64-67, 2011.

[14] Hossein Pourghassem,”A Hierarchical Logo Detection and Recognition Algorithm Using Twostage Segmentation and Multiple Classifiers”, Fourth International Conference on Computational Intelligence and Communication Networks,2012,PP.228-233.

[15] David G. Lowe, “Distinctive Image Features from Scale-Invariant KeyPoints”, International Journal of Computer Vision, Netherlands, pp. 91-110, 2004.

[16] S. Yasser Arafat, Muhammad Saleem and S. Afaq Hussain, “Comparative Analysis of Invariant Schemes for Logo Classification”, International Conference on Emerging Technologies, Islamabad, Pakistan, pp. 256-261, 2009.

[17] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in International Conference on Computer Vision & Pattern Recognition, 2005, pp. 886–893.

[18] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, 'Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection,' In: 12th International IEEE Conference On Intelligent Transportation Systems, 2009, St. Louis, 2009. V. 1. P. 432-437.

[19] DATASET: Logo Database for Research http://lampsrv02.umiacs.umd.edu/projdb/project.php?id=47

[20] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. cvpr, 1:511, 2001.