EFFICIENT OBJECT RECOGNITION USING DISCRIMINATIVE WEIGHT LEARNING

B. Ramesh Naik
Department of CSE, GST, GITAM University, Bengaluru

Dr. T. Venu Gopal
Department of CSE, JNTUH College of Engineering, Hyderabad

ABSTRACT

Object recognition is the process of identifying and detecting an object or a feature in a digital image or video. But it is also challenging vision problems because objects often suffer from significant scale, illumination variations, material recognition, pose changes, background clutter and partial occlusion. The scale invariant feature transform (SIFT) is a leading feature extraction approach which generates high dimensional features from regions selected based on pixel values. We proposed novel approach based Discriminative Weight Learning to discriminate the image features efficiently which enjoy the important applications of computer vision. This approach extracts image features using SIFT algorithm and weights are assigned to features based Discriminative Weight Learning. Extensive experiments were conducted to evaluate thoroughly this approach and the result showed that this approach is very competitive in object recognition.

Key word: SIFT, Discriminative Weight Learning, Log Euclidean Multivariate Gaussian

http://www.iaeme.com/ijcet/issues.asp?JType=IJCET&VType=8&IType=3

1. INTRODUCTION

Object Recognition has been massively studied in the last two decades with many application areas like face recognition, finger print recognition, image classification, driving assistance [12]. However objects often suffer from significant scale, illumination variations, material recognition, pose changes, background clutter and partial occlusion [2], [3], [19]. Object may include cases, in which the object's appearance changes quickly and significantly or even the situations where the object is not directly observable due to the occlusion or it is behind the border of the image. Recognition of objects without a companion can be very difficult [3], [4], [12], e.g, a glass held in hand as shown in Figure 1. The environment is not stable usually in
the real applications, e.g., illumination or view may change, and the object appearance changes with its motion, there may be clutter and occlusions [21]. The unrecognized companion may randomly merge with the object model causing the model to not describe the object anymore. It may also be assigned to the background distracting its model. Though many sophisticated Object recognition algorithms based on template matching, color, shape, local global features have been designed to extract visual features, these approaches cannot adequately model image semantics and have many limitations when dealing with large volume of image Databases. The Image retrieval quality is sufficient for some retrieval tasks but there is still a semantic gap between the low-level visual features (textures, shapes, colors) and automatically extracted high-level concepts that users normally search for [5], [7], [18]. More specifically, the ‘semantic gap’ is referred to as the discrepancy between the limited descriptive power of low-level visual features and the richness of user semantics. Comprehensive surveys exist on this topic of semantic gap in Content Based Image Retrieval [1], [2]. In particular, densely sampled descriptors have proven to achieve outstanding performance in image-based classification tasks such as object classification, texture recognition, scene categorization and image retrieval). However, it is challenging to develop image descriptors with high distinctiveness for general object recognition.

Our goal is to propose a novel approach that discriminate the image visual features efficiently which enjoys the important application of recognizing objects in given image database. The proposed approach is based on Discriminative Weight Learning [17] which takes input as SIFT features [6], [8], [9] and assigns weights to the extracted features. Our work is inspired by popular distribution based descriptors such as SIFT and HoG. This approach initially extracts image features using SIFT algorithm. These features are again described using Discriminative weight Learning.

The remainder of this paper is organized as follows. Section 2 reviews the details SIFT descriptors. Section 3 introduces Discriminative Weight Learning. Section 4 presents the experiments to evaluate and analyze our Descriptors. Finally Section 5 concludes the paper.

2. SCALE INVARIANT FEATURE TRANSFORM

Primary aspect of CBIR is image matching, which is important aspect of many problems of computer vision and pattern recognition communities. But it is also challenging vision problems because images often suffer from significant scale, illumination variations, material recognition, pose changes, background clutter, partial occlusion [6], [8], [11], [18]. The scale invariant feature transform (SIFT) is a feature extraction approach which generates high dimensional features from patches selected based on pixel values which can then be compared and matched to other features. This approach has a set of parameters, which can be varied and the choice, modification can be used to improve the quality of the results. In the original paper by [6] a set of default parameters is given with a variety of images but whether or not these are optimal is not clear.
This section introduces basic background of SIFT. The original SIFT feature detection algorithm developed by [6] is a four stage process that creates unique and highly descriptive features from an image. The features extracted using SIFT are designed to be invariant to rotation, changes in scale, illumination, noise and small changes in viewpoint. The features can be used to indicate if there is any correspondence between areas within images. Clusters of features from an image that are similar to a cluster of features from another image may indicate, with a high likelihood, areas that match. This allows object recognition to be implemented by comparing features generated from input images to features generated from images of target objects. The four stages of the SIFT algorithm is given in [6] paper. The four stages of the SIFT algorithm are as follows.

2.1. Scale-space extrema detection
The first step to find the SIFT features is to create a Gaussian scale-space [10] pyramid for the image. Multiple octaves are created from blurred images using convolution of Gaussian. Difference between two consecutive images within an octave is referred as Difference of Gaussian. The scale space is defined by the function:

\[ L(x, y, \sigma) = G(x, y, \sigma \ast I(x, y) \] (1)

Where \( \ast \) is the convolution operator, \( G(x, y, \sigma) \) is a variable-scale Gaussian and \( I(x, y) \) is the input image.

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

Stable key point locations in the scale-space are detected by various techniques. Among which Difference of Gaussians is one such technique, locating scale-space extrema, \( D(x, y, \sigma) \) by computing the difference between two images, one with scale \( k \) times the other. \( D(x, y, \sigma) \) is then given by:

\[ D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \]

To detect the local maxima and minima of \( D(x, y, \sigma) \) each point is compared with its 8 neighbours at the same scale, and its 9 neighbours up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

2.2 Feature localisation
The next step in SIFT is Feature Localization. The number of features which are less important is reduced in this stage. Interpolation occurs to locate the exact, sub pixel, location of the candidate features and points that are in areas of low contrast or those that are localised along edges are eliminated. The location of extremum, \( z \) is given by:

\[ z = -\frac{\partial^2 \sigma^{-1}}{\partial x^2} \frac{\partial \sigma}{\partial x} \] (2)

If the function value at \( z \) is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localization it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the key point, the key point is rejected.
2.3. Orientation assignment
The image gradient directions of the pixels in a feature’s neighbourhood are calculated and added to an orientation histogram with 36 bins [10]. The values in the neighbourhood are Gaussian weighted so those nearer the centre have a greater effect on the resulting orientation. One key orientation is selected for each feature. The approach taken to find an orientation is as followed: Use the key points scale to select the Gaussian smoothed image \(L\), from above.

Gradient magnitude \(m\) is

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

Orientation \(\theta\) is

\[
\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right)
\]

Form an orientation histogram from gradient orientations of sample points.

Find out the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation some points will be assigned multiple orientations. Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position.

2.4. Creating the feature descriptor
A Feature Descriptor is considered with 128 dimensional Vector which describes pixel properties of area surrounding a feature is shown in Fig-2. A 4 by 4 array of 16 histograms is centred on the feature and rotated to match the key orientation calculated in the previous step. The gradient magnitudes are given a Gaussian weighting, added to the histograms and normalised to create the descriptor.

The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is then used to create a set of histograms over a window centred on the keypoint.

![Figure 2 Image gradients- Key Point Descriptor](image)

Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the midpoints of these directions. These results in a feature vector containing 128 elements.

3. DISCRIMINATIVE WEIGHT LEARNING
But not all features, even ones that are carefully engineered, improve performance. Adding more features to a model can hurt its accuracy on unseen testing data. One well-known reason for this is over fitting: a model with more features has more capacity to fit chance regularities in the training data. For example, wheel regions are more important than uniform patches to
distinguish a bicycle from a mug. Here, we adapt framework described in [17], [14], [20], [22] for learning region weights. Given a pattern \( I \) containing one object instance and a query \( J \), denote \( f_i^I = 1, 2, 3, \ldots, M \) and \( f_j^J = 1, 2, 3, 4, \ldots, N \) their bags of region features. The distance from \( I \) to \( J \) is defined as:

\[
D(I \rightarrow J) = \sum_{i=1}^{M} w_i^I d_i^I = (w_i, d_i^I)
\]

(4)

where \( d_i^I = \min \{d(f_i^I, f_j^J) \} \) is the elementary distance between \( f_i^I \) and the closest feature in \( J \), \( w_i^I \) is the weight for feature \( f_i^I \). Note that the pattern-to-query distance is asymmetric, i.e., \( D(I \rightarrow J) \neq D(J \rightarrow I) \).

In the weight learning stage, supposing \( I \) is an object of category \( C \), we find a pair of \( J \) and \( K \) such that \( J \) is an object of the same category \( C \) and \( K \) is an object of a different category. The learning algorithm enforces the following condition:

\[
D(I \rightarrow K) > D(I \rightarrow J)
\]

(5)

\[
\Rightarrow (w_i^I, d_i^{IK}) > (w_i^I, d_i^{IJ})
\]

(6)

\[
\Rightarrow (w_i^I, x_i^{IK}) > 0
\]

(7)

where \( x_i^{IK} = d_i^{IK} - d_i^{IJ} \). Supposing we construct \( T \) such pairs for \( I \) from the training set, thus \( x_1, x_2, \ldots, x_T \). Note that the most discriminative regions (leaf and body of the apple logo, handle of the mug) have the highest weights from learning. Optimization is formulated as follows:

\[
\min_{w, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^{T} \xi_i
\]

(8)

s.t.: \( w^T \geq 1 - \xi_i, \xi_i \geq 0, \forall i = 1, 2, 3, \ldots, T \)

(9)

\( w \geq 0 \)

When integrating multiple cues for a single region, we learn one weight for each cue. As in [13], we model the probability of query \( J \) being in the same category as pattern \( I \) by a logistic function:

\[
p(I, J) = \frac{1}{1 + e^{-(\alpha_i d(I, J) - \beta_i)}}
\]

(10)

where \( \alpha_i \) and \( \beta_i \) are parameters learned in training.

4. EXPERIMENTAL SETUP

For a given image \( M \), we first extract features through densely sampling in a regular grid or using an interest point detector. Let \( X = \{x_1, \ldots, x_N\} \) be a set of local features in \( M \) which are extracted using SIFT algorithm [6] described in section 2. i.e \( x_1 \) is feature vector of \( M_1 \), \( x_2 \) feature vector of \( M_2 \), \ldots, \( x_N \) is feature vector of \( M_N \). In Fig-3, (a) is original Image, its region is shown in (b), set if keypoints shown in (c) which are extracted using SIFT algorithm [6], [8].

The number of features of given object extracted using SIFT algorithm [6], [8] is not same compared to another object. Thus these features are normalized using Log Euclidean Multivariate Gaussian using eq-11. The Multivariate Gaussian distribution [4], [13], [15] of a set of \( d \)-dimensional vectors \( X \) is calculated as.
Efficient Object Recognition Using Discriminative Weight Learning

![Image](image.png)

**Figure 3** (a) Original Image (b) Image region. (c) Key Points

where \( j \) is the determinant, \( \xi \) is the mean vector and \( K \) is the covariance matrix with space of real symmetric positive semi-definite matrices [6] defined as follows:

\[
\xi = \frac{1}{N} \sum_{i=1}^{N} x_i, K = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \xi)(x_i - \xi)^T
\]

Let \( x_1 \) and \( x_2 \) set features described by log Euclidean Multivariate Gaussian Descriptors of given two objects, the difference between \( x_1 \) and \( x_2 \) is calculated using \( D(I \rightarrow J) \) described in eq-4.

5. RESULTS ON BENCHMARK DATASETS

We carry out experiments on other popular benchmark datasets, but this approach focus on Image Feature comparison. Our experiments involve object detection on WANG Database [16] containing 1000 Corel images in JPEG format shown in fig-4. The image set comprises 100 images in each of 10 categories. In our experiment, we have selected 100 images randomly, containing 10 images in each category. Within this database, it is known whether any two images are of the same category. In particular, a retrieved image is considered a match if and only if it is in the same category as the query. The retrieval results of our approach and SIFT is shown in fig-5.

![Image](image.png)

**Figure 4** Snapshot of WANG Database
Figure 5 (a) Retrieval Result with our approach (b) Retrieval Result with SIFT

We compare with the method in [6], and it involves mapping or transforming of the local features (e.g., SIFT) and achieve improved performance over SIFT with Discriminative Weight Learning. With this approach we further achieve an accuracy improvement of 2.2% on average.

7. CONCLUSION

This paper presented a Discriminative Weight Learning approach to discriminate object features of one image to another image. Unlike popular histogram-based descriptors, which, based on feature space quantization, collect zero-order (occurrence) information, Log Euclidean Multivariate Descriptors is continuous and models higher-order statistics. It can naturally leverage multiple cues or other descriptors (e.g., SIFT) as raw features. We have further shown that discriminative weight learning with SIFT combination significantly boosts recognition performance.

REFERENCES


Efficient Object Recognition Using Discriminative Weight Learning


[16] http://wang.ist.psu.edu/docs/related


