3rd International Conference on Research Developments in Applied Science, Engineering & Management
The Indian Council of Social Science Research (ICSSR) AEM–2018
Panjab University Campus, Chandigarh (India)
19th August 2018 www.conferenceworld.in ISBN : 978-93-87793-43-9

DRLBP based edge texture features for Object Recognition with SIFT

Divyani S. Hadoltikar[1], D. J. Tuptewar[2]

[1]*Department of Electronics and Telecommunication, MGM COE, Nanded-431 606, India*
[2]*Department of Electronics and Telecommunication, MGM COE, Nanded-431 606, India*

ABSTRACT

Category recognition system will be developed for application to image retrieval. This paper proposes two sets of novel edge-texture features, Discriminative Robust Local Binary Pattern (DRLBP) and Ternary Pattern (DRLTP), for object recognition. By finding the limitations of Local Binary Pattern (LBP), Local Ternary Pattern (LTP) and Robust LBP (RLBP). DRLBP and DRLTP are proposed features by analyzing with HOG and SIFT features for better performance. For classification purpose linear SVM classifier is used. Furthermore, the proposed features retain contrast information necessary for proper representation of object contours that LBP, LTP, and RLBP discard. Our proposed features are tested on CALTECH 256 Data set. Results demonstrate that the proposed features outperform the compared approaches on most data sets.

Keywords- *DRLBP, DRLTP, SIFT, HOG, Feature extraction and SVM*

1. INTRODUCTION

Object recognition is divided in two parts category recognition and detection. Category recognition is used to classify an object into one of several predefined categories. Detection is used to distinguish objects from the background. Typically, objects have to be detected against cluttered, noisy backgrounds and other objects under different illumination and contrast environments. Performance of object recognition can be improved by discriminating the object from the background or other objects in different lightings and scenarios.

Object recognition features are categorized into two groups sparse and dense representations. For sparse feature representations, interest-point detectors are used to identify structures such as corners and blobs on the object. A feature is created for the image patch around each point. Dense feature representations, which are extracted at fixed locations densely in a detection window, are gaining popularity as they describe objects richly compared to sparse feature representations.

LBP is robust to illumination and contrast variations as it only considers the signs of the pixel differences. However, it is sensitive to noise and small fluctuations of pixel values. To handle this, Local Ternary Pattern (LTP) has been proposed. In comparison to LBP, it has 2 thresholds which creates 3 different states as compared to 2 in LBP. It is more resistant to noise and small pixel value variations compared to LBP. LBP and LTP

differentiate a bright object against a dark background and vice versa. They have also been used for texture classification and face detection. But it increases the object intra-class variations which is undesirable for most object recognitions. To solve this problem propose Robust LBP (RLBP) to map a LBP code and its complement. It is therefore desirable to represent objects using both texture and edge information. However, in order to be robust to illumination and contrast variations, LBP, LTP and RLBP do not differentiate between a weak contrast local pattern and a similar strong one. They only capture texture information. Discriminative Robust LBP (DRLBP) and DR LTP are proposed.

The proposed features solve the issues of LBP, LTP and RLBP. They alleviate the intensity reversal problem of object and background. Furthermore, DRLBP discriminates local structures that RLBP misrepresent. In addition, the proposed features retain the contrast information of image patterns. They contain both edge and texture information which is desirable for object recognition. We have discussed the issues object recognition in this paper and suggested a new feature extraction with combining HOG and SIFT features, we can increases the performance. Histogram of Oriented Gradient (HOG) descriptors provides excellent performance relative to other existing features including wavelets. Scale Invariant Feature Transform (SIFT) is used for image matching.

2. PROPOSED SYSTEM

The proposed scheme has been divided into two main modules. First is the feature Extraction includes preprocessing and local binary pattern extraction of images and second is classifier, which includes learning methods and classification of images using SVM classifier.

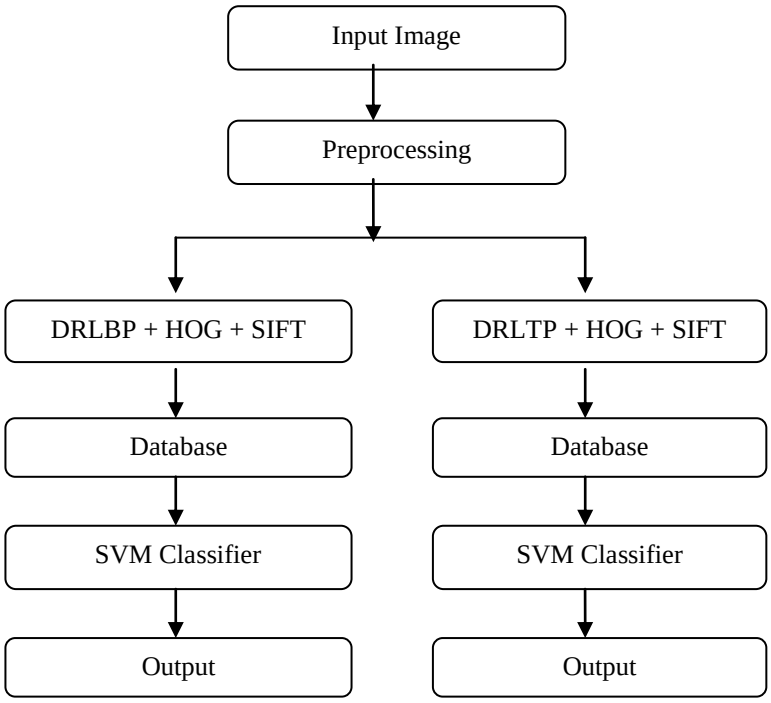

Fig.1 Proposed system

A. Feature Extraction process:

After preprocessing is done, image is given to the feature extraction process where two main algorithms are present DRLBP and DRLTP. LBP is main feature extraction process in DRLBP and DRLTP algorithm.

$$LBPx, y = \sum_{b=0}^{B-1} s(p_c - p_b)2^b \ , \qquad (1)$$

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

Where p_c is the pixel value at (x, y), p_b is the pixel value estimated using bilinear interpolation from neighbouring pixels in the b^{th} location on the circle of radius R around p_c and B is the total number of neighbouring pixels.

In this way, if a LBP code covers both sides of a strong edge, its gradient magnitude will be much larger and by voting this into the bin of the LBP code, we take into account if the pattern in the local area is of a strong contrast. Thus, the resulting feature will contain both edge and texture information in a single representation. The value of the i^{th} weighted LBP bin of a M × N block is as follows

$$h_{lbp}(i) = \sum_{x}^{M-1} \sum_{y}^{N-1} \omega_{x,y}\, \delta(LBP_{x,y}, i) \ , \qquad (2)$$

$$\delta(m, n) = \begin{cases} 1, & m = 0 \\ 0, & \text{otherwise} \end{cases}$$

The RLBP histogram is created from both LBP codes and their complements.

$$h_{rlbp}(i) = h_{lbp}(i) + h_{lbp}(2^B - 1 - i), \quad 0 \leq i < 2^{B-1} \qquad (3)$$

Where h_{dlbp} (i) is the i^{th} DLBP bin value. The number of DLBP bins is 128 for B = 8. Using uniform codes, it is reduced to 30. For blocks that contain structures with both LBP codes and their complements, DLBP assigns small values to the mapped bins. It differentiates these structures from those having no complement codes within the block.

$$h_{dlbp}(i) = \left| h_{lbp}(i) - h_{lbp}(2^B - 1 - i) \right|, \quad 0 \leq i < 2^{B-1} \qquad (4)$$

The 2 histogram features, RLBP and DLBP, concatenated to form Discriminative Robust LBP (DRLBP) as follows

$$h_{drlbp}(j) = \begin{cases} h_{rlbp}(j), & 0 \leq j < 2^{B-1} \\ h_{dlbp}(j - 2^{B-1}), & 2^{B-1} \leq j < 2^B \end{cases} \qquad (5)$$

The LTP code at (x, y) is calculated as follows

$$LTP_{x,y} = \sum_{b=0}^{B-1} s'(p_b - p_c)3^b ,$$

(6)

$$s' = \begin{cases} 1, & z \geq T \\ 0, & -T < z < T \\ -1, & z \leq -T \end{cases}$$

LTP code is divided into "upper" and "lower" LBP codes. The ULBP and LLBP are calculated as follows

Upper code, ULBP, is computed as follows

$$ULBP = \sum_{b=0}^{B-1} f(p_b - p_c)\, 2^b, \quad (7)$$

$$f(z) = \begin{cases} 1, & z \geq T \\ 0, & \text{otherwise} \end{cases}$$

The lower code, LLBP, is computed as follows

$$LLBP = \sum_{b=0}^{B-1} f'(p_b - p_c)\, 2^b, \quad (8)$$

$$f'(z) = \begin{cases} 1, & z \leq -T \\ 0, & \text{otherwise} \end{cases}$$

By doing so, the dimensionality of the feature is reduced from 6561 bins to 512 bins. Using uniform LBP code representation, the number of bins is further reduced to 118 bins. The RLTP code is divided into upper and lower LBP codes. The URLBP is calculated as follows

$$URLBP = \sum_{b=0}^{B-1} h(RLTP_{x,y,b})2^b, \quad (9)$$

$$h(z) = \begin{cases} 1, & z = 1 \\ 0, & \text{otherwise} \end{cases}$$

Where $RLTP_{x,y,b}$ represents the RLTP state value at the b^{th} location. The lower code, LRLBP is computed as follows

$$LRLBP = \sum_{b=0}^{B-1} h'(RLTP_{x,y,b})2^b, \quad (10)$$

$$h'(z) = \begin{cases} 1, & z = -1 \\ 0, & \text{otherwise} \end{cases}$$

Here, LRLBP only has 7 bits as the state at $(B-1)^{th}$ location of RLTP is always 0 or 1.

Consider a LTP histogram for M×N image block. The value of the k^{th} bin of the weighted LTP histogram is

$$h_{ltp}(k) = \sum_{x=0}^{M-1} \cdot \sum_{y=0}^{N-1} w_{x,y}\delta(LTP_{x,y}, k),$$

(11)

It is not difficult to see that the RLTP histogram can be simply created from above equation is

$$h_{rltp}(k) = \begin{cases} h_{ltp}(0), & k = 0 \\ h_{ltp}(k), +h_{ltp}(-k), & 0 < k < \frac{3^B+1}{2} \end{cases}$$

(12)

Where $h_{rltp}(k)$ is the k^{th} bin value of RLTP. We consider the absolute difference between the bins representing a LTP code and its inverted representation to form Difference of LTP histogram as follows

$$h_{dltp}(k) = \left| h_{ltp}(k) - h_{ltp}(-k) \right|, \quad 0 \le k < \frac{3^B+1}{2}$$

(13)

Where $h_{dltp}(k)$ is the k^{th} bin value of DLTP.

RLTP and DLTP are concatenated to form Discriminative Robust LTP as follows

$$h_{drltp}(l) = \begin{cases} h_{rltp}(l), & 0 < l < \frac{3^B+1}{2} \\ h_{dltp}\left(l - \frac{3^B-1}{2}\right), & \frac{3^B+1}{2} \le l < 3^B \end{cases} \quad (14)$$

DRLTP produces different features for the structures. It also resolves the issue of brightness reversal of object and background. Consider the ULBP and LLBP codes for an image block. The value of the s^{th} bin, $0 < s < 2^B$, of URLBP can be generated from ULBP and LLBP codes as follows

$$h_{urlbp}(s) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta(\max(ULBP, LLBP), s) \quad (15)$$

$$h_{lrlbp}(s) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta(\min(ULBP, LLBP), t) \quad (16)$$

The split LBP histograms, UDLBP and LDLBP, for DLTP can also be generated from the ULBP and LLBP codes. For every LTP code whose ULBP and LLBP representations are swapped, the corresponding values of UDLBP and LDLBP bins are decremented by 1 accordingly. Otherwise, the bins are incremented by 1. The s^{th} bin value, $0 < s < 2^B$, of UDLBP is expressed as follows

$$\square_{udlp} = \left| \sum_{x=0}^{M-1} . \sum_{y=0}^{N-1} w_{x,y} \delta'(\lambda(ULBP, LLBP), s) \right|, \quad (17)$$

$$\lambda(p,q) = \begin{cases} p, & p > q \\ -q, & p < q \end{cases}$$

$$\delta'(m,n) = \begin{cases} 1, & m = n, m > 0 \\ -1, & |m| = n, m < 0 \\ 0, & ot\square erwise \end{cases}$$

The function λ (•) determines whether the ULBP and LLBP codes are being swapped. If a swap occurs, the negative maximum code is assigned to the result. The function $\delta'(\bullet)$ checks the value output from λ with s. If the value is positive and matches s, the s^{th} bin value is incremented. Otherwise, it is decremented. The t^{th} bin value of LDLBP is determined as follows

$$\square_{ldlbp} = \left| \sum_{x=0}^{M-1} \cdot \sum_{y=0}^{N-1} w_{x,y} \delta''(\lambda(ULBP, LLBP), t) \right|,$$

(18)

$$\lambda'(p,q) = \begin{cases} q, & p \geq q \\ -p, & p < q \end{cases}$$

$$\delta''(m,n) = \begin{cases} 1, & m = n, m \geq 0 \\ -1, & |m| = n, m < 0 \\ 0, & ot\square erwise \end{cases}$$

The function $\lambda'(\bullet)$ determines whether the ULBP and LLBP codes are being swapped. If a swap occurs, the negative minimum code is assigned to the result. The function $\delta''(\bullet)$ checks the value output from λ' with t. If the value is zero or positive and matches t, the t^{th} bin value is incremented. Otherwise, it is decremented. The URLBP, LRLBP, UDLBP and LDLB histograms are then concatenated to form DRLTP.

B. SVM Classifier

Separation of classes. That's what SVM does. It finds out a line/ hyper-plane (in multidimensional space that separate outs classes). SVM classification uses different planes in space to divide data points using planes. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories or classes are divided by a dividing plane that maximizes the margin between different classes. This is due to the fact if the separating plane has the largest distance to the nearest training data points of any class, it lowers the generalization error of the overall classifier. As we have stated in the introduction, the recognition system is based on a supervised learning technique. Hence, we have used a set of training image described by their HOG and SIFT features to learn their performance. In our case, we have used a Support Vector Machines classifier.

3. OVERVIEW OF THE METHOD

A. Histogram of Oriented Gradients

In the context of object recognition, the use of edge orientation histogram has gain popularity. However, the concept of dense and local histograms of oriented gradients (HOG) is a method introduced by Dalal. The aim of such method is to describe an image by a set of local histograms. These histograms count occurrences of gradient orientation in a local part of the image. In this work, in order to obtain a complete descriptor of an infrared image, we have computed such local histograms of gradient according to the following steps

1) *Gradient computation*: Detector performance is sensitive to the way in which gradients are computed, but the simplest scheme turns out to be the best. We tested gradients computed using Gaussian smoothing followed by one of several discrete derivative masks. Several smoothing scales were tested including σ=0 (none). Masks tested included various 1-D point derivatives (uncentred [−1,1], centred [−1,0,1] and cubic corrected [1,−8,0,8,−1]) as well as 3×3 Sobel masks and 2×2 diagonal ones (the most compact centred 2-D derivative masks). Simple 1-D [−1,0,1] masks at σ=0 work best. Using larger masks always seems to decrease

performance, and smoothing damage sit significantly for Gaussian derivatives. For colour images, we calculate separate gradients for each colour channel, and take the one with the largest norm as the pixel's gradient vector.

2) *Spatial/Orientation Binning* : The next step is the fundamental nonlinearity of the descriptor. Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centred on it, and the votes are accumulated into orientation bins over local spatial regions that we call cells. Cells can be either rectangular or radial (log-polar sectors). The orientation bins are evenly spaced over 0^o–180^o ("unsigned" gradient) or 0^o–360^o ("signed" gradient). To reduce aliasing, votes are interpolated bilinear between the neighboring bin centers in both orientation and position. The vote is a function of the gradient magnitude at the pixel, either the magnitude itself, its square, its square root, or a clipped form of the magnitude representing soft presence/absence of an edge at the pixel.

3) *Normalization and Descriptor Block* : Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local contrast normalization turns out to be essential for good performance. We evaluated a number of different normalization schemes. Most of them are based on grouping cells into larger spatial blocks and contrast normalizing each block separately. The final descriptor is then the vector of all components of the normalized cell responses from all of the blocks in the detection window.

4) *Block Normalization schemes* : We evaluated four different block normalization schemes for each of the above HOG geometries. Let v be the unnormalized descriptor vector, $\|v\|_k$ be its k-norm for k=1, 2, and ε be a small constant. The schemes are:

(a) L2-norm, $v \rightarrow v/\sqrt{\|v\|_2^2 + \varepsilon^2}$;

(b) L2-Hys, L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing,

(c) L1-norm, $v \rightarrow v/(\|v\|_1 + \varepsilon)$ and

(d) L1-sqrt, L1-norm followed by square root $v \rightarrow \sqrt{v/(\|v\|_1 + \varepsilon)}$ which amounts to treating the descriptor vectors as probability distributions and using the Bhattacharya distance between them. L2-Hys, L2-norm and L1-sqrt all perform equally well, while simple L1-norm reduces performance and omitting normalization entirely reduces it. Some regularization ε is needed as we evaluate descriptors densely, including on empty patches, but the results are in sensitive to ε's value over a large range.

B. Scale Invarient Feature Transform

The SIFT algorithm is a local feature extraction algorithm, which finds extrema points in scale space, and extracts a position, scale, rotation invariant feature vector for each extrema point. In order to increase SIFT's power and enhance the efficiency of object recognition, extensions of the SIFT descriptor have been proposed by researchers. The improvement techniques have included different histograms and different region shapes than used for the standard SIFT, and the use of dimensionality reduction methods to reduce the dimension of the

standard SIFT descriptor. The SIFT algorithm finds extrema points in scale space, and extracts position, scale, rotation invariant feature vectors. The major stages of computation of SIFT descriptors are divided into four major stages (1) scale-space extrema detection (i.e identifying keypoints) (2) keypoint localization (3) orientation assignment and (4) keypoint descriptor computation. These stages are used to produce the set of image features. The sections below provide details for these stages.

(1) Scale-Space Extrema Detection : The first stage of calculation is to search over all scales and image locations. The difference-of-Gaussian function is used to detect stable keypoint locations in scale space. This stage attempts to find those locations and scales that are identifiable from different views of the same object. This can be efficiently achieved by using a scale space function. Furthermore, it has been shown under reasonable assumptions that it must be based on a Gaussian function. The scale space of a 2-dimensional image is defined by equation as shown below

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{19}$$

Where * is the convolution operator, G(x, y, σ) is a variable-scale Gaussian, and I(x, y) is the input image. The parameter σ is the scale of the keypoint and is also the standard deviation of the Gaussian function.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{20}$$

The difference of Gaussians function, D(x, y, σ), is used to detect stable keypoint locations in scale space; D(x, y, σ) is computed by using the difference between two images, one with scale k times the other. Then, D(x, y, σ) is given by equation

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) \tag{21}$$

$$= L(x, y, k\sigma) - L(x, y, \sigma)$$

To detect the local maxima and minima of D(x, y, σ), each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema. The extrema is used as a SIFT keypoint.

(2)Keypoint Localization : In order to enhance the stability of the follow-up image feature matching and increase the algorithm's anti-noise ability, we need to remove the low-contrast and unstable keypoints. This stage attempts to eliminate these unstable keypoints from the final list of keypoints by finding those that have low contrast or are poorly localized on an edge. This may be achieved by calculating the Laplacian value for each keypoint found in stage one. The location of extremum z is given by equation

$$z = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \tag{22}$$

(3) Orientation Assignment: This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor can then be represented relative to this orientation, achieving invariance to rotation. The gradient magnitude m and orientation μ of (x, y) are given in equations

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (23)$$

$$\mu(x,y) = \tan^{-1}\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \quad (24)$$

(4) The Keypoint Descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. 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 a variance of 1.5 * the keypoint scale. These data are then used to create a set of histograms over a window centered on the keypoint.

4. RESULT & DISCUSSION

The datasets consists of 40 test images in axial plane and 256×256 in plane resolution, which were downloading from the Caltech-256 object category dataset. Table 1shows the classification accuracy with proposed system using HOG and SIFT features.

$$\text{Classification Accuracy} = \frac{\text{Correctly Classified images}}{\text{Test Images}}$$

The MATLAB tic and toc functions were connected to clock. Each tic command would get the current time as a local time date vector generated by clock and save it. The toc command would call clock again to get the current time and subtract it from the saved local time date vector to get the elapsed time. Clock calls a time service using API provided by the operating system.

Table 1 Recognition accuracy on CALTECH 256 Data set

Approach	**Accuracy**	**Time**
DRLBP + SVM	60.80 %	12.69 sec
DRLTP + SVM	81.89 %	12.48 sec
DRLBP+HOG+SVM	71 %	14.59 sec
DRLTP+HOG+SVM	70 %	14.01 sec

DRLBP+SIFT+SVM	78.9 %	14.31 sec
DRLTP+SIFT+SVM	82.6 %	13.42 sec
DRLBP+HOG+SIFT+SVM	98.8 %	16.50 sec
DRLTP+HOG+SIFT+SVM	99.01 %	16.40 sec

5. CONCLUSION

We have present a robust DRLBP and DRLTP edge texture analysis for image recognition performance. In this paper we have analyzed DRLBP, DRLTP, HOG and SIFT techniques for feature extraction. The query image features and database image features are compared with recognition accuracy. It was observed that recognition accuracy is improved, but execution time and complexity increases. Experimental result indicates that the proposed method gives excellent recognition accuracy of different image datasets. We present results of the proposed features on CALTECH 256 Data set and compare them with 3 methods for object recognition. Results demonstrate that the proposed method achieves improvement in recognition accuracy than using DRLTP and DRLBP techniques. Finally, the performance factors such as accuracy and execution time is evaluated and it is improved upto 99%.

REFERENCES

[1] Amit Satpathy, Xudong Jiang, How-Lung Eng, "LBP-Based Edge-Texture Features for Object Recognition",IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 5, MAY 2014

[2] Rasika Raikar, Shivani Pandita, "Discriminative Robust Local Binary Pattern based Edge Texture Features for Object Recognition", International Journal of Scientific Engineering and Research (IJSER), Impact Factor (2014)

[3] Pranita R. Chavan, Dr. Dnyandeo J. Pete, "Face Recognition using Local Derivative Pattern Face Descriptor", International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 3 Issue 10 October, 2014

[4] Duc Thanh Nguyen, Zhimin Zong, Philip Ogunbona, and Wanqing Li,"OBJECTDETECTIONUSINGNON-REDUNDANTLOCALBINARYPATTERNS", Proceedings of 2010 IEEE 17th International Conference on Image Processing September 26-29, 2010, Hong Kong

[5] Jim Mutch and David G. Lowe, "Multiclass Object Recognition with Sparse, Localized Features", Department of Computer Science University of British Columbia #201 - 2366 Main Mall Vancouver, B.C., Canada, V6T 1Z4

[6] Navneet Dalal and Bill Trigg, "Histograms of Oriented Gradients for Human Detection", INRIA Rhone-alps, 655 avenue del'Europe, Montbonnot 38334, France

[7] F. Suard, A. Rakotomamonjy, A. Bensrhair, "Pedestrian Detection using Infrared images and Histograms of Oriented Gradients", Intelligent Vehicles Symposium 2006, June 13-15, 2006, Tokyo, Japan

[8] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", Computer Science Department University of British Columbia Vancouver, B.C., Canada lowe@cs.ubc.ca January 5, 2004

[9] Piotr Doll´ar, Christian Wojek, Bernt Schiele, and Pietro Perona, "Pedestrian Detection: An Evaluation of the State of the Art", SUBMISSION TO IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

[10] Yuehua Tao, Marjorie Skubic, Tony Han, Youming Xia, and Xiaoxiao Chi, "Performance Evaluation of SIFT-Based Descriptors for Object Recognition"

[11] Xiaoyang Tan and Bill Triggs, "Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions", INRIA & Laboratoire Jean Kuntzmann, 655 avenue de l'Europe, Montbonnot 38330, France

[12] Ms. Rajashri. A. Kolhe1, Prof. A. S. Deshpande, "DRLBP and DRLTP Based Object Recognition for Image Retrieval Systems", International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 5, Issue 9, September 2016