Object Cataloging Using Heterogeneous Local Features for Image Retrieval

Mohammad Khairul Islam[1] , Farah Jahan[1] and Joong Hwan Baek[2]
[1]Department of Computer Science & Engineering, University of Chittagong, Chittagong-4331, Bangladesh.
[2]Department of Information and Telecommunication Engineering, Korea Aerospace University, Goyang-city, Gyeonggi-do, 412-791, South Korea.
[e-mail: mkislam@cu.ac.bd, farah_csc@cu.ac.bd, jhbaek@kau.ac.kr]
*Corresponding author: Mohammad Khairul Islam

Received April 1, 2015; revised July 26, 2015; accepted August 25, 2015; published November 30, 2015

Abstract

We propose a robust object cataloging method using multiple locally distinct heterogeneous features for aiding image retrieval. Due to challenges such as variations in object size, orientation, illumination etc. object recognition is extraordinarily challenging problem. In these circumstances, we adapt local interest point detection method which locates prototypical local components in object imageries. In each local component, we exploit heterogeneous features such as gradient-weighted orientation histogram, sum of wavelet responses, histograms using different color spaces etc. and combine these features together to describe each component divergently. A global signature is formed by adapting the concept of bag of feature model which counts frequencies of its local components with respect to words in a dictionary. The proposed method demonstrates its excellence in classifying objects in various complex backgrounds. Our proposed local feature shows classification accuracy of 98% while SURF,SIFT, BRISK and FREAK get 81%, 88%, 84% and 87% respectively.

Keywords: Color Histogram, SIFT, SURF, BRISK, FREAK, and Bag of Words.

http://dx.doi.org/10.3837/tiis.2015.11.016 **ISSN : 1976-7277**

1. Introduction

Given an image containing an object, the tasks of object cataloging is to map the imagery to one of a set of object categories. Extensive research has been performed for object classification. Some works on object class models have focused on the problems of classification. The class of "bag-of-features" model fall into this category. Representative examples of this class of model include the work of Csurka et. al. [1], Sivic et. al. [2], and Grauman et al. [3][4]. Research on hierarchical models has been highly popular in recent years. The line of research by Sudderth et. al. [5][6][7][8] falls into this category. These models use hierarchical Dirichlet processes for detecting and recognizing objects with variable number of parts, and scenes with variable numbers of objects. Fei-Fei and Perona [9] used a bayesian hierarchical model for learning natural scene categories. Part based representation forms the basis of some computational theories of object detection [10] where object parts are local groupings of common primitive features such as edge fragments [11][12]. Due to huge success in part-based concept in image representation, we adapt this concept for object classification.

Most works that have used learning in object classification have been done at the pixel level [13][14][15]. The primitive features at pixel level are edge fragments, and color. However, these features are not robust in the presence of illumination changes and non-rigid motion. Recently, scale invariant feature transform (SIFT), a scale and rotation invariant descriptor proposed by D. Lowe [16] has been successfully applied in various general object recognition tasks [17]. This approach extracts blob-like local features from an image, and represents each blob structure at an appropriate scale with a mechanism of automatic scale selection [16]. It is computationally expensive. Regarding computational speed, another robust feature named SURF[18] outperforms SIFT on general purpose computers. Recent developments on local descriptors include FREAK[19], BRISK[20] etc. Comparison between the recent developmes are illustrated by Schaeffer et. al. [21] and Canclini et. al. [22]. P. Viola and M. Jones [23] used rectangle features to capture the presence of edges, bars and other simple structures. In [24] simple relations of local image patches are used for histogram of gradient (HOG) descriptor. Lazebnik et. al. [25] classified images of both scenes and isolated objects by partitioning tche image into increasingly fine regions, and then computing histograms of features for each region. C. Gu. et al. [26] combined heterogeneous descriptors such as the region and shape together for object detection. In this paper, we investigate various types of features extracted from local regions and considering the computational efficiency as well as performance metrics we propose an extended feature model which is composed of multiple heterogeneous features.

In order to make the article more self-contained, we briefly discuss our proposed method in the following sections. Section 2 briefly explains our proposed method for object classification. Section 3 and 4 presents feature extraction, and global signature generation methods. We present classification technology in section 5 folloed by our experiments in section 6. And finally, in section 7 we make our comments and future research plans to resolve currently available issues.

2. Proposed Approach

Given an object imagery, our object recognition approach aims to classify it into one of a set of predefined object categories. In this approach, we extend one of the state-of-the-art feature extraction technologies for object recognition. **Fig. 1** gives a pictorial description of our proposed method. The primary components in this problem domain include feature extraction, object representation and classification. Feature extraction deals with generating local descriptors from low-level information, whereas object representation is supposed to generate a global signature from the local descriptors in order to represent the object.

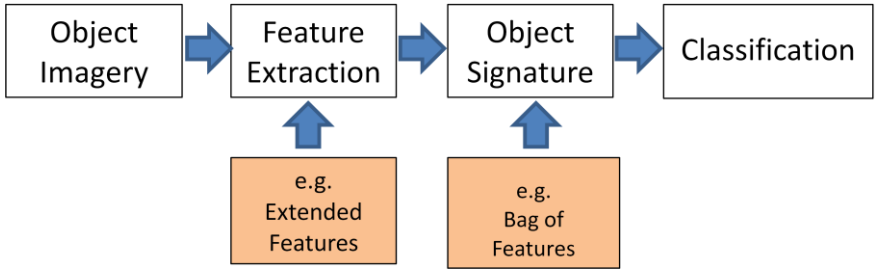

Fig. 1. Block diagram of object detection and classification approach.

3. Feature Extraction

We work with heterogeneous features for describing a patch around an interest point. We aim to strengthen the discriminative power of our feature using heterogeneous attributes of local patches. Considering acceptable computational complexity we use sum of wavelet responses and weighted color statistics in our approach.

3.1 Sum of Wavelet Response

Being inspired by speeded up robust feature [27] we calculate sum of wavelet response. In this method, an integral image is constructed for fast computation. Given an input image $I(p,q)$ of resolution *mxn* where (p,q) is spatial position of a pixel. An integral image I_Σ is calculated as in Eq. (1).

$$I_\Sigma = \sum_{p=1}^{p\le m}\sum_{q=1}^{q\le n} I(p,q) \tag{1}$$

The Hessian matrix $H(\mathrm{P},\sigma)$ in $\mathbf{X}$, where $P=(p,y)$, at scale σ is calculated as Eq. (2)

$$H(P,\sigma) = \begin{bmatrix} L_{pp}(P,\sigma) & L_{pq}(P,\sigma) \\ L_{pq}(P,\sigma) & L_{qq}(P,\sigma) \end{bmatrix} \tag{2}$$

Where, $L_{pp}(\mathrm{P},\sigma)$is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial p^2} g(\sigma)$ with the image I in point P, and similarly for $L_{pq}(\mathrm{P},\sigma)$ and $L_{qq}(\mathrm{P},\sigma)$. A pixel at which the Hessian's determinant satisfies a certain threshold is considered as an interest point.

Circular neighborhood of radius $6s$ around an interest point, where s is the scale at which the point was detected, is selected to find dominant orientation. Haar wavelet responses with side length of $4s$ in horizontal and vertical directions are computed. Sum of all responses within a sliding orientation window covering an angle of 60 degree yields a vector. The longest vector among all possible orientations is the dominant orientation.
For the extraction of the descriptor, the first step consists of constructing a square region centered around the interest point and oriented along the orientation selected in the previous section. The size of this window is 20s. An interest region is split into 4x4 square sub-regions with 5x5 regularly spaced sample points inside. Haar wavelet responses d_p and d_q weighted with a Gaussian kernel centered at the interest point are calculated. Sum of responses at P for d_p, $|d_p|$, d_q, and $|d_q|$ creates a feature vector of 16x4 or 64 elements and the sum of responses d_p, and $|d_p|$computed separately for $d_q < 0$ and $q > 0$ and similarly sum of d_q, and $|d_q|$ creates a feature vector of length 128.

3.2 Color Histogram

Color histograms are widely used for describing the color of objects [28]. A color histogram is a representation of the distribution of colors in an image. It is easy to compute and effective in characterizing both the global and local distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space made up of a set of possible colors. In our experiment, we construct a color descriptor for each key point. In this case, a 16x16 window around a keypoint is considered as a patch. The color values are calculated from the patch and put into a n-bin histogram. Any value in the range 0 to $255/n$ is added to the first bin, $(255/n) + 1$ to $\left(\frac{255}{n}\right) * 2$ is added to the next bin and so on. **Fig. 2** displays a symbolic example of the RGB and HSV color planes and there corresponding histograms obtained from an input image. **Fig. 3** displays color histograms obtained from a real image.

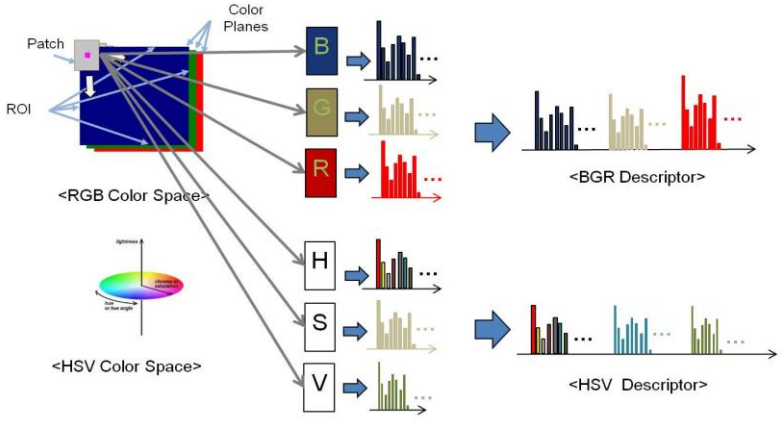

Fig. 2. Symbolic color planes and there corresponding histograms.

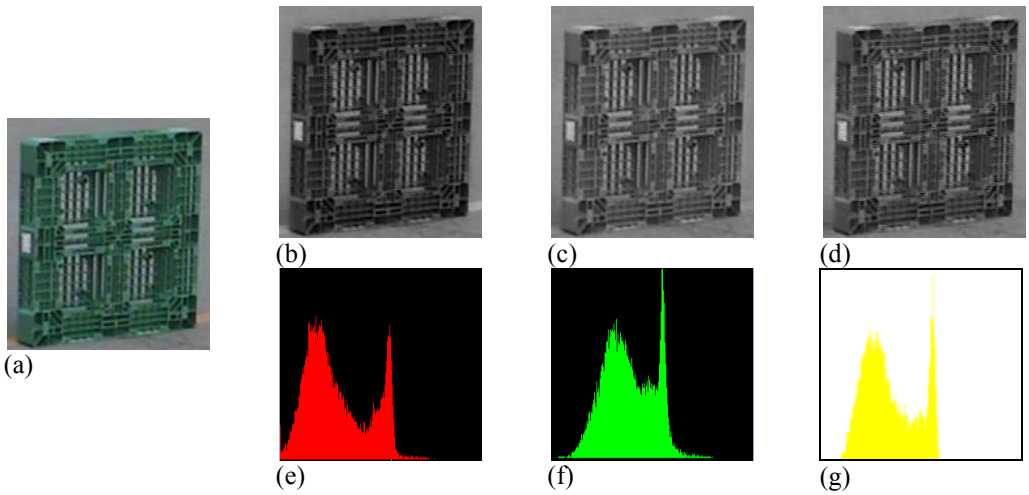

Fig. 3. Color Histograms. (a) A color image. (b), (c), and (d) are R, G, and B planes obtained from the image (a). (e), (f), and (g) are intensity histograms of (b), (c) and (d) respectively.

3.3 Weighted Color Histogram

In our experiment, the amount added to a bin depends on the color value of the pixel and the distance from the keypoint. So, the pixels far away from the keypoint will add less contribution to the histogram. Given a color image I of N pixels, a M-bin histogram of the image can be produced by discretization of the intensity values in the image into M bins. Each bin $b \in \{1, \dots, M\}$ counts the number of occurrences into itself. Let us denote a bin count as n_b, then it can be defined as Eq. (3).

$$n_b = \sum_{k=1}^{N} \delta_{kb} \tag{3}$$

Where, $\delta_{kb} = 1$, if the k^{th} pixel falls into the bin b or $\delta_{kb} = 0$ otherwise.

To establish the assumption that the points staying closer to the interest point in the local patch has higher importance than those of the farther distance, we consider a Gaussian kernel for weighting the pixels of the local patch. Suppose that we have a local patch $P(x, y)$ centered and a Gaussian kernel denoted by $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$. Also, suppose that $P_c(x, y)$ is the intensity value at position (x, y) in a color channel c and a pixel $P_c(x, y)$ falls into the bin $b = \left\lfloor \frac{P_c(x,y)}{c_k} \right\rfloor$ where, c_k is the total number of bins for c. The pixel is then weighted by the respective value in the Gaussian before counting into the bin. For instance, a pixel $P_c(x, y)$ falling into a bin b counts the value $G(x, y)$ at the bin. If we denote the histogram of c plane as H_c, then for the pixel $P_c(x, y)$ the b-th bin is updated as Eq. (4).

$$H_c(b) = H_c(b) + G(x, y) \tag{4}$$

All histograms from the color plances are initially set to zero. The calculated histograms

are then concatenated to make a color descriptor.

3.4 Combined Descriptor Generation

In our previous research [29] we integrated texture and color-based descriptors by concatenating them together. In such case, each element of the newly generated combined descriptor has same weight. For example, if we have a texture-based descriptor of m elements, denoted by T, defined as $T = (t_1, t_2, \dots, t_m)$ and a color-based descriptor of n elements, denoted by C, defined as $C = (c_1, c_2, \dots, c_n)$, then the combined descriptor can be defined by a vector of length $m + n$. Let us denote the combined vector as D, then it can be defined as $D = (t_1, t_2, \dots, t_m, c_1, c_2, \dots, c_n)$. That combined approach also showed improvements in discriminative power. We can see in the definition of the combined vector that each element in the vector has same weight independent of its source type. But, every feature type has different discriminative power as a signature. Keeping this hypothesis in our mind we analyze our feature set and discover their relative strengths in representing a same problem domain. After discovering their relative strength in a given dataset or problem we weight individual feature type by their strength. In our research, we represent each sample images in a training dataset by using a single feature type, then find correct classification rate as its strength. For instance, suppose we have the above two types of features and their correct classification ratios are a and b respectively. If we consider their performances combinedly then they have strengths $\frac{a}{a+b}$, and $\frac{b}{a+b}$ respectively. We use these strengths as weight of respective descriptor before combining them. Mathematically, we denote the combination for the above descriptors as Eq. (5). **Fig. 4** shows example of combined descriptor.

$$D = \left(\frac{a}{a+b} * (t_1, t_2, \dots, t_m), \frac{b}{a+b} * (c_1, c_2, \dots, c_n)\right) \tag{5}$$

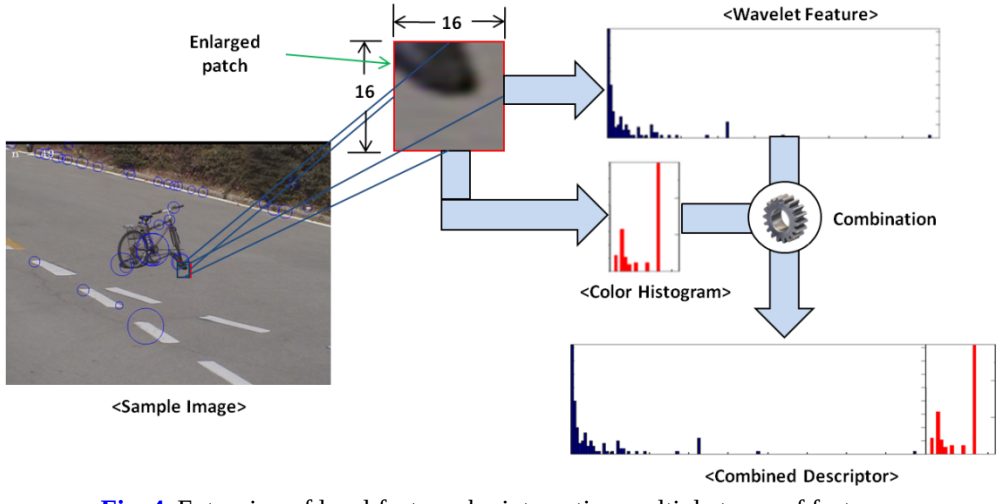

Fig. 4. Extension of local features by integrating multiple types of features.

4. Signature Generation

A codeword is considered as a representative of several similar patches. One simple method of generating codebook or dictionary is to perform k-means clustering [2] over all the local descriptors. Thus, each patch in an image is mapped to its closest codeword to generate a signature. A signature is a frequency histogram that counts number of patches closest to each codeword into a corresponding bin. **Fig. 5** and **Fig. 6** show codebook and signature generation processes in Bag of Feature respectively.

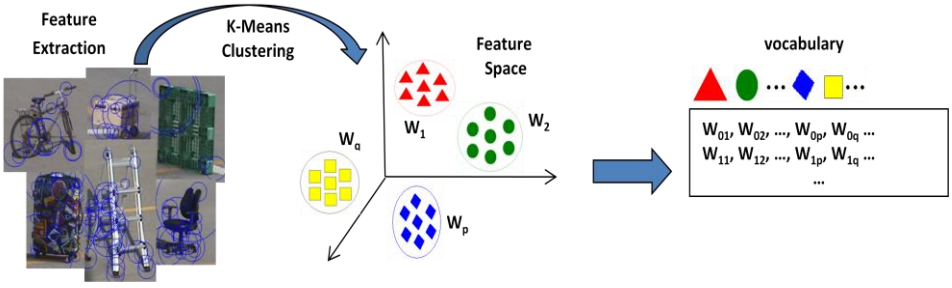

Fig. 5. Codebook generation using Bag of Features Model.

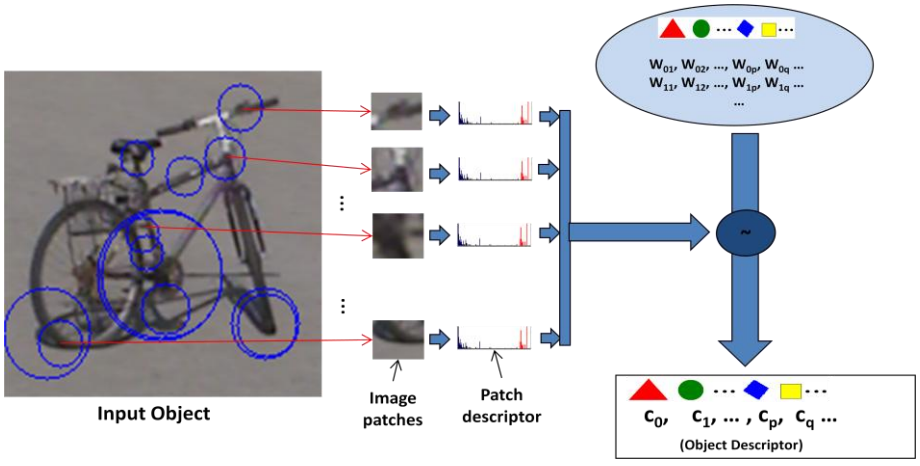

Fig. 6. Local features are mapped to global space for signature generation.

5. Classification

Given a signature, we prepare object models using Naïve Bayes algorithm which is discussed below in detail. **Fig. 7** displays an example of object classification system using NB.

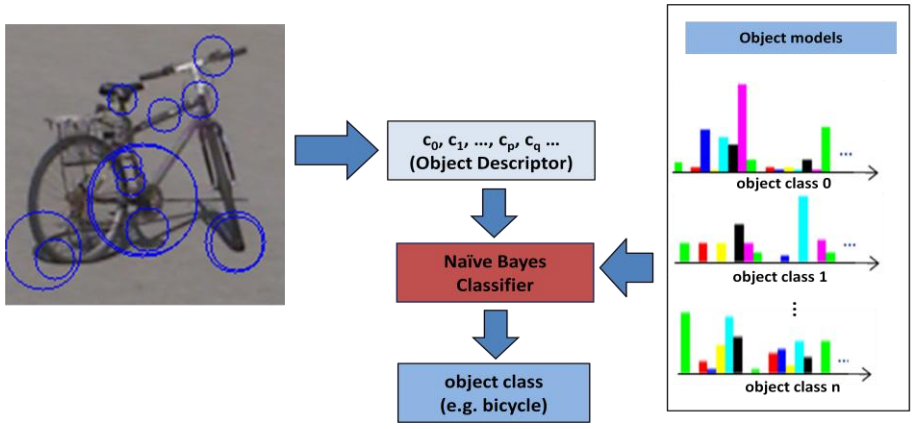

Fig. 7. Example of object classification using Naïve Bayes.

Given a hypothesis h and data D which bears on the hypothesis. The posterior probability of h given D, denoted by $p(h|D)$, can be estimated as $p(h|D) = \frac{p(D|h)p(h)}{p(D)}$. Given a set of hypotheses, defined as $H = \{h_i | i = 1,2,\dots,n\}$, the maximum of posteriors of all hypotheses is calculated to determine the category of the data D. Maximum A Posterior (MAP), here denoted as H_{MAP}, serves this purpose. MAP for H can be calculated as Eq. (6).

$$\begin{aligned} H_{MAP} &= \underset{h_i \in H}{\operatorname{argmax}} \; p(h_i|D) \\ &\equiv \underset{h_i \in H}{\operatorname{argmax}} \; \frac{p(D|h_i)p(h_i)}{p(D)} \\ &\equiv \underset{h_i \in H}{\operatorname{argmax}} \; p(D|h_i)p(h_i) \end{aligned} \tag{6}$$

Assuming a uniform prior, i.e., $P(h_i) = P(h_j)$ for all h_i, h_j belonging to H, we obtain Maximum Likelihood, $H_{ML} = \underset{h_i \in H}{\operatorname{argmax}} \, p(D|h_i)$. Consider $D = (d_1, d_2, \dots, d_n)^T$, then H_{MAP} can be calculated using Eq. (7).

$$H_{MAP} \equiv \underset{h_j \in H}{\operatorname{argmax}} \; p(d_1, d_2, \dots, d_n|h_j)p(h_j) \tag{7}$$

An independence assumption that all attribute elements in D are conditionally independent given the target value is called Naïve Bayes (NB). Then $p(d_1, d_2, \dots, d_n|h_j)$ can be writen as Eq. (8).

$$p(d_1, d_2, \dots, d_n|h_j) = \prod_i p(d_i|h_j) \tag{8}$$

To prevent underflow NB uses the formulation in Eq. (9).

$$C_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log p(c_j) + \sum_{i \in positions} \log p(x_i|c_j) \tag{9}$$

To avoid overfitting NB applies normalization as Eq. (10).

$$\hat{p}(x_i|c_j) = \frac{N(d_i=x_i, C=c_i)+1}{N(C=c_i)+k} \quad (10)$$

Where k is some constant.

Eq. (11) shows the classification process of the given component D where H denotes a set of categories.

$$\text{classify}(D) = \underset{h_i \in H}{\text{argmax}}\, p(h) \prod_{i=1}^{n} p(D_i|h) \quad (11)$$

6. Experiments

We tested our method on two different data sets. One is self collected data set (Data Set-I) using Sony PTZ camera (EVID70) and the other is bench mark Caltech101 data set (Data Set-II).

Data Set-I: We capture images of 6 categories such as Bicycle, Chair, Box, Ladder, Luggage, and Pallet using 2 PTZ cameras. For each category, we capture images in 8 different views, and 3 different zoom factors. Thus we have a total of 6×2×8×3 or 288 images each with resolution of 640×480. We manually crop the object area from every image and thus construct the Data Set-I. **Fig. 8** shows a few examples of images and cropped area from the data set-I. From now on, Data Set-I will represent the object area from this six categories with 2-fold cross validation arragement which alternatively takes 50% of the samples for training and 50% for testing.

Fig. 8. (Top row) Examples of objects from self-collected images. One sample object from each category such as Bicycle, Chair, Box, Pallet, Luggage, and Ladder. (Bottom row) object area cropped from the images in the top row.

Data Set-II: It has images of 101 distinct objects categories and one background category. The Caltech 101 data set consists of a total of 9,146 images, split between 101 different object categories, as well as an additional background/clutter category. Each object category contains between 40 and 800 images. Each image is about 300x200 pixels. **Fig. 9** shows a few examples of images from Caltech101 Data Set. From now on, Data Set-II will represent 4 classes of objects such as Car_Side, Laptop, Sunflower, and Motorbike each having 40 images in 2-fold cross validation arrangement.

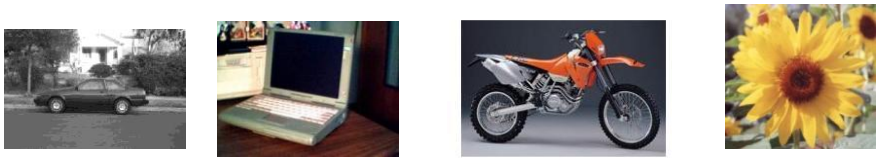

Fig. 9. Images from Caltech101 Dataset. (Left to right) Car_Side, Laptop, Sunflower, and Motorbike.

6.1 Parameter Optimization

Performances of these descriptors are dependent on real world challenges; therefore several parameters for them need tobe adjusted for optimum performance. The most important parameters include the blurring parameters '*sigma*' in SIFT, '*hessian threshold*" in SURF, number of dimensions in global feature space or "*codebook size"*, and time complexities as well. In this section, we investigate and show their influences in performance metrics by experimental results.

Fig. 10 displays plot of elapsed time required for generating codebook or dictionary varying the number of clusters in it. Here, codebook in Bag of Words model means a new feature space which clusters image patches and creates codewords. Codebook size represents the number of codewords in the codebook. As the number of codewords increases the time spent for codebook generation increases. It should be noted that codebook generation is performed offline. That is why we are not much concerned about it always.

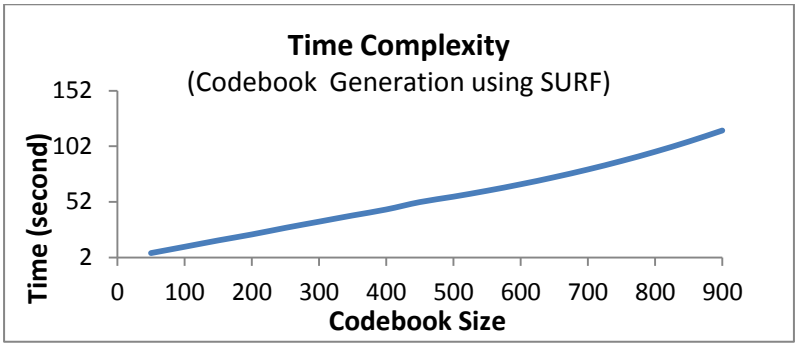

Fig. 10. Elapsed time required for codebook generation varying number of clusters.

Fig. 11 compares time spent for object classification depending on codebook size. As we have described, in the global signature generation method of classification stage, we need a global signature from object imagery. The length of global signature in this case is same as the length of codebook size or number of codewords in the codebook. Thus, the time spent

for signature generation process is directly related to the codebook size. Since, signature generation is an online process in real world applications, the time spent for signature is one of the major concerns. This figure shows that as the number of codewords increases the time for signature generation and classification increases, but the amount is still reasonable.

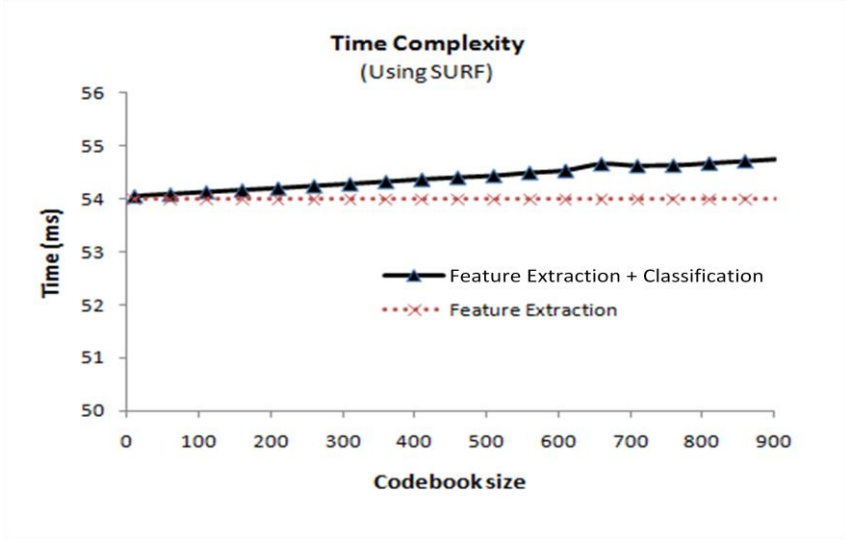

Fig. 11. Time complexities in feature extraction and classification.

Fig. 12 depicts a curve illustrating the level of impurity denoted by 'entropy' depending on codebook size. When we generate codebook, we cluster all the local patches together without any knowledge of where they come from. That means, a cluster may be formed with samples from multiple types of object categories. Even though local patches or samples from multiple object categories are visually different, if they come to same cluster they mean same thing to the classification system. As a cluster is highly mixed up with patches from various object categories, its impurity increases and thus it becomes less distinctive or representative. When we extract features from training images tens of thousands of features are extracted and when we cluster them into smaller number of codewords they get mixed up with each other. That is why, in the following figure we can see that when number of codewords increases the impurity or entropy decreases. Lower entropy means better codeword. This figure shows average impurity of a codeword after codebook generation.

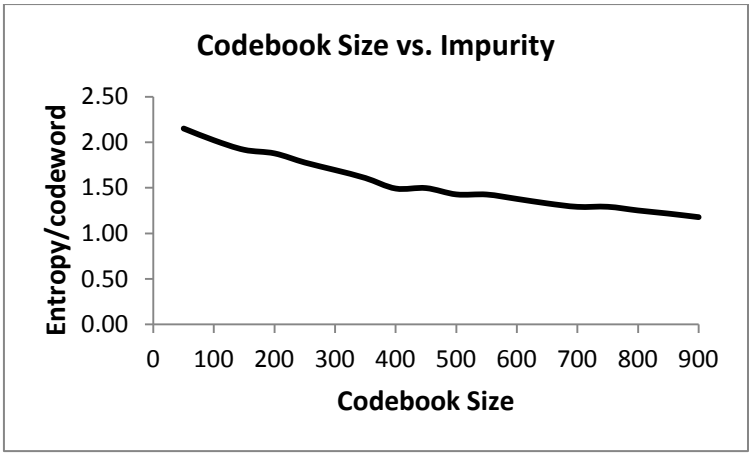

Fig. 12. Codebook size vs. Entropy/codeword.

Fig. 13 compares object classification accuracy in train and test stages using SURF feature. In this case, half of the object samples are used for building object models and remaining are used for testing. The curve for 'Train' is obtained by classifying the training samples by their own models. In 'Test' remaining samples from the set are tested using the object models built in train stage. Naturally, the 'Train' curve outperforms 'Test' curve. It is seen in the figure that as the number of codebook increases the overall classification accuracy increases at the beginning. As it reaches to some codebook size its performance seems saturated. So, we select the point where the curves seem to be saturated. In this figure, the shaded area represents the saturation point. So, in the following experiments we used codebook size of 430. The fluctuations in this figure can be fixed by increasing the number of training test samples. **Fig. 14** is obtained by increasing the number of test samples. In this case we test 1525 samples while in the previous case we tested 300 samples.

Fig. 15 explores classification accuracy depending on the number of bins in color histogram. This optimization is performed by observing its impact both on training and test stages. In our proposed descriptor we integrate color information with state of the art texture based descriptors. The descriptor size is directly related to the number of bins obtained from color descriptor. That is why we try to explore its impact here. From this figure, we can conclude that when we use small number of bins in color histogram, the accuracy is less. As the number of bins increases classification accuracy increases, but becomes almost steady at some points. In this figure, it is around 16 bins. That is why we use 16 bin histograms in later experiments.

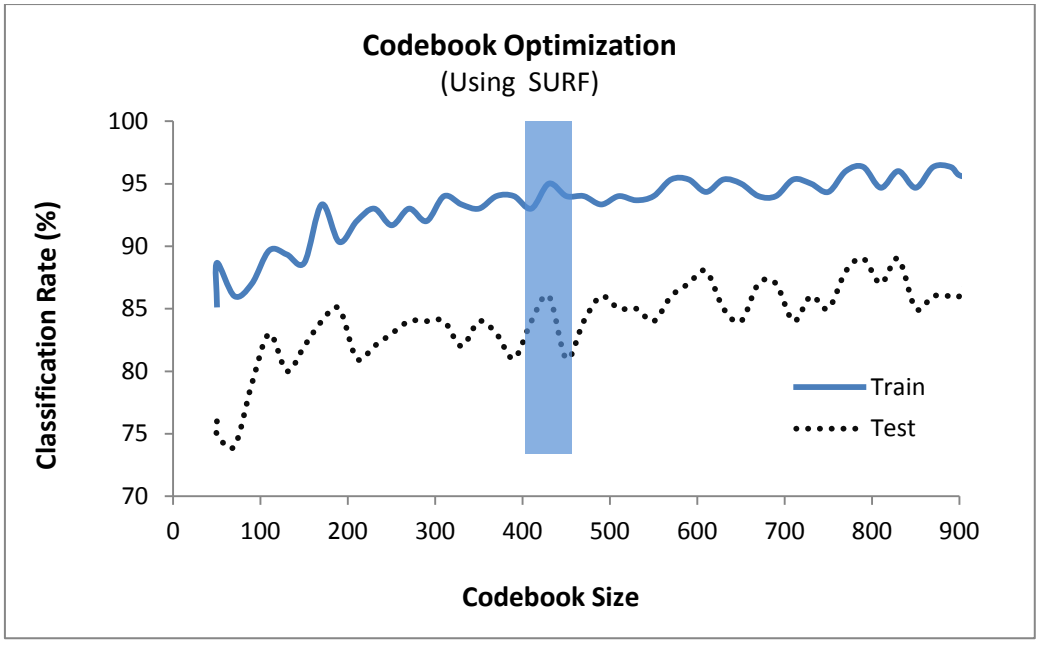

Fig. 13. Optimization of codebook size.

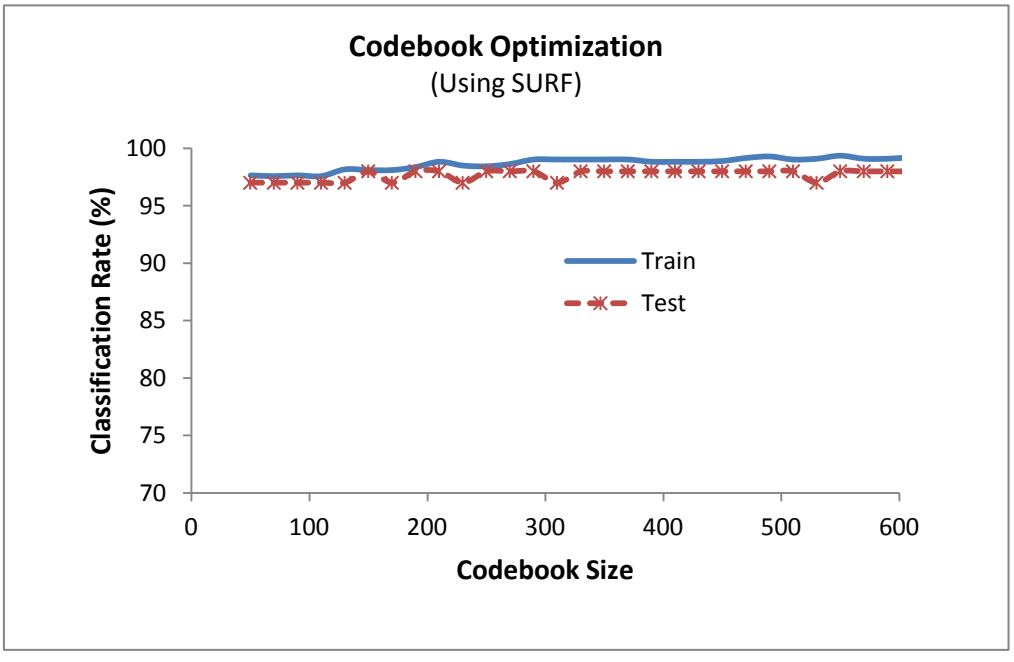

Fig. 14. Optimization of codebook size using large set of test samples.

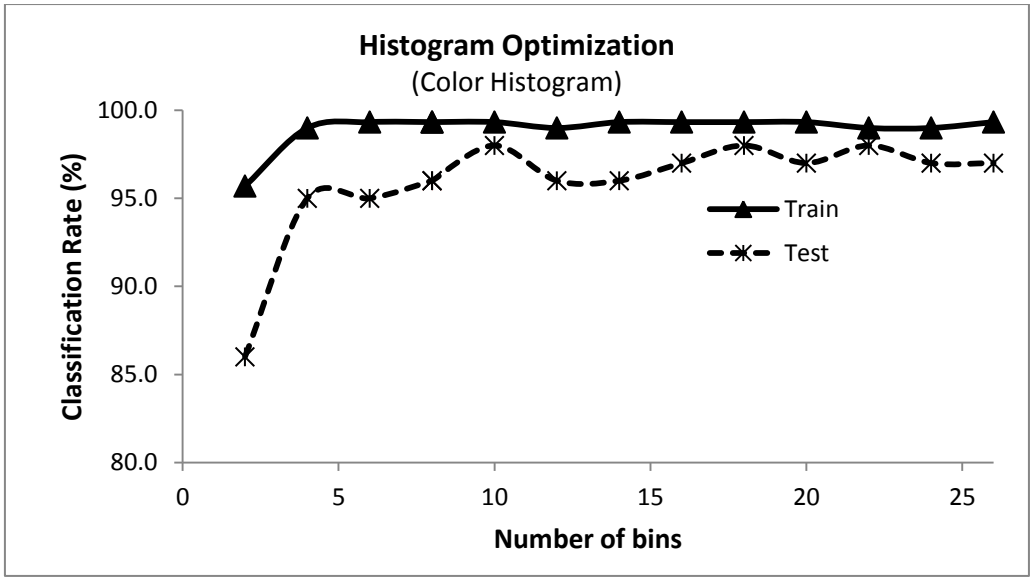

Fig. 15. Classification accuracy depending on the number of bins in color histogram.

6.2 Experimental Results

Object classification is highly biased to the nature of descriptor generated from the sample images. **Fig. 16** illustrates such result when applied on data set-I. Here, we use the same descriptor SUWCH in the Artificial Neural Network, Support Vector Machine and Naïve Bayes classifiers. It is seen that Naïve Bayes outperforms the other two state of the art classifers. Naïve Bayes is a classification technology which works with the concept of probabily in the basement. Since our global descriptor is generated using bag of features method, it also holds the concept of probibilistic idea. This is why Naïve Bayes suits most in this purpose. Here, SUWCH is an integration of SURF and Color Histogram.

Fig. 17 discovers the strength of our descriptor. In this experiment we compare performances of different features using bar charts. We integrate color descriptor with texture-based descriptors such as SIFT or SUFT and form SURFRGB, SURFHSV, SIFTRGB, and SIFTHSV. Here, SURFRGB is an integration of SURF and RGB color histogram, SURFHSV is an integration of SURF and HSV color histogram, similarly for SIFTRGB and SIFTHSV. In test results we see that SURF, SIFT, SURFRGB, SURFHSV, SIFTRGB, and SIFTHSV obtains 81, 82, 95, 98, 89, and 94 percent of accuracy respectively where SURFHSV is the highest. That is why we select SURFHSV as our local descriptor. From now on, we will refer SURFHSV as SUWCH.

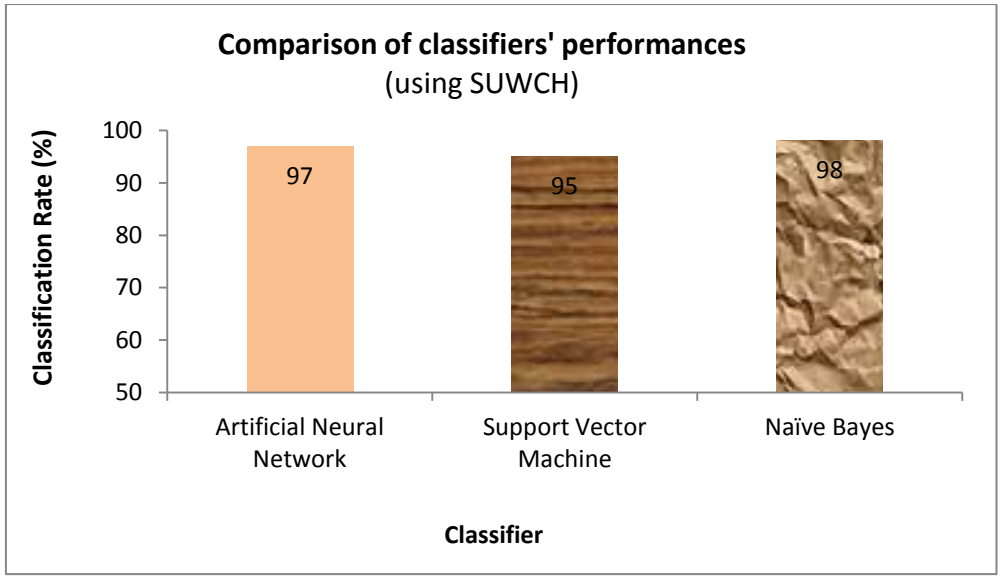

Fig. 16. Classification accuracy achieve for SUWCH descriptor using Artificial Neural Network, Support Vector Machine, and Naïve Bayes classifiers.

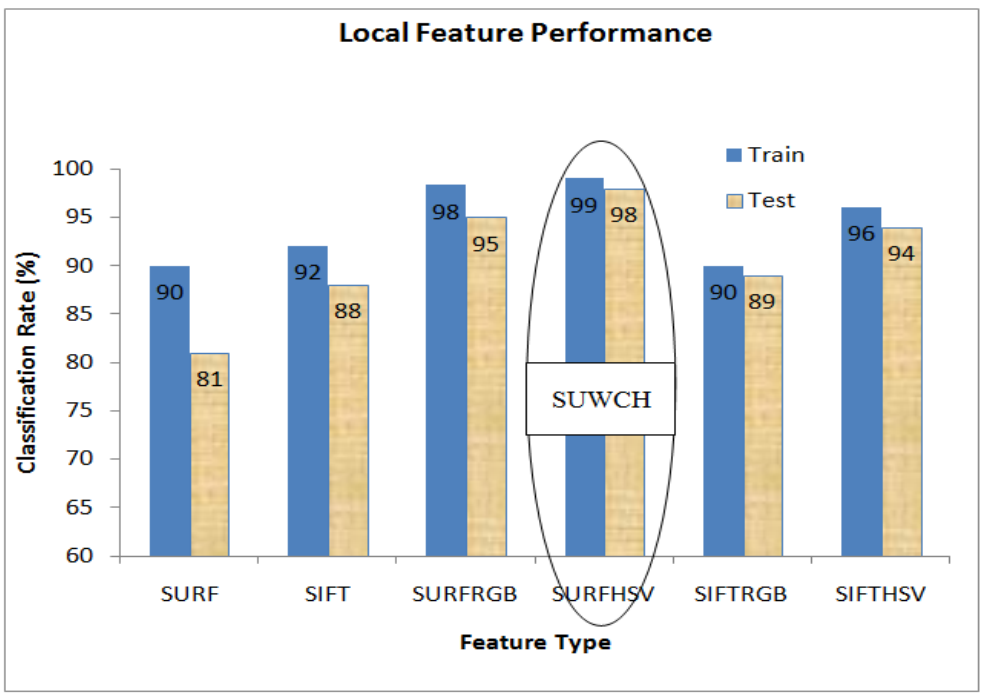

Fig. 17. Classification accuracy for various feature types from our self-collected dataset.

Classification accuracy is highly biased by the properties of the samples. Since we use same descriptor for all object categories, it works well for some categories and bad others. **Fig. 18** illustrates such experimental result. It compares classification efficiencies for state of the art SURF, SIFT and our proposed SUWCH descriptors. We can apparently see SUWCH outperforms other descriptors. We can also see the large variation in classification rates for different object categories. We are fortunate enough that our proposed method does not suffer much with this issue. Even we have improved it using inra-class spliting.

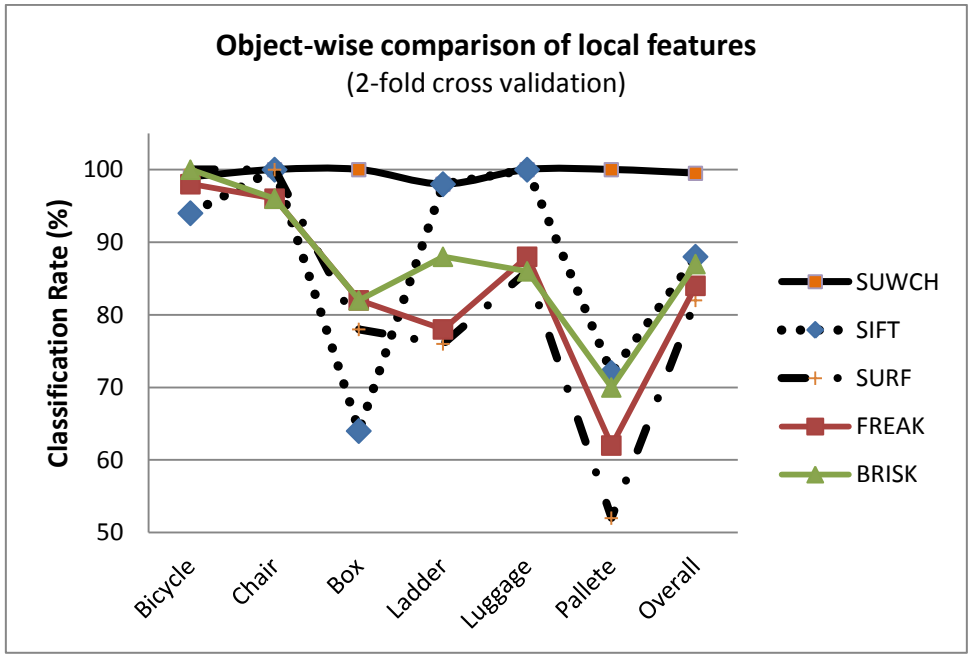

Fig. 18. Classification accuracy of different object categories using SIFT, SURF and SUWCH descriptors.

Fig. 19 compares the classification rates obtained before intra-class splitting and after splitting. Here, intra-class splitting means dividing object samples of a category into multiple smaller groups. It is required when any category has very large variation in appearances. Thus increases confusions while classified. In this example, we can see that the object category ‘pallet’ obtains very low classification rate compared to others such as bicycle, chair, box, ladder, and luggage. Thus it reduces overall classification performance. When we analyze these samples we find that ‘pallet’ has very inconsistent patterns. So, the representative codewords dominated by ‘pallet’ has much higher impurity than others. To overcome this issue, we split the pallet samples depending on visual appearances such as ‘front-view’ and ‘side-view’ and treat them as different object categories while modeling in train stage. This method helped to reduce false positives and thus increased classification accuracy. In this Fig. we can clearly see the impact of intra-class split. We enhance

classification performance by intra-class spit. In future we shall device some method for automatic intra-class spliting.

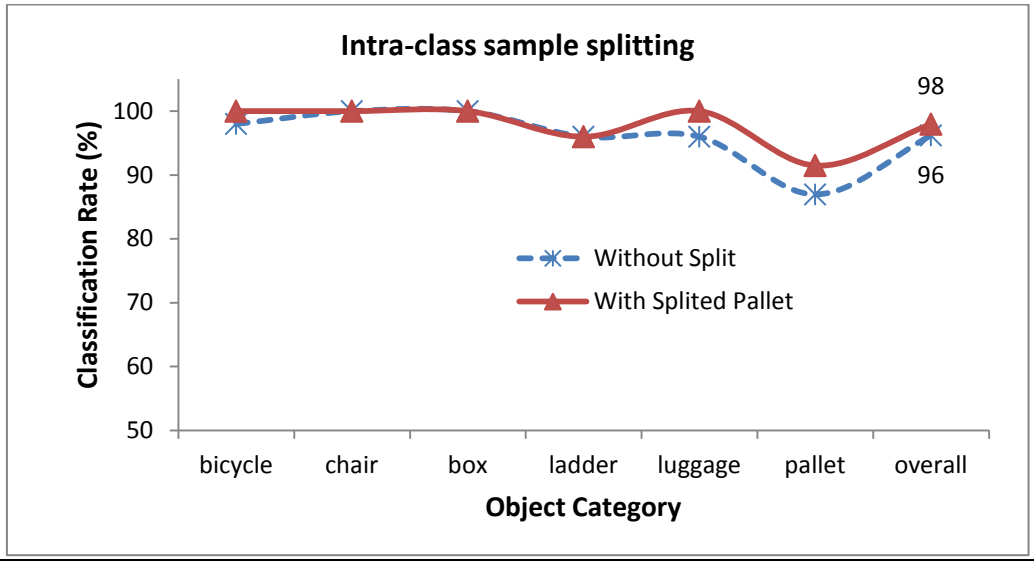

Fig. 19. Impact of intra-class splitting.

Usually as the number of classes increases the classification accuracy decreases. But in case of the proposed descriptor does not decrease much. It is satisfactory even if we increase the number of classes to 6. **Fig. 20** illustrates the result. . **Fig. 21**, we compare the efficiencies of our proposed descriptor with other descriptors in data set-II.

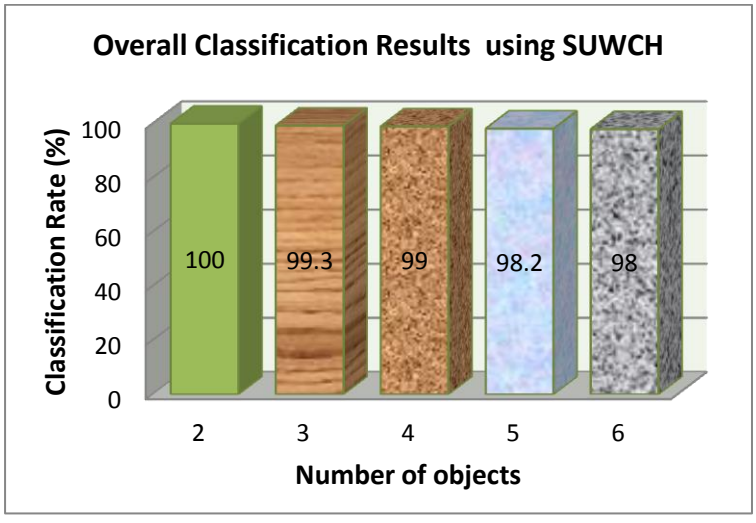

Fig. 20. Overall classification rate obtained for various number of object categories in data set-I.

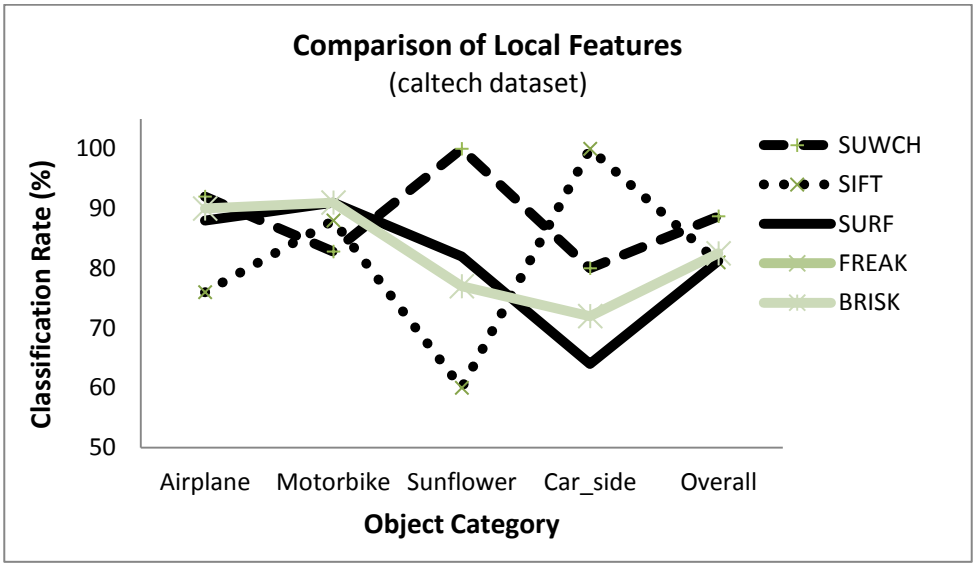

Fig. 21. Comparison of descriptors' efficiencies using 4 object categories from benchmark Caltech101 data set.

Besides, classification accuracy another important aspect comes to the light of discussion. It is computation time which is very important for real-time application. Now a days several descriptors are in battles with their computation time beside accuracy. The computation time depends on the number of local keypoints detected by the key point detectors. Among the descriptor SURF and SIFT are both detector and descriptor but FREAK and BRISK are only descriptor. According to literature in [21] 3x slower than FREAK while used in INRIA pedestrian dataset. In our future work we will present detail analytical result of state of the art descriptors.

Fig. 22 displays a bar chart of elapsed time required for various types of feature extraction from an image in an average. Here, elapsed time means total time spent for locating interest regions and feature extraction. In this figure we can see that feature extraction time in SURF is much shorter than SIFT extraction. Usually SURF spends almost one third of SIFT extraction time. When we integrate color histogram (here we denote by 'CH') with SURF it adds very small amount of time compared to SIFT and SURF. SUWCH is an integration of SURF and Color Histogram.

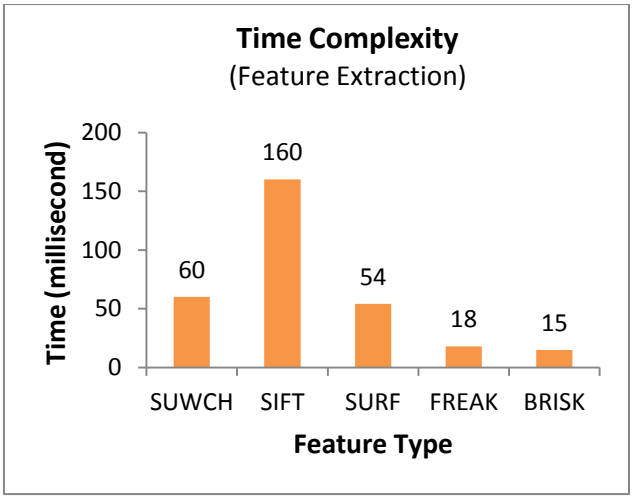

Fig. 22. Elapsed time required for extraction of various types of features.

7. Conclusions

We integrate sum of wavelet responses of SURF and color histogram from nearby regions and form speeded up wavelet response and color histogram (SUWCH). Our approach can be evaluated with the quantitative performance of object cataloging, and computational complexities. We show that our proposed approaches help the overall system to outperform cutting edge technologies in respect of computational time and performance. For our self collected dataset, our contributions in feature extraction increases classification accuracy to 98% while state of the art descriptors SURF, SIFT, FREAK and BRISK achieve 81% 88%, 82%, 84% and 87% respectively. The issues with computational complexity along various environmental challenges such as variation is scale, rotation, affine, illumination, blurring etc will be addressed in future work. Generalization of object categories deserves more research on it in order to be used in real world applications. In future, we would like to evaluate our approach on more natural datasets, since this is more likely to reveal the advantage of having models with a large number of parts.

Acknowledgments

This study was conducted with the assistance of the Korea Aerospace University Technical Research Center of the next generation broadcast media by the GRRC(Gyeonggi-do Regional Research Center) program.

References

[1] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. of the Workshop on Statistical Learning in Computer Vision*, Czech Republic , pp.1-22, 11-14 May, 2004. http://www.xrce.xerox.com/Research-Development/Publications/2004-010

[2] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Proc. of the IEEE International Conference on Computer Vision (*ICCV*)*, San Diego, USA, pp. 370-377 , 2005. Article (CrossRef Link)

[3] K. Grauman, and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. of the IEEE International Conference on Computer Vision (*ICCV*)*, San Diego, USA, pp. 1458 - 1465, 17-20 October, 2005. Article (CrossRef Link)

[4] K. Grauman, and T. Darrell, "Unsupervised learning of categories from sets of partially matching image features," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), vol. 1, pp. 19 – 25, 17-22 June, 2006. Article (CrossRef Link)

[5] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Describing visual scenes using transformed dirichlet processes," in *Proc. of the International Conference on Neural Information Processing Systems* (NIPS), pp. 1299-1306, 2005. http://papers.nips.cc/paper/2772-describing-visual-scenes-using-transformed-dirichlet-processes

[6] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *Proc. of the IEEE International Conference on Computer Vision* (ICCV), San Diego, USA, pp. 1331-1338 , 17-20 October, 2005. Article (CrossRef Link)

[7] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Depth from familiar objects: A hierarchical model for 3d scenes," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), New York, USA, pp. 2410 - 2417 , 17-22 June, 2006. Article (CrossRef Link)

[8] E. Sudderth, A. Torralba, W. T. Freeman, and A. Willsky, "Describing visual scenes using transformed objects and parts," *International Journal of Computer Vision* (IJCV), vol. 77, no. 1-3, pp. 291–330, 2008. Article (CrossRef Link)

[9] L. Fei-Fei, P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), San Diego, USA, vol. 2, pp. 524 – 531, 20-26 June, 2005. Article (CrossRef Link)

[10] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, vol. 94, no. 2, pp. 115-147, 1987. Article (CrossRef Link)

[11] Y. Amit, and D. Geman, "A computational model for visual selection," *Neural Computation*, vol. 11, pp. 1691-1715, 1999. Article (CrossRef Link)

[12] D. Roth, M. H. Yang, and N. Ahuja, "Learning to recognize 3D objects", *Neural Computation*, vol. 14, no. 5, pp. 1071-1104, 2002. Article (CrossRef Link)

[13] A. J. Colmenarez, and T. S. Huang, "Face detection with information-based maximum discrimination," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), San Juan, Puerto Rico, pp. 782-787, 17-19 June, 1997. Article (CrossRef Link)

[14] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI), vol. 20, no. 1, pp. 23-38,

1998. Article (CrossRef Link)
[15] M. H. Yang, D. Roth, and N. Ahuja, "A SNoW-based face detector," *Advances in Neural Information Processing Systems* (NIPS), vol. 12, pp. 855-861, 2000. http://papers.nips.cc/paper/1747-a-snow-based-face-detector
[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* (IJCV), vol. 60, pp. 91–110, 2004. Article (CrossRef Link)
[17] D. G. Lowe, "Local feature view clustering for 3D object recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) , Kauai, Hawaii, pp. 682-688, Dec. 2001. Article (CrossRef Link)
[18] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," *Journal of Computer Vision and Image Understanding* (CVIU), vol. 110, no. 3, pp. 346-359, 2008. Article (CrossRef Link)
[19] A. Alahi, R. Ortiz and P. Vandergheynst, "FREAK: Fast Retina Keypoint," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp. 510-517, 2012. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6247715&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6247715
[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. of IEEE International Conference on Computer Vision* (ICCV), pp. 2548-2555, 2011. Article (CrossRef Link)
[21] C. Schaeffer, "A comparison of keypoint detectors in the context of pedestrian detection: FREAK vs. SURF vs. BRISK," 2013. http://cs229.stanford.edu/proj2012/Schaeffer-ComparisonOfKeypointDescriptorsInTheContextOfPedestrianDetection.pdf
[22] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," *International Conference on Digital Signal Processing* (DSP), pp. 1–7, 2013. Article (CrossRef Link)
[23] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), Hawaii, vol. 1, pp. 511-518, 2001. Article (CrossRef Link)
[24] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR), San Diego, USA, pp. 886-893, 20-26 June, 2005. Article (CrossRef Link)
[25] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), New York, USA, vol. 2, pp. 2169 - 2178, 17-22 June, 2006. Article (CrossRef Link)
[26] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *Proc. of the IEEE Computer Vision and Pattern Recognition* (CVPR), Miami, FL, USA, pp. 1030-1037, 20-25 June, 2009. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5206727&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5206727
[27] R. Hess, A. Fern, and E. Mortensen, "Mixture-of-parts pictorial structures for objects with variable part sets," in *Proc. of the IEEE International Conference on Computer Vision* (ICCV), Rio de Janeiro, Brazil, pp.1-8, 14-20 October , 2007. Article (CrossRef Link)
[28] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Proc. of the IEEE International Conference on Computer Vision* (ICCV), Nice, France, pp. 952-957, 13-16 October, 2003. Article (CrossRef Link)

[29] M. K. Islam, F. Jahan, J. H. Min, and J. H. Baek, "Object classification based on visual and extended features for video surveillance application," in *Proc. of the 8th Asian Control Conference (ASCC)*, Kaohsiung, Taiwan, pp. 1398 – 140, May 15-18, 2011. http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Object%20classification%20based%20on%20visual%20and%20extended%20features%20for%20video%20surveillance%20application

Mohammad Khairul Islam received his Bachelor of Science (Engineering) degree in Electronics & Computer Science in December 1998 from Shahjalal University of Science & Technology, Bangladesh. He obtained his Master of Engineering and Doctor of Engineering degrees in Information and Telecommunication Engineering in August 2007 and August 2011 respectively from Korea Aerospace University, South Korea. He presently serves as an Associate Professor in the Department of Computer Science and Engineering, University of Chittagong, Bangladesh. His research areas are multimedia, image processing, and computer vision.

Farah Jahan received her Bachelor of Science (Honors) degree in Computer Science & Engineering in December 2005 from the University of Chittagong, Bangladesh and Master of Engineering degree in Information and Telecommunication Engineering in August 2011 from Korea Aerospace University, South Korea. Currently she is pursuing her Doctoral program under the School of Information and Communication Technology, Griffith University, Brisbane, Australia. She presently holds the position of Assistant Professor in the Department of Computer Science and Engineering, University of Chittagong, Bangladesh. Her research areas include multimedia, image processing, and computer vision.

Joong-Hwan Baek received his B.S. degree in Telecommunication Engineering from Korea Aerospace University in 1981. He received his M.S. and Ph.D. degrees from Oklahoma State University in 1991 and 1987, respectively. He has been a professor of School of Electronics and Information Engineering at Korea Aerospace University since 1992. His research interests include image processing, computer vision, pattern recognition, and multimedia.