A COMPARATIVE REVIEW OF DWT AND SURF FOR RETRIEVING IMAGE DATA ON A LARGE SCALE

Vipin Gopalakrishnan Nair[1], Nadir N. Charniya[2]

[1]ME Student, Electronics and Telecommunication Department, V.E.S Institute of Technology, Maharashtra, India
[2]Professor, Electronics and Telecommunication Department, V.E.S Institute of Technology, Maharashtra, India

Abstract

The advent of digital era and advancement in the field of imaging technology has increased the amount of images present in the domain to billions. More and more images are created every moment through one source or the other. It is highly inconvenient in such a case to search for a few images of our interest. It is in such a scenario that image search algorithms gain prominence. Research is highly focused on improving the efficiency and ability of algorithms to handle large datasets as well as minimizing the time for searching a query drastically. Bag of Words model is the approach employed by almost every state of the art method. Algorithms work by detecting the most distinguishing feature in an image, representing it in compact form and matching it with features in other images. Accuracy of the result depends on how well the features are detected, extracted and matched. This paper presents two different methods for content based image retrieval: Discrete Wavelet Transform (DWT) and Speeded Up Robust Features (SURF). DWT employs decomposition of an image by wavelets while SURF represents an image in scale space by blurring and using box filters. In this paper we look into detail of how the mechanism of image search is achieved by both the methods.

Keywords: *Content Based Image Retrieval, Discrete Wavelet Transform, Wavelet Decomposition, Speeded Up Robust Features (SURF), Hessian Matrix.*

---***---

1. INTRODUCTION

Feature detection, description of the interest region and feature matching is the core of any image search algorithm. Information contained in images that are difficult to correlate and understand by human beings can be transformed into numerical values called features. The dimension of features extracted from an image is of far lesser magnitude than the original image. The features of an image as a whole are termed as global features while those extracted from the image patches are called local features. Blobs, corners, edges are some of the local features in an image. Selecting regions of an image that are distinguishable from its immediate neighborhood is termed as feature detection. Features from accelerated segment test (FAST) and Binary Robust Invariant Scalable Keypoints (BRISK) detect corners while SURF [1] detects blobs within an image. The most popular local descriptors for image searching are Scale Invariant Feature Transform (SIFT), SURF, Oriented FAST and Rotated BRIEF (ORB), BRISK, Maximally Stable Extremal Regions (MSER) and Fast Retina Keypoint (FREAK) [2]. Bag of Words model [3] on which most algorithms are based, detect features by any of the detectors. The features are then extracted by the local feature descriptors and the features are represented in compact vector form. Using efficient initialization methods, the k-means clustering [4] algorithm groups the feature vector into clusters which are then converted to visual words. SIFT [5] detects features which are invariant to scale and rotation by using scale space representation. The scale space is generated by blurring the image repeatedly with a Gaussian function. SURF [6] uses integral images which simplify computations greatly and the fast Hessian matrix detects corners. The performance of SURF is superior as compared to SIFT. DWT [7] uses wavelet decomposition to separate both the upper and lower half of a signal spectrum by bandpass and lowpass filters. The process is iterated to give the information stored at different resolutions, which is feature extraction. The energy and standard deviation is calculated at each of the subbands in the decomposed image to obtain the features.

2. RELATED WORK

The most widely used approach for image search is the content based image retrieval. In this section, we look into some of the work in this domain. In [8], Ekta Gupta *et al* decomposed an image into horizontal, vertical and diagonal regions by Haar Discrete Wavelet Transform and the features extracted by the Gray Level Co-occurrence Matrix (GLCM) classified it using Support Vector Machine. S. Agarwal *et al* in [9] used both DWT and Edge Histogram Descriptor (EHD) to enhance the performance of image retrieval. The images are retrieved on the basis of shape and texture features alone. Edge Histogram Descriptor is used on select wavelet coefficients to collect information about dominant edge orientations. In [10] Hua-Zhang Wang proposed Dual Tree Complex Wavelet Transform (DT-CWT) to extract texture features which are rotationally invariant. DT-CWT orients the texture information in six different directions, improving the overall orientation as compared to DWT. In [11] Ahmed Naser Hussein used WT to analyze texture and measured entropy to identify a query image. Bhuvana Shanmugam *et al* in [12] segmented an image using quad tree segmentation and extracted features using SURF. The images are classified by Multiple Instance Learning Support Vector Machine and the visual codebook is designed by Lindae-Buzo-Gray (LBG) algorithm. Histogram Intersection (HI) is then used to measure the similarity between query and dataset images. In [13] Kai Chen *et al* identified a product using LIRe and SURF. Faster feature extraction of SURF combined with image resizing and Lucene indexing speeds up the process of image retrieval. N. Gopal and R. S. Bhooshan in [14] proposed an enhanced SURF descriptor which computes the Hu moments and eigen values in the interest area of detected keypoints to extract the image features. The descriptor could ably distinguish images of the same object with similar grayscale properties but different colors.

3. OVERVIEW OF METHODS

3.1 DWT

3.1.1 Wavelet Transform

A wavelet [15] is a wave whose amplitude varies between zero and a maximum value. It starts at zero, its amplitude increases and reaches a peak value and again returns back to zero. When wavelets are convolved with parts of a known signal, the information stored in an unknown signal can be extracted. Hence wavelets can be used as a mathematical tool to extract information from images. Wavelet decomposition [16] algorithms generate a set of complementary wavelets to decompose the image data without gaps and overlaps. A wavelet transform decomposes a signal into functions that are localized in real and Fourier space. Wavelet Transform [15] can be expressed by the following equation:

$$F(a,b)=\int_{-\infty}^{\infty} f(x)\psi^{*}{}_{(a,b)}(x)dx \qquad (1)$$

ψ is any arbitrary function and ψ^{*} is its complex conjugate. Wavelet Transforms are broadly classified as Discrete Wavelet Transform (DWT) [17] and Continuous Wavelet Transform. DWT decomposes a function into mutually orthogonal wavelets sets which are orthogonal to its translations and scaling. Pyramidal algorithm is the most prominent type of implementation of the DWT algorithm which employs two filters, constructed from the wavelet coefficients. These filters are used repeatedly to obtain information for all the scales. Haar Wavelet [18] and Daubechies Wavelet [18] are the most used wavelet transforms. DWT using Haar wavelets for a length 4 is as below:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \qquad (2)$$

Wavelets possess both location and frequency. The first wave completes zero cycles and the next, one complete cycle. The frequency of the second and third wave equals twice of the first wave. They don’t differ in frequency but location.

3.1.2 Wavelet Decomposition

In Wavelet Packet Decomposition (WPD) [16], the sampled discrete time signal is passed through more number of filters as compared to DWT. In WPD, the approximation coefficients and (cHj, cVj, cDj) for the 2-D case are decomposed to form the binary tree. Using a second order bandpass filter and a low pass filter, the upper and lower half of the signal spectrum is separated. The process is iterated to subsample the lower half of the spectrum. The first bandpass filter gives the difference between consecutive resolutions 2j and 2j-1. The next bandpass filter gives the difference between 2j-1 and 2j-2 and so on.

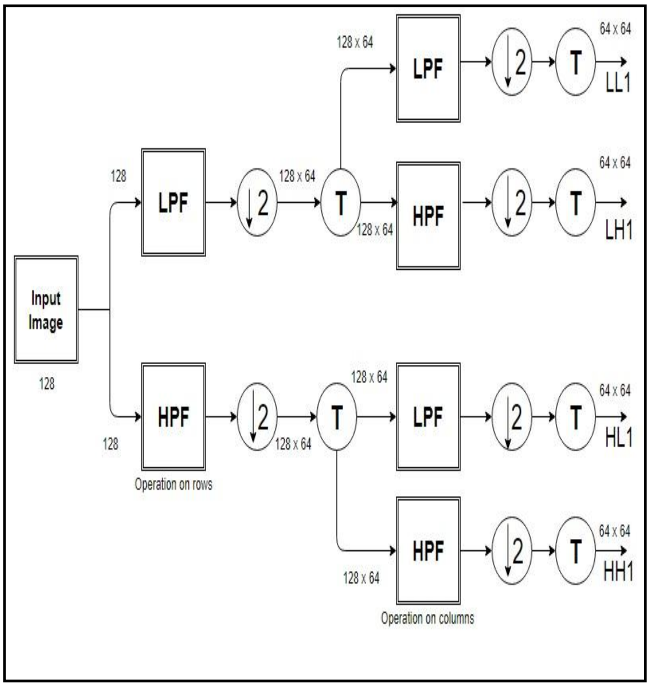

Fig. 1: First level of WT implementation scheme

3.1.3 Feature Extraction

Processing large amount of feature data which is irrelevant can consume much time of an algorithm making it slow. Hence the voluminous features need to be represented in a compact form for processing to take place faster. The features are reduced to set of compact representation in the form of feature vectors. Detecting and retrieving the most prominent features within an image is termed as feature extraction. The feature vectors are such that they represent the most relevant information from the input image and hence the desired task can be performed with this reduced representation. Feature extraction is performed on the wavelets after the wavelet decomposition stage. The main features that are extracted include color, texture and shape.

3.1.4 Texture Extraction

The visual characteristic of a surface or description of the arrangement of color or intensities spatially within an image is termed as texture. The amount of texture within an image can be represented in the form of a histogram; however it doesn't give the relative pixel location with respect to one another. For representing texture information correctly, we need to consider the distribution of intensities along with the relative position of pixels.

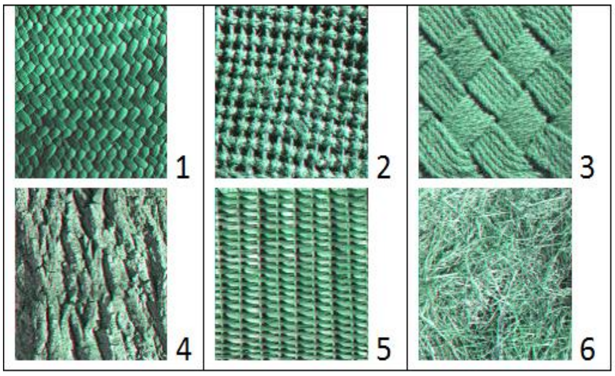

Fig.2: Various Texture Patterns

Gray Level Co-occurrence Matrix (GLCM) [19] extracts texture along with the information about relative position of pixels. Gray Level Co-occurrence Matrix (GLCM) is computed for a given image and then entropy of the matrix is calculated. Entropy of the GLCM denotes the Texture Descriptor. The entropy is calculated as follows:

$$Entropy = -\sum_{i}\sum_{j} c_{ij}\log(c_{ij}) \qquad (3)$$

c_{ij} is an element of GLCM matrix which denotes the number of times a given image element, F(x, y) = i and F(x + 1, y + 1) = j

3.1.5 Edge Descriptor

A given image is partitioned into 16 parts (4x4) in total. The edges in each part are classified into one of the following type: vertical, horizontal, 45° diagonal, 135° diagonal. If an edge doesn't fall into any of the above mentioned categories, it is termed as non-directional edge. Filtering with appropriate masks gives the prominent edge in each sub-image. All the prominent edges in the 16 sub-images are added to the corresponding bins in the edge histogram. The sum of all edges in the image gives the normalization factor. The ratio of the horizontal edge component of the image to the normalization factor gives the normalized horizontal edge. The prominent edge in each part is assigned a value of 1 and all others 0.

3.1.6 Color Descriptor

The most visual distinguishing factor in an image is color as human beings differentiate images on the basis of color feature. A given RGB image is converted to HSV plane as hue distinguishes color while saturation measures the amount of white light in a pure color.

3.1.7 Steps for Image Retrieval

Feature Extraction

1. Select each image from the database and decompose it with DWT up to the 3rd level.
2. Compute energy and standard deviation at each sub-band and store the results in vector form. Hence construct the feature vector for each image.

$$Energy = E_k = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|W_{i,j}\right| \qquad (4)$$

$$STD = \left[\frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(W_{i,j}-\mu_{i,j})^2\right]^{\frac{1}{2}} \qquad (5)$$

3. Length of the feature vector is calculated as the product of number of feature measures in combination and the number of sub-band.
4. Repeat step 1 to 3 for all the images in the dataset.
5. The resulting feature vector should be stored in matrix form.

Query Processing and Image Matching

1. Prompt a query image from the user and decompose it with DWT to construct its feature vector.

$$y_i = \left[E_1 \quad E_2 \quad E_3 \cdots E_k \quad \sigma_1 \quad \sigma_2 \quad \sigma_3 \cdots \quad \sigma_k\right] \qquad (6)$$

2. For an image I in the database, choose the value of a feature vector and denote it as

$$x_i = \left[E_1 \quad E_2 \quad E_3 \cdots E_k \quad \sigma_1 \quad \sigma_2 \quad \sigma_3 \cdots \quad \sigma_k\right] \qquad (7)$$

3. Calculate the distance between both the feature vectors given by

$$d_c(x,y) = \sum_{i=1}^{d}\frac{\left|x_i - y_i\right|}{\left|x_i\right| + \left|y_i\right|} \qquad (8)$$

4. Store the value of distance and corresponding index of image in a distance vector.
5. Sort the distance vector and display the most relevant results.

3.2 SURF

Pyramid representation [20] is used to transform an image into coordinates. In this technique, the image is subjected to repeat smoothing using a Gaussian pyramid and we obtain many samples of the image with same size but reduced bandwidth. A Gaussian pyramid weights down subsequent images using a Gaussian blur. Thus the original image is blurred ensuring that the interest points are scale invariant.

3.2.1 Detection

SURF approximates the determinant of the Hessian Matrix [6] to detect interest points by using integral image [6]. An integral image S(x, y) is defined as an image where each point stores the sum of all pixels in a rectangular region fit between the origin and the corresponding point.

$$S(x,y)=\sum_{i=0}^{x}\sum_{j=0}^{y}I(i,j) \qquad (9)$$

SURF uses box filter as an approximation to Gaussian smoothing. Using integral images facilitates faster filtering of the image with a box filter. The determinant of the Hessian matrix gives the scale and measure of local change around a point. Only those points where the determinant is maximum are chosen. For any given point p(x, y) within an image, the Hessian matrix H (p, σ) at point p and scale σ is given as:

$$H(p,\sigma)=\begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix} \qquad (10)$$

where,

$$L_{xx}(p,\sigma)=I(p)*\frac{\partial^2}{\partial x^2}g(\sigma) \qquad (11)$$

$$L_{xy}(p,\sigma)=I(p)*\frac{\partial^2}{\partial xy}g(\sigma) \qquad (12)$$

$L_{xx}(p,\sigma)$ represents convolution of the image with second order derivative of Gaussian. The convolutions are computationally expensive but can be speeded up using integral images and discrete kernels.

$$Det(H_{approx})=D_{xx}D_{xy}-(\omega D_{xy})^2 \qquad (13)$$

D_{xx} is the approximated kernel for $L_{xx}(p,\sigma)$
D_{xy} is the approximated kernel for $L_{xy}(p,\sigma)$

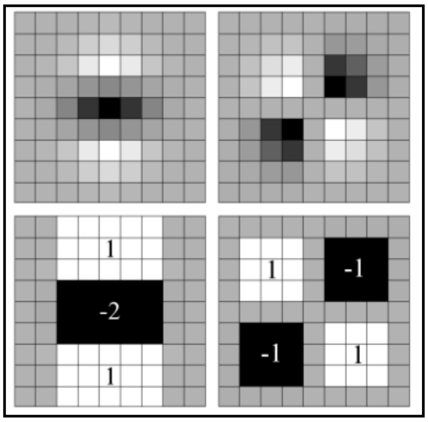

Fig 3: $L_{yy}(p,\sigma)$ and $L_{xy}(p,\sigma)$ and the approximations D_{yy} and D_{xy} [6]

3.2.2 Scale Space Representation and Location of Interest Points

The interest points are detected by capturing the image at different scales, and the same effect can be introduced by repeat blurring of the original image with a Gaussian function. Scale space representation of SURF is achieved by employing box filters of various sizes. The filter size is up-scaled to analyze the scale space. The filter size corresponding to size 9x9 is the initial scale layer for a scale s = 1.2. The image is then progressively filtered with bigger masks considering the discrete nature of integral images. The various filter sizes used are of size 9x9, 15x15, 21x21, 27x27 and so on. Finally the maxima of the Hessian matrix determinant are interpolated over scale and space.

3.2.3 Descriptor

A descriptor which is generated on the basis of pixels surrounding an interest point provides a unique description of a feature. The SURF feature descriptor computes the sum of Haar wavelet [18] response around the corner points. The computations can be simplified by use of integral images. To achieve rotational invariance, a unique orientation has to be determined for each interest point. The region of interest around each keypoint is then rotated to its direction. The area around each interest point is divided into 4x4 subareas which are described by the Haar wavelet response in the x and y direction.

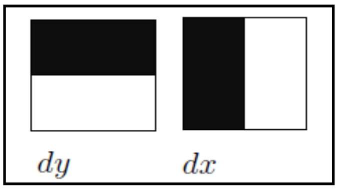

Fig. 4: Haar Wavelet responses in x and y direction. Black area corresponds to a weight of -1 and white area to size 1 for the Haar kernels [6]

dx - wavelet response in x direction
dy - wavelet response in y direction

A vector v is calculated for each subarea based on 5x5 samples given as

$$v = \left\{ \sum dx, \sum |dx|, \sum dy, \sum |dy| \right\} \tag{14}$$

A total of 16 vectors representing the subareas form the descriptor of an interest point.

3.2.4 Orientation Assignment

The Haar wavelet responses are weighted by a Gaussian function around the point of interest. The responses are then plotted in a two dimensional space and the sum of all responses within a sliding window of size (π/3) gives the dominant orientation. Within the window, both the horizontal and vertical responses are added to yield a local orientation vector, the longest of which defines the orientation of the interest point.

3.2.5 Matching

The distance between two feature vectors is a parameter to find the match between two descriptors. The vector with the least distance is chosen as the perfect match. The Laplacian is the trace of the Hessian matrix given as:

$$\nabla^2 L = tr(H) = L_{xx}(p,\sigma) + L_{yy}(p,\sigma) \tag{15}$$

The sign of the Laplacian speeds up the matching process as it distinguishes bright spots on dark background. Only vectors with the same sign of Laplacian needs to be compared which significantly cuts down the time for matching.

3.2.6 Steps for Retrieval

Training

1. Before executing the algorithm, batch convert all the dataset images to a fixed and common aspect ratio.
2. For a given image in the dataset, convert it into grayscale
3. Detect SURF features over the image. Use detectSURFFeatures(I) function readily available in MATLAB, where I is the image.
4. Retain only the valid SURF points by points.selectStrongest () in MATLAB and specify the number of points required as arguments.
5. Calculate SURF descriptors around each keypoint by the extractFeatures(I, points) function and store the feature vectors.
6. Repeat the above steps for all images in the dataset and store the feature vectors and valid points into a codebook.

Testing

1. Prompt for a query image from the user.
2. Convert the query image into grayscale.
3. Detect the SURF points over the query image similar to the training stage.
4. Calculate the SURF descriptors around each keypoint and store the feature vectors.
5. Compare and match the database feature vectors with the query image ones. Match the features and generate an index table for pairing query image features with the dataset.
6. Define a threshold for the maximum number of points matched and class identification. The class with the maximum matches qualifies as the output image.

4. EXPERIMENTS

We evaluate both the approaches of DWT and SURF on the UKBench Dataset [21]. The dataset contains 1072 images in groups of four which are placed adjacent to each other. We select 240 unique images from the dataset and use it for training purpose. 720 variations of these images are used for the testing purpose. Each of the 240 images is labelled into 240 classes.

4.1 DWT Method

We provide an arbitrary image from the testing data to check the effectiveness of the algorithm. DWT identifies the correct class to which the query image belongs to, but in addition to that, certain irrelevant images also get matched which are shown in the results.

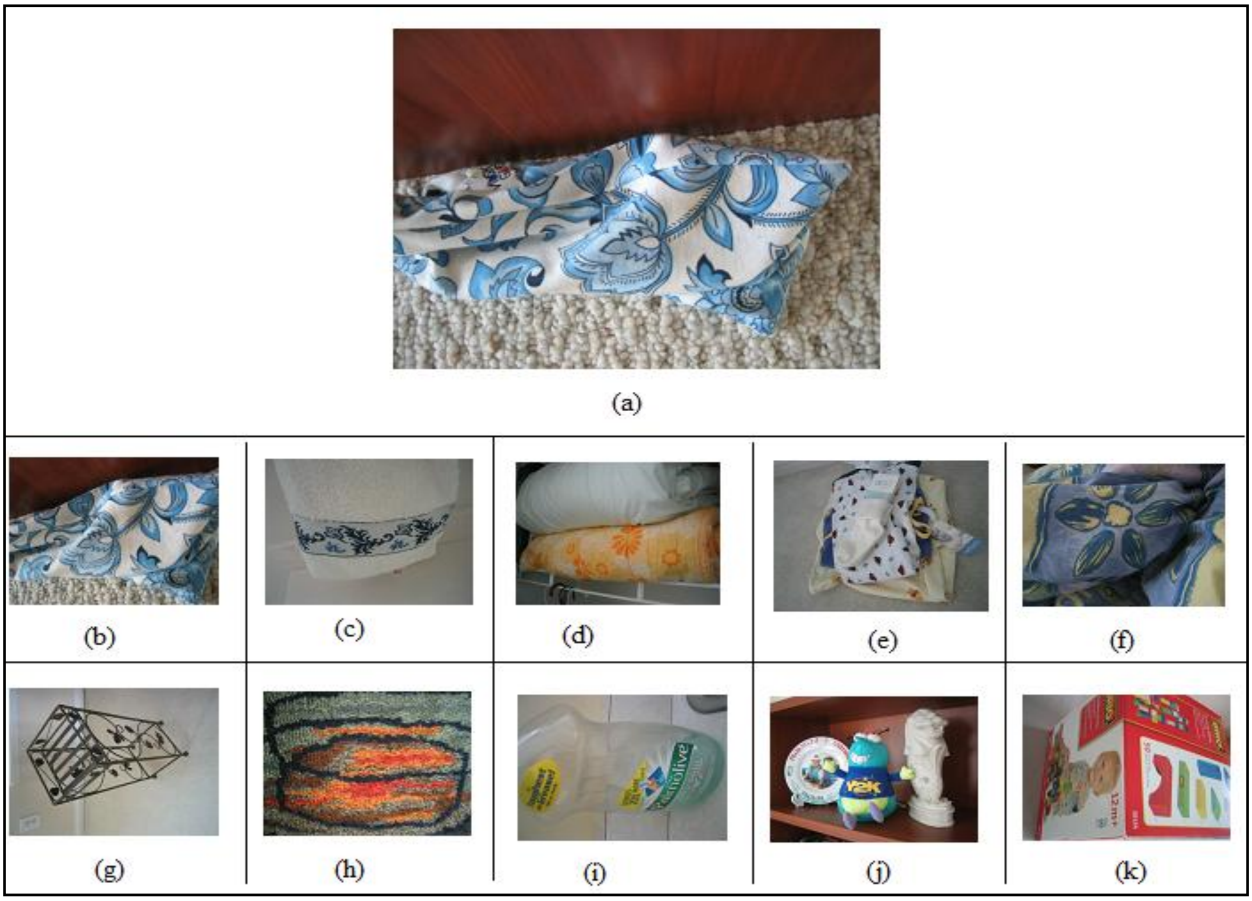

Fig.5: (a) Query Image. (b) – (k) : Retrieval Results for DWT method

Fig.5 (a) is the query image and 5 (b) is the correct matched result, however incorrect matches also crop up in the result stage as depicted by fig. 5 (c) to fig. 5(k)

4.2 SURF Method

An arbitrary image which belongs to Class 18 is given as query image to check if the algorithm can correctly identify its match.

Fig. 6: Query Image

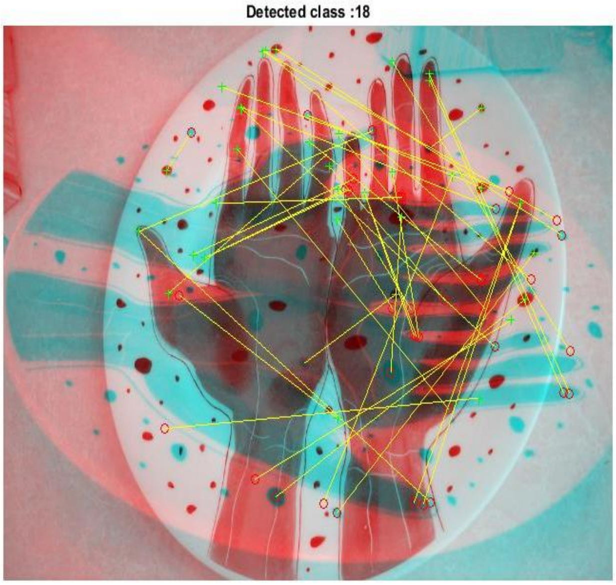

Fig. 7: Feature matching to identify the correct class

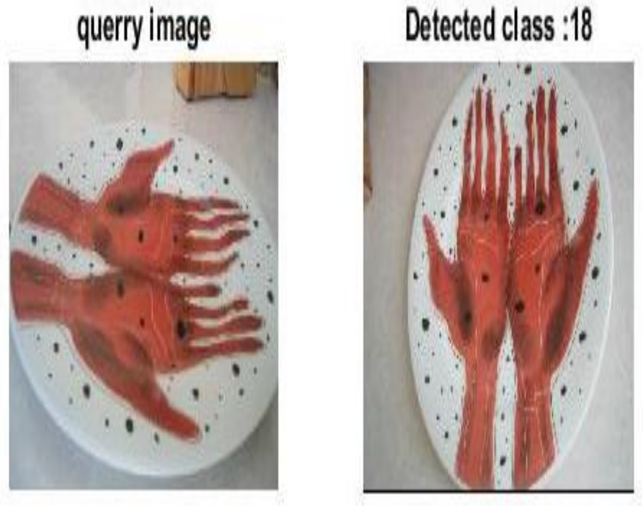

Fig. 8: Matched Results

SURF clearly discriminates among the features as shown by the matched lines in fig. (7) and displays only the correctly matched result. The performance parameters for both the methods are listed below.

Table 1: Performance Comparison of both the Approach

Method	DWT	SURF
Average Time Cost per Query (sec)	0.91	1.187
Mean average Precision (mAP)	0.54	0.73

5. CONCLUSION

The ever increasing size of image datasets has triggered the development of efficient and faster image retrieval systems. Systems using DWT extract color and texture features and rather than relying on textual indexing within an image. These features are invariant to change in scale and rotation. However while working with large datasets, the state of the art retrieval systems use SURF for feature detection and extraction as SURF is computationally faster and has a higher rate of mean average precision. The features detected are robust to variations in scale, illumination, rotation and viewpoint.

REFERENCES

[1] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Tsukuba, pp. 2681-2684, Nov. 2012

[2] Dibyendu Mukherjee, QM Jonathan Wu, and Guanghui Wang, "A comparative experimental study of image feature detectors and descriptors." in *Machine Vision and Applications*, vol. 26, no. 4, pp. 443-466, May. 2015.

[3] Jialu Liu, "Image retrieval based on bag-of-words model," in *arXiv preprint arXiv:1304.5168* , Apr. 2013.

[4] M. Emre Celebi, Hassan A. Kingravi, and Patricio A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm." in *Expert systems with applications*, vol. 40, no. 1, pp. 200-210, 2013.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, vol.2, pp. 1150-1157, 1999.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features." in *Computer vision–ECCV 2006*, pp. 404-417, 2006.

[7] Ammar Huneiti, and Maisa Daoud, "Content-based image retrieval using SOM and DWT." in *Journal of Software Engineering and Applications*, vol.8, no. 2, pp. 51, 2015.

[8] Ekta Gupta and R. S. Kushwah, "Combination of global and local features using DWT with SVM for CBIR," in *4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Noida, pp. 1-6, 2015.

[9] S. Agarwal, A. K. Verma and P. Singh, "Content Based Image Retrieval using Discrete Wavelet Transform and Edge Histogram Descriptor," in *2013 International Conference on Information Systems and Computer Networks*, Mathura, pp. 19-23, 2013.

[10] Hua-Zhang Wang, Xiao-Hai He and Wen-Jiao Zai, "Texture image retrieval using dual-tree complex wavelet transform," in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, Beijing, pp. 230-234, 2007.

[11] Ahmed Naser Hussein, S. Mashohor and M. Iqbal Saripan, "A texture-based approach for content based image retrieval system for plant leaves images," in *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, Penang, pp. 11-14, 2011.

[12] Bhuvana Shanmugam, Radhakrishnan Rathinavel, Tamije Perumal, and Subhakala Subbaiyan, "An Efficient Perceptual of CBIR System using MIL-SVM Classification and SURF Feature Extraction," in *International Arab Journal of Information Technology (IAJIT)* ,vol.14, no. 4, Jul. 2017.

[13] Kai Chen, and Jean Hennebert, "Content-based image retrieval with LIRe and SURF on a smartphone-based product image database," in *Mexican Conference on Pattern Recognition*, pp. 231-240. Springer, Cham, 2014.

[14] N. Gopal and R. S. Bhooshan, "Content Based Image Retrieval using enhanced SURF," in *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, Patna, pp. 1-4, 2015.

[15] Paul S Addison, “The Continuous Wavelet Transform” in *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance*, 2nd ed. CRC press, 2017, Edinburgh, Scotland.

[16] B. Siva Kumar, and S. Nagaraj, "Discrete and stationary wavelet decomposition for image Resolution Enhancement," in *International Journal of Engineering Trends and Technology (IJETT)*, vol.4, no.7, 2013.

[17] S. Agarwal, A. K. Verma and N. Dixit, "Content Based Image Retrieval using Color Edge Detection and Discrete Wavelet Transform," in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Ghaziabad, pp. 368-372, 2014.

[18] Dipalee Gupta and Siddhartha Choubey, "Discrete wavelet transform for image processing." in *International Journal of Emerging Technology and Advanced Engineering*, vol.4, no. 3, pp. 598-602, 2015.

[19] A. Gebejes, and R. Huertas, "Texture characterization based on grey-level co-occurrence matrix." in *Proceedings in Conference of Informatics and Management Sciences*, vol.2, no. 1, 2013.

[20] Azriel Rosenfeld, “Image Pyramid and their Uses” in *Multiresolution image processing and analysis*, 1st ed. vol. 12, Springer Science & Business Media, 2013.

[21] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol.2, pp. 2161-2168.