Volume 3, No 2; April 2015

Advances in Image And Video Processing

ISSN: 2054-7412

Society for Science and Education - United Kingdom

TABLE OF CONTENTS

DISCLAIMER

Society for Science and Education United Kingdom
AIVP Advances in Image and Video Processing
Volume 3, Issue 2
ISSN 2054-7412

Medical Image Segmentation Based on Edge Detection Techniques

[1]Bnar M. Ghafour, [2]Nassir H. Salman and [3]Gullanar M. Hadi
[1,3]Department of Software Engineering – Salahaddin University, Erbil, Iraq
[2]Department of Computer Science, Cihan Univ,Erbil, Iraq
bnar_e@yahoo.com; nahu64@yahoo.com; gullanarm@yahoo.com

ABSTRACT

In this article a new combination of image segmentation techniques including K-means clustering, watershed transform, region merging and growing algorithm was proposed to segment computed tomography(CT) and magnetic resonance(MR) medical images.

The first stage in the proposed system is "preprocessing" for required image enhancement, cropped, and convert the images into .mat or png ...etc image file formats then the image will be segmented using combination methods (clustering , region growing, and watershed, thresholding). Some initial over-segmentation appears due to the high sensitivity of the watershed algorithm to the gradient image intensity variations. Here, K- means and region growing with correct thresholding value are used to overcome that over segmentations. In our system the number of pixels of segmented area is calculated which is very important for medical image analysis for diseases or medicine effects on affected area of human body also displaying the edge map.

The results show that using clustering method output to region growing as input image, gives accurate and very good results compare with watershed technique which depends on gradient of input image, the mean and the threshold values which are chosen manually. Also the results show that the manual selection of the threshold value for the watershed is not as good as automatically selecting, where data misses may be happen.

Keywords: Medical image segmentation, K-means Clustering, Watershed, Region growing and merging, CT and MR images

1 Introduction

Medical imaging is an essential part of modern healthcare. Medical imaging technologists take X-rays, mammograms, ultrasounds and computed tomography images to help diagnose patients' injuries and diseases. on another hand, medical imaging (diagnostic radiography) is the practice of taking medical images of patients' internal body parts. A range of technologies are used, including radiography, fluoroscopy, angiography, computed tomography, mammography, ultrasound, magnetic resonance imaging and nuclear medicine.

Medical imaging professionals are a vital part of a healthcare team as the images they collect are usually used to confirm or exclude a medical diagnosis, to advise on the treatment of illness, monitor patient progress, or provide medical screening information for doctors and other medical specialists. Finally and as a conclusion; Medical imaging is the technique, process and art of creating visual representations of the interior of a body for clinical analysis and medical intervention. Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and

DOI: 10.14738/aivp.32.1006
Publication Date: 25th March, 2015
URL: http://dx.doi.org/10.14738/aivp.32.1006

treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities. Although imaging of removed organs and tissues can be performed for medical reasons, such procedures are usually considered part of pathology instead of medical imaging.

2 Segmentation of medical images

Segmentation is the process dividing an image into regions with similar properties such as gray level, color, texture, brightness, and contrast.[1] and [2-4]. The role of segmentation is to subdivide the objects in an image; in case of medical image segmentation the aim is to:

- Study anatomical structure, Diagnosis
- Identify Region of Interest i.e. locate tumor, lesion and other abnormalities
- Measure tissue volume to measure growth of tumor (also decrease in size of tumor with treatment)
- Help in treatment planning prior to radiation therapy; in radiation dose calculation
- Surgery planning

3 Methods available for MR image segmentation:

MR imaging is specifically used in brain imaging and thus lot of research work has been done particularly in the areas of MR brain image segmentation [5-8]. The main goal in brain MR segmentation is to segment gray matter, white matter and cerebrospinal fluid. Segmentation is also used to find out the regions corresponding to lesions tumors, cyst, edema, and other pathologies and for this mostly T1- weighted images are used.

Most of the segmentation methods available for CT and MR images segmentation are intensity based i.e. gray level based hence; the segmentation results are affected by (1) intensity in-homogeneities and (2) partial volume effects. Accordingly, different researchers have proposed methods for correction of these problems. For more details see [1].

3.1 Region growing method

Region growing is a bottom-up procedure that starts with a set of seed pixels. The aim is to grow a uniform connected region from each seed. Segmentation by growing a region from seed point using intensity mean measure. The region is iteratively grown by comparing all unallocated neighbouring pixels to the region. The difference between a pixel's intensity value and the region's mean, is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the region. This process stops when the intensity difference between region mean and new pixel becomes larger than a certain threshold. Starting from the seed the intensity values of each pixel is compared with its neighbors and if it is within the threshold, it'll be marked as one.A pixel is added to a region if and only if:

- It has not been assigned to any other region.
- It is a neighbor of that region.
- The new region created by addition of the pixel is still uniform.

*[9]*the region growing method we used after modify is proposed by Danial kellner Mathwork :Recursive region growing algorithm for 2D/3D grayscale images with polygon and binary mask output (2011).*

3.2 Limitations of region growing

Note that a complete segmentation of an image must satisfy a number of criteria:

- All pixels must be assigned to regions
- Each pixel must belong to a single region only
- Each region must be a connected set of pixels
- Each region must be uniform
- Any merged pair of adjacent regions must be non-uniform

Region growing satisfies the third and fourth of these criteria, but not the others. It fails to satisfy the first and second criteria because, in general, the number of seeds defined by the user will not be sufficient to create a region for every pixel. The fifth criterion is not satisfied because the regions grown from two nearby seeds are always regarded as distinct, even if those seeds are defined in a part of the image that should be segmented as a single region.

3.3 Theory of Watershed Method

Any grayscale image can be viewed as a topographic surface where high intensity denotes peaks and hills while low intensity denotes valleys. You start filling every isolated valleys (local minima) with different colored water (labels). As the water rises, depending on the peaks (gradients) nearby, water from different valleys, obviously with different colors will start to merge. To avoid that, you build barriers in the locations where water merges. You continue the work of filling water and building barriers until all the peaks are under water. Then the barriers you created gives you the segmentation result applying the watershed transform method was explained well in the results section. We used watershed transform proposed by [10]

Figure (1) shows the diagram of the processing steps of the medical images segmentation proposed system.

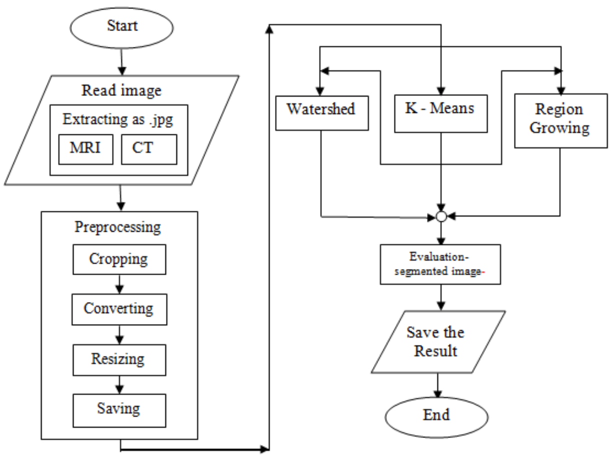


Figure 1: shows the diagram of the proposed medical images segmentation system.

4 Results of Tested Examples

Example 1- Spine image

After using K-means to divide the spine image into 4 regions see figure (2) , we have got four values represent 4 regions as in the figure below(maximum value 216 and minimum is 32) the display range is 0-255.

Figure 2 (a) origin image

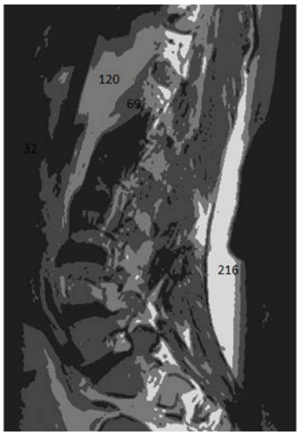
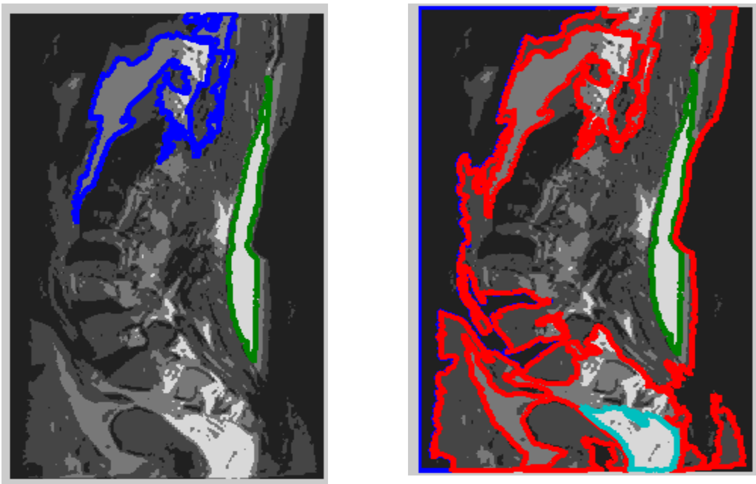


Figure 2 (b)
segmented image into 4 regions using clustering Technique (see their intensities(32, 69,120,216)

By choosing a seed point in the input image ,then using region growing method to find the region boundary and the number of pixel represents this region within the threshold range depend upon the image .number of pixels is very useful to calculate the area of the effected region in the patient image before and after diagnostic by the hospital. See figure (3)



(a) (b)

Figure 3: a) chose two regions seed and segmented(blue and green boundaries)
b) segmented image into 4 regions(red, blue, green light blue edges)

Example-2 Knee image

Using the region growing algorithm to choose the medical image (MRI, or CT, Jpeg or .mat formats) that we want to segment. inter the region number, then click somewhere inside the medical image(initial position), the region growing algorithm (regionGrowing1) was called as below, where Bn is the input image, and the threshold value is 2500(using .mat image file format here)

Poly = regionGrowing1 (Bn, [], 2500);

The output as follows, see figure (4), where the algorithm determines the initial position of seed point then calculate number of pixel represent the segmented region.

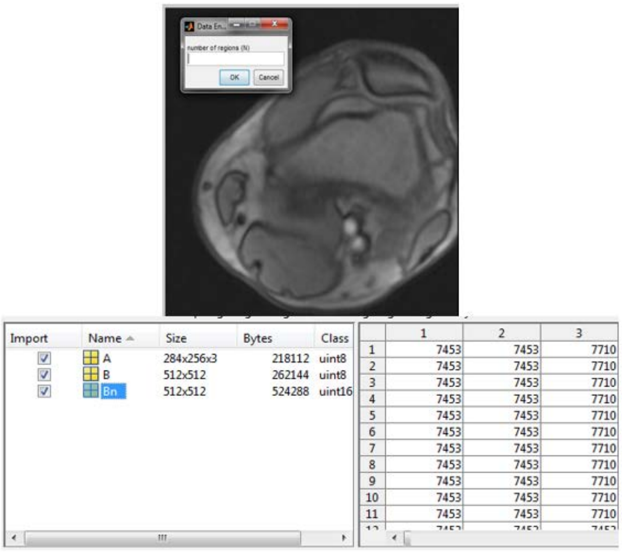


Figure 4: show the input knee image with its information as a .mat file

The segmentation result as follow, see figure (5).

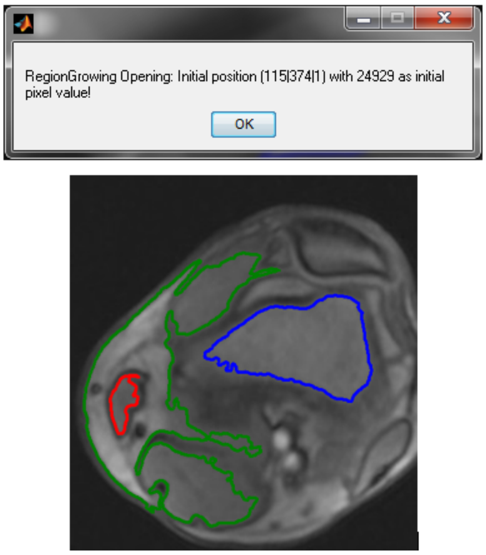


Figure 5: shows three segmented regions of origin image in figure(4)

We choose just three regions here, also there are many message box to show you our result in details. for 4 segmented region see the following figure(6).

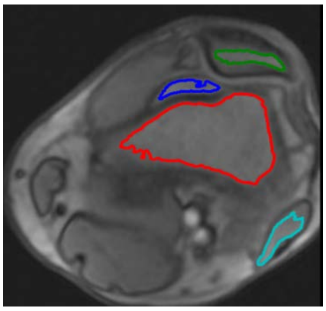


Figure 6: shows 4 segmented regions of figure(4)

Another test for another choosing region in the same image as below.

From these results we can calculate:

- area in pixel for effected region interested region
- after take medicine and take new image ,we can calculate the affected area again and compare it with first one, to see the effect of medicine or if no difference we can say the medicine is not correct.
- we can evaluate how many days the patient can be good health
- we can color the regions with their edge

here we use .mat format with uint16 format for fast processing using Matlab

Example 3-Abdomen image

Another test image, 5 regions are chosen just for example to segment abdomen image as in figure(7).

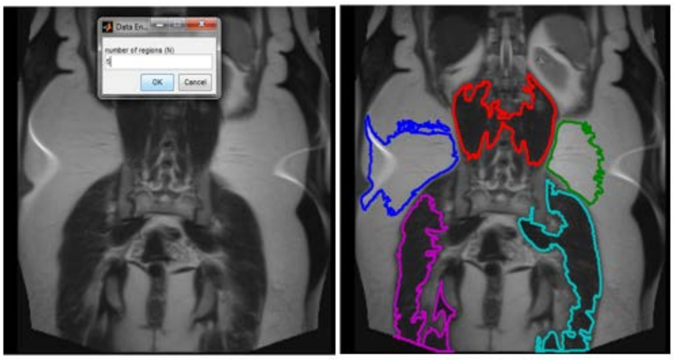


Figure 7: abdomen image segmentation into 5 chosen regions

4.1 CT images sample using .mat and .jpg files formats

- For .mat file format of input medical image, we used uint16 class, where the display range is 0-65535, it is easy to get pixel information (location + values). we used gray image, its name 002284c9.jpg from Rezqari hospital -Erbil-Iraq, it is color image converted into gray , resize into 512 x 512 , save it on D:\ Bnrgray.mat to easy manipulate using Matlab

- For .jpg image file format

  we read the original color image , then convert it into gray , also resize it into 512 x 512 , and finally write it on D:\ as .jpg image format

Example 4 Knee image with watershed transform

In this example watershed method was used for medical image, the result is oversegmented image see the figure (8) below.

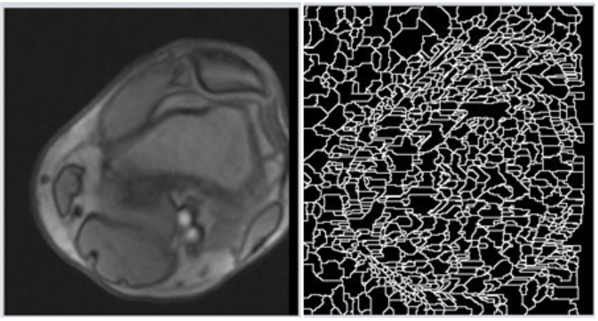
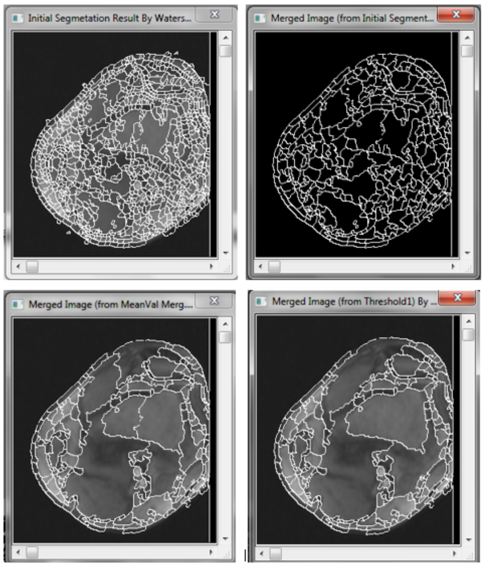


Figure 8: show the over-segmentation results using watershed algorithm

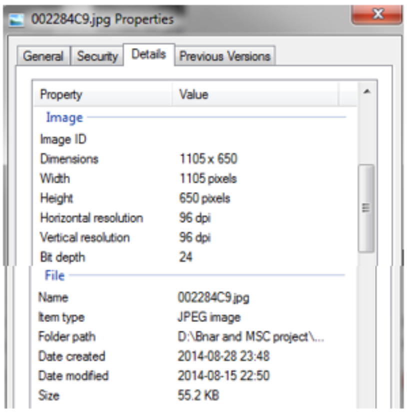
The watershed transform is executed on the gradient image. This implemented by defining the first derivative of an image (measuring the change of gray levels) using 3 x 3 gradient operator in x and y directions, where the aim of the watershed transform is to search for regions of high intensity gradients (watersheds), the result is initial segmentation. With this method the marker image was used to calculate the number of pixels of each region then calculate region mean intensity value. In the merging step (on the initial segmentation image) if the difference of means value of the two regions is less than our test threshold, the two regions are merged. also we used two threshold values T1, T2 , and the edge strength value compare with these values as follows: , if it is for example the edge strength less than T1 , the two adjacent regions are merged and the region becomes large(growing), and the same for T2 .

From above we can say shortly: 1st we get merging image using mean values (initial segmentation) , then using T1 on the mean value merging image to get new merge image. Finally the output again used with T2 to get the final merging region (final segmented image). The results are shown in the figure (9).



Figure(9) watershed algorithm result using mean value and two thresholding values for merging

We found that threshold 1 and threshold2 values between 0-5 (i.e in our case above T1=T2=4) give us good results, this because the nature of the image got it from Rezgari hospital, see the figure below. i.e we do simple processing on image 002284C9.jpg and so on for all other medical images. The properties of the medical image is shown in the figure below.



5 Conclusions

Applying our automated proposed segmentation system on MRI and CT images give the following conclusions:

- It is easy to read and process the .mat file and .jpg or other file formats for medical images (i.e., CT, MRI).
- The program determines the number of pixels of segmented area which is very important for medical image analysis for diseases or medicine effects for affected area of human body.
- The affected area (i.e.; region of the interest) with their edge can be colored and segmented very well with correct edge positions of each region.
- Using K-means clustering method output to region growing as input, gives accurate and very good results compare with watershed technique which depends on gradient of input image, the mean and the threshold values.

- The manual selection of the threshold value for the watershed is not as good as automatically selecting, data misses may be happen
- Images can be taken in a period; before and after receiving a medicine; so after taking medicine and taking new image, the affected area can be calculated again and compare with the first one, to see the effect of medicine or if no difference appears it can be concluded the medicine did not match the disease, so the medicine is not correct.
- Evaluate during how many days the patients can acquire their health.

REFERENCES

[1] Neeraj Sharma and Lalit M. Aggarwal,.2010:Automated medical image segmentation techniques, Journal of Medical Physics / Association of Medical Physicists of India. Jan-Mar 2010; 35(1)3-14.

[2] Gonzalez RC, Woods RE. Digital image processing. 2nd ed. 2004. Pearson Education.

[3] Pratt KW. Digital image processing. 3rd ed. Willey; 2001. pp. 551–87.

[4] Pal NR, Pal SH. A review on image segmentation techniques. Pattern Recog. 1993;26:1277–94.

[5] Song T, Gasparovicc C, Andreasen N. A hybrid tissue segmentation approach for brain MR images. Med Biol Eng Comput. 2006;44:242–9.

[6] Liao L, Lin T, Li B. MRI brain image segmentation and bias field correction based on fast spatially constrained kernel clustering approach. Pattern Recog Lett. 2008;29:1580–8.

[7] Kuo WF, Lin CY, Yung-Nien Sun YN. Brain MR images segmentation using statistical ratio: Mapping between watershed and competitive Hopfield clustering network algorithms. Comput Met Prog Biomed.2008;9:191–8.

[8] Cuadra MB, Craene MD, Duay V. Dense deformation field estimation for Atlas-based segmentation of pathological MR brain images. Comput Met Prog Biomed. 2006;84:67–75.

[9] Danial kellner :Recursive region growing algorithm for 2D/3D grayscale images with polygon and binary mask output (2011). Mathwork.

[10] L. Vincent, P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991 , 13(6): 583-593.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

AIVP ADVANCES IN IMAGE AND VIDEO PROCESSING

VOLUME 3, ISSUE 2
ISSN 2054-7412

Efficient Modern Description Methods by Using SURF Algorithm for Recognition of Plant Species

[1]Masoud Fathi Kazerouni, [2]Jens Schlemper and [3]Klaus-Dieter Kuhnert
Institute of Real-Time Learning Systems, University of Siegen, Siegen, Germany
[1]masoud.fathi@uni-siegen.de; [2]schlemper@fb12.uni-siegen.de; [3]kuhnert@fb12.uni-siegen.de;

ABSTRACT

Plants are one of one of most valuable natural resources. There would be no life on earth without plants. They unlike humans and animals manufacture their own food by photosynthesis. In every food chain, the plants occupy the first position and lead the chain as source of food. Environment and climate are largely interlinked with plants. Rainfall, humidity and temperature are influenced by presence of plants. Cutting down plants also imbalance the environment which will indirectly affect human life. Even the economic importance of plants is also quite large to mankind. Plants are great contributors of economy. Many countries rely on agriculture as one of the main source of revenue. Other benefits of plants are significant applications in different fields. Medical and agricultural applications are just some instances of plants application.

Plant recognition can be done by using unique characteristic parts of plants. The used part is leaf. Shapes of leaves are useful to do plant recognition and find the species. Bag of words (BoW) and support vector machine (SVM) methods are applied to recognize and identify plants species. Visual contents of images are used and four steps are performed: (i) image preprocessing, (ii) BoW, (iii) train, (v) test. Three combined methods are used on Flavia dataset. The proposed approach is done by Speed-up robust features (SURF) method and two combined method, HARRIS-SURF and features from accelerated segment test-SURF (FAST-SURF). The accuracy of SURF method is higher than other applied methods. It is 92.28395 %. In addition to visional comparison, some quantitative results are measured and compared.

Keywords: Text Component; SURF; combined methods; HARRIS-SURF; FAST-SURF; feature extraction; feature detection; plant recognition.

1 Introduction

Over 3,000 million years ago, the first living-organism which resembled a plant appeared. It was a blue-green algae which lived in the sea and can still be found in the water today. When the plants made their first appearance on Planet Earth, the atmosphere was not appropriate for all oxygen breathing creatures. The air was made out of carbon dioxide, a gas which to us is deadly. Then photosynthetic plants came along and slowly over several million years, cleaned the atmosphere and filled it with oxygen. Ever since early man rubbed two sticks together to make fire, plants have played a vital role in the history of mankind. Over time, utilization and application of plants are increased in different fields. For a few decades at the beginning of the 20th century, growing and learning from a garden of medicinal plants was part of the pharmacy curriculum [1]. A sharply increasing interest in the commercial growth of medicinal plants led to more attention to plants.

DOI: 10.14738/aivp.32.1151
Publication Date: 28th April, 2015
URL: http://dx.doi.org/10.14738/aivp.32.1151

Medicinal application of plants are still increasing and playing a fundamental role in this field. Researchers are trying to promote profitable and sustainable agrifood, fibre and horticultural industries, developing new plant products and improving natural resource management. Also recent developments of agriculture have changed traditional agriculture and added new applications and techniques to increase agricultural production and protect environment. Application and existence of plant are undeniable in human life and environment. Hence, recognition and distinguish of plant species are increased.

Determination of plant species is a challenge for non-botanists. Recognition process may take a long time and be grueling. For instance, traditional farmers do not have enough knowledge of different plant species that may grow between their products. Existence of unwanted plants like weed, which is a plant in the wrong place, can damage products of a farm. Due to plant industry and increase of new plant products, necessity and need of a plant recognition system are increased. In context of biological applications, health of plants is very important during a research and accurate and fast recognition of plant disease are significant. Accurate identification of a cultivated plant can be very helpful in knowing how it grows (e.g., size shape, texture, etc.) as well as how to care and protect it from pests and diseases. Also, a plant recognition system would let non-professionals to obtain scientific knowledge of botanists. These are only some aspects of plant recognition system's need.

Since last decade, plant recognition has become a popular topic in computer vision and image processing. Design and implantation of a plant recognition system are urgent needs. An efficient study is vital to achieve this goal. The characteristics of plants give us clues as to how some plants are similar to and different from others. Classification of plant species depends upon common and unique features that are used to identify plants by their characteristics. Scientists and plant experts have collected data on numerous plant species from studying the plants in their natural habitats and recording information about their characteristics in scientific literature and databases for future reference. Although plants can be identified by the shape and color of their fruits, flowers, stems, color of stems, and the seeds they contain, but leaves are the most useful characteristics for recognition and identification of plants. Plants leaves contain important amount of information. Leaf features include the color, texture, shape, size and orientation of the plant's leaves on the stem, end etc. Some features like texture and color may vary with different environmental conditions and time. So they are not useful because of their variations. Shape of leaves is the best feature of plants to do classification and identification of various species. By using digital images of leaves, it would be easy to collect and use them for further process. Information of leaves will be used for performing the methods.

Plant recognition was done by using leaf shapes-a case study of the Acer family in [2]. A polygon approximation was used to classify different species of Acer family. In [3], simple shape features as centroid-contour distance (CCD) curve, eccentricity and angle code histogram (ACH) were applied to recognize plants of 140 Chinese medical plants. Douglas-Peucker approximation algorithm was used to extract leaf shape features in [4]. After that, basic geometric features and digital morphological features were used in [5]. In [6], snakes technique with cellular neural networks (CNN) was employed to do recognition. Also, shape context has been used for doing leaf classification [7], [8]. A widely used keypoint based detector and descriptor is SURF [9]. Hessian-matrix approximation of integral image is used by SURF. In "[10]", a comparative study of SURF and Scale-invariant feature transform (SIFT) [11] has been performed.

Bag of features [12] representation is very popular for content based image classification due to good performance and simplicity. SIFT and SURF can be used in image categorization. Generally, better performance and efficiency of training and classification depend on better representation and clustering of features.

Bag of Words (BoW) model [13, 14, 15, 16, 17] is originated from natural language processing tasks and information retrieval. For image analysis, a visual analogue of a word is used in the BoW model, which is based on the vector quantization process by clustering low-level visual features of local regions or points, such as color, texture, and so forth [18]. Representation of it uses image patches as visual words. One important benefit of using BoW model is increase of accuracy in classification as a result of building a large dictionary.

In [19], a plant recognition algorithm is proposed which is implemented with three different modern description methods. Now, the purpose is to have a better feature extraction that leads to higher accuracy of the algorithm.

The first requirement of designing an automatic plant classification is to find and select a suitable dataset. The used dataset is Flavia dataset [20]. It is comprised of 32 different plant species. The dataset is divided into two datasets, training dataset and test dataset. The training dataset consists of leaf images of training dataset for in [19]. Also the test dataset is built with the same images of test dataset in [19]. All proposed methods are tested by the dataset.

The used methods are originated from computer vision and machine learning fields. To achieve the goal, advanced and modern methodologies are investigated to design and implement the desired plant recognition system with high efficiency and accuracy.

Using useful and advanced methods and techniques of pattern recognition, image processing, and machine learning lead to implementation of an automatic plant recognition system with acceptable accuracy. Identification and recognition of different plant species can be used in different fields of science and industry.

The present paper explains an automatic recognition system which is implemented by modern description methods and algorithms. The approach includes four phases: Image pre-processing, Feature detection and extraction, Train, and Test.

This paper is organized as follows: Section II deals with general overview of the used architecture for plant classification and related works, Section III introduces the applied algorithm with explanation on detection and extraction of feature and the schemes of classification, Section IV describes the results and the experiments conducted using the proposed methods, and Section V presents the conclusion and next steps of research.

2 General Review

Shapes of plant leaves can help botanists and biologist to identify and recognize various plants species as they have an important amount of information. Because of large number of plants species all around the world, an automatic plant recognition system is helpful to classify them. Advancement and progress of computer vision can help to develop an accurate and reliable system for this purpose.

An important part of a recognition system is to handle information. Leaves of plants have significant information. It is essential to extract information for recognition systems. Developing an efficient and fast system needs to find an effective method for extracting information of leaves helps.

In image classification, an image is classified according to its visual content. The principle basis is contents of the image. Classification is based on the similarity of the image contents. The image contents are described via image features. Visual contents are needed for object recognition and identification. Detecting keypoints with rich information should be done as a basis part. It can be done automatically by using different detection methods and represented by descriptors [21]. Then, keypoints are grouped into a number of clusters with those with similar descriptors assigned into the same cluster. They would be shown by one single visual word. So, all keypoints will be mapped to a limited number of visual words.

A popular and widely used method for feature representation is Bag-of-Words (BoW). It is used for document representation in information retrieval. This methodology was first proposed in the text retrieval domain problem for text document analysis [22], and it was further adapted for computer vision concepts and applications [22]. For image analysis, a visual analogue of a word is used in the model, which is based on the vector quantization process by clustering low-level visual features of local regions or points, such as color, texture, and so forth.

Detection of local interest points and features has an important role in different image processing concepts and fields. It is also the first step of BoW model. Automatic detection of features is performed to detect meaningful points in images. They are quite unique to the objects of images. Due to this process, any object will be detected based on its own features in each image. Several well-known region detectors have been described in the literature [23, 24].

The next step of the model is to compute feature descriptors for the detected features. It is a main part of the procedure. In general, feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which over fits the training sample and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. In this specific purpose, the importance of it is not deniable. Extracting keypoints and descriptors is done by using SURF algorithm. In [19], SIFT algorithm was used for combined methods. Now SURF algorithm is used and its influence on results is investigated. SURF creates 64-dimensional vectors or 128-dimensional vectors when SIFT creates only 128-dimensional vectors. The order of difference vectors is of no significance. The implemented SURF is used by 64-dimensional vectors.

After detection and extraction of features, the next step is based on vector quantization. For this task, K-means clustering is performed and the number of visual words generated is based on the number of clusters. Clustering algorithm is used to find the centers of clusters of the feature descriptors and visual vocabulary is the collection of the visual words. For each feature descriptor in an image the nearest visual word from the vocabulary is assigned to it. The distribution of visual words in the image is represented as histogram.

After the BoW feature is extracted from images, it is entered into a classifier for training or testing. SVM [27] is used to train the classifier. Using a training dataset is necessary in this step. After generating a classifier, the test process is done. In this step, a test dataset is needed too. Experiments are done with different combined methods and the results are obtained. The number of classes is 32.

Figure 1 shows a general scheme of plant recognition system.

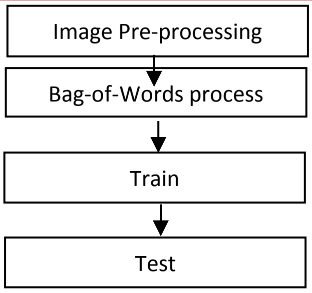


Figure 1: General scheme of plant recognition system.

3 Leaf Recognition Approach

3.1 Image Pre-processing

The leaves images of dataset should be converted to gray scale images to be used as input images. Conversion RGB images to gray scale images is the first step to do preprocessing and the algorithm.

The following equation, Equation (1), is the used formula to convert RGB pixel values to gray scale values.

$$Gray\ scale = 0.299.R + 0.587.G + 0.114.B \quad (1)$$

Where R, G, and B correspond to the color of the pixel.

3.2 Bag of Words

Local feature extraction involves interest point detection and computation of descriptors in region surrounding those interest points.

The current step is to perform feature detection. Features can be detected manually or, preferably, it can be detected automatically using some specific techniques. Feature must be prominent, easily detectable and spread over the whole image. Feature detection method should have good localization accuracy and should not be sensitive to the assumed image degradation. The used method should be able to detect features regardless of image deformation. Transformation such as scale, and rotation should be detectable.

Each feature detection method has its own characteristics. For instance, Harris [25] method is a rotation-invariant method for feature detection. When rotation is occurred, finding the same corners is its important characteristic. It is one of the used methods. SURF method is a speed-up version of SIFT method and is rotation and scale invariant. In SIFT, Lowe approximated Laplacian of Gaussian (LoG) with Difference of Gaussian (DoG) for finding scale-space. SURF goes a little further and approximates LoG with Box Filter. One big advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images. And it can be done in parallel for different scales. Also SURF method relies on determinant of Hessian matrix for both scale and location. As a robust local feature detection method, it works much faster than SIFT method. For feature detection, another used method is FAST (Features from Accelerated Segment Test) [26] method which was proposed by Edward Rosten and Tom Drummond in their paper “Machine learning for high-speed corner detection” in 2006 (later revised it in 2010). FAST method is a weapon choice to do detection faster. Its utilization is undeniable in real time systems. This method finds a lot of features and it is one of important characteristics of it. Generally, the goal of a descriptor is to provide a unique and robust description of an image feature, e.g. by describing the intensity distribution of the pixels within the neighborhood of the point of interest. Most descriptors are computed thus in a local manner; hence, a description is obtained for every point of interest identified previously. The SURF descriptor is based on the similar properties of SIFT, with a complexity stripped down even further. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. The second step is constructing as square region aligned to the selected orientation, and extracting the SURF descriptor from it. A descriptor vector is computed for every keypoint. The dimension of the descriptor is 64 and it is less than SIFT method with the dimension of 128. Although it seems to be high, lower descriptors than it do not perform the task as well as it. Also computational cost is another aspect of the process. Gained descriptors should be rich enough to be usable at the category level. SURF method has lower dimension, higher speed of computation and provides better distinctiveness of features. Obtained descriptors will be used to find similarity between different images.

Sets of keypoint descriptors are used to represent images. Then, vector quantization (VQ) technique is used to cluster the keypoint descriptors in their feature space into a large number of clusters. K-means clustering is applied and each keypoint is encoded. K-means algorithm is used to assign of points to their closest cluster centers and computation of the cluster centers. Each cluster is a visual word and represents a special pattern by the keypoints in the cluster. Now a visual word vocabulary is generated. It contains local patterns in images. After mapping the keypoints to visual words, each image can be represented as a bag of visual words. Thus, a novel image's features can be translated into words by determining which visual word they are nearest to in the feature space (based on the Euclidean distance between the cluster centers and the input descriptor). As a point, vocabulary size should be large enough to recognize changes in image parts. Increase of size may lead to distinguish irrelevant variations and quantization artifacts.

3.3 Classifier Train

After generation of vocabulary, each image is represented by a histogram of how often the local features are assigned to each visual word. The representation is known as bag-of-visual-words in analogy with the bag-of-words (BOW) text representation where the frequency, but not the position, of words is used to represent text documents. Applying a classifier is the next step of the approach. The classifier is typically a support vector machine (SVM) [27] which is it often known to produce state-of-the-art results in high dimensional problems. The output of BoW method is used to do classification and train a SVM.

In general, SVMs are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. SVM performs classification tasks by constructing hyper-planes (support vectors) in a multidimensional space that separates cases of different class labels. It supports both estimating the relationships between variables (regression) and data classification and can handle multiple continuous and categorical variables. Also, SVM benefits from two good ideas: maximizing the margin and the kernel trick. These good ideas can guarantee high testing accuracy of classifiers and overcome the problem about curse of dimensionality. In other words, SVM constructs hyper-planes either in input space or in feature space from a set of labeled training dataset. The hyper-plane will try to split the positive samples from the negative samples. The linear separator is commonly constructed with maximum distance from the hyper-plane to the closest negative and positive samples. Intuitively, this causes correct classification for training data which is near, but not equal to the testing data.

Consider a binary classification problem with N training samples (data). Each sample is indicated by a tuple (xi, yi) and (i = 1, 2, …, N), where xi=(xi1, xi2, …, xin) corresponds to the attribute set for the ith sample. Conventionally let yi ε {-1, 1} and it is considered as its class label. The decision boundary of a linear classifier can be written as follows:

$$w^T x + b = 0 \tag{2}$$

Where w is weight vector and b is a bias term.

The SVM implementation goal is to define a decision boundary that is maximally far away from any data point. For training process, SVM needs an input matrix and labels each samples as either belonging to a given class (positive) or not (negative), and then treats each sample in the matrix as a row in an input space or high dimensional feature space, where the number of attributes identifies the dimensionality of the space. Determination of the best hyper-plane to separate each positive and negative training sample is one of important steps. The obtained and trained SVM will be used to perform predictions of the test dataset and find the labels.

SVM is an efficient algorithm for classification due to high accuracy, nice theoretical guarantees regarding overfitting, and with an appropriate kernel. There are several types of SVM. The type of used SVM is υ-Support Vector N-class classification. Kernel type of SVM is Radial basis function (RBF), because support vector machines employing the kernel trick do not scale well to large numbers of training samples or large numbers of features in the input space, several approximations to the RBF kernel (and similar kernels) have been devised. Like other types of SVM, it has some parameters. Instead of C parameter, υ parameter is used, which is in the range 0..1. When it increases, the decision boundary becomes smoother. The training set is divided into kFold subsets. KFolds is cross-validation parameter and the SVM algorithm is executed kFold times. This parameter equals 10 for the approach. Automatic training is done by choosing optimal SVM parameters such as gamma, p, nu, coef0, degree from parameters. When the cross-validation estimate of the test set error is minimal, parameters ate optimal.

The final step of classification is testing. Prediction of different existed leaves in testing dataset is performed and the results of the combined methods are acquired.

4 Experiments

32 different plant species of Flavia dataset are applied to do experiments. Each method is tested with the following machine and the accuracy of them is evaluated. The machine is Intel® Core™ i7-4790K, CPU @ 4.00 GHz, and installed memory (RAM) 16.0 GB. Investigation of the experiments is done on the dataset by SVM classifier with RBF kernel. Also, test dataset comprised of 648 images. The procedure of experiments is illustrated in below.

SURF, HARRIS-SURF, and FAST-SURF are proposed as three methods to obtain results on the dataset and evaluate accuracy of tests.

Table 1 shows the accuracy result of three performed methods. SURF method has the highest accuracy between the used methods and the methods that are used in [19]. FAST-SURF method has a higher accuracy when it is compared to SIFT method in [19].

Table 1: Accuracy of classification.

Method	Accuracy of Classification
SURF	92.28395
FAST-SURF	89.6605
HARRIS-SURF	87.1914

As it is shown, the maximum accuracy belongs to SURF method. SURF is a speed-up version of SIFT. It is invariant with regard to scale, orientation, and illumination. In SIFT, Lowe approximated Laplacian of Gaussian (LoG) with Difference of Gaussian (DoG) for finding scale-space. SURF goes a little further and approximates LoG with Box Filter. One important advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images. And it can be done in parallel for different scales. Also the SURF relies on determinant of Hessian matrix for both scale and location.

To assign orientation, SURF uses wavelet responses in horizontal and vertical direction for a neighborhood of size 6s. Adequate Gaussian weights are also applied to it. Then, they are plotted in a space. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. Interesting part is that, wavelet response can be found out using integral images very easily at any scale.

For feature description, SURF uses Wavelet responses in horizontal and vertical direction (again, use of integral images makes things easier). A neighborhood of size 20sX20s is taken around the keypoint where s is the size. It is divided into 4x4 sub-regions. For each sub-region, horizontal and vertical wavelet responses are taken and a vector is formed. The represented vector gives SURF feature descriptor with total 64 dimensions. Lower the dimension, higher the speed of computation and matching, but provide better distinctiveness of features.

For more distinctiveness, SURF feature descriptor has an extended 128 dimension version. The sums of dx and |dx| are computed separately for dy < 0 and dy > 0. Similarly, the sums of dy and |dy| are split up according to the sign of dx, thereby doubling the number of features. It doesn't add much computation complexity.

Using sign of Laplacian (trace of Hessian Matrix) for underlying interest point is one of important improvement. It adds no computation cost since it is already computed during detection. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. In the matching stage, features comparison is only possible if they have the same type of contrast. This minimal information allows for faster matching, without reducing the descriptor's performance. Totally, SURF adds a lot of features to improve the speed in every step.

Corner detection can be done by Harris detector. Harris points are preferable when looking for exact corners or when precise localization is required. It basically finds difference in intensity for a displacement of (u, v) in all directions. This method is not reliable, because detected points of the method do not have the required level of invariance for image matching. Although the performance of the method is less than other methods, it has been used widely for different computer vision applications.

To have a faster detection, FAST detector is applied. It is based on the accelerated segment test (AST), which is a modification of the SUSAN [28] corner detector. FAST is not robust to the presence of noise and high level noise. Beside this disadvantage, it is many times faster than other existing corner detectors. Also, it has high levels of repeatability under large aspect changes and for different kinds of features.

The second experiment is calculation of number of keypoints for four different species. These species are selected from the dataset. Complexity of the species is specified by human vision. The labels are named Simple, approximately simple, approximately complicated, complicated. The number of keypoints is calculated for SURF and FAST-SURF methods (Table 2).

Table 2: Number of keypoints.

Number of keypoints in method	Simple leaf	Approximately simple leaf	Approximately complicated leaf	Complicated leaf
SURF	161	47	350	829
FAST-SURF	528	322	1165	6180

Keypoints are represented for one leaf by SURF and FAST-SURF methods. The images are shown in Figure 2.

(a)

(b)

Figure 2: (a) Representation of key points for SURF method. (b) Representation of key points for FAST-SURF method.

The FAST method finds thousands of keypoints, while other used methods find only hundreds. In above, it is mentioned that detection with FAST method generates some noise keypoints. Large number of keypoints, key points mixed up with noisy keypoints, cause decrement of accuracy. Detected keypoints of SURF are enough and also accurate to have a good result.

The needed time for performing any method is computed and shown in Table 3. The needed time per image is calculated. FAST-SURF method has the minimum required time and it is the fastest one. In comparison with SIFT method [19], SURF method needs less time for the same dataset. The needed time for HARRIS-SURF method is lower than HARRIS-SIFT method in [19].

Table 3: Test time per image.

Method	Needed test time per image (ms)
SURF	445.268
FAST-SURF	345.512
HARRIS-SURF	528.756

By using SURF method, faster computation is obtained without sacrificing performance. In short, SURF adds a lot of features to improve the speed in every step. Analysis shows it is faster than SIFT while performance is comparable to SIFT. SURF is good at handling images with blurring and rotation, but not good at handling viewpoint change and illumination change.

Each detected point has a little information. Descriptors use their relationship to have enough information and a useful model. Fast-SURF method finds more keypoints than HARRIS-SURF method. Thus, more information is obtained which helps to have a better performance. Although FAST algorithm detects some noisy keypoints, increase of the number of keypoints leads to perform better. FAST-SURF method has repeatability and this characteristic has a good influence on its performance. Due to the characteristics of HARRIS and FAST algorithm, it is anticipated to obtain better results when FAST-SURF method is used.

The detected keypoints affect the results of plant recognition. In SURF method, number of keypoints is not too high or low. It is completely enough to extract information and have an accurate method with acceptable results.

SVM parameters can be changed to investigate their effects on the methods. Two SVM parameters are chosen to consider their variation effects on final error of the methods. The selected parameters are Nu and Gamma parameters.

Nu parameter and C parameter are equivalent regarding their classification power, but Nu parameter has a more meaningful interpretation. This is because Nu presents an upper bound on the fraction of training samples which are errors (badly predicted) and a lower bound on the fraction of samples which are support vectors. Nu parameter has a value between 0 and 1. Nu parameter is changed, while Gamma parameter is held fixed and equals 1.0. When Nu parameter increases, error of methods is increased. It is shown in Figure 3. Increment of Nu parameter has the least influence on SURF method in comparison to other methods. Robustness of SURF method against Nu parameter variations could be a helpful characteristic.

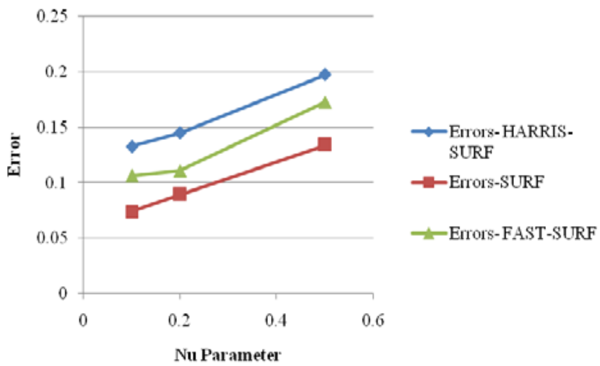


Figure 3: Variation of Nu parameter for used methods.

The next investigated parameter is Gamma. Gamma parameter variations are performed to consider the effects on final error of each method. G*amma* parameter defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close. For this experiment, Nu parameter is kept fixed at 0.1. Increase of Gamma parameter causes error’s increment. It is shown is Figure 4. The diagrams are ascending, but the values of slopes are not high. The minimum slope value belongs to SURF method that indicates the robustness of the method in this experiment. In general, increase of Gamma parameter has less impact than increase of Nu parameter.

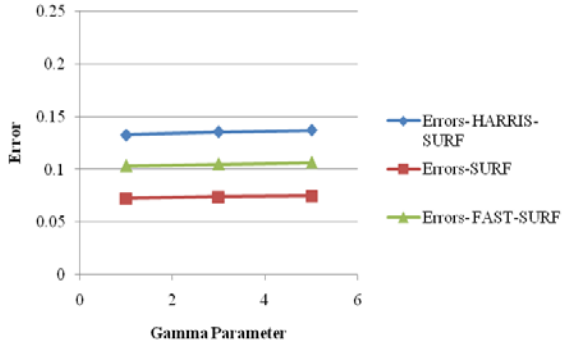


Figure 4: Variation of Gamma parameter for used methods.

Visualization of the performance of methods is performed by construction of confusion matrix. A confusion matrix contains information about actual and predicted classifications done by classification methods. It is one $n \times n$ matrix. 32 different plant species are used for the methods. Thus, n equals 32.

Precision and recall values are calculated for each label of the used methods by using confusion matrix information. Generally, precision is a measure of accuracy provided that a specific class has been predicted. Recall is a measure of the ability of a prediction model to select instances of a certain class from a dataset. The following equations, equation 3 and equation 4, are used to obtain precision and recall values from confusion matrixes.

$$Precision_i = \frac{M_{ii}}{\sum_j M_{ji}} \tag{3}$$

$$Recall_i = \frac{M_{ii}}{\sum_j M_{ij}} \tag{4}$$

Where i is i^{th} row of confusion matrix and j is the j^{th} column of it.

In both Figure 5 and Figure 6, HARRIS-SURF method has the minimum values. It is predictable as it has the least accuracy percentage between three methods. Variation of SURF method values is less than other methods in both figures. Precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned. The SURF method has the best performance due to the figures. FAST-SURF method has the second rank between these three methods. Its performance is in the middle. Due to the measurements, the sequence is SURF, FAST-SURF, and HARRIS-SURF.

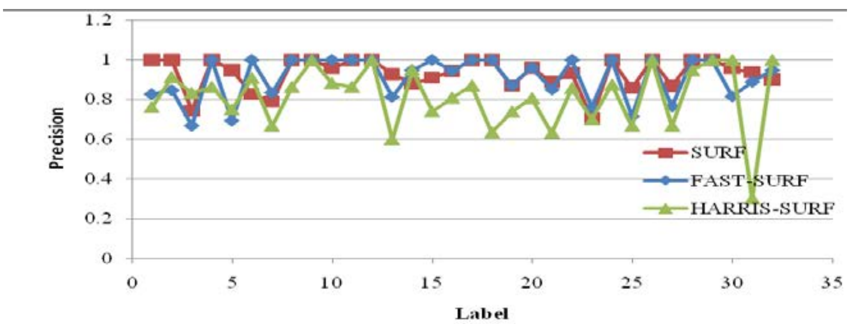


Figure 5: Precision measurement for SURF, FAST-SURF, HARRIS-SURF methods.

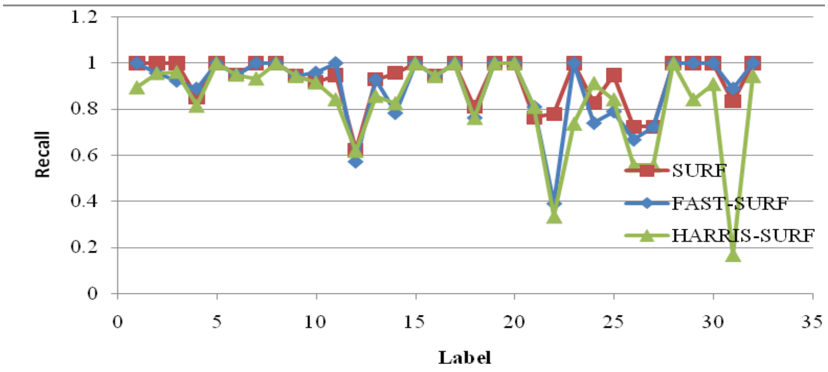


Figure 6: Recall measurement for SURF, FAST-SURF, HARRIS-SURF methods.

Another concept to compare the methods is surrounded area. Higher area under the curve represents both higher precision and higher recall. A lower false positive rate relates to high precision and a lower false negative rate relates to high recall. Due to these concepts in Figure 5 and Figure 6, SURF method has better performance of the method because higher areas belong to it. Both high scores show that the method is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

The relationship between recall and precision is shown for each method.

For SURF method, the minimum value of recall is 0.619048 and the minimum value of precision is more than 0.703704. Due to experiment, variation of values in SURF method is less than the SIFT method used in [19]. In Figure 7, these variations are shown.

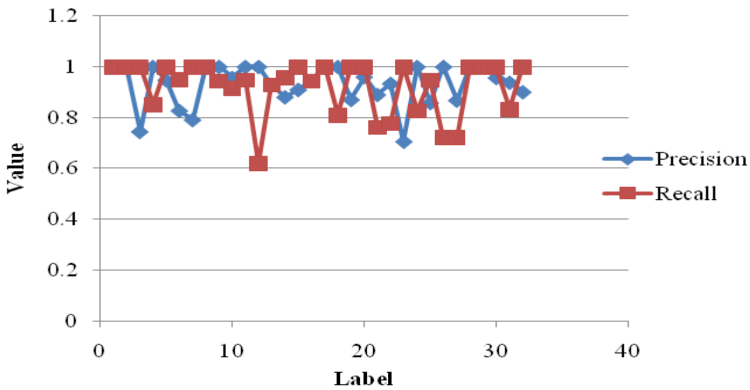


Figure 7: Precision and recall values for SURF method.

Also, more labels have values in [0.9, 1], which is the highest possible interval.

In Figure 8, precision and recall values are shown for FAST-SURF method. In this method, the minimum values of precision and recall are the same and equal to 0.666667. In comparison to SURF method, there are more variations in obtained results of precision and recall. After investigation of minimum value of recall, it is found that the value is less than 0.4 and equals 0.3333. In comparison to the used FAST-SIFT method in [19], this combined method has better performance.

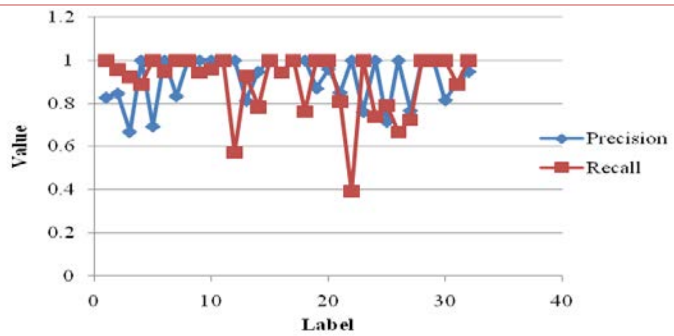


Figure 8: Precision and recall values for FAST-SURF method.

The Figure 9 shows precision and recall of HARRIS-SURF method. The minimum value of precision is 0.3 while the minimum value of recall is 0.166667. Difference between maximum and minimum is high in this method for both precision and recall values. Larger intervals are covered with HARRIS-SURF method for obtained precision and recall values. The reason is completely obvious, because the accuracy of this method is lower than other methods, SURF and FAST-SURF methods.

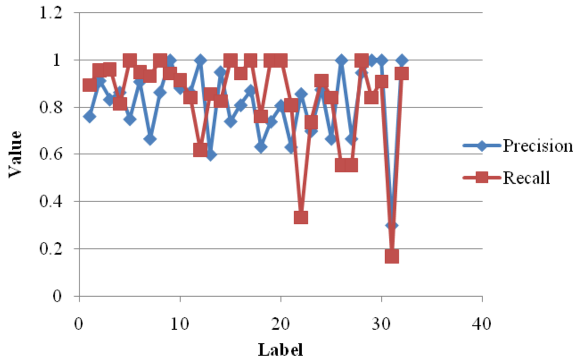


Figure 9: Precision and recall values for HARRIS-SURF method.

5 Conclusion

In this paper, three methods are taken into consideration for plant recognition and identification on Flavia dataset. SURF method and two combined methods, HARRIS-SURF and FAST-SURF, are taken into consideration for plant recognition and classification. Accuracy measurement and efficiency of each method are described. Experimental results are also compared with some quantitative results and discussed according to human vision for four different species. Experiments on the dataset, demonstrate that SURF method has the best performance between proposed methods. In comparison with used methods in [19], SURF and FAST-SURF have better performance and accuracy. Improvement can be done for next step and obtain higher accuracy.

REFERENCES

[1]. Allen I. White, *The History of the Washington State University College of Pharmacy 1891-1991*. 1996. pp. 63-65.

[2]. T. K. C. Im, H. Nishida, *Recognizing plant species by leaf shapes-a case study of the acer family*. Proceedings of IEEE International Conference on Pattern recognition, 1998. pp. 1171.

[3]. Z. Wang, Z. Chi, and D. Feng, *Shape based leaf image retrieval*, in Vision. Image and Signal Processing, IEE Proceedings, 2003. 150(1): pp. 34–43.

[4]. J. Du, D. Huang, X. Wang, and X. Gu, *Computer-aided plant species identification (CAPSI) based on leaf shape matching technique*. Transactions of the Institute of Measurement and Control, 2006. 28 (3): pp. 275-284.

[5]. S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, *A leaf recognition algorithm for plant classification using probabilistic neural network*. IEEE International Symposium on Signal Processing and Information Technology, IEEE explore library, 2007. pp. 11–16.

[6]. Y. Li, Q. Zhu, Y. Cao, and C. Wang, *A leaf vein extraction method based on snakes technique*. Proceedings of IEEE International Conference on Neural Networks and Brain, 2005.

[7]. S. Belongie, J. Malik, and J. Puzicha, *Shape Matching and Object Recognition Using Shape Contexts*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. 24(4): pp. 509–522.

[8]. H. Ling and D. W. Jacobs, *Shape Classification Using the Inner-distance*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. 29(2): pp. 286–299.

[9]. Herbert Bay, Tinne Tuytelaars and Luc Van Gool, *SURF: Speeded Up Robust Features*. Computer Vision – ECCV 2006, Lecture Notes in Computer Science, 2006. 3951: pp. 404-417.

[10]. P. M. Panchal, S. R. Panchal, S. K. Shah, *A Comparison of SIFT and SURF*. International Journal of Innovative Research in Computer and Communication Engineering, 2013. 1(2).

[11]. Lowe, D. G., *Object Recognition from Local Scale-Invariant Features*. Proc. Of the International Conference on Computer Vision, 1999. pp. 1150—1157.

[12]. G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. *Visual categorization with bags of keypoints*, in. ECCV'04 workshop on Statistical Learning in Computer Vision, 2004. pp. 59–74.

[13]. Xin Chen, Xiaohua Hu1, Xiajiong Shen, *Spatial Weighting of Bag-of-Visual-Words*. Proceedings of the 13th pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 2009. pp. 867-874.

[14]. Yi Yang, Shawn Newsam, *Bag-of-visual-words and spatial extensions for land-use classification*. Proceedings of SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010. pp. 270-279.

[15]. Michael Villamizar, Jorge Scandaliaris, Alberto Sanfeliu, and Juan Andrade-Cetto, *Combining Color-based Invariat Gradiet Detector with HoG Descriptors for Robust Image Detection in Scenes under Cast Shadows*. IEEE International Conference on Robotics and Automation, 2009. pp. 1997-2002.

[16]. Jun Yang, Yu-gag Jiang, *Evaluating bag-of-Visual-words representations in scene classification*. Proceedings of the International Workshop on Multimedia Information Retrieval, 2007. pp. 197-206.

[17]. David Picard, Nicolas Thome and Matthieu Cord, *An Efficient System for Combinig Complementary Kernels in Complex Visual Categorization Tasks*. Proceedings of IEEE International Conference of Image Processing (ICIP), 2010. pp. 3877-3880.

[18]. C.-F. Tsai, *Bag-Of-Words Representation in Image Annotation: A Review*. International Scholarly Research Network, 2012. pp. 1-19.

[19]. Masoud Fathi Kazerouni, Jens Schlemper, and Klaus-Dieter Kuhnert, *Comparison of Modern Description Methods for the Recognition of 32 Plant Species*. Signal & Image Processing: An International Journal (SIPIJ), 2015.

[20]. H. Laga, S. Kurtek, A. Srivastava, M. Golzarian, and S. Miklavcic, *Ariemannian elastic metric for shape-based plant leaf classification*. Digital Image Computing: Techniques and Applications, 2012.

[21]. K. Mikolajczyk, and C. Schmid, *Scale and Affine Invariant Interest Point Detectors*. International Journal of Computer Vision, 2004. 60(1): pp. 63-86.

[22]. A. Bosch, X. Munoz, and R. Marti, *Which is the best way to organize/classify images by content?*. Image and Vision Computing, 2007. 25(6): pp. 778–791.

[23]. K. Mikolajczyk, B. Leibe, and B. Schiele, *Local features for object class recognition*. Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05), 2005. pp. 1792–1799.

[24]. T. Tuytelaars and K. Mikolajczyk, *Local invariant feature detectors: a survey*, Foundations and Trends. Computer Graphics and Vision, 2007. 3(3): pp. 177–280.

[25]. C. Harris, M. Stephens, *A Combined Corner and Edge Detector*, in Proceedings of the Fourth Alvey Vision Conference, 1988. pp. 147-151.

[26]. E. Rosten, T. Drummond, *Machine Learning for High-Speed Corner Detection*, European Conference on Computer Vision, 2006. 1: pp. 430–443.

[27]. V. Vapnik, *The natural of statistical learning theory*, Springer-Verleg, New York, USA, 1995

[28]. S. M. Smith and J. M. Brady, *SUSAN — A new approach to low level image processing*, International Journal of Computer Vision, 1997. 23(34): pp. 45–78.

SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
AIVP ADVANCES IN IMAGE AND VIDEO PROCESSING
VOLUME 3, ISSUE 2
ISSN 2054-7412

Analysis and Optimization of Parameters used in Training a Cascade Classifier

[1]Abhishek Kumar Annamraju and [2]Akash Deep Singh
Bits Pilani KK Birla Goa Campus, Goa, India
[1]abhishek4273@gmail.com; [2]akash.singh011235@gmail.com

ABSTRACT

Training a cascade classifier for object detection using Local Binary Pattern (LBP) and Histogram of Gradients (HOG) features is computationally exorbitant. If the parameters of training are not chosen appropriately, the training may take weeks to complete with the output of an inefficient classifier. The state-of-the-art face recognition applications demand accurate and reliable cascade classifiers. Open Computer Vision (OpenCV) organization provides libraries which accomplish the training task once all parameters are given as inputs. In this paper we analyze the parameters experimentally and concluded with an optimal range of values for each of these parameters. Testing of the generated classifiers with optimal parameters values is performed on a dataset of 4000 test images. The training of these classifiers with optimal parameters takes an average training time of 25000 sec and provides average true positive detection of 88%.

Keywords: Cascade classifier, Local Binary Pattern (LBP), Histogram of Gradients (HOG), Object detection, face detection, OpenCV

1 Introduction

Face detection is one of the major topics in computer vision, with a wide range of applications like surveillance, robotics, and Human computer interaction (HCI). Developing a face detection system is still an open problem, but over the past decade several successful methods have been developed.

Detecting the human face is more complicated than other objects because of its dynamic nature with many forms and colors [1]. The tough part of facial recognition is isolating it from the background. Although several algorithms exist to perform face detection, each has its own weaknesses and strengths. However all of the algorithms have a major setback that they are computationally expensive [2]. An image is a collection of the color and/or light intensity values of pixels. Analyzing every pixel of an image for face detection is computationally redundant and difficult to accomplish because of the variability associated with a human face. Pixels often require reanalysis for scaling and precision. Viola and Jones devised the method called Haar Classifiers, for rapid detection of any object, including human faces, using AdaBoost classifier cascades that are based on Haar-wavelet features and not pixels [3].

Training a classifier on the realm of less knowledge regarding parameters of cascade training often leads to increased complexity in the process. This paper helps to understand the effect and importance of every training parameter, while it discovers the optimal values. Training a classifier for face recognition, with not restricting the type of feature to Haar, but with Local Binary Pattern (LBP) and Histogram of Gradients (HOG) and optimized parametric values reduces training time drastically.

DOI: 10.14738/aivp.32.1152
Publication Date: 28th April, 2015
URL: http://dx.doi.org/10.14738/aivp.32.1152

Also this study analyses the testing parameters of every trained cascade classifier to reach to the conclusion of optimized parameters.

The outline of the paper is simple and understandable. Section 2 illustrates the work that has been carried out in the past related to this study while Section 3 defines the various testing and training parameters in detail. The following section, Section 4, gives in depth understanding of the experiments carried out to reach the optimal parametric values. Section 5 lists out the best experimental results of the classifiers trained with the optimal parametric values and Section 6 provides a conclusion to the paper.

2 Related works

Fast and efficient face detection has been of great interest in the field of Computer Vision [4]. The number of approaches to this problem have increased over the last decade [5][6][7] providing reliable methods. However, after the work by Viola and Jones [7] describing a fast multi-stage object classification approach, and the release of an open source implementation in form of Open Computer Vision (OpenCV) [8], an increasing number of applications have emerged, particularly in the field of facial detection.

The notations for the cascade-based face detector of Viola and Jones [3] are explained in subsequent sections. Cascade is a one-branch tree, of which every node is a classifier designed to reject a large portion of the Non-Face sub-windows and pass all of the Face ones. Consequently, most Non-Face sub-windows are rejected quickly before reaching the final node, resulting in very fast face detection performance. The node classifiers are constructed using an algorithm similar to AdaBoost [9]. The algorithm combines an ensemble of weak classifiers to produce a final boosted classifier with very high accuracy. Much of the recent work using the cascade paradigm has tended to focus on improving the underlying boosting algorithm, such as AsymBoost [10][11], FloatBoost [5], GentleBoost [12], and RealBoost [13][14]. There are two major setbacks in Viola-Jones classifier training methods if the parameters are not selected wisely, firstly, the training algorithm goes quite slow and secondly the classifiers developed produce a high false positive detection.

2.1 Our contribution:

Our contribution to the approach of training a cascade classifier using OpenCV is in the form of getting the optimal parameters of the process. Various systematically planned experiments have been carried to understand the importance of single parameter involved in training a cascade classifier using viola-Jones classifier training method. Optimization of parameters results in training an efficient classifier and reduces the training time as compared to training with unknown random values of parameters. In this study, 250 classifiers have been trained to obtain a set of optimal parameter values and more than 50 classifiers have been trained using these obtained parameters to test the reliability and efficiency of the classifiers. The results prove to be in extreme favor of the research, with large reduction in detection of false positives and increased detection of object of interest. The entire training process has been carried out using OpenCV and the system specifications are as follows:

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- Byte Oder: Little Endian
- On-line CPU(s): 16
- Thread(s) per core: 1

- Core(s) per socket: 8
- CPU socket(s): 2
- Vendor ID: GenuineIntel
- CPU family: 6
- CPU MHz: 2701.000
- RAM (MB): 32855848
- Platform: IBM Platform HPC 3.2 (build 7680)
- Linux: Red Hat Enterprise Linux Server release 6.2 (Santiago)
- OpenCV version: OpenCV 2.4.8
- Code language: C++
- System Owner: Bits-Pilani KK Birla Goa Campus, Goa, India

3 Parameter description

The efficiency of Cascade classifier training depends on two set of parameters: training set parameters and testing set parameters. The training approach is based on a single AdaBoost classifier which involves multiple stages. To start training of next stage, a few thresholds are cross-checked in order to train using only the important features. The new stage takes samples of negatives from the lot of negative images available and the new set is selected based on a fitness functions which determines how badly is the previous stage trained. To make sure the process is smooth a basic understanding of the parameters is essential. In this section a brief introduction of the notations involved is provided.

3.1 Basic definitions

Before jumping on to the parameters, a few intrinsic definitions are needed to be mentioned.

3.1.1 Object of interest

This is the part of an image for which the Cascade is trained and detection is carried out. In real-time image processing it can be objects ranging from as complex as a human face, pedestrians, automobiles, to a spherical ball.

3.1.2 Positive images

It is a set of images containing the object of interest cropped out from the background. In this paper we use a private dataset provided by Ayonix Inc., Japan, containing a set of 11716 images. A set of 145 such images is shown in Figure1.

Figure 1: A set of 145 positive images. Each image 20x20 pixels in dimension.

3.1.3 Negative Images

This set of images are void of the object of interest. The size of these images generally has to be greater than those of the positive images. A set of 4 downscaled negative images is shown in Figure2. The negative image dataset has been gathered from various online available datasets[15-18] and is composed of 27694 images. These images are sometimes also referred as background images.

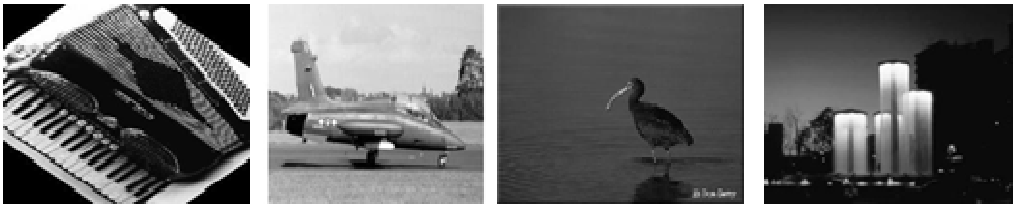


Figure 2: A set of 4 negative images. Each image is 200x150 pixels in dimension.

3.1.4 Samples

A sample is an image in which background is a negative image with positive images embedded at random positions over it. It is represented by a VEC file. The number and dimension of the sample set plays an important role in the training process.

3.1.5 Test image

A test image is an image which contains single or multiple objects of interest with a varied background and is used for testing the effectiveness of the classifier trained.

3.2 Training set parameters

Training set parameters define how a cascaded classifier is to be trained. A set of 11 parameters structure the process of training:

3.2.1 Number of positives used (Npos)

States the number of samples actually used in the training process. This variable can range between a finite non-zero value to a maximum of number of positives actually available, in our case 11716.

3.2.2 Number of negatives used (Nneg)

States the number of negative images actually used in the training process. This variable can range between a finite non-zero value to a maximum of number of negatives actually available, in our case 27694.

3.2.3 Number of stages used (Nstage)

States the number of stages to be involved in the training process of the cascade classifier. It can be as small as 1 to as high as possible. Common conception is that higher the number of stages, better is the classifier trained, but as the number gets beyond 150, training technically gets so computationally expensive it almost comes to a halt.

3.2.4 Feature type (FType)

States which features have to be extracted during the training process, currently Open Computer vision platform serves Viola-Jones Haar features, Histogram of Gradients (HOG) features, Local binary pattern (LBP) features. This study critically examines the parametric analysis of the later two features.

3.2.5 Sample dimensions (Sdimensions)

States the width and the height specifications of the samples being created for the training purpose. Minimum specifications involve a width of 20 pixels and a height of 20 pixels.

3.2.6 Boosted classifier type (Btype)

The type of boosted classifier to be trained, namely Discrete AdaBoost (DAB), Read AdaBoost (RAB), LogitBoost (LB), and Gentle AdaBoost (GAB).

3.2.7 Minimum hit rate(MHRate)

Hit rate is defined as the ratio of the number of objects detected in the test image to that of the total objects present. While training a minimum required hit rate is specified, this also depicts the quality of training. MHRate is sometimes also referred as sensitivity.

3.2.8 Maximum false alarm rate (MFARate)

False alarm rate decides the maximum allowable number of false detections to be present in the training process. MFARate is the parameter to set this value.

3.2.9 Weight trim rate (WTRate)

Weight trimming was proposed by Friedman [19], according to which after every stage of cascading, the samples with the smallest weight are ignored. WTRate indirectly specifies the number of samples to be eliminated at each stage.

3.2.10 Weak count (Wcount)

States the maximum number of weak trees to be involved in every stage of the training.

3.2.11 Memory

The memory being used to store the features and later the classifier vector parameters, in Mb, is managed using this parameter.

3.3 Testing set parameters

Testing set parameters are used in judging the quality of the trained classifier being tested. Test data set is provided by Ayonix Inc., which has 4000 multi-face images. Each image is equipped with data giving information of number of faces present and consecutively the coordinates of eyes of each image in the format "<#faces>; <face1_leyeX>; <face1_leyeY>; <face1_reyeX>; <face1_reyeY>; <face2_leyeX>; <face2_leyeY>; ....". Consider a downscaled Figure3., it has three faces, the data file with have the information "3; 1346.614014; 314.388092; 1412.677490; 309.363983; 326.637787; 433.988586; 391.727936; 421.614502; 853.516846; 528.516846; 945.516846; 527.483154".

Figure 3: Test image with three faces with original resolution 1920x1060

A set of 5 parameters structure the process of testing, and they are as follows:

3.3.1 Training time (TT)

States the time taken for the classifier to get trained in sec. This is totally dependent on the training set parameters.

3.3.2 Detection time (DT)

States the time taken for the trained classifier to detect all the faces in sec/Mega-pixel of image.

3.3.3 False positive ratio (FPR)

False positive is a detection of a non-desired object as a desired one. FPR states the ratio of the number of false positives detected for every test image to that of the total detections for that image.

3.3.4 True positive ratio (TPR)

True positive is a detection of a desired object in a test image. TPR states the ratio of the number of true positives detected for every test image to that of the actual number of object of interest present in that image.

3.3.5 True positive index (TPI)

TPR states the ratio of the number of true positives detected for every test image to that of the total detections for that image.

As shown in Figure4. there are 3 faces (objects of interest) and total number of detections is 17. Any detection (red colored circles) which covers both the eyes of face is counted as a true positive, and the count where detection of same person occurs is not counted. Number of faces detected is 2, thus number of false positives turns out to be 15. Therefore,

FPR = 0.882353, TPR = 0.666667, and TPI = 0.117647.

Figure 4: Image containing both false and true detections.

Note that in any test, where nothing is detected, all the testing parameters are simultaneously set to 0.

4 Training experiments

4.1 Histogram of Gradient Classifier (HOG)

This section includes data regarding the training experiments for HOG classifier which were used to arrive at the best optimal values of the parameters of training.

4.1.1 Variation in Npos

Experiments were carried out to see the effect of the value of Npos on training. Npos was varied from 100 to 11500 in regular intervals of 100, while testing for every 110 classifiers was done on Figure5. and all the testing parameters were calculated. The training parameters are shown in Table.1.

Figure 5: Test image with original dimensions 1920X1060

Table 1: Parameter selection for training with every other parameter fixed except Npos.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	10
Ftype	HOG
Memory	10240
Sdimensions	20x20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	Variable
Nneg	27694

The results of the test are plotted in Figure6. As value of Npos increases, TT decreases. This decrement is at the cost of increased FPR and decreased TPI. On the other hand TPR shows an almost steady graph. Detection time is almost unaffected by the variations in Npos. In a special case where Npos is 100 and other parameters are as specified as in Table.1., the training stops at 8^{th} stage as false alarm rate reaches its maximum value set equal to MFARate. There exists a tradeoff between TT and TPI, and considering that the optimal value can be anything between 8000 and 10000 for a total available 11716 positives, i.e. , Npos must be approximately between 68% and 85% of total positives available. Results with other images also depicted the same.

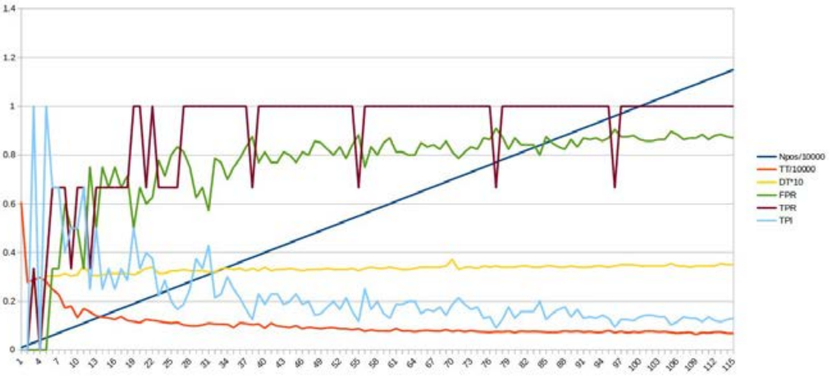


Figure 6: Image representing graphical data of the tests done with varying the value of Npos from 100 to 11500 in regular intervals of 100.

4.1.2 Variation in Nstages

Experiments were carried out to see the effect of the value of Nstages on training. Nstages was varied from 1 to 100 in irregular intervals, while testing for every 57 classifiers was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.2.

Figure7. Image representing graphical data of the tests done with varying the value of Nstages from 1 to 100 in irregular intervals.

Table 2: Parameter selection for training with every other parameter fixed except Nstages.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	Variable
Ftype	HOG
Memory	10240
Sdimensions	20x20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	8000
Nneg	27694

The results of the test are plotted in Figure7. In this experiment, the value of Nstages is increased from 1 to 52 in regular intervals of 1, and after that the next 5 selected values are 60, 70, 80, 90, 100 respectively. As Nstages increases, DT shows a very gradual increment after every step. On the other hand average value of TPI increases and thus average value of FPR decreases. Similar to DT, with the increase in Nstages, the value of TT increases. The limit to the value of Nstages is the computational specifications of the training platform, for a normal computer with Intel Core i5-3120M CPU (@ 2.5GHz * 4 processor speed), the optimal value of Nstages for a decent training is between 40 to 60.

4.1.3 Variation in Nneg

Experiments were carried out to see the effect of the value of Nneg on training. Nneg was varied from 1000 to 27000 in regular intervals of 1000, while testing for every 27 classifiers was done on

Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.3.

The results of the test are plotted in Figure8. With the increase in Nneg, TT shows a steady growth, while DT almost remains constant. Nneg has very less effect on TRI initially but, it shows a gradual increment when Nneg increases. Since rest of the parameters are not optimized yet, the effect of Nneg cannot be witnessed on a large scale. The optimum value of Nneg lies between 25000 and 27000 when 27694 negatives are present. It can be concluded that optimal value of Nneg is between 90% and 98% of the negatives available.

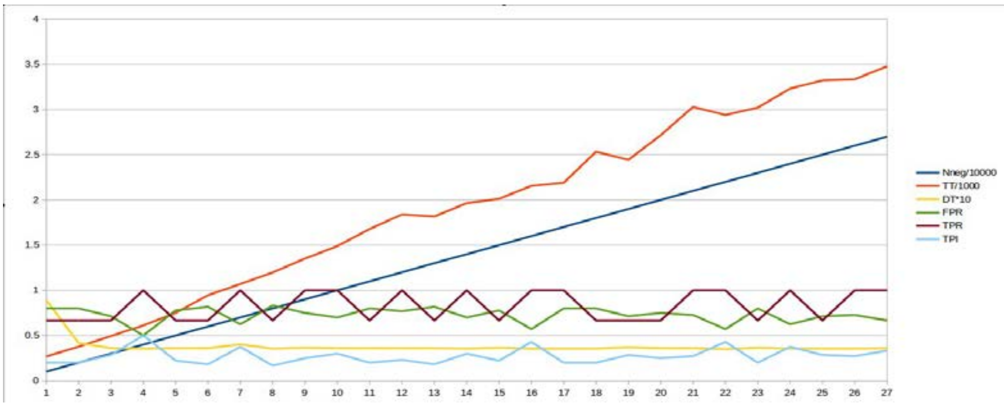


Figure 8: Image representing graphical data of the tests done with varying the value of Nneg from 1000 to 27000 in regular intervals of 1000.

Table 3: Parameter selection for training with every other parameter fixed except Nneg.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	20
Ftype	HOG
Memory	10240
Sdimensions	20x20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	9000
Nneg	Variable

4.1.4 Variation in MFARate

Experiments were carried out to see the effect of the value of MFARate on training. MFARate was varied from 0.01 to 0.99 in irregular intervals, while the testing for every 20 classifiers was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.4.

Table 4: Parameter selection for training with every other parameter fixed except MFARate

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	Variable
Nstages	10
Ftype	HOG
Memory	10240
Sdimensions	20x20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	9000
Nneg	27000

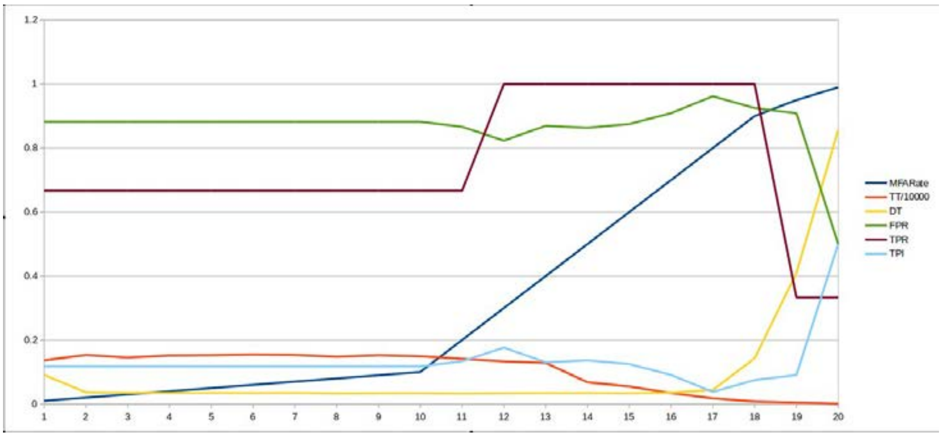


Figure 9: Image representing graphical data of the tests done with varying the value of MFARate from 0.01 to 0.99 in irregular intervals.

The results of the test are plotted in Figure9. In the experiment, the value of MFARate is initially increased from 0.01 to 0.1 in regular intervals of 0.01, later from 0.1 to 0.9 in regular intervals of 0.1, and last two classifiers have MFARate 0.95 and 0.99 respectively, thus training a total of 20 classifiers. TPR shows a sharp increase when MFARate is in the range 0.2 to 0.3 and stays high till MFARate is between 0.3 and 0.9. TT almost remains constant, but decreases gradually till MFARate has value between 0.4 and 0.9. Unexpectedly, TT falls steeply and DT rises for values 0.95, and 0.99 as the training stops at stage 8(false alarm reached). TPI and FPR almost remain constant but FPR shows a deep trough when the value of MFARate lies between 0.3 and 0.5. The optimal value of MFARate is between 0.3 and 0.5, but for higher number of stages, lower value of MFARate is preferred.

4.1.5 Variation in MHRate

Experiments were carried out to see the effect of the value of MHRate on training. MHRate was varied from 0.1 to 0.999 in irregular intervals, while the testing for every 11 classifier was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.5.

Table 5: Parameter selection for training with every other parameter fixed except MHRate.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	Variable
MFARate	0.5
Nstages	10
Ftype	HOG
Memory	10240
Sdimensions	20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	1000
Nneg	27694

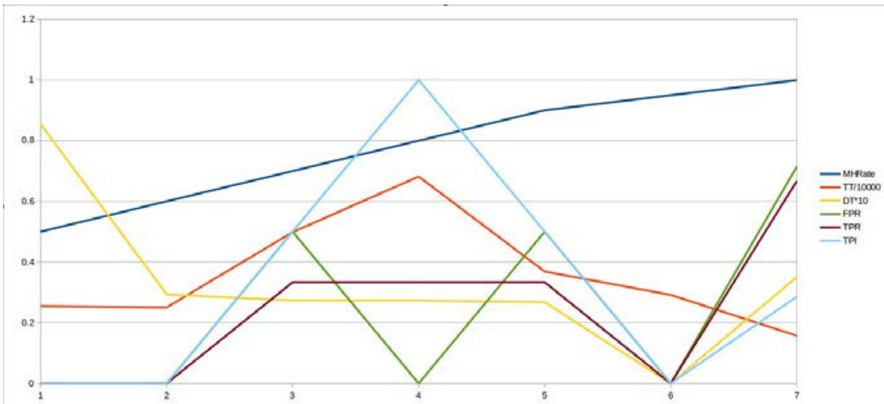


Figure 10: Image representing graphical data of the tests done with varying the value of MHRate from 0.01 to 0.99 in irregular intervals.

The results of the test are plotted in Figure10. In this experiment, 11 classifiers were trained with different values oh MHRate. First 9 classifiers had MHRate equally spaced between 0.1 and 0.9 and the last two had vales equal to 0.95 and 0.999. First four classifiers with MHRate between 0.1 and 0.4 fail to train as the later stages fail to get new samples. With lesser value of MHRate, more samples are wasted for every stage, leaving no samples for stages greater than 5 or 6. TT increases till MHRate reaches a value of 0.8 from 0.5 and then for every increment in MHRate, TT decreases. Till MHRate equal to 0.9, the classifiers trained reach the maximum allowable false alarm rate at stage 5. Only successfully trained classifiers are with MHRate value 0.95 and 0.999 and higher the MHRate lesser is the TT and greater is the TPR. Optimal value of MHRate lies between 0.98 and 0.999.

4.1.6 Varying Sdimensions

Experiments were carried out to see the effect of the value of Sdimensions on training. Sdimensions was varied from 10x10 to 40x40 in regular intervals with the interval spacing 5x5, while the testing for every 7 classifier was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.6.

Table 6: Parameter selection for training with every other parameter fixed except SDimensions

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.999
Nstages	10
Ftype	HOG
Memory	10240
Sdimensions	Variable
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	9000
Nneg	27694

The results of the test are plotted in Figure11. Greater the dimension of the samples created, higher is the requirements for efficient computational platform. In this experiment efforts were made to train 7 classifiers with varied dimensions as specified earlier. With Sdimension values 10x10 and 15x15, the classifier cannot be trained. Increase in Sdimension values leads to increase in TT and the classifier with Sdimension 40x40 stuck at stage 8. TPI shows gradual increment till Sdimension value reaches 30x30, and then starts increasing with a greater positive slope at a cost of training time and efficient platform requirements. Thus a decent Sdimension value lies between 20x20 and 35x35. This value may vary with dimensionality of object of interest, but it will lie in the specified range.

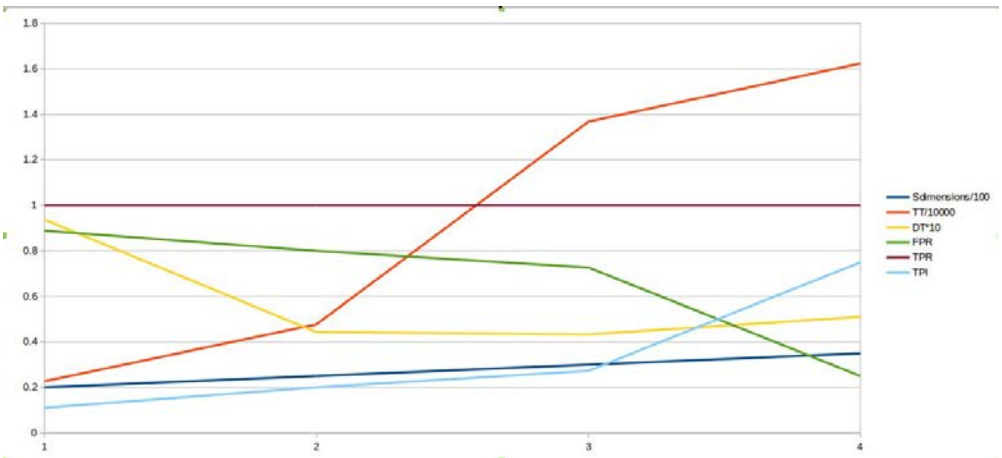


Figure 11: Image representing graphical data of the tests done with varying the value of SDimensions from 10x10 to 40x40 in regular intervals with the interval spacing 5x5.

4.1.7 Variation in Memory

Experiments were carried out to see the effect of the value of Memory on training. Memory was varied from 256 MB to 51200 MB in irregular intervals, while the testing for every 16 classifier was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.7.

Table 7: Parameter selection for training with every other parameter fixed except Memory

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	5
Ftype	HOG
Memory	Variable
Sdimensions	20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	9000
Nneg	27694

The results of the test are plotted in Figure12. It can be observed that increase in Memory has no persistent effect on TPR, FPR, and TPI. O the other hand, average value of TT tends to decrease with increase in Memory, the effects become clearer when the value of Nstages lies between 20 and 30. Thus the optimal Memory specification is to use 90% of the total RAM memory of the system for the training process.

4.1.8 Variation in Wcount

Experiments were carried out to see the effect of the value of Wcount on training. Wcount was varied from 10 to 150 in regular intervals of 10, while the testing for every 15 classifier was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.8.

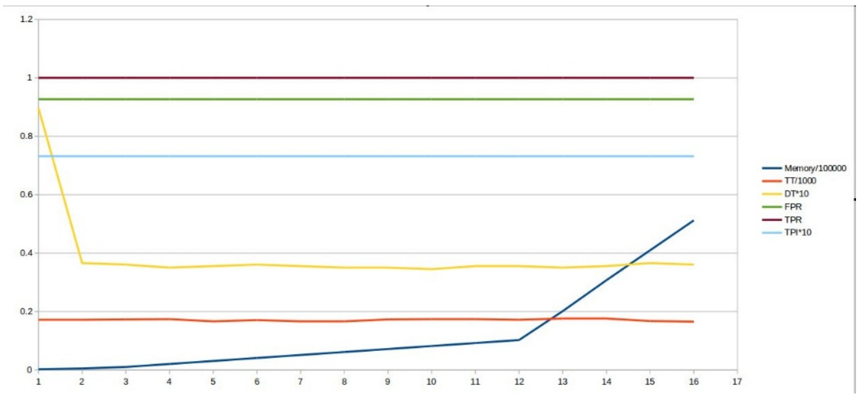


Figure 12: Image representing graphical data of the tests done with varying the value of Memory from 256 MB to 51200 MB in irregular intervals.

Table 8: Parameter selection for training with every other parameter fixed except Wcount.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	10
Ftype	HOG
Memory	10240
Sdimensions	20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	Variable
Npos	9000
Nneg	27000

The results of the test are plotted in Figure13. As the value of Wcount increases, the value of TT increases steadily. The benefit of increase in Wcount is that FPR decreases and TPI increases. But as it can be observed in Figure13, when Wcount crosses the value 130, TPR reduces. DT tends to decrease till the value of Wcount lies between 10 and 70, but it is not affected once Wcount has crossed the value 70. The optimal value of Wcount lies between 80 and 120.

4.1.9 Variation in WTRate

Experiments were carried out to see the effect of the value of WTRate on training. WTRate was varied from 0.05 to 0.99 in irregular intervals, while the testing for every 12 classifier was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.9.

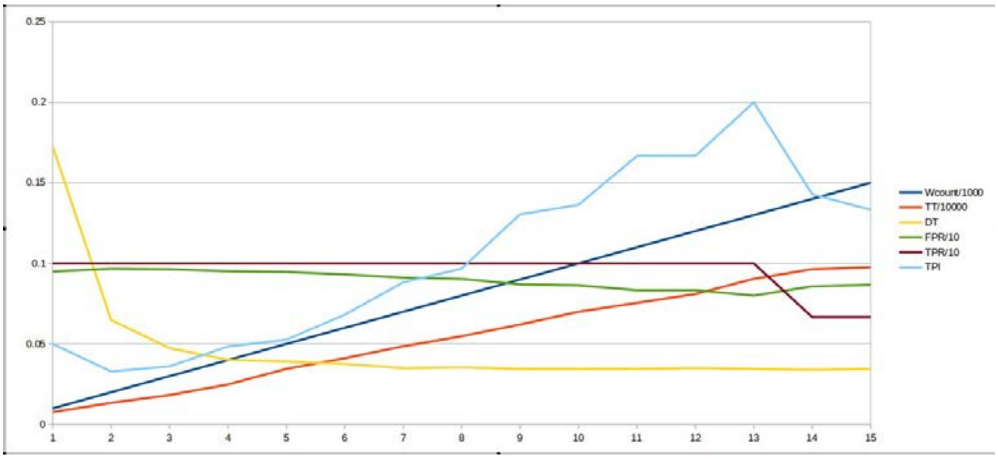


Figure 13: Image representing graphical data of the tests done with varying the value of Wcount from 10 to 150 in regular intervals of 10.

Table 9: Parameter selection for training with every other parameter fixed except WTRate.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	10
Ftype	HOG
Memory	10240
Sdimensions	20
Stage type	Boost
Btype	GAB
WTRate	Variable
Wcount	100
Npos	9000
Nneg	27000

The results of the test are plotted in Figure14. With the increase in WTRate, TT initially remains constant, but increases after WTRate crosses the value 0.4. It appears that FPR increases after this point, but before that point TPR is very low, this states that the number of detections is very low till WTRate lies below 0.4. DT falls steeply once WTRate crosses the value 0.5. Once WTRate reaches a value of 0.9, the DT continues to fall, FRP decreases, TPI increases and TPR continues to stay at a value of 1. Thus, the optimal value of WTRate lies between 0.9 and 0.99.

4.1.10 Variation in Btype

Experiments were carried out to see the effect of the value of Btype on training. Four classifiers were trained with Btypes GAB, DAB, RAB, and LB respectively, while the testing for every 4 classifier was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.10.

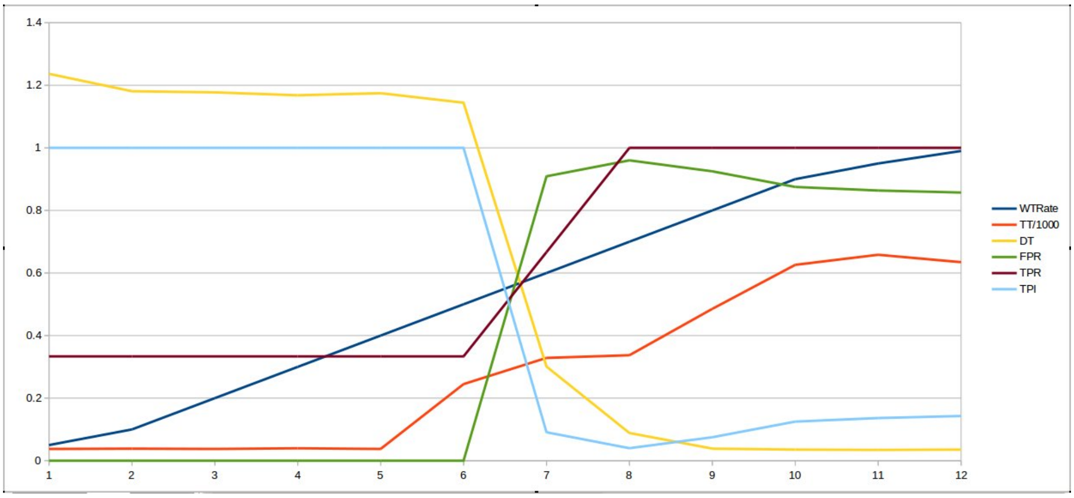


Figure 14: Image representing graphical data of the tests done with varying the value of Wcount from 0.05 to 0.99 in irregular intervals.

Table 10: Parameter selection for training with every other parameter fixed except Btype.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	10
Ftype	HOG
Memory	20480
Sdimensions	20
Stage type	Boost
Btype	Variable
WTRate	0.95
Wcount	100
Npos	9000
Nneg	27000

The results of the test are plotted in Figure15. Btype GAB classifier results in highest DT. TPR remains constant for all the 4 cases. RAB results in highest TPI but at the cost of highest TT. It can be clearly concluded that best Btype is RAB.

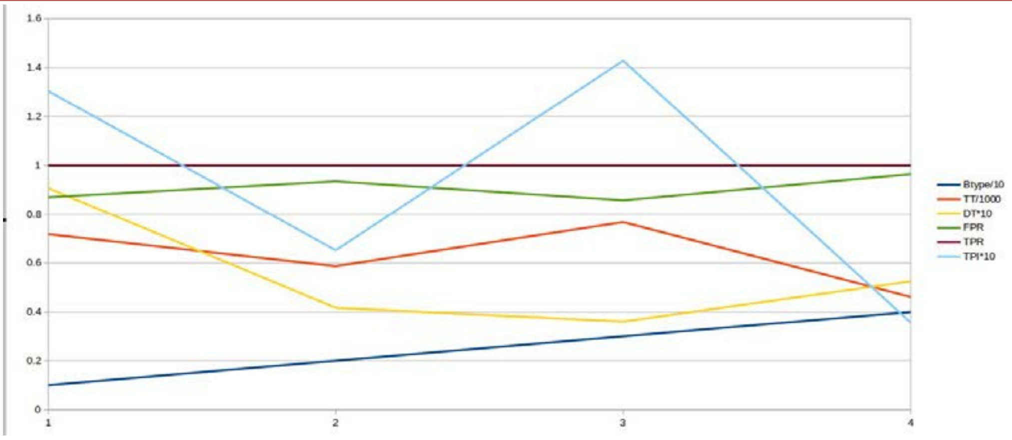


Figure 15: Image representing graphical data of the tests done with varying the value of Btype. The x-axis value 1 corresponds to GAB, 2 to DAB, 3 to RAB, and 4 to LB.

4.2 Local binary pattern (LBP)

This section includes data regarding the training experiments for LBP classifier which were used to arrive at the best optimal values of the parameters of training. The parameters on which experiments were conducted are Npos, Nstages, and Nneg, and for rest of the parameters optimal values stated in Section 4.1 are used.

4.2.1 Variation in Npos

Experiments were carried out to see the effect of the value of Npos on training. Npos was varied from 100 to 11500 in regular intervals of 500, while the testing for every 24 classifier was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.11.

Table 11: Parameter selection for training with every other parameter fixed except Npos.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	10
Ftype	LBP
Memory	5120
Sdimensions	20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	Variable
Nneg	27000

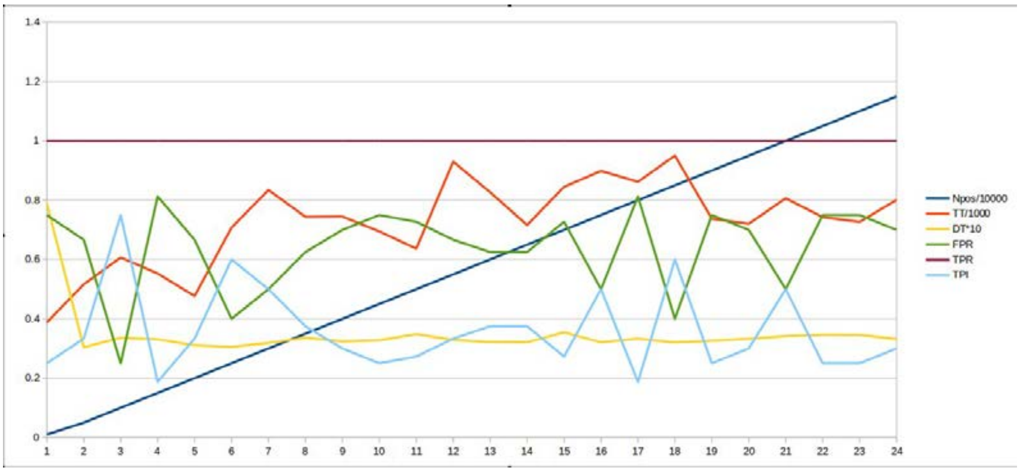


Figure 16: Image representing graphical data of the tests done with varying the value of Npos from 100 to 11500 in regular intervals of 500.

The results of the test are plotted in Figure16. In training a LBP classifier, Npos has no effect on TPR. Average TT increases with the increase in Npos, while DT remains almost constant. TPI keeps fluctuating between 0.3 and 0.7, thus concluding that the optimal value of Npos is same as mentioned in Section 4.1.1

4.2.2 Variation in Nstages

Experiments were carried out to see the effect of the value of Nstages on training. Nstages was varied from 10 to 50 in regular intervals of 10, while testing for every 5 classifiers was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.12.

Table 12: Parameter selection for training with every other parameter fixed except Nstages.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.5
Nstages	Variable
Ftype	LBP
Memory	10240
Sdimensions	20
Stage type	Boost
Btype	GAB
WTRate	0.95
Wcount	100
Npos	9000
Nneg	27000

The results of the test are plotted in Figure17. With the increase in Nstages, TT shows an increment as expected. FPR reduces between Nstages value 20 and 50, but increases after that. Similarly, TPR remains constant in that region, but decreases after Nstages crosses the value 40. Thus it can be concluded that the optimal value of Nstages lies between 20 and 40 as stated in Section 4.2.2.

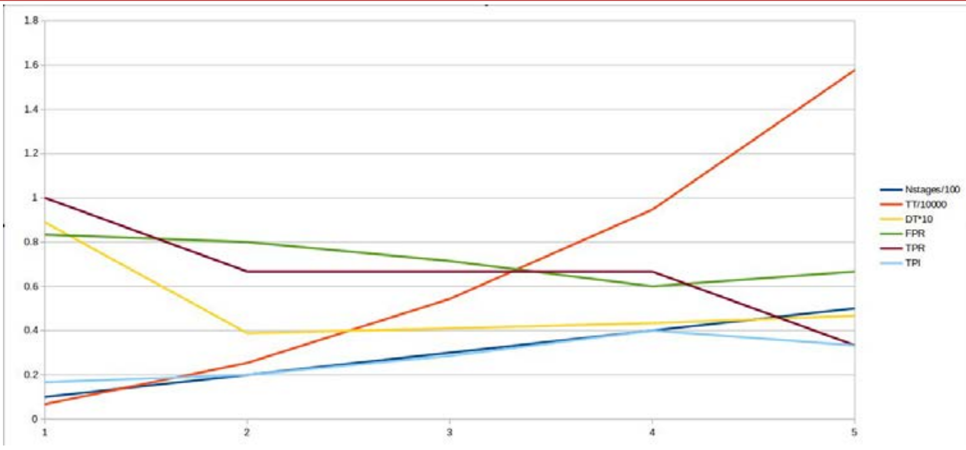


Figure 17: Image representing graphical data of the tests done with varying the value of Nstages from 10 to 50 in regular intervals of 10.

4.2.3 Variation in Nneg

Experiments were carried out to see the effect of the value of Nneg on training. Nneg was varied from 1000 to 27000 in regular intervals of 1000, while testing for every 27 classifiers was done on Figure5, and all the testing parameters were calculated. The training parameters are shown in Table.13.

Table 13: Parameter selection for training with every other parameter fixed except Nneg.

Number of positives	11716
Number of Negatives	27694
Samples created	11716
MHrate	0.999
MFARate	0.3
Nstages	10
Ftype	LBP
Memory	10240
Sdimensions	20
Stage type	Boost
Btype	RAB
WTRate	0.95
Wcount	100
Npos	9000
Nneg	Variable

The results of the test are plotted in Figure18. The parameters used in this experiment provide exceptional results with TPR equal to 1 and FPR equal to 0. As stated earlier, with the increase in Nneg, TT increases. Thus it can be concluded that the optimal value of Nneg is between 90% and 98% of the negatives available as explained in Section 4.1.3.

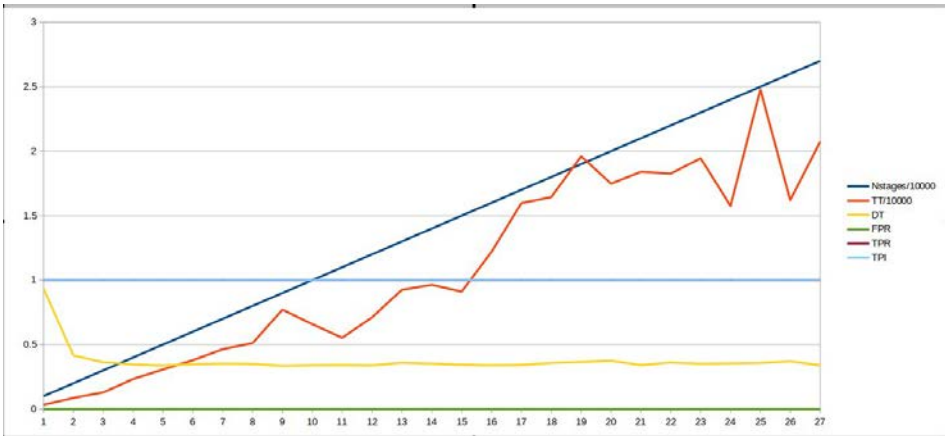


Figure 18: Image representing graphical data of the tests done with varying the value of Nneg from 1000 to 27000 in regular intervals of 1000.

5 Testing Experiments

All the testing experiments include training of more than 50 highly efficient cascade classifiers based on the optimal values obtained from the previous section. The efficiency of the classifiers has been tested on 4000 test images provided by Ayonix Inc, Japan as a private dataset to be used for research purpose.

Figure 19: Test Image with face detection applied on by a trained classifier with optimal parametric values and feature type HOG

Classifier used to detect faces in image shown in Figure19 is trained with feature type HOG. The training parameters include MHrate as 0.999, MFARate as 0.4, Nstages as 45, Memory as 20480 MB, Sdimensions as 30, Npos as 9000 and Nneg as 27000. This classifier has an average DT of 0.2724 sec per image, average FPR value as low as 0.0171, and average TPR value of 0.887, which reflects the high efficiency of the classifier.

Figure 20: Test Image with face detection applied on by a trained classifier with optimal parametric values

The faces in Figure 20 were detected using a classifier trained with feature type HOG. The training parameters include MHrate as 0.999, MFARate as 0.3, Nstages as 20, Memory as 20480 MB, Sdimensions as 25, Npos as 9000 and Nneg as 27000. This classifier has an average DT of 0.2892 sec per image, average FPR value of 0.293, and average TPR value of 0.714. The efficiency of this classifier was seen to increase if the parameters had values with MHrate as 0.999, MFARate as 0.4, Nstages as 50, Memory as 20480 MB, Sdimensions as 35, Npos as 9000 and Nneg as 27000. The results can be observed in Fig 21-22.

Figure 21: Test Image with face detection applied on by a trained classifier with optimal parametric values

Figure 22: Test Image with face detection applied on by a trained classifier with optimal parametric values

A few one of the best results can be observed in Figure 23. The results are invariant to orientation of face, facial expressions, use of spectacles, and moment of eyes and mouth.

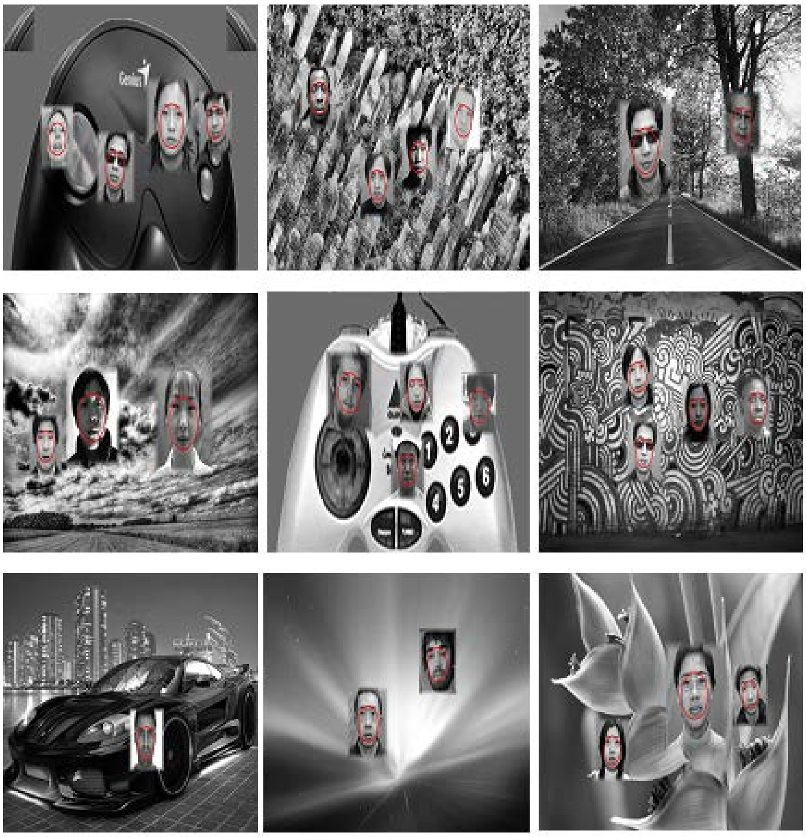


Figure 23: A set of 9 images, each acted upon by different HOG cascade classifiers representing the a few best experimental results.

Classifiers trained with feature type LBP provide better results than trained as HOG classifiers in case of face recognition. Classifier used to detect faces in image shown in Figure24 is trained with feature type HOG. The training parameters include MHrate as 0.995, MFARate as 0.4, Nstages as 55, Memory as 20480 MB, Sdimensions as 25, Npos as 9000 and Nneg as 25500. This classifier has an average DT of 0.1781 sec per image, average FPR value as low as 0.0144, and average TPR value of 0.8851 when tested on 4000 images.

Figure 24: Test Image with face detection applied on by a trained classifier with optimal parametric values

Similar to Fig 23, Fig 25 represents a set of one of the best results with classifiers trained using feature type LBP.

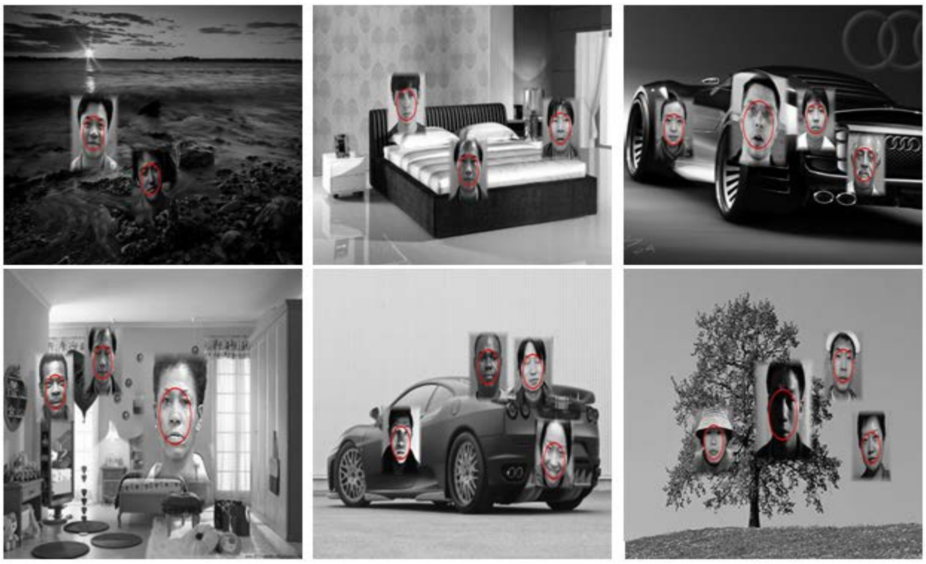


Figure 25 A set of 6 images, each acted upon by different LBP cascade classifiers representing the a few best experimental results.

6 Conclusion

The paper aimed at understanding and finding the optimality in the parameters involved in Viola-Jones cascade classifier training using OpenCV libraries. Two feature types were used name, HOG, and LBP. Table 14 provides the range of optimal values of every parameter in the training process.

Table 14: Optimal range of training parameters for viola-Jones based cascade classifier training.

Number of positives	>10000
Number of Negatives	>2*Number of positives
Samples created	=Number of positives
MHrate	Range [0.98 0.999]
MFARate	Range [0.3,0.5]
Nstages	Range [40,60]
Ftype	LBP/HOG/HAAR
Memory	Higher the better
Sdimensions	Range [20x20, 35x35]
Stage type	Boost
Btype	RAB
WTRate	Range [0.9, 0.99]
Wcount	Range [80, 120]
Npos	Range [68%, 85%] of Number of positives
Nneg	Range [90%, 98%] of Number of negatives

These parametric values are also applicable to any object training and recognition. The trained classifiers had the average values for DT, FPR, TPR, and TPI as shown in Table 15.

Table 15: Average values of testing parameters for viola-Jones based cascade classifier training in this paper.

Average DT (Sec/image)	0.2269938
Average FPR	0.0689601
Average TPR	0.8380987
Average TPI	0.8407661

The results obtained in the experiments are very much reliable and accurate and match the high-end complex face recognition systems. The training process is achievable by normal computer systems and takes average time of 25000 sec for training. This research could be further used in training for different objects, which could further be used in various research grounds globally.

REFERENCES

[1] Muller, N., Magaia, L. and Herbst B.M. Singular value decomposition, eigenfaces, and 3D reconstructions. SIAM Review, 2004, 46(3): p. 518-545.

[2] Bradski, G. Computer vision face tracking for use in a perceptual user interface, Intel Technology Journal, 2nd Quarter, 1998.

[3] Viola, P. and Jones, M. Rapid object detection using boosted cascade of simple features, IEEE Conference on Computer Vision and Pattern Recognition, 2001, 1: p.511-518

[4] Pentland, A, Looking at People: Sensing for ubiquitous and wearable computing, IEEE trans. on Pattern Analysis and Machine Intelligence, 2000, 22(1): p. 107-119.

[5] Li, S. Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., and Shum, H., Statistical learning of multi-view face detection, In 7th European Conference Computer Vision, 2002, part iv: p. 67–81.

[6] Schneiderman, H. and Kanade, T, A statistical method for 3d object detection applied to faces and cars, In IEEE Conference on Computer Vision and Pattern Recognition, 2000, vol 1: p. 1746–1759.

[7] Viola, P. and Jones, M. J., Robust real-time face detection, International Journal of Computer Vision, 2004, 57(2): p.151–173.

[8] Lienhart, R. and Maydt, J., An extended set of haarlike features for rapid object detection, In IEEE International conference on Image processing, 2002, vol 1, p. 900–903.

[9] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm, In proc. of International conference on Machine Learning, 1996, p.148-156

[10] M.-T. Pham and T.-J. Cham, Online learning asymmetric boosted classifiers for object detection, In proc. of Computer vision and pattern recognition, 2007, p. 1-8.

[11] P. Viola and M. Jones., Fast and robust classification using asymmetric adaboost and a detector cascade, In Neural information processing systems, MIT Press, Cambridge, MA, 2002.

[12] R. Lienhart, A. Kuranov, and V. Pisarevsky., Empirical analysis of detection cascades of boosted classifiers for rapid object detection, In DAGM-Symposium, 2003, p. 297–304.

[13] C. Huang, H. Ai, Y. Li, and S. Lao, High-performance rotation invariant multiview face detection, in IEEE Transactions on Pattern Analysis and Machine Intelligence, TPAMI, 2007, 29:671–686.

[14] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In proc. of International Conference on Automatic face and Gesture recognition, 2004, p. 79–84.

[15] Dataset-1, Caltech 101 Object categories, (available online), http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[16] Dataset-2, Amos Storkey, Image sequence segmentation data, (available online) http://homepages.inf.ed.ac.uk/amos/afreightdata.html

[17] Dataset-3, Oriental Bird Images, (Available online), http://orientalbirdimages.org/

[18] Dataset-4, Computational Visual Cognition Laboratory, MIT, (Available online), http://cvcl.mit.edu/database.htm

[19] Jerome H. Friedman, Greedy function approximation: A gradient boosting machine, IMS 1999 Reitz lecture (Available online), https://statweb.stanford.edu/~jhf/ftp/trebst.pdf