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.

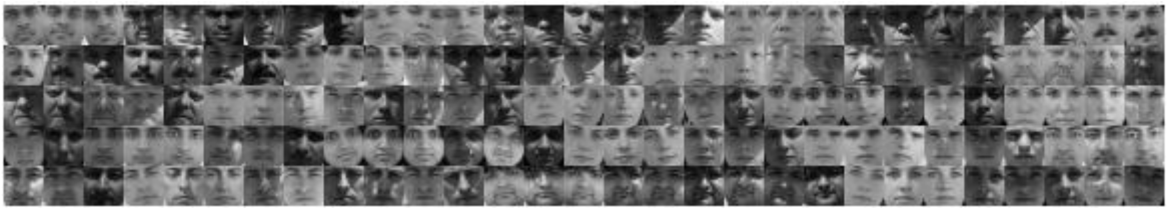

Figure 1: A set of 145 positive images. Each image 20x20 pixels in dimension.

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".

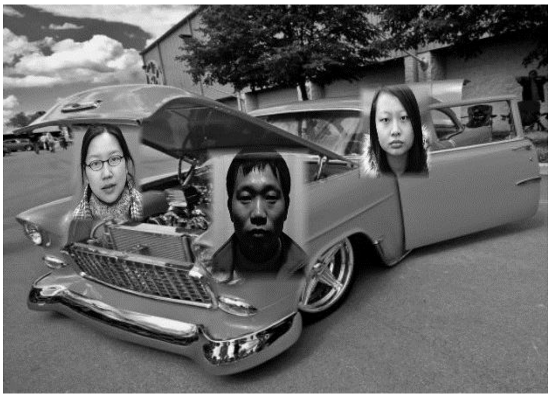

Figure 3: Test image with three faces with original resolution 1920x1060

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.

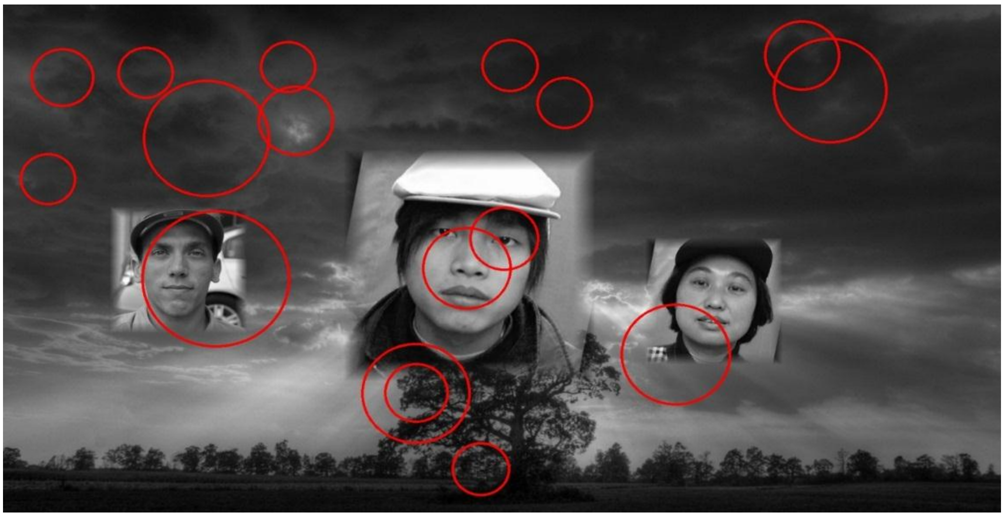

Figure 4: Image containing both false and true detections.

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.

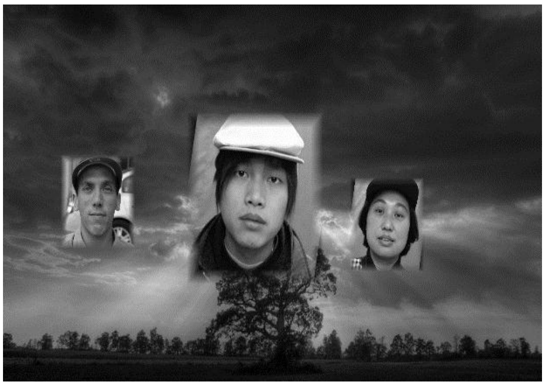

Figure 5: Test image with original dimensions 1920X1060

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 8th 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.

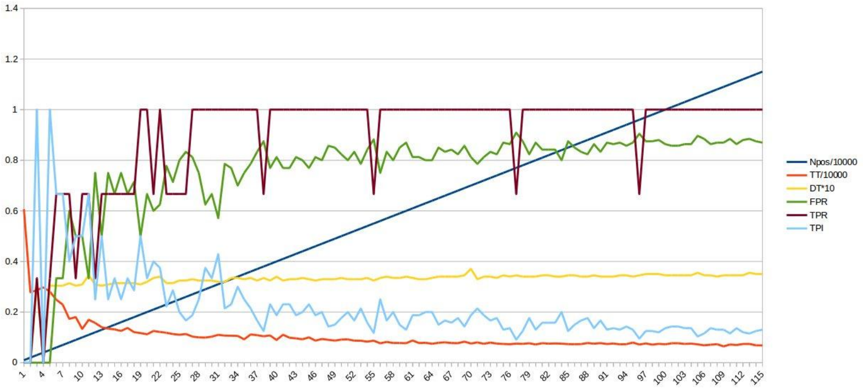

Figure 6: Image representing graphical data of the tests done with varying the value of Npos from 100 to 11500 in regular intervals of 100.

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.

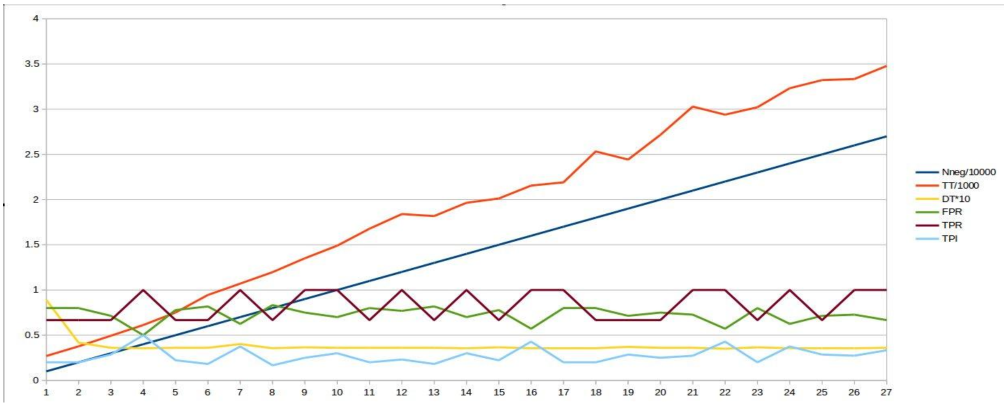

Figure 8: Image representing graphical data of the tests done with varying the value of Nneg from 1000 to 27000 in regular intervals of 1000.

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

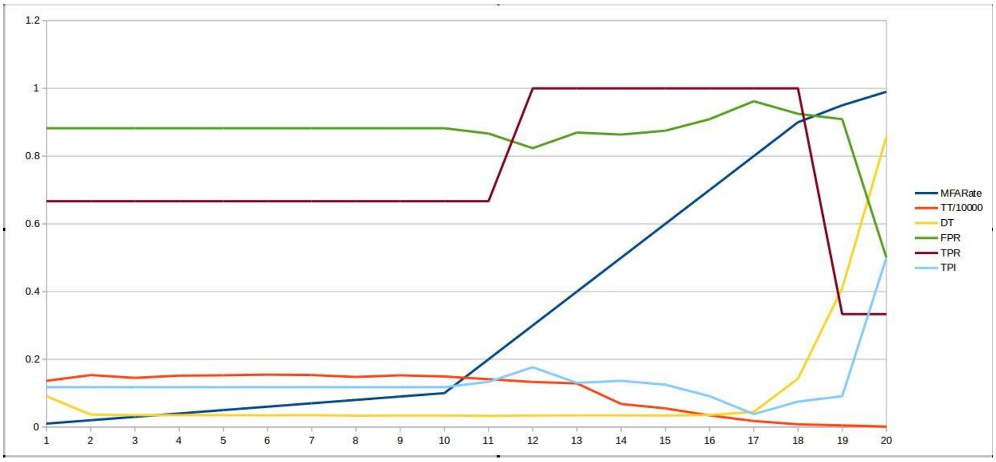

Figure 9: Image representing graphical data of the tests done with varying the value of MFARate from 0.01 to 0.99 in irregular intervals.

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

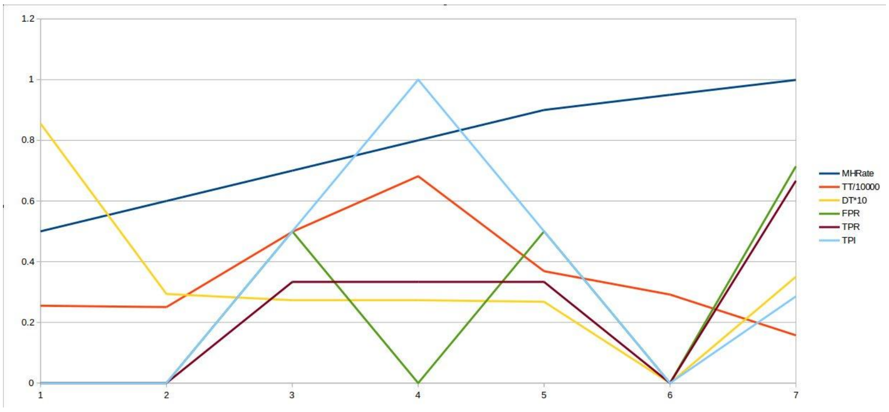

Figure 10: Image representing graphical data of the tests done with varying the value of MHRate from 0.01 to 0.99 in irregular intervals.

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.

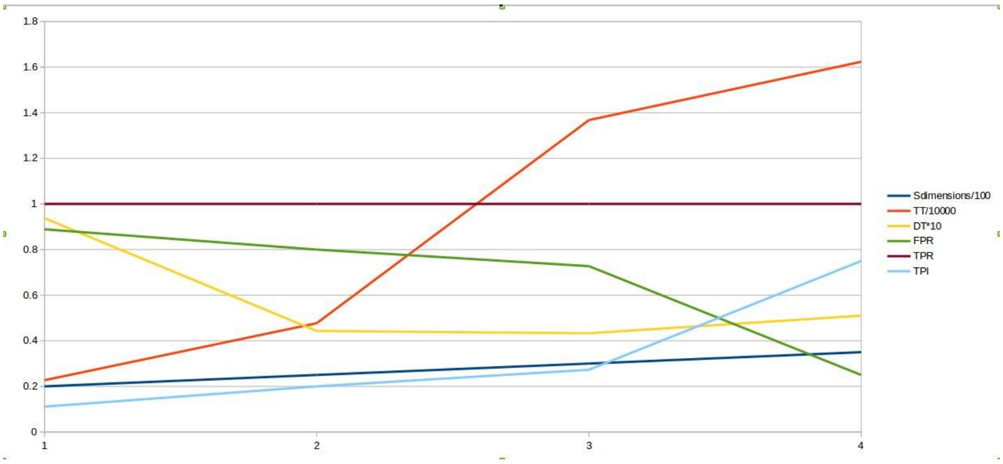

Figure 11: Image representing graphical data of the tests done with varying the value of SDimensions from 10x10 to 40x40 in regular intervals with the interval spacing 5x5.

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.

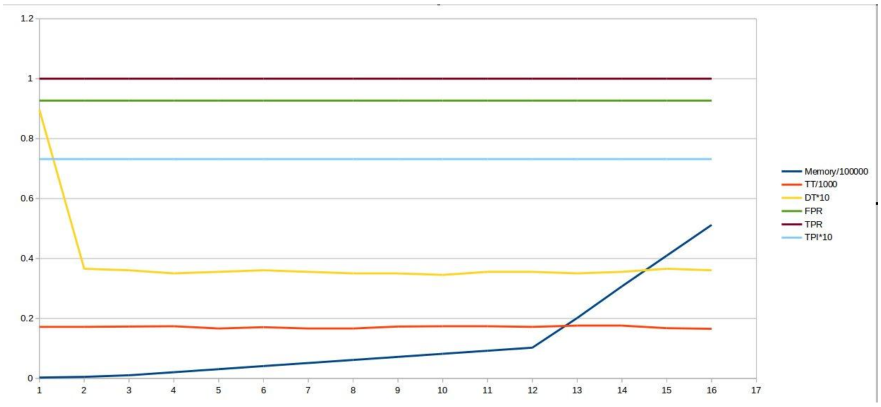

Figure 12: Image representing graphical data of the tests done with varying the value of Memory from 256 MB to 51200 MB in irregular intervals.

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.

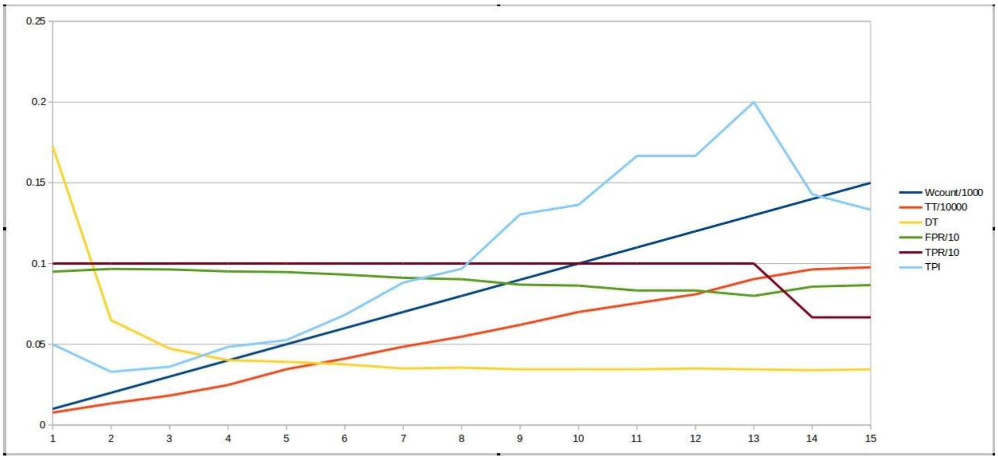

Figure 13: Image representing graphical data of the tests done with varying the value of Wcount from 10 to 150 in regular intervals of 10.

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.

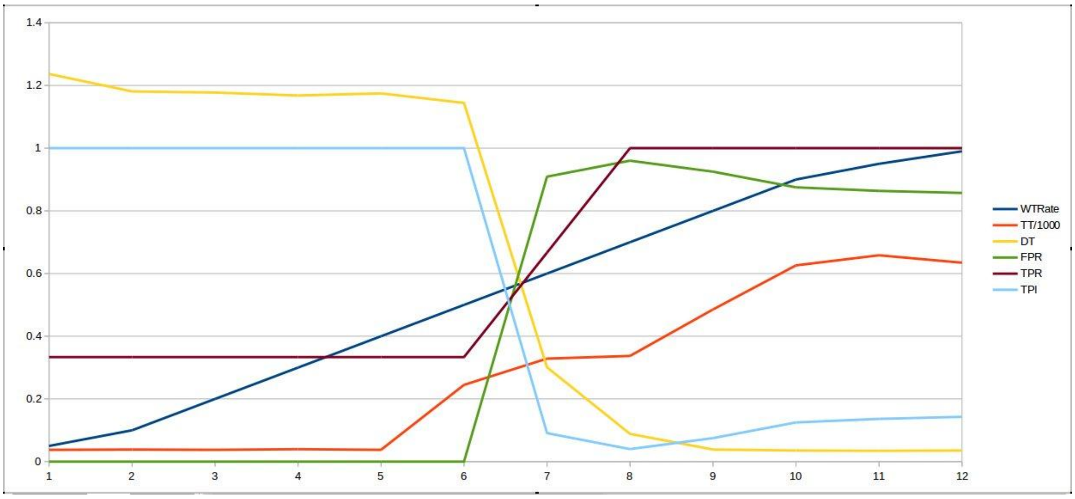

Figure 14: Image representing graphical data of the tests done with varying the value of Wcount from 0.05 to 0.99 in irregular intervals.

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.

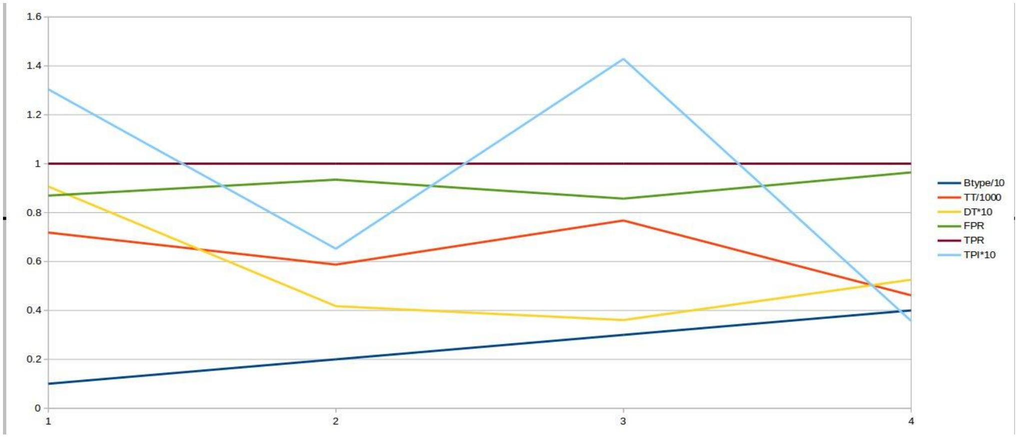

Figure 15: Image representing graphical data of the tests done with varying the value of Btype. The x-axis value 1 corresponds to GAB, 2 to DAB, 3 to RAB, and 4 to LB.

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

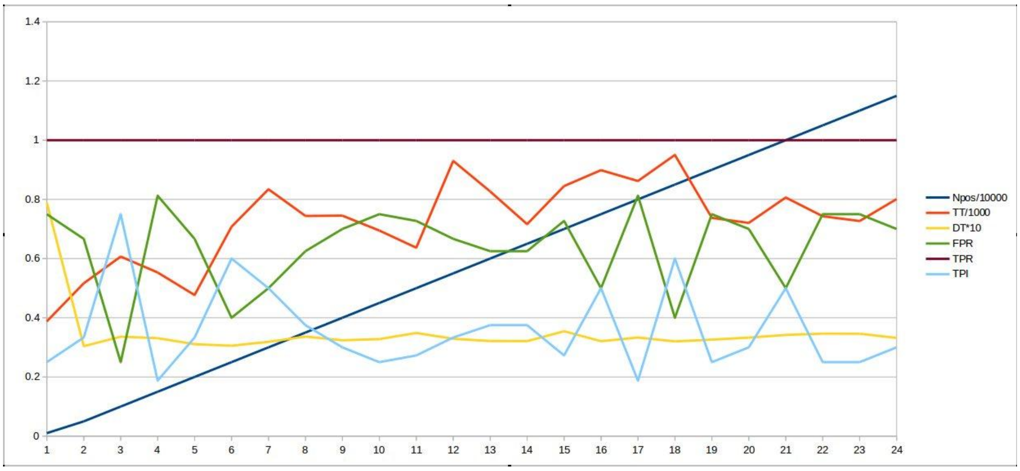

Figure 16: Image representing graphical data of the tests done with varying the value of Npos from 100 to 11500 in regular intervals of 500.

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.

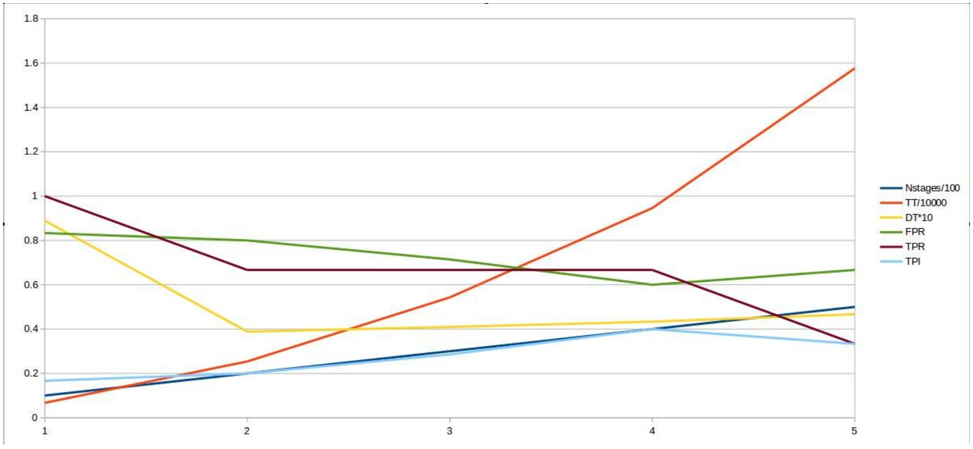

Figure 17: Image representing graphical data of the tests done with varying the value of Nstages from 10 to 50 in regular intervals of 10.

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.

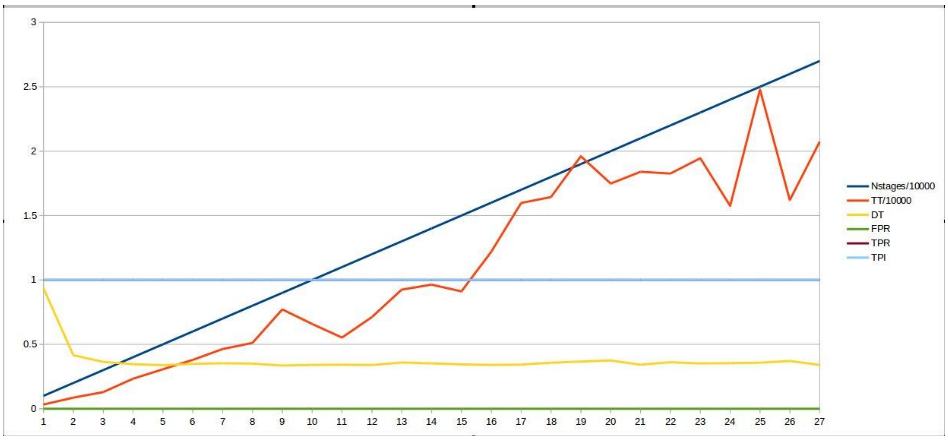

Figure 18: Image representing graphical data of the tests done with varying the value of Nneg from 1000 to 27000 in regular intervals of 1000.

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.

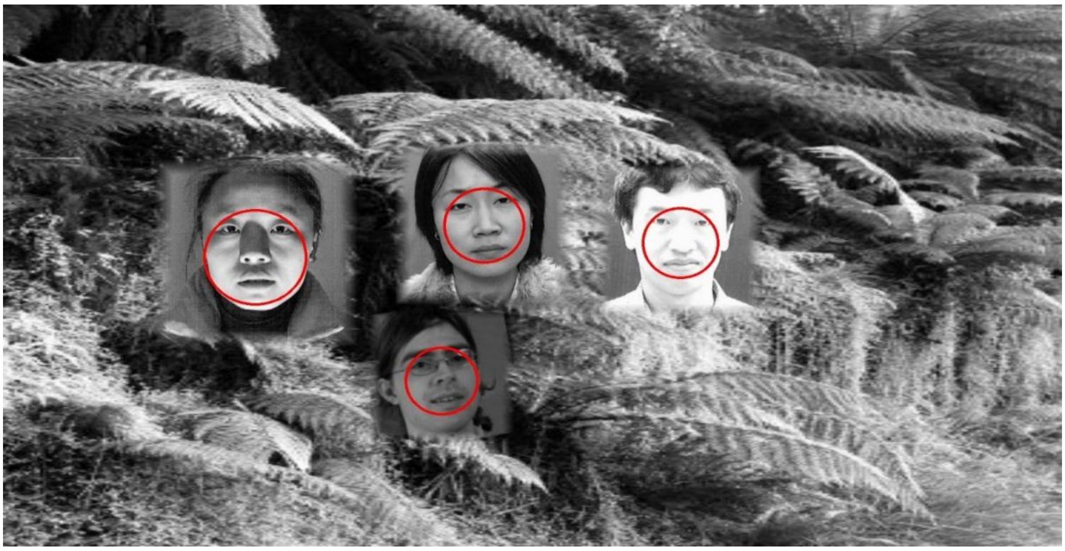

Figure 19: Test Image with face detection applied on by a trained classifier with optimal parametric values and feature type HOG

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.

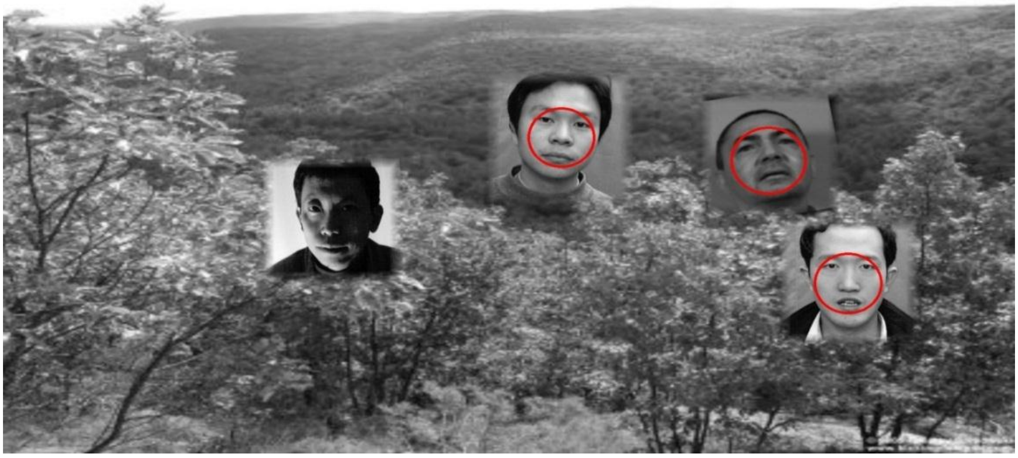

Figure 20: Test Image with face detection applied on by a trained classifier with optimal parametric values

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.

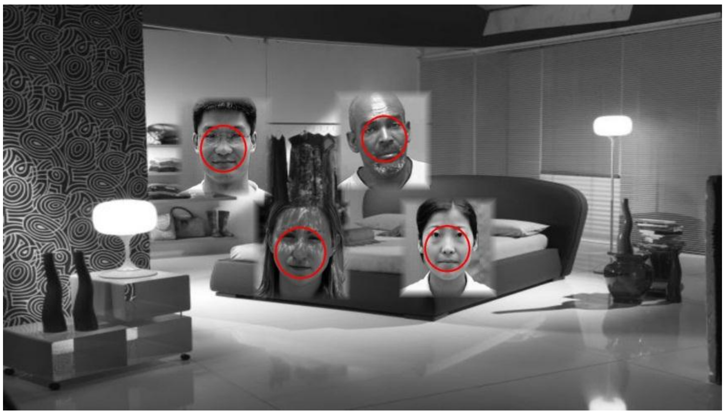

Figure 21: Test Image with face detection applied on by a trained classifier with optimal parametric values

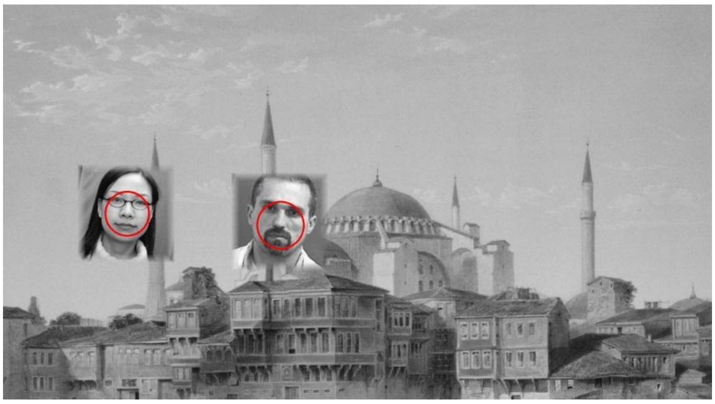

Figure 22: Test Image with face detection applied on by a trained classifier with optimal parametric values

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.

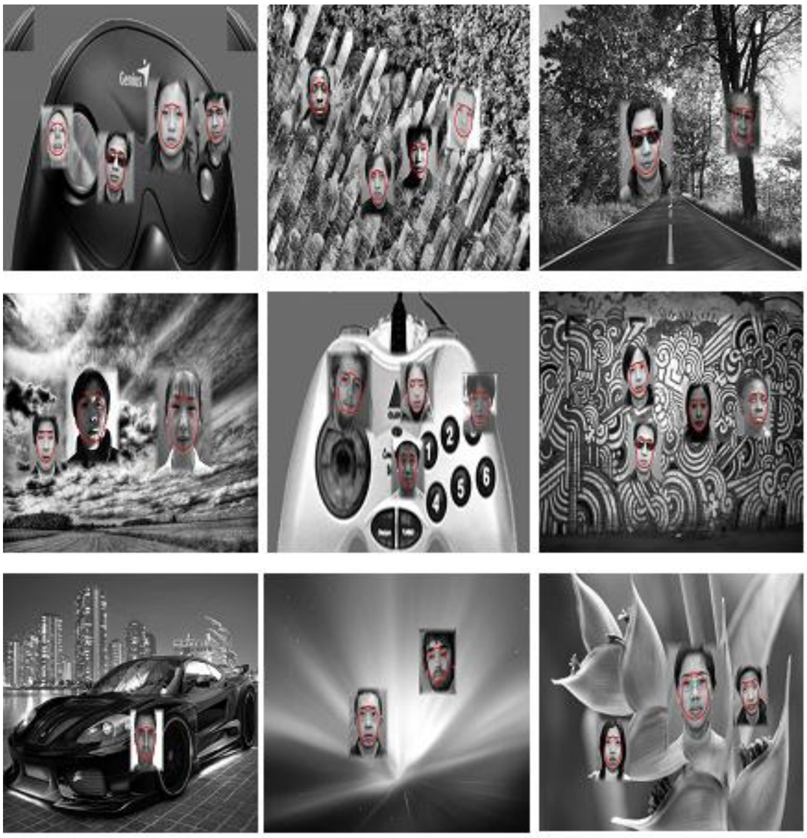

Figure 23: A set of 9 images, each acted upon by different HOG cascade classifiers representing the a few best experimental results.

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.

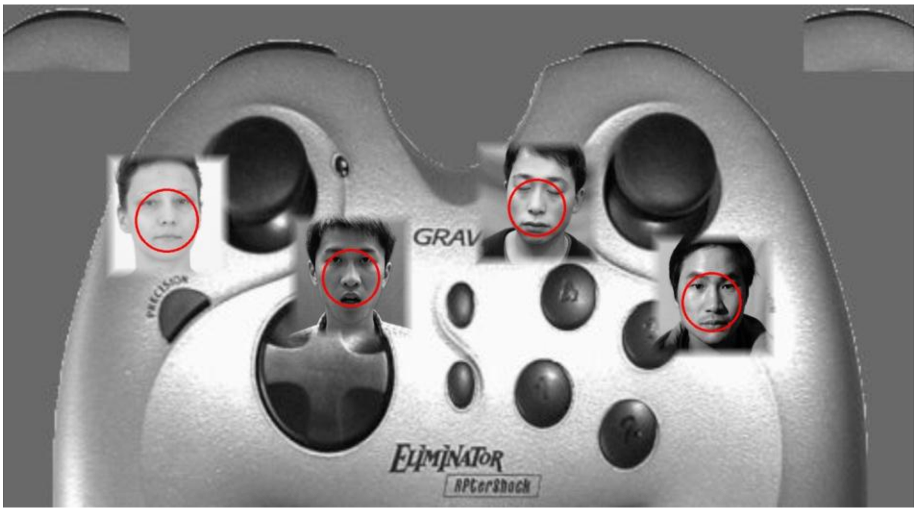

Figure 24: Test Image with face detection applied on by a trained classifier with optimal parametric values

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

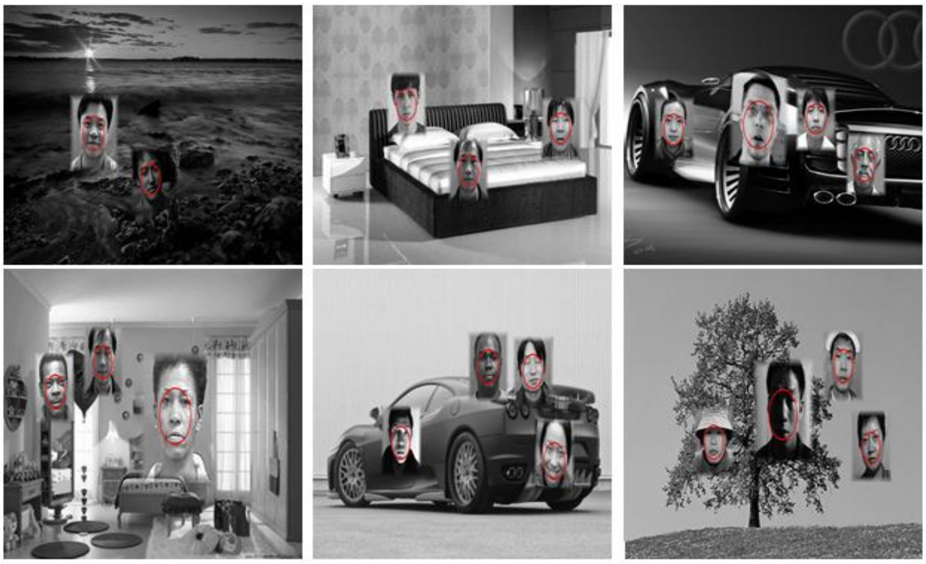

Figure 25 A set of 6 images, each acted upon by different LBP cascade classifiers representing the a few best experimental results.

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