International Journal of Latest Trends in Engineering and Technology
Vol.(10)Issue(1), pp.086-095
DOI: http://dx.doi.org/10.21172/1.101.15
e-ISSN:2278-621X

SURF DESCRIPTOR BASED OPTIMIZED FACE DETECTION USING BOOSTED SVM CASCADE AND SKIN COLOR FILTERING

Jayasree M[1], N K Narayanan[2], Kabeer V[3], Arun C R[4]

Abstract- An enhanced and optimized face detection system robust to variations in pose and illumination, along with minimum false positive rate has been proposed in this paper. SURF descriptors are used as feature vectors and SVM is used as the base classifier. A number of SVMs are trained using a boosting algorithm and stored as an ensemble of weighted SVMs. A cascade of different stages is built to make the final strong classifier. Optimization is a main criterion in designing this system, as far as false positive rate and processing time are concerned. These two features are achieved by a skin color filtering module integrated to the proposed face detection system, which outputs a skin-color segmented binary image. The current detection window is processed only if there are enough skin color pixels in that window. Further, the detection window is classified using boosted SVM cascade classifier. Typical post processing operations completes the face detection procedure. Testing on FDDB benchmark dataset shows that the proposed system outperforms classic boosted cascade face detection methods on ROC curve evaluation and state-of-the-art detectors on comparison based on detection rate.
Keywords – Face Detection, Boosted SVM Cascade, Skin color filtering, ROC Curve

1. INTRODUCTION

Face detection is a major research area in Digital Image Processing due to its widespread applications in various fields. It is an indispensable initial step towards all automated face analyzing algorithms like face recognition, age recognition, gender identification, facial expression recognition and many more. Given an arbitrary image, the goal of face detection is to determine whether any faces are present or not in the image and, if present, return the image location and extent of each face [1].

Face detection became practically feasible in real world applications with the seminal work by Viola and Jones [2]. Boosted-cascade methods are made popular in the field of face detection by Viola-Jones. Three core ideas in their algorithm are 1) integral image 2) classifier learning with a boosting algorithm and 3) cascade structure. These ideas seeded to a number of researches based on original Viola-Jones algorithm [2]. A method to address challenges in unconstrained face detection has been introduced by Liao et al., [5] using a new image feature called Normalized Pixel Difference (NPD), which is scale invariant, bounded, and is able to reconstruct the original image. A novel method to detect and correct the failures in appearance based facial tracking is proposed by Wang et al., using a sparse coding strategy to learn an efficient feature representation of the difference between the face template and the warped image [6]. Unlike the traditional methods, new meta-heuristic optimization algorithm for face detection has been proposed by Gao et al., [7] in which, face tracking is treated as an optimization problem and a, differential harmony search (DHS) is introduced to solve face tracking problems.

A collaborative tracking framework was introduced to robustly track faces under large pose and expression changes and to learn their appearance models online [4]. Kawulok et al., has presented a new method for precise detection of frontal human faces and eyes using a multi-level ellipse detector combined with a support vector machine verifier [8]. A new face tracking system in an illumination insensitive feature space, called the gradient logarithm field (GLF) feature space was proposed by Zou et al [9].

Even though a lot of research has been carried out in this area, a face detection system that has achieved optimal detection rates and minimum error rate, for faces including those under varying pose and illumination, is yet to be realized. This article presents an enhanced face detection system which is robust to variations in poses and illumination to a tolerable extends. In the proposed method, Speeded Up Robust Features (SURF) descriptor is used as the feature vector representing face and non-face. It was introduced by H. Bay et al., [10] and proved to be efficient for representing human faces in an innovative work by J. Li et al., [11]. Only a few hundreds of local SURF patches are considered for a detection window instead of hundreds or thousands of Haar-like features. Also SURF is a multi-dimensional descriptor while Haar-like feature has only a single dimension.

[1] Department of Computer Science & Engineering, Govt. Engineering College, Thrissur, Kerala, India
[2] College of Engineering, Vadakara, Kerala, India
[3] Department of Computer Science, Farook College, Kozhikode, Kerala, India
[4] Department of Electronics & Comm. Engineering, Model Engineering College, Kochi, Kerala, India

In the proposed system, the basic Adaboost algorithm is adopted for building the face detection system. A number of SVMs are trained using a boosting algorithm and stored as an ensemble of weighted SVMs. A cascade of different stages is built to make the final strong classifier. Initially, the input image is passed through a skin color segmentation module, which outputs a skin-color segmented binary image. The current detection window is processed only if there are enough skin color pixels in that window. Further, the detection window is classified using boosted SVM cascade classifier. Typical post processing operations completes the face detection procedure.

SURF and SVM are combined for face detection and face component localization by D. Kim and R. Dahyot [12]. In this method, SURF key-point detection is used to find interest points automatically and store this information in SURF descriptors, which is finally used for classifying a face using a trained SVM. In our method, we do not use SURF interest point detection. SURF descriptors for the whole detection window are given as input to the training algorithm, and relevant descriptors are automatically selected during the boosted cascade training procedure.

Based on the milestone work of Viola-Jones, a number of boosted-cascade detection systems were introduced [2]. Most of them vary in extracted features, classifier and in the boosting method. In the research work of Jianguo Li et al., SURF cascade based face detection technique was introduced [11, 13]. The SURF descriptor is trained using boosted logistic regression classifiers. They use Area under Receiver Operating Characteristic (ROC) curve as a criterion for convergence, and have achieved a lot of improvement in training speed and detection accuracy. In the proposed work, SVM is used as the base classifier instead of a weak classifier like logistic regression, thereby reducing the number of boosting rounds and negative samples trained.

Face detection using skin-tone segmentation is realized in [11] [14]. In this paper, rules have been introduced to find skin and non-skin pixels. It was proved that the use of color model rules for skin-color detection is easy and effective. Skin-color segmentation is adopted from their work as a preprocessing step for the proposed face detector.

Support Vector Machine (SVM) [15] is used in this work as the base classifier in the form of boosted ensembles. SVM is used instead of a weak classifier, because of its classification accuracy. This in turn reduces the need of number of boosting rounds and number of stages in the attentional cascade structure. One of the difficulties with boosted cascade methods is the need of a large number of negative samples. Most of the boosted-cascade methods make use of the fine-grained Haar-like feature [2]. Size of feature search space is very high for these features. This in turn requires more training time [11]. Also high false positive rate is another drawback of existing methods. These problems motivated us to develop a new method for face detection and to come up with the proposed face detection system. This work adopts ideas from original Viola-Jones method for face detection [2] and also from many advanced research in this area [11, 13, 14].

Original SURF descriptors only stores information about gradient space. But, skin-color of different people is found to be clustered in the chromatic color space [16]. So this information can be used to limit the search space to only skin color areas in the image there by reducing false positive rates and improving speed of detection. Rule based skin-color segmentation [14] is used as the preprocessing step in the detection process.

The rest of the paper is organized as follows: Section II presents the proposed scheme in detail, Section III describes the implementation details, Section IV discusses the evaluation methods and their results and Section V draws the conclusion

2. PROPOSED ALGORITHM

In any face detection system, there are mainly two phases: Training phase and Detection phase. In the training phase, the face detection system is trained using known images, whereas in detection phase, the actual detection of the face takes place. An overview of the proposed system is given in Figure 1.

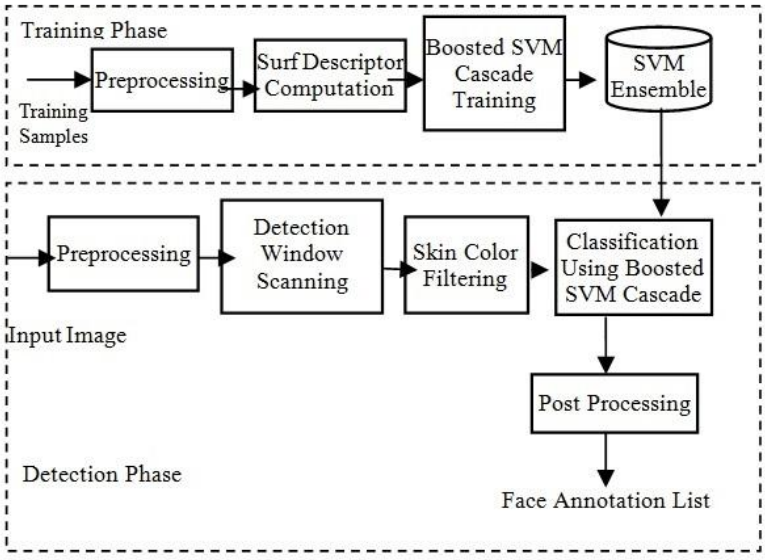

Figure 1. Proposed Face Detection System Overview

Input to the training module is positive and negative samples of images of size equal to the used target detection window. Positive samples are used to train face and negative samples are used to train non-face. Initially, noise filtering is performed on the images to improve training. If the training image is color, it is converted to a grayscale image.
In the feature extraction phase of training, SURF descriptors are computed for every positive and negative sample and stored as a training matrix. At the end of this procedure a number of SVM ensembles are generated. An SVM ensemble consists of a number of weighted SVM models, which are used later for detecting faces from the input images.
Detection phase is where the actual face detection is carried out using the trained face detector. The input image is first passed through the preprocessing step, where it is converted from a color image to a grayscale image. At the same time color image is passed through a skin color segmentation procedure, which store skin color information in the form of binary values. After pre-processing, the input image is scanned for faces in various resolutions. Extracted target detection windows are then passed through a skin-color filtering procedure, where non-skin color windows are rejected. If a detection window passes skin color filtering test, it goes into the classification step, in which trained SVM ensembles are used for the cascaded detection procedure, where most of the false positives are eliminated.
Finally, a post processing step removes further false positives. The result of this face detection system is a list of detected faces with necessary parameters like location and extend. The various steps of the proposed face detection system are explained in the rest of this section.

2.1 SURF Descriptor Extraction

Speeded Up Robust Features (SURF) was introduced as a detector and descriptor of local scale and rotation invariant image features [11]. In our system, SURF is used only as a descriptor. SURF descriptors are computed from a detection window and are used as a feature vector for that window.
SURF descriptor is defined in the gradient space. Let dx be the horizontal gradient image obtained with the filter kernel [-1, 0, 1] and dy be the vertical gradient image obtained with the filter kernel $[-1, 0, 1]^T$. dx and dy are summed over a sub-region to form a first set of entries into a feature vector. The polarity of the intensity changes is also extracted by the sum of the absolute values of the responses, |dx| and |dy|. Hence, each sub-region has a four-dimensional descriptor vector v for its underlying intensity structure as given below:

$$v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$$

A target detection window of fixed size is defined for scanning an input image or for training samples and for computing SURF descriptor. It is a portion of the actual image, where the presence of a face is examined. In the case of training, each image size is made equal to the size of the detection window. In the case of actual operation of the system, this detection window frame is moved over the test image to find portions of the image that has a face. The procedure for generating SURF description is given below.
Each target detection window of size 40 × 40 is densely sampled into local patches of different sizes. Patch sizes ranging from 12 × 12 to 40 × 40 in a step of 4 pixels is chosen. Also slide the same patch over the target detection window with a fixed step, say 4 pixels. Additionally patch height-width ratio is allowed to be 1:1, 1:2 and 2:1. In this work, a total of 284 local patches from a given detection window is generated. Each local patch generated using the above method is represented by 2 × 2 = 4 cells of SURF descriptors. Each cell has a four-dimensional SURF descriptor vector v for its underlying intensity structure. Concatenating these feature vectors in 2 × 2 cells will yield SURF descriptor as the feature vector for a local patch. Thus, each local patch in the target detection window is represented using 4 × 2 × 2 = 16 dimensional feature vector. Configuration of local patches and cells are illustrated in Figure 2.

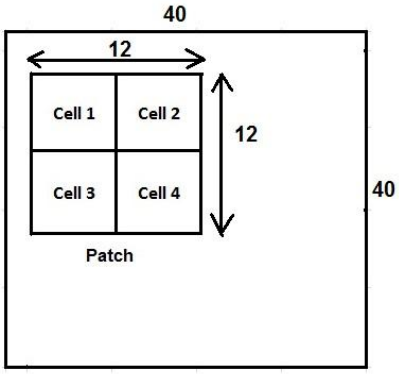

Figure 2. Patch and Cell Configuration inside a detection window

To make the descriptor invariant to the patch size and to reduce impact of illumination and contrast variations, feature vector of each local patch is normalized using L2 normalization and thus created the unit vector. L2 unit vector is found first by finding

the Euclidean distance of the vector from the origin and then dividing the whole vector, element by element with this normalized vector.

2.2 Boosted SVM Cascade Training

A strong classifier is trained from positive and negative samples using a boosting algorithm at various stages. Positive samples are images of size 40 × 40, each of them contain exactly one face represented as a grayscale image. Negative samples also have size 40 × 40 which can be any gray-level pattern other than a face. SURF descriptors are computed from each of these samples, and stored as training matrix. These positive and negative feature matrices and their respective classes (1 for face, -1 for non-face) are given as inputs to the learning algorithm. Finally, ensembles of SVM classifiers are returned along with other parameters that are needed for classification. Procedure for one stage training is given in Algorithm 1.

Algorithm 1: Boosting SVM on SURF local patches in one stage

1. Given training set:

$$\left\{\begin{matrix} Np \\ i=1 \end{matrix}(x_i^k, 1)\right\}; \left\{\begin{matrix} Nn \\ j=1 \end{matrix} y_j^k, -1)\right\}, k = 1{:}K$$

Where Np: number of positive samples, Nn: number of negative samples, K: total number of local patches. x^k; $y^k \in R^d$, d-dimensional SURF descriptor representation of k^{th} local patch.

2. Initialize weights for positive and negative samples

$$w_{1,i}^p = \frac{1}{N_p}; i = 1{:}N_p \qquad w_{1,j}^n = \frac{1}{N_n}; j = 1{:}N_n$$

3. For t = 1 : T boosting round

3.1. If t = 1, select n_p random positive samples and n_n random negative samples from training set; including previous misclassified samples if any. Else select n_p positive samples and n_n negative sampes from the training set according to their weight.

3.2. For k = 1 : K

3.2.1. Train SVM with local patch k from n_p positvie and n_n negative samples.

3.2.2. Test trained SVM on whole training set to get misclassified samples and add weights of misclassified samples to get classification error for current patch. Error of positive samples:

$$e_{t,p}^k = \sum w_{t,i}^p \mid h_t^k(x_i) \neq 1$$

Error of negative samples: $e_{t,n}^k = \sum w_{t,j}^n \mid h_t^k(y_j) \neq -1$

Total error for k^{th} classifier: $e_t^k = e_{t,p}^k + e_{t,n}^k$

3.3. Find k, such that e_t^k is minimum and set $h_t(x) = h_t^k(x)$; $e_t = e_t^k$ $\qquad \alpha_t = \frac{\frac{1}{2}\log(1-e_t)}{e_t}$

3.4. Update weights for next iteration:

$$w_{t+1,i}^p = w_{t,i}^p \, X \, e^{\alpha t}; if \; h_t(x_i) \neq 1 \;, w_{t+1,j}^n = w_{t,j}^n \, X \, e^{\alpha t}; if \; h_t(y_j) \neq -1$$

3.5. Normalize weights such that: $\sum_i w_{t,i}^p = 1 \; and \; \sum_j w_{t,j}^n = 1$

4. Output final strong classifier for this stage: $H_T(x) = \sum_{t=1}^{T} \alpha_t \, h_t(x)$

Suppose there are N_p positive training samples, N_n negative training samples and K possible local patches, where local patches are represented using 16 dimensional SURF feature x. Each stage is a boosted learning procedure with SVM as base learner. Output from a stage is the strong classifier H_T which is a linear combination of a T number of base SVM learners (h_t) as given below.

$$H_T(x) = \sum_{t=1}^{T} (a_t \, h_t(x))$$

T is the number of boosting round in a stage. Base learners in boosting rounds are assigned a weight α_t during training. Training happens in continuous boosting rounds. In each boosting round, N_p number of positive samples and N_n number of negative samples are selected from the total training set. One SVM per local patch is trained with these selected samples. At the end of current boosting round, local patch with lowest error is considered as the patch for current round and SVM trained for this patch is considered as the classifier for current round. Each boosting round t produces the classifier ht with its weight α_t. The weight of a classifier is determined based on its accuracy which in turn is based on the weight of misclassified samples. Every training sample is associated with a weight in every boosting round t. Initially, all weights are equally distributed over all positive and negative samples separately. Weights of misclassified samples are increased in the next iteration. So the probability of selecting misclassified samples for training in the next iteration would be higher. At the end of each boosting round, weights are separately normalized.

Misclassified samples of each stage are given as mandatory input to the next stage. For each stage, the number of boosting rounds, T may be different. For the first boosting round of a stage, all the misclassified samples from the previous stage are added to the training sample set. So the misclassified samples of previous stage have a higher probability of correct classification in the current stage. The number of stages trained depends on the tolerable limit of false positive rate. Experiments showed that the false positive rate falls rapidly.

2.3 Skin Color Filtering

Generally, human skin-color pixel spans only a small area of a test image. Scanning only the suspected area improves speed of detection and reduces false positive rate.

For skin color filtering, the input image is pre-processed with skin color segmentation. The skin color information of the whole image is saved. When a detection window scans an input image, span of skin color pixels is found by referring this saved skin-color image. If the skin region is below a threshold, no processing is done in that window. The preparation of this skin color filter image is given in Figure 3.

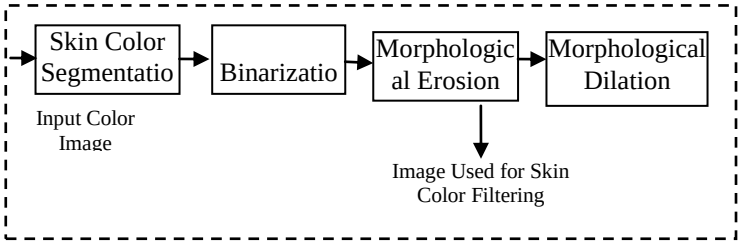

Figure 3. Preparation of Skin Color Filter Image

A three rule combination [14] is used for correctly identifying skin color. It uses RGB, YC_bC_r, and HSV color spaces. Each rule is defined for each color space. A combination of these rules is used to separate skin color from the background.

The segmented image is then passed through binarization procedure, where the whole image is limited to have only two values of gray-levels. One gray-level represents skin and other represents non-skin area. The resultant image is passed through a morphological erosion process with a relatively small structuring element (SE). This process removes most of the skin-color pixels that is spread over different parts of the image and retains continuous skin-color regions. Final step is to dilate the resultant image with a SE larger than that used for erosion, but much smaller compared to face size. This removes smaller holes (like eyes) in continuous skin-color region. The resultant image is used for skin-color filtering.

2.4 Detection of Faces

Detecting faces is the process of scanning an input image for presence of faces and storing detected faces into an annotated list. A digital color image from which faces are to be detected is given as the input image and annotation list is the list of five attributes associated with the faces being detected. Attributes selected are X-coordinate position in the original image, Y-coordinate position in the original image, height and width of the detected face and a detection score. First four parameters are directly obtained from detection window. Detection score is a real value obtained from the detection procedure. The procedure used for face detection is given in Algorithm 2 in which Img is the input image and Win is the current subset of images under consideration for detecting faces. Size of Win is fixed as 40 X 40. ScaleVal is used to scale down the input image to a lower size.

Algorithm 2 : Face Detection Using Proposed Method

```
Input: Image Img
Output: List of Detections FaceList

While size(Img) ≥ size(Win) do
    for all Detection window Win in Img do
        if Win passes skin color filtering then
                stage = 0
                score = 0
        for s = 1 : S do
        Compute Ts number of patch FV from Win in x needed for H_T^s (x)
        res = H_T^s (x)
         if sign (res) = + then
         score = score + res
        stage = stage + 1
        else
        break
        end if
  end for
        if stage = S then
            Add current Win and score as a new entry in FaceList
        end if
    end if
  end for
Scale down Img with ScaleV al
end while
FaceList = PostProcess(FaceList)
```

$H^S_T(x)$ is the T^{th} stage strong classifier, which is a weighted ensemble of SVM classifiers, obtained through the Adaboost learning procedure. $H^S_T(x)$ returns a positive value, if the current window is detected as a face, otherwise a negative value is returned.

An input image is scanned in the form of image pyramids. Initially, an image with the given resolution is scanned by the detection window Win. Then, the image is resized to a certain ratio and again the whole image is scanned with a new resolution. This process is repeated until the image is resized to the size of the detection window or even lesser. Image is consecutively resized by the factor ScaleVal. Detection happens as follows. The input image is scanned at current resolution from top-left to bottom-right using detection window Win. Then, it is passed through skin-color filtering. If Win gets through this step, then it passes through different stages one after another. This process is given in Figure 4 taken from Viola-Jones [2], where T represents True and F represents False. This is called cascaded detection process.

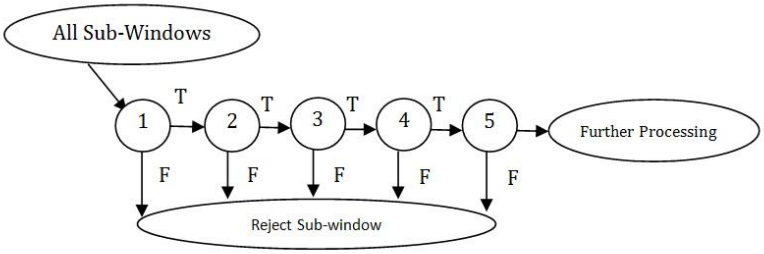

Figure 4. Schematic depiction of the detection cascade. [2]

In each stage, local patch feature vectors used for the current stage are computed. Sign of the classification result indicates the probability of the result being a face. If current stage returns a positive value, it is passed to the next stage for further processing. If the current window passes all stages successfully by returning a positive value in every stage, then that would be added to detected face list FaceList along with its detection score. Detection score is the sum of results of classifications from all stages.

This process is repeated for detecting faces of size larger than the size of the detection window. This type of image scanning allows detecting faces of any scale in the input image. Resultant FaceList undergoes post processing operation to get the final FaceList. This face list is later used for annotating faces in images.

2.5 Post Processing

Final detector is insensitive to small changes in translation and scale. So, a face may be detected multiple times, both in translational movement and in various resolutions. This happens because there is only small change in the feature vector while moving some the window right or down and with consecutive resolutions. To solve this, two post-processing operations are performed on the detected list, to group nearby responses and to reduce false positive rates. They are Merging Multiple Detections and Non-Maximum Suppression.

Merging Nearby Detections: It is the process of combining detections that probably indicate same face. Two detections D_1 and D_2 are grouped together if they satisfy the following condition.

$$\frac{\text{IntersectionArea}(D_1; D_2)}{\text{UnionArea}(D_1; D_2)} > \text{Threshold}$$

All entries in the face list are grouped in such a way that one face appears only in one group. After successful grouping, entries in the same group are merged together as a single entry and other entries are deleted. First four parameters of the new entry will be the mean value of x-coordinates, y-coordinates, width and height respectively computed from all face entries in the group. Score for new entry is the maximum score in that group. New face list contains these merged entries from all groups. This process is repeated until no further merging is possible.

Non-Maximum Suppression: If two detections are highly overlapping and the difference in size is huge, one of them should be eliminated. It is done through non-maximum suppression. For example, a very small false positive region can be caught under real face detection using non-maximum suppression, which could not have been removed by merging nearby detections. If detections D_1 and D_2 satisfy the following conditions, then detection with smaller detection score is removed, and detection with the highest score is retained.

$$\frac{\text{IntersectionArea}(D_1; D_2)}{\text{Area}(D_1)} >> \frac{\text{IntersectionArea}(D_1; D_2)}{\text{Area}(D_2)}$$

3. IMPLEMENTATION DETAILS

Positive samples for the face detector were collected from GENKI [17] and the FaceTracer dataset [18]. Each face image is converted to 8 bit gray image, and resized to 40 x 40 without affecting the aspect ratio of the face. For generating negative sample windows, grayscale images without any faces are used, mainly from UMD negative dataset [19]. As a total, 5000 positive samples and 8000 negative samples were generated for training purpose and stored as positive and negative training matrices respectively. In a training matrix, each row is made by concatenating all the patch features for each sample. In the training procedure, required patch features were selected from the samples using matrix indexing.

Feature extraction phase extracts four values per cell of a local patch: Σdx, Σdy, $\Sigma |dx|$ and $\Sigma |dy|$ to create SURF descriptor. For efficient computation of these values, four separate integral images are maintained. Using these tables, sum of values under any rectangle can be calculated using only 4 array references.

Number of positive samples used for training (N_p) are 5000 and the number of negative samples used for training (N_n) is 8000 since larger number of negative samples is desirable. In each boosting round, N_p and N_n are set to 10% of the total number of samples. Total number of patches (K) is 284 as it is generated using the SURF descriptor extraction procedure.

Number of boosting round in each state depends upon the total number of local patches and desired speed of evaluation. It is set to 5 in the first three stages and 10 in the next three stages. If the number of boosting round is less, then speed of evaluation for that stage increases, at the same time false positives in that stage also increase. So a balance is made between these two factors and the number of boosting rounds is selected appropriately. The main advantage of Adaboost algorithm is that concentration is given only to those image portions where there is a possibility of detecting a face. For getting advantage of this property, the value of T is made as minimum as possible in first 3 stages while the true-positive and false-positive rates are kept at least at 99% and 1% respectively. Generally it is observed that the false positive rate falls rapidly while increasing the number of boosting rounds. Experiments proved that for optimum true-positive and false-positive rates, six stages are ideal.

4. EVALUATION

The proposed face detection system is trained using the collected face and non-face images from various standard data sets and private photo collections. The learnt model, in the form of ensembles of weighted SVMs is used for evaluating face detection accuracy. For illustration purpose, here sample outputs use a rectangle around detected faces. For evaluation purpose two popular public datasets are used: Caltech [20] and UMass FDDB [21].

4.1 Caltech Dataset Evaluation

Caltech Dataset is a face dataset collected by Markus Weber at California Institute of Technology [20]. It contains 450 face images 27 unique people under different lighting, expressions and backgrounds. For evaluating the implemented system with this dataset, each image in the data set is given as input and output from the system is stored in a face annotated form of original images. Manual evaluation is also done based on these outputs. Some sample detections are given in Figure 5. Evaluation results are given in Table 1.

Figure 5. Sample detection results on Caltech Dataset

Table -1 The Evaluation Result of Face Detection using Caltech Dataset

Detection Type	Number of Detections	Percentage of detection (Total 450 images)
True Positive (TP)	393	87.33
False Positive (FP)	60	13.33
False Negative (FN)	57	12.67

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

From 450 faces, 393 faces were correctly classified. So True Positive (TP) is 393 and False Negative (FN) is 450 - 393 = 57. But 60 non-face detection windows were incorrectly classified as face. This yields False Positive (FP) as 60. From these values, precision and recall are calculated using the above equations as 0.868 and 0.873 respectively.

4.2 UMASS FDDB Evaluation

Face Detection Dataset and Benchmark (FDDB) is a benchmark for face detection algorithms [21]. They provide dataset containing 2845 images with a total of 5171 faces. A wide range of challenging scenarios including occlusions, difficult poses, and low resolution and out-of-focus faces are contained in this dataset. Figure 6 shows some detection on FDDB dataset.

Figure 6.Sample detection results on FDDB Dataset

For performance evaluation on FDDB dataset, an experiment with unrestricted training has been proposed [21] which has been adopted in this current work. In FDDB, algorithms are compared using Receiver Operating Characteristic (ROC) curves. Two types of curves are proposed which are Discrete Score (DS) Curve and Continuous Score (CS) Curve. In discrete score, a face is considered as correct if it spans over 50% of original elliptical annotation. In continuous score, detection score is same as amount of overlapping with original annotation.

In the benchmark, ROC curves for Viola-Jones detector [22] and C. Mikolajczyk's detector [23] are given for comparison purposes. In Mikolajczyk's detector, humans are modeled as flexible assemblies of parts, and part detection is done using Adaboost trained classifiers. Open CV implementation of the Viola-Jones detector is the other face detection method for

comparison. Figure 7 shows CS ROC Evaluation curve and DS ROC evaluation curve on FDDB dataset in which, X-axis denotes number of false positives and Y-axis denotes fraction of true positives. When false-positive number is allowed to increase, true positive rate also increases drastically. In DS evaluation, true positive rate approaches to 0.6 when false detections are 100. At the same time, other methods are only just around 0.3. Compared to DS evaluation, CS evaluation appears to have a lower performance. But this performance drop is common for all detectors and still proposed method outperforms the other compared methods.

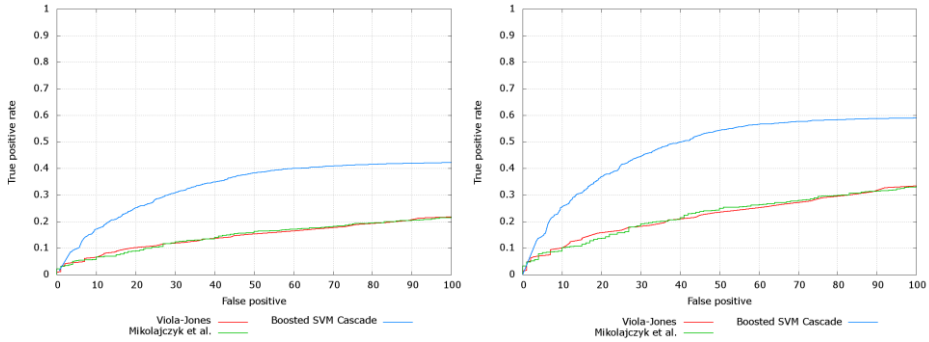

Figure 7.Evaluation Curves for FDDB Dataset

Thus, Boosted SVM Cascade outperforms Viola-Jones and Mikolajczyks face detectors both in DS evaluation and CS evaluation. The implemented method has more advantages over other methods. SURF descriptors are somewhat coarse feature descriptor compared with Haar-like features. So the number of SURF features is comparatively very less. This reduces the time complexity for selecting best feature in each round and thereby improving overall training efficiency. Also, SVM is used as the base classifier which increases the accuracy tremendously. Use of SVM also helps to reduce the number of training samples by fast convergence. The false positive rate is further reduced by scanning only skin-like regions through skin-color filtering, thus improving accuracy of boosted SVM cascade face detection system

Evaluation based on the detection rate was also performed, and the proposed method has been found superior to the state of the art face detectors. Table- 2 shows the detection rates on FDDB dataset compared with the existing recent classiers.

Table- 2: The Evaluation Result of Face Detection using FDDB Dataset

Detection Type	Percentage of detection (Total 5171 faces)
Local Binary patterns (LBP)	71.6
Modied Census Transform (MCT)	75.3
Semi-Local Binary Pattern (SLBP)	75.5
Semi-Local Modied Census Transform (SMCT)	76.9
Proposed Method	81.03

Figure 8 shows the evaluation on FDDB dataset graphically, which demonstrates that the proposed method outperforms existing state-of-the-art classifiers.

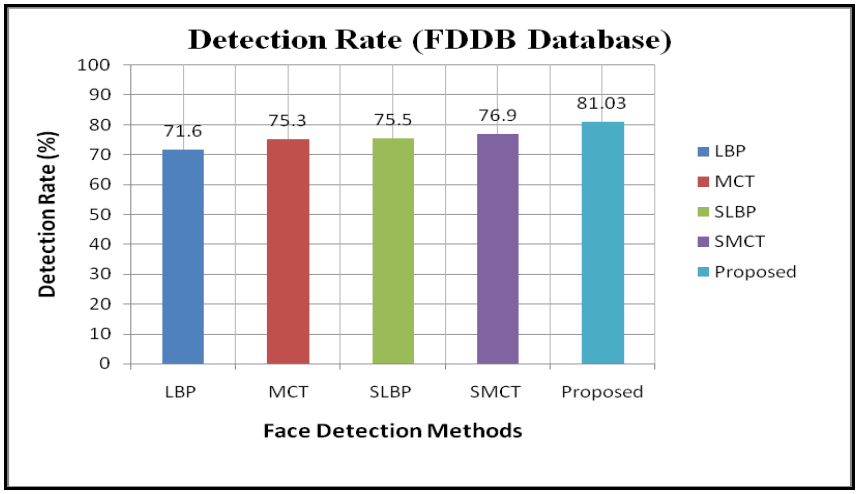

Figure 8. Performance evaluation on FDDB dataset, demonstrated graphically

The implemented method has more advantages over existing methods. SURF descriptors are somewhat coarse feature descriptor, so the number of SURF features needed is very less. This reduces the time complexity for selecting best feature in each round and thereby improving overall training efficiency. Also, SVM is used as the base classifier which increases the accuracy tremendously. Use of SVM also helps to reduce the number of training samples by fast convergence. The false positive rate is further reduced by scanning only skin-like regions through skin-color filtering, thus improving accuracy of boosted SVM cascade face detection system.

5. CONCLUSION

This research work aimed to develop a robust method for the face detection system, learnt through machine learning. Rule based Skin color filtering is used to filter out non-face regions before processing. SURF descriptor is used as the feature vector. A variation of Adaboost algorithm is used for boosting a number of SVM classifiers in different stages. Connecting all stages in a cascade manner completed the proposed face detection system. Boosted SVM Cascade based face detection system has been implemented, trained and evaluated. Performance evaluation has been done on Caltech [20] and FDDB [21] datasets. From Caltech evaluation, it is understood that this face detection system is good at detecting upright front faces. Evaluation on a more difficult dataset from FDDB showed that, this method outperforms certain popular approaches for face detection like that of Viola-Jones. Many future improvements are possible as an extension to this research work. Color information can be coded in the SURF descriptor itself, instead of using skin color as a pre-filter. Extended SURF descriptors [10] can be used with this cascade classifier system for more representational ability. Number of features can be increased by varying local patch ratios and movement inside a detection window. This concept can be extended to build a multi-view object recognition system.

6. REFERENCES

[1] D. J, Kriegman., M.-H, Yang., N, Ahuja.: 'Detecting faces in images: A survey', IEEE Trans. on PAMI., 2002,v24, (1), pp. 34-58

[2] Paul, Viola., Michael, Jones.: 'Rapid object detection using a boosted cascade of simple features', Proc. Computer Vision and Pattern Recognition, Hawaii, USA, Dec 2001. pp. I–511.

[3] Cha, Zhang., Zhengyou, Zhang.: 'A survey of recent advances in face detection', Technical Report, MSR-TR-2010-66, 2010.

[4] Wang, Peng., Qiang, Ji.: 'Robust face tracking via collaboration of generic and specific models', Image Processing, IEEE Transactions on,2008, 17,(7) ,pp. 1189-1199.

[5] Liao, Shengcai., Anil K, Jain., Stan Z, Li.: 'A fast and accurate unconstrained face detector', IEEE transactions on pattern analysis and machine intelligence, 2016, 38, (2), pp. 211-223.

[6] Wang, Lei., Yixiong, Liang., Wangyang, Cai., Beiji, Zou.: 'Failure Detection and Correction for Appearance Based Facial Tracking', Chinese Journal of Electronics , 2015, 24, (1), pp. 20-25.

[7] Gao, M.L., Li, L.L., Sun, X.M.; et al.:'Face tracking based on differential harmony search', IET Computer Vision, 2014, 9, (1), pp.98-109.

[8] Kawulok, Michal., Janusz, Szymanek.:'Precise multi-level face detector for advanced analysis of facial images.', Image Processing, IET, 2012, 6, (2), pp. 95-103.

[9] Zou, Wilman W., Pong C, Yuen., Rama, Chellappa.: 'Low-resolution face tracker robust to illumination variations.', Image Processing, IEEE Transactions on, 2013, 22, (5), pp. 1726-1739.

[10] H, Bay., A, Ess., T, Tuytelaars., et al.: 'Surf: Speeded up robust features', Computer Vision and Image Understanding (CVIU), 2008, 110,(3), pp. 346–359

[11] Jianguo, Li., Tao, Wang., Yimin, Zhang.: 'Face detection using surf cascade.', Computer Vision Workshops (ICCV Workshops), Proc. IEEE International Conference on, Barcelona, Spain, Nov 2011, pp. 2183-2190

[12] Donghoon Kim, Rozenn Dahyot, "Face Components Detection using SURF Descriptors and SVMs", Proc. International Machine vision and Image Processing Conference, IEEE, Coleraine, Northern Ireland, Sep 2008, pp. 51-56

[13] Yimin, Zhang., Jianguo, Li.: 'Learning surf cascade for fast and accurate object detection', Proc. Conference on Computer Vision and Pattern Recognition, IEEE, Portland, Oregon, Jun 2013, pp. 3468-3475

[14] Sayantan, Thakur., Sayantanu, Paul., Ankur, Mondal., et al.: 'Face Detection Using Skin Tone Segmentation', Proc. World Congress on Information and Communication Technologies, IEEE, Mumbai, India ,Dec 2011, pp. 53-60

[15] Corinna, Cortes., Vladimir, Vapnik.: 'Support-vector networks', Machine learning, 1995, 20, (3), pp. 273–297.

[16] Padma Polash, Paul., Marina, Gavrilova.:'Pca based geometric modeling for automatic face detection', Proc. Computational Science and Its Applications (ICCSA) International Conference on, IEEE, Santander, Spain, Jun 2011, pp. 33–38.

[17] 'The MPLab GENKI Database', http://mplab.ucsd.edu, accessed 1 December 2016

[18] [18] N, Kumar., P, Belhumeur., S. K, Nayar.: ' Facetracer: A search engine for large collections of images with faces', European conference on computer vision , Marseille, France, Oct 2008, pp. 340-353

[19] 'UMD negative images', http://tutorial-haartraining. googlecode.com/svn/trunk/data/negatives/, accessed 1 Jun 2013

[20] 'Caltech Face Dataset', http://www.vision.caltech.edu/html-files/archive.html, accessed 1 December 2016

[21] Vidit, Jain., Erik, Learned-Miller.: 'Fddb: A benchmark for face detection in unconstrained settings', UMass Amherst Technical Report, Tech. Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010, pp. 1-7

[22] Paul, Viola., Michael J, Jones.: 'Robust real-time face detection', International journal of computer vision., 2004, 57, (2), pp. 137–154.

[23] Krystian, Mikolajczyk., Cordelia, Schmid., Andrew, Zisserman.: 'Human detection based on a probabilistic assembly of robust part detectors', Proc. Computer Vision-ECCV, Prague, Czech Republic, May 2004,pp. 69–82.