

# Enhanced Face Detection Based on Haar-Like and MB-LBP Features

Tarek Dandashy<sup>1</sup>, Moustapha El Hassan<sup>2</sup> and Amine Bitar<sup>3</sup>

<sup>1</sup>M.Sc Student, Computer Science Department, University of Balamand, Koura, LEBANON

<sup>2</sup>Associate Professor, Electrical Engineering Department, University of Balamand, Koura, LEBANON

<sup>3</sup>Assistant Professor, Computer Science Department, University of Balamand, Koura, LEBANON

<sup>1</sup>Corresponding Author: tarek.dandashy@std.balamand.edu.lb

## ABSTRACT

The effective real-time face detection framework proposed by Viola and Jones gained much popularity due to its computational efficiency and its simplicity. A notable variant replaces the original Haar-like features with MB-LBP (Multi-Block Local Binary Pattern) which are defined by the local binary pattern operator, both detector types are integrated into the OpenCV library. However, each descriptor and its evaluation method has its own set of strengths and setbacks. In this paper, an enhanced two-layer face detector composed of both Haar-like and MB-LBP features is presented. Haar-like features are employed as a coarse filter but with a new evaluation involving dual threshold. The already established MB-LBPs are arranged as the fine filter of the detector. The Gentle AdaBoost learning algorithm is deployed for the training of the proposed detector to reach the classification and performance potential. Experiments show that in the early stages of classification, Haar features with dual threshold are more discriminative than MB-LBP and original Haar-like features with respect to number of features required and computation. Benchmarking the proposed detector demonstrate overall 12% higher detection rate at 17% false alarm over using MB-LBP features singly while performing with  $\times 3$  speedup.

**Keywords**— Face Detection, Machine Learning, Boosting, Real-Time Systems

## I. INTRODUCTION

In the area of computer vision, object detection is particularly important and in widespread utilization. Face detection is a fundamental case of object detection and is required as the primary module in face and gesture recognition systems, tracking and more. Due to its potential importance, it is a hot topic in computer vision and is under extensive research with many proposed approaches and their variants that have shown increasingly better performance. The factors which make this task non-trivial are due to a large variance of face instances found in real images, which are attributed to human face, scale, position, orientation, pose, lighting, shadowing, occlusion, expression, image quality, background clutter and color. The main issues addressed are the detection ability and computational density which limit usability.

A notable advance in this area was introduced by

the influential framework of Viola & Jones [1] and is the basis framework of the work in this paper. Many variants of the Viola and Jones framework have been proposed, notably the variant that implements Multi-Block Local Binary Pattern (MB-LBP) features [2] is better suited to the problem and is found in real applications.

Although both MB-LBP and Haar-like features have been shown to be somewhat effective, their dependence on a single threshold model is not best suited to summarize the leading image content. Typically, almost all the processing is spent on rejecting candidate image regions, thus the extended utilization of feature extracted data can be valuable to quickly reject non-promising regions. A two-layer face detector is proposed, which implements Haar-like and MB-LBP features in each layer respectively. A new evaluation for Haar-like features is defined by deploying dual thresholds and composes the first layer. The second layer implements the well-known MB-LBP features to achieve efficient face and object detection.

The rest of the paper is structured as follows. In Section II, an overview of the Related Work is summarized. Next, the proposed (Dual-Threshold Evaluation) and MB-LBP features are defined and illustrated in Section III, showing their potential advantages. Section IV, the (Classifier Construction) of the proposed method for the selected features is explained. For the validation of this study, Experiments are carried out and the Conclusion is presented in Section V and Section VI, respectively.

## II. RELATED WORK

Face detection is a fundamental and classical problem in computer vision. The pioneering work of Viola and Jones [1] enabled unprecedented advancement into providing a real time and effective solution framework. They introduced the combined use of AdaBoost [3] machine learning, simple Haar-like features arranged in an attention cascade and the integral image representation to enable fast calculation. Along this axis, many other works have provided better results by developing different kinds of features or different machine learning algorithms.

Other methods have been developed to assist in this axis, improving detection ability and decreasing computation. Skin color detection [4, 5, 6] is selected to

identify potential regions and thus to prune much of the image while also decreasing false positives. Multi-view detection [7, 8] is enabled by the divide and conquer strategy, where a set of face detectors are trained on separate facial pose images. The detectors are the run concurrently or in multi resolution on images to detect faces of various poses. Further improvements involve face alignment and detection jointly in a single cascade where the face pose is progressively estimated via boosted regression as in [9, 10].

The very simple Haar-like feature was quick to be replaced by more complex features in subsequent works. The feature was found to be weak in discrimination and resulted in a suboptimal detector [2]. Due to its simple structure and use of single threshold it is weak in discriminating the distribution of binary class data [11]. Succeeding features have replaced Haar-like with improved results like MB-LBP [2] which has a more informative structure and is defined by a look up table (LUT) to encode its 256 different values along with their learned classification values.

An efficient multi-threshold AdaBoost approach to detecting faces in images using Haar features was presented [12]. The method is a multi-threshold weak classifier, constructed using an intelligent method of finding thresholds based on points of optimization. The Kadane algorithm is exploited to solve the optimization problem and has similar time complexity  $O(n)$  to that of training a single-threshold weak classifier. The boundary thresholds of both positive and negative ranges in the feature space are found. Each of which is represented mostly of either positive or negative training samples. The final feature's dual threshold is derived from these four boundary values.

By using multiple feature extractors, benefits of each feature type improves overall performance. Here [13] faces are detected using Viola and Jones method then a Shi-Tomasi detector finds potential corner points of eyes and lastly k-means allows clustering of neighbor corner points to determine eye regions. Infrared imaging takes advantage of the generally constant distribution of face temperatures to achieve more reliable detection. Work [14] proposed the application of AdaBoost to a mixture of local features like Haar-like, MB-LBP and (Histogram of Oriented Gradient) HOG to detect faces captured in infrared cameras. MB-LBP was extended by fitting a margin around the reference giving better noise immunity. In another method [15], MB-LBP and linear Support Vector Machine (SVM) was applied to gender classification. Different SVM learning models were used to process and analyze the results, which outperformed MB-LBP implemented with Radial Basis Function (RBF).

A simple feature named Normalized Pixel Difference (NPD) was introduced [16] for face detection. It is defined as the difference to sum ratio of two pixel values. A deep quadratic tree is applied to learn an optimal set of NPD features which represent complex face manifolds. A multi-view face detector based on a

cascaded classifier that is supported by Convolutional Neural Network (CNN) is presented by [17]. The CNN is deployed to filter out false positives and perform pose estimation. The arrangement allowed the system to maintain a high speed despite the more complex CNN.

A funnel structured cascaded multi-view face detector [18] consists of three cascade classifiers. Multiple fast Locally Assembled Binary (LAB) classifiers, a coarse Multilayer Perceptron (MLP) classifier and lastly a fine MLP classifier allowed detection refinement at a low cost.

### III. DUAL-THRESHOLD

The original Haar-like feature found in the Viola and Jones face detector is composed of a few rectangles, where each rectangle represents the average intensity of the area it is placed over in an image. The Haar-like feature set is composed of three different structures of either two, three or four equally sized rectangles placed in direct proximity of each other. The features can then be rotated in right angles, displaced and resized to form a large usable set. The calculation of a Haar-like feature output is performed by mathematical addition and subtraction, where the rectangles' average intensity values are added or subtracted from one another in a specified meaningful arrangement. By selecting the position, dimension, size and arrangement of a feature, it is capable of concluding large and small scale intensity differences at any location and of several orientations within an image [1]. Using the integral image representation, Viola and Jones were able to evaluate any of these feature sub-types at any size, dimension and position in constant time.

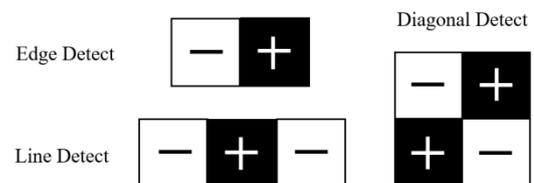
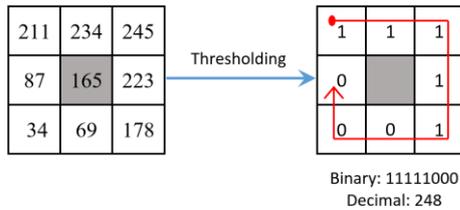


Figure 1: Haar-Like Features Basis Set

On the other hand, MB-LBP features are composed by a singular  $3 \times 3$  rectangle structure of any integer pixel ratio and scale, which is better suited for modeling complex image structure over the simpler Haar-like features consisting of between two to four rectangles [2]. The LBP operator is then applied to encode the data and produce the output, which adds to their advantages and simplicity by the threshold of each of eight outer rectangles to the center rectangle and assigning an eight-bit code as the output. Due to the dependence of MB-LBP on rectangular features, it also benefits from the integral image representation for the fast evaluation of features in constant time.



**Figure 2: MB-LBP Feature and Operator**

The exhaustive set of MB-LBP features that can be enumerated in a window of size 20x20 pixels is 2049, which is 1/20 compared to the number of instances of Haar-like features amounting to 45891. It is also to be noted that the output value of a MB-LBP feature is an encoded value or non-metric and does not directly represent any intensity magnitude information from the image structures. The 8-bit code resulting from the evaluation of each MB-LBP feature is accepted or rejected by the learned binary classification model using a 256 bin LUT. In contrast, the output of the Haar-like features is an integer value that represents actual average intensity difference values calculated from the image structures, which is then classified based on learned threshold values in the classification model. The properties of MB-LBP result in the use of fewer features, consequently lowering computational requirements and significant complexity reduction in the training phase to match the effectiveness of a Haar based classifier [2]. However, this study shows that during the early stages of classification, Haar features are more effective than and can supplement MB-LBP features. In contrast, the later stages of classification are more suitably performed by MB-LBP features.

As MB-LBP features output a binary code representing image structure and Haar-like features use a single threshold, both cases disregard effective exploitation of intensity magnitude information from the appearance of structures in image content. A typical object class, under different lighting scenarios, should maintain regular intensity magnitude relationships within the appearance of its dominant structures, which can be machine learned. However, to enable the utility of such information, the challenges of face detection (Introduction) must be taken into consideration.

In this method, a new evaluation for Haar-like features using dual thresholds is introduced. By using the same types and structure of the original Haar-like feature but with a dual as opposed to single threshold evaluation, the descriptor is more naturally suited to model image structures for the binary classification problem. The incorporation of Haar-like features with their propose devaluation model is complemented with the use of MB-LBP features in a two-stage method, respectively. The Haar-like feature based classifier is used as an effective coarse filter and is limited to just that, subsequently the MB-LBP feature based classifier is applied and acts as a fine filter to achieve the objective of enhanced face detection.

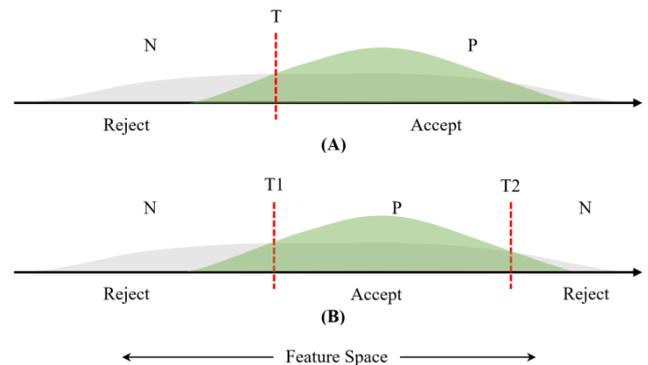
#### IV. CLASSIFIER CONSTRUCTION

In order to construct a classifier for the new Haar-like features, the GentleAda Boost algorithm [19] is adopted. The boosting algorithm solves the problems of: (1) Identifying the most effective features from the entire feature set, (2) Constructing weak classifiers by learning the most effective feature's thresholds, (3) Boosting the weak classifiers to form a strong classifier by cascading and learning the stage thresholds. For the learning of each weak classifier, an optimal threshold classification function is used to determine the optimum threshold for the evaluation of the corresponding feature.

The success of the viola and jones approach is presented by the dependence on simple rectangular Haar-like features, integral image for efficient feature computation, cascade for efficiency and the training algorithm that is able to construct a cascaded strong classifier from hundreds of these features. The integral image S allows fast computation of pixel sums within any rectangular area of an image I in constant time. The integral image representation requires one pass over all the pixels in an image and is calculated using  $S(x,y) = \sum_{x' \leq x, y' \leq y} I(x',y')$  where  $(x,y)$  and  $S(x',y')$  denote pixel locations. Once the integral image is calculated, any rectangular feature ABCD can be computed in four array access and 3 additions as  $\sum_{(x,y) \in ABCD} i(x,y) = ii(D) + ii(A) - ii(B) - ii(C)$ .

Since there is a very large number of possible features in a small window size of 24x24, only the ones that present highest discriminating abilities are selected to form the strong classifier. Each stage of the final cascade classifier or strong classifier is composed of several features or weak classifiers and constructed using the AdaBoost algorithm.

As seen in Fig. 3., deployment of two thresholds more closely bounds the target distribution and thus filters out more negative samples. Using a single threshold presents inefficient classification of available data and results in a weak fit for the distribution of a sample in a feature's space. More appropriately, identifying two thresholds to bound and discriminate the target distribution lump presents a more data efficient arrangement.



**Figure 3: Feature Space with One (A) Or Two Thresholds (B) To Fit a Distribution**

Fig. 3. shows the distribution of positive (P) and negative (N) samples in an effective feature's space. Each of the P and N graphs are a probability distribution having area equal 1. Typically, in Fig. 3. (A), a single threshold  $\theta_j$  dissects the range into two parts. The left sub-range represents a small portion of the positive samples but a large portion of the negative samples. The second range represents a large portion of positive samples and the remaining portion of the negative samples. A feature is effective when its space can be threshold such that a portion of the negative samples can be separated from a non-equal portion of the positive samples. In other words, an effective feature should be able to significantly increase the ratio of negative to positive samples or vice versa. What is not useful is when a feature maintains the ratio of positive to negative samples, hence it is non-discriminatory.

Although the probability graphs of a large number of data samples over each distinct feature have large variations, Fig. 3. provides a usable overall visualization. The use of dual threshold to bound the positive samples is shown in Fig. 3.(B). It is evident from the graph that a better fit for the samples is in effect. Threshold T2 enables further exclusion of bulk negatives. To define the Haar feature with dual threshold, let  $x$  be a training instance,  $j$  be the  $j^{th}$  feature of the feature set and  $p_j = \{+1, -1\}$  be the parity of the inequality. Then  $f_j(x)$  is the raw output of the feature as evaluated on a training instance,  $\theta_{j1}$  and  $\theta_{j2}$  are the thresholds of this particular feature and  $h_j(x)$  is the confidence output of this transaction giving binary value of either 0 or 1.

$$h_j(x) = \begin{cases} 1, & p_j f_j(x) < p_j \theta_{j2} \\ 1, & p_j f_j(x) > p_j \theta_{j1} \\ 0, & otherwise \end{cases} \quad (1)$$

By employing the parity variable  $p_j \{-1, +1\}$  of (1), the evaluation becomes simpler. It is sufficient to evaluate just one of the two equations. Taking the possible cases for the placement of the positive and negative range can be seen in Fig. 4. For dual thresholds where  $\theta_1 < \theta_2$ . In the case of (Fig. 4A) the parity is set at +1 and for the case of (Fig. 4B) the parity is set at -1.

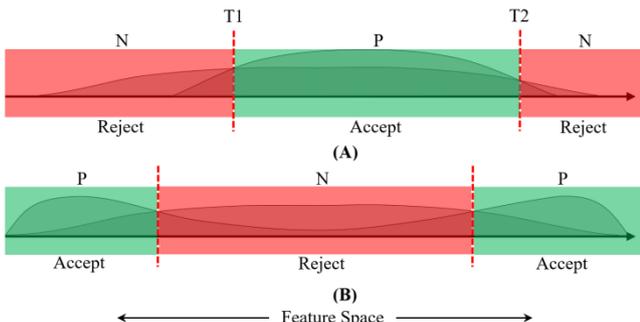


Figure 4: Range Selection in Feature Space Using Two Thresholds

By splitting the feature space of a Haar feature  $h_j$  into two parts using one or two thresholds  $\theta_{j1}$  and  $\theta_{j2}$  as shown in Fig. 4(B), the training sample will also be partitioned into two subsets. The formulated weighted error which is related to the target sample categories, enables the determination of optimal thresholds. Let  $T^-$  equal the total weights of the negative instances and  $T^+$  equal the total weight of the positive instances in the training set. The weight sum of the negative instances that fall outside the threshold range  $\theta_{j1}$  and  $\theta_{j2}$  is denoted as  $S^-$ . The weight sum of the positive instances that fall outside the same range  $\theta_{j1}$  and  $\theta_{j2}$  denoted as  $S^+$ . The minimum weighted error can then be calculated by finding the thresholds resulting in a minimum weight sum of all incorrectly classified samples (positives and negatives). Depending on the parity or whether the inner range represents the positive confidence or a negative one, the minimum weighted error function can be summarized in (2).

$$e = \min(S_j^+ + (T_j^- - S_j^-), S_j^- + (T_j^+ - S_j^+)) \quad (2)$$

For each Haar feature, the return value evaluated over each training sample is collected then sorted in increasing order, similarly to the case of building single threshold classifiers. The list now consists of the sorted feature return values corresponding to the weight  $w$  and classification label  $(w_k, y_k), K = 1, 2, \dots, N$ , of the training sample instances of either target or non-target. The dual optimal thresholds for each feature are found successively by applying the minimum weighted error function over the entire sample distribution in the feature's space. The first optimal threshold is deduced by applying the minimum weighted error function, then applying it again over the resulting range from the first round to derive the second optimal threshold. The time complexity of this dual optimal threshold discovery is upper bounded by  $O(2n)$ , while the discovery of a single threshold is exactly  $O(n)$ .

Subsequently, the gentle AdaBoost algorithm is chosen over other variants to build the strong classifier due to its simple implementation and numerical robustness. In each iteration of the Gentle AdaBoost algorithm, the strong classifier is constructed by exhaustively identifying and adding a new optimized weak classifier which presents the lowest weighted error. Thereafter, the weights of previously misclassified training instances are updated such that they now have higher weight. In the next iteration, those instances will acquire more attention by the new weak classifier. After many AdaBoost iterations, the final and complete strong classifier is built and optimized from the utilization of many weak classifiers. The algorithm can be seen in detail below.

**Algorithm 1** Gentle AdaBoost

1. Start with weight  $w_i = \frac{1}{N}, i = 1, 2, \dots, N, F(x) = 0$

2. Repeat for  $m = 1, \dots, M$  (Hypothesis)
  - a. Fit the regression function by weighted least squares fitting  $Y$  to  $X$  using:

$$J_{wse} = \sum_{i=1}^N w_i (y_i - f_m(x_i))^2$$

- b. Update  $F(x) \leftarrow F(x) + f_m(x)$
- c. Update  $w_i \leftarrow w_i e^{-\gamma f_m(x_i)}$  and normalization

The weighted least squares error function of Algorithm 1 above, calculates the performance of a weak classifier over the training sample. In each iteration of the Gentle AdaBoost algorithm, the weak classifier resulting in the lowest error is selected and combined into the strong classifier. Given a weak classifier  $f(x)$ , the weighted least squares function is calculated as the sum of the error on each of the instances  $i$  in the training sample with respect to a specific weak classifier. The error is calculated as the product of the instance weight by the square of label minus confidence value for that instance, as seen above.

During the training of stage classifiers in a cascade, each stage is set to have higher than 99% acceptance of positives and rejection of at least 50% of negatives. This is often required in order to converge to a strong classifier with good detection ability within 20 to 40 stages.

## V. EXPERIMENTS

To evaluate the characteristics and performance advantage of the proposed Boosted based (Dual Threshold Haar and Multi-Block Local Binary Pattern) face detector, two experiments are carried out. Firstly, for finding a good optimization for the joining of the two cascades in the proposed method. Secondly, to benchmark the proposed method in order to compare its performance.

For the experiments, two face detector cascades are required, the authors prepared 5,672 positive images each with 24x24 pixel size and 8,255 negative images for training. The positive and negative training samples are derived from multiple sources including internet and face detection databases. The positive images selected are then randomly transformed by shifting, scaling, rotating and mirroring to generate a total of 28,360 positive training instances.

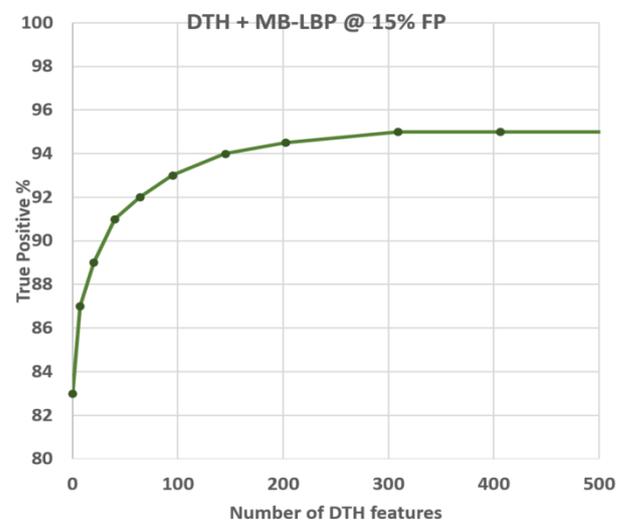
The first strong classifier is based on DT-HF features and is generated using the gentle AdaBoost algorithm. A second strong classifier is based on MB-LBP features is also generated using the gentle AdaBoost algorithm. During the training procedure, the construction and optimization of each stage is required to provide no less than 99.5% pass through for positives and no more than 50% pass through for negatives of the training sample. The generated DT-HF strong classifier consists

of 2217 Haar features arranged in 20 stages. In contrast, the original Viola and Jones detector required 4297 Haar features arranged in 32 stages. Furthermore, the generated MB-LBP strong classifier consists of 20 stages of 156 features.

After the strong classifiers are built and optimized, a benchmark over a popular face detection database is carried out to compare their performance. The MIT+CMU face detection database is selected for carrying out the performance benchmark due to its popularity and widespread usage. The image database consists of 130 greyscale images of various sizes containing a total of 507 upright faces for an average of four faces per image.

### 5.1 Experiment 1

In this approach, two independent face detectors based on DT-HF and MB-LBP are built and joined. The choice of optimized joining of these detectors depends on how many stages of the DT-HF classifier should be elected and what remains is disregarded. The MB-LBP classifier is then appended to result in two serial classifiers acting as one stronger classifier.



**Figure 5: Performance Improvement of Proposed Classifier by Increasing the Number of DT-HF Weak Classifiers**

Fig. 5 summarizes the detection rate with false positives anchored at 15%, while increasing the number of DT-HF weak classifiers pretended to the MB-LBP classifier. Exclusively, the MB-LBP classifier provides a detection rate of 82% but that figure increases as DT-HF classifiers are added. By adding just 40 DT-HF weak classifiers or features, the detection rate increases to 91% and 145 weak classifiers results in 94%. Eventually the detection rate saturates for the proposed method at 95% using 309 features. In the final testing version, 202 DT-HF weak classifiers have been included, or the first 7 stages of the DT-HF strong classifier. This has shown optimum performance with respect to speed and detection accuracy. This choice can also be suitably justified in Fig. 5.

5.2 Experiment 2

After optimizing the proposed classifier as seen in experiment 1, a benchmark is required to evaluate its performance with respect to the existing methods. The experimental results in Fig. 6 show the performance of the proposed detector (DT-HF + MB-LBP) and of the MB-LBP, DT-HF and Haar detectors. It is observed that the proposed detector is able to provide good detection ability, better than purely MB-LBP based.

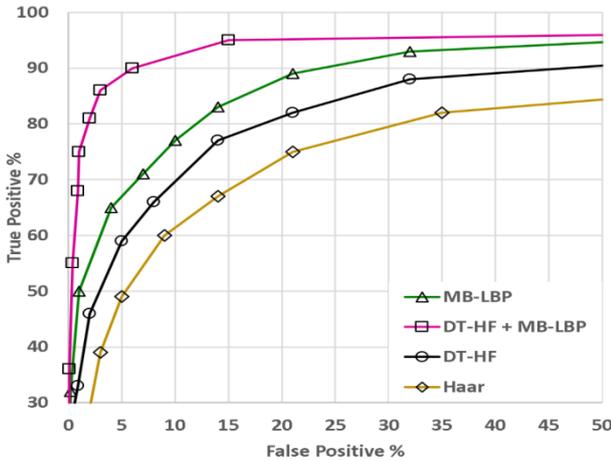


Figure 6: Performance Comparison of Classifiers Based On MB-LBP, Haar and DT-H Features Vs. Proposed Method Over MIT+CMU Dataset

The MB-LBP and proposed detector approach similar 95% detection rate limit as their sensitivity is increased. However, as sensitivity is increased to yield beyond 90% TP, the harboured false positives become inordinate for typical usage. It is also noticed that a classifier utilizing only DT-HF out performs the original Haar classifier but falls slightly short of the MB-LBP classifier. When detection rate is selected at 90%, the proposed detector returns only 7% FP while MB-LBP returns 24%. Thus the proposed method is able to reduce FP by 17% over MB-LBP, resulting in just 7% FP when detection rate is chosen at 90%. Roughly 90% detection rate shows a reasonable trade-off between TP and FP. These values can be traced in Fig. 6.

5.3 Discussion of Results

In the detection performance comparison of Fig.6, the proposed detector is served by both DT-HF and MB-LBP features and consistently presents a lower false positive rate indicating its ability to better discriminate the target object class from background regions. Eventually, both classifiers converge to the same functional performance when detection rate is required above 95%. At that point onwards, the false positives become quite high at 50% and continue to increase while detection rate is unable to increase. This effect is probably due to the reduced ability of the classifiers to discriminate small portion of the faces present from background regions in the test dataset.

It should also be noted that both classifiers were

built and optimized separately for sake of simplicity, resulting in a slightly sub-optimized DT-HF and MB-LBP mixture classifier. The two level coarse-fine classifier is simply a concatenation of a subset of the DT-HF classifier and the complete MB-LBP classifier and serves as proof of concept.

However, the proposed detector shows the potential advantages of using these two types of features together. Each feature type is able to extract different and relevant information from structures in an image, providing better results when used together as compared to using either singly. DT-HF is able to sample intensity differences which are then compared to the learned or expected range. MB-LBP is able to sample minute intensity patterns of a rectangle to its surrounding space which is also compared to learned patterns.

The detection speed of the classifiers also shows significant improvement to the favor of the proposed method. The testing was performed on an intel i7 mobile processor of the 4<sup>th</sup> generation utilizing a single core with 16gb RAM onboard. The program was written in C++ and run in Visual Studio 2015. The testing dataset was also the MIT+CMU benchmark, in which the proposed detector performed a better detection job while spending roughly one third of the time of MB-LBP as seen in Table 1. The MB-LBP detector required 28 seconds to detect 83% of the faces while the proposed method was able to detect 95% of the faces and requiring only 9 seconds. The major speed improvement indicates that the operation of DT-HF features enabled higher utilization of image data with respect to computation required, effectively used for the target discriminatory function.

TABLE 1  
TIME REQUIREMENT COMPARISON OF CLASSIFIERS OVER MIT+CMU DATASET

Method	True Detections	False Detections	Time Required
MB-LBP	83%	15%	28 seconds
Proposed	95%	15%	9 seconds
DT-HF	77%	15%	89 seconds
Haar	68%	15%	173 seconds

In other words, DT-HF is able to better discriminate image data while not introducing complex computation requirements, in fact a single DT-HF is simpler in computation than a MB-LBP feature. But abstracted data of a single DT-HF is more oriented towards a qualitative intensity change and less towards quantitative intensity changes, contrary to MB-LBP. It is these differences in the utilized features and their placement in earlier or later stages that allow them to perform optimally.

Functionally, DT-HF carries out comparison

between two to four areas, but MB-LBP always carries out comparison between eight areas. Subsequently, DT-HF retrieves the value of the differences using subtraction while MB-LBP uses Boolean comparison. Almost all of the processing time is spent in the rejection of background regions, DT-HF features are able to reject them more effectively by using different data metrics as compared to metrics extracted by MB-LBP. It can be deduced that DT-HF features placed in the early stages of classification allow for much efficient data processing and reduced computation, which resulted in three times speed up.

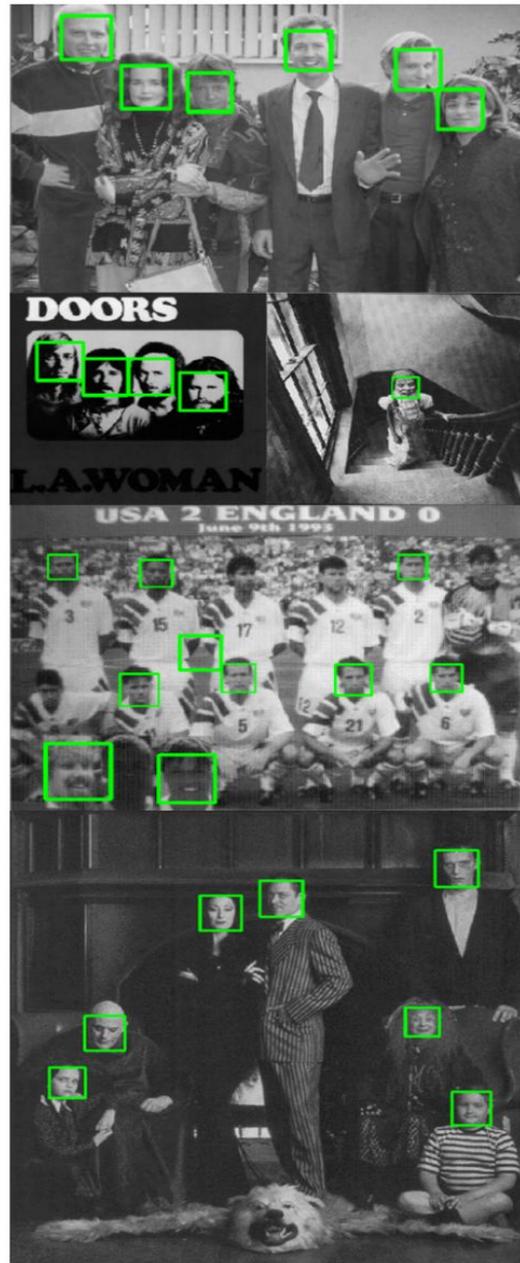
**TABLE 2**  
**RECALL, PRECISION AND F-SCORE COMPARISON OF CLASSIFIERS OVER MIT+CMU DATASET**

@ 15% FP	Recall	Precision	F-score
<b>MB-LBP</b>	0.83	0.85	0.84
<b>Proposed</b>	0.95	0.86	0.90
<b>DT-HF</b>	0.77	0.84	0.80
<b>Haar</b>	0.68	0.82	0.74

Taking a look at the recall (R), precision (P) and the overall system accuracy or F-score (F) evaluation of the detectors (Table 2), using their respective formal equations in (3).

$$R = \frac{TP}{TP + FN}, \quad P = \frac{TP}{TP + FP}, \quad F = 2x \frac{P \times R}{P + R} \quad (3)$$

The proposed method presents a 0.90 F-score which is significantly higher than the 0.84 of MB-LBP. The higher score is attributed to the recall scores of 0.95 and 0.83 respectively, while the precision scores are similar. The recall metric is a gauge produced as a fraction of the true positives to the real positives. Precision metric is a gauge of the detection error and since the scores are reported at a fixed 15% for the detectors, it is unsurprising that the precision is similar at 0.85 and 0.86 respectively. It is calculated as the portion of the detections that are true positives.



**Figure 7: Some Detection Results of the Proposed Method**

Finally, detections from sample images in the MIT+CMU image database by the proposed method is shown in Fig. 7. It is noticed that the proposed method reduces false positives and increases true positives in most of these examples.

## VI. CONCLUSION

This paper explores dual threshold evaluation for Haar features (DT-HF) alongside MB-LBP features for detecting faces in images. Each of these features presents differences along with their strengths and shortcomings that prompt their combined deployment for the face detection task. In the proposed approach, Haar features

are utilized using two thresholds as opposed to one in their output evaluation. Using two thresholds enables more efficient classification of available data and results in higher discrimination. The DT-HF and MB-LBP feature cascades are boosted independently then combined roughly optimally such that the latter would replace a portion of the former, behaving as a coarse to fine filter. Experimental results on public datasets like MIT+CMU reveal that the use of the proposed feature types and feature evaluation method enables the composition of a high performance face detector. Mainly, the detector is capable of lowering false positives by 17% while maintaining high detection rate at 90% and with a three times speedup, over the dependence on either these features singly. Future work will include more optimized learning of a mixture of these and possibly new features, also using neural network models to carry out complimentary tasks.

## REFERENCES

- [1] Viola, P. & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2), 137-154.
- [2] Zhang, L., Chu, R., & Xiang, S., et al. (2007). Face detection based on multi-block lbp representation. *In Proc. International Conference on Biometrics*, pp. 11-18.
- [3] Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119-139.
- [4] Amjad, A., Griffiths, A., & Patwary, M. N. (2012). Multiple face detection algorithm using colour skin modeling. *IET Image Processing*, 6(8), 1093-1101.
- [5] Ban, Y., Kim, S. K., & Kim, S., et al. (2014). Face detection based on skin color likelihood. *Pattern Recognition*, 47(4), 1573-1585.
- [6] Dhivakar, B., Sridevi, C., & Selvakumar, S., et al. (2015). Face detection and recognition using skin color. *In Proc. Third International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1-7.
- [7] Wu, B., Ai, H., & Huang, C., et al. (2004). Fast rotation invariant multi-view face detection based on real adaboost. *In Proc. Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 79-84.
- [8] Huang, C., Ai, H., & Li, Y., et al. (2007). High-performance rotation invariant multi view face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4), 671-686.
- [9] Chen, D., Ren, S., & Wei, Y., et al. (2014). Joint cascade face detection and alignment. *In Proc. European Conference on Computer Vision*, pp. 109-122.
- [10] Ren, S., Cao, X., & Wei, Y., et al. (2014). Face alignment at 3000 fps via regressing local binary features. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1685-1692.
- [11] Overett, G. & Petersson, L. (2007). Improved response modelling on weak classifiers for boosting. *In Proc. IEEE International Conference on Robotics and Automation*, pp. 3799-3804.
- [12] Li, Z., Wang, W., & Liu, X., et al. (2015). An efficient multi-threshold adaboost approach to detecting faces in images. *Multimedia Tools and Applications*, 74(3), 885-901.
- [13] El Kaddouhi, S., Saaidi A., & Abarkan, M. (2017). Eye detection based on the viola-jones method and corners points. *Multimedia Tools and Applications*, 76(21), 23077-23097.
- [14] Ma, C., Trung, N. T., & Uchiyama, H., et al. (2017). Adapting local features for face detection in thermal image. *Sensors*, 17(12), 2741.
- [15] Tianyu, L., Fei, L., & Rui, W. (2018). Human face gender identification system based on MB-LBP. *In Proc. Chinese Control and Decision Conference (CCDC)*, pp. 1721-1725.
- [16] Liao, S., Jain, A. K., & Li, S. Z. (2016). A fast and accurate unconstrained face detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 211-223.
- [17] Zhang, C. & Zhang, Z. (2014). Improving multiview face detection with multi-task deep convolutional neural networks. *In Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1036-1041.
- [18] Wu, S., Kan, M., & He, Z. (2017). Funnel-structured cascade for multi-view face detection with alignment-awareness. *Neurocomputing*, 221, 138-145.
- [19] Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2), 337-407.