Face detection and tracking using hybrid margin-based ROI techniques

Bacha Rehman[1], Ong Wee Hong[1], Abby Tan Chee Hong[1] , Trung Dung Ngo[2]

Abstract

This study is to solve the problem of low accuracy and slow processing speed for real-time face detection and tracking systems. A margin-based region of interest (MROI) approach with fixed and dynamic margin concepts is proposed to speed up the processing time. In addition, a hybrid system is developed to boost the accuracy and overcome the deficiency of the main detection algorithm. This approach consists of two routines, i.e. main and escape routines. Three algorithms are used independently as the main routine to evaluate the effectiveness of the proposed hybrid approach. These algorithms are Haar cascade, Joint cascade, and Multi-task Convolutional Neural Networks (MTCNN). The escape routine based on template matching (TM) algorithm is designed to evaluate the effectiveness of the proposed hybrid approach and improve detection accuracy. Two RGB video datasets with diversity and variations in face poses, video backgrounds, illuminations, video resolutions, expressions, over exposed faces, and occlusions of people within various unseen environments have been used for experiments and evaluation. The experiment results confirm that the hybrid approach is capable of detecting and tracking faces in non-frontal orientation with better accuracy and faster processing speed, i.e. four times faster than the conventional full frame scanning techniques.

Keywords Face detection, Joint-cascade, Convolutional neural network, Haar cascade, Template matching, Region of interest, Hybrid model, Dynamic margin, Face tracking, Processing time

1. Introduction

Face detection is a vital research field in human computer interaction (HCI) and computer vision (CV) [1][2][3]. It is considered to be the basic step for any system dealing with face analysis. A number of researches have been done targeting automatic face detection [3]. Recent research work in computer-vision primarily focuses on face detection under uncontrolled environments because variations in the face appearance (i.e. illuminations and pose changes) could lead to poor robustness of the system.

The Haar cascade based system [4] can detect near frontal faces with simple Haar like features based boosted cascade as primary principles. The simple features enable the system to quickly evaluate and reject false positive detection in early stages. These principles make the Haar cascade based system very effective and popular for many real-time face detection frameworks. However, the simple nature of the features makes it less effective for faces with non-frontal orientation and in uncontrolled environment (i.e. lighting, diverse poses, exaggerated expression, and occlusion). The huge number of features involved also leads to heavy processing, which makes it unfit for time critical face analysis systems [5]. Improvements to the Haar cascade detector have been made to speed up and to improve non-frontal face orientation detection accuracy in the

Bacha Rehman
bachapk@gmail.com

Ong Wee Hong
weehong.ong@ubd.edu.bn

Abby Tan Chee Hong
abby.tan@ubd.edu.bn

Trung Dung Ngo
dungnt@ieee.org

1 Faculty of Science, Universiti Brunei Darussalam, Brunei Darussalam
2 The More-Than-One Robotics Laboratory, University of Prince Edward Island, Canada

form of joint face detection and alignment [5]. This method is taken as one of the main routine algorithms for this study.

Apart from including alignment, researchers are also looking into more complex features based methods i.e. convolutional neural network (CNN) to achieve better results [2]. The CNN requires extra computation time to compute complex features for improving accuracy [6]. However, the added computation load can be normalized by reducing the number of cascade stages. The reduction in the number of cascade stages reduces the computation load without affecting the accuracy. The decrease in cascade stages makes the whole computation to remain roughly unchanged. This observation encourages using advanced features based method (i.e. CNN) for real-time face detectors.

The CNN-based methods are in contrast to the hand-crafted features based methods [7]. It can deal with tough visual variations by leveraging large training data. CNN works similarly as the normal Artificial Neural Networks (ANN) [6], however, the neurons in a CNN layer are connected to a specific sub-region of the previous layers. On the other hand, each layer's entire neurons are fully connected with each other in ANN [8]. The neurons within a CNN's layer are arranged in three dimensions, i.e. width, height, depth.

Multi-task learning (MTL) is a method for solving several learning tasks in parallel by using commonalities and differences across all tasks. For task-specific models, the prediction accuracy and learning efficiency can be improved using MTL [9]. The Multi-task Convolutional Neural Network (MTCNN) is an advanced type of CNN having the ability to use MTL for task based learning. Three stages of MTCNN architecture has been used in this paper as explained in Section 3.2.1.

In this work, Haar cascade [4], joint-cascade [5], and MTCNN [6] have been applied and tested for face detection and tracking as main routines. It has been observed that all the three approaches often lack the ability to perform fast as well as not able to handle non-frontal face orientation during real-time video analysis environment. This observation motivated us to develop a hybrid face detection and tracking system with margin-based region of interest (MROI) to improve detection accuracy and processing speed. An escape routine incorporating template matching (TM) algorithm [10] has been proposed in a hybrid approach to make the proposed face detection and tracking system more accurate, robust and reliable, and the margin-based region of interest (MROI) has been proposed to improve the processing speed for applications in real-time systems such as video analysis, surveillance, and human robot interactions (HRI) systems.

This paper also presents two variants to compute MROI, i.e. fixed margin (FM) and dynamic margin (DM). In FM, a fixed percentage extra pixels is added around the face area for detected face. In DM, fixed percentage with extra pixels is added with proportion to the change in the face position between two consecutive frames. The results show significant improvement in processing speed during face detection process.

A total of five algorithms have been proposed and implemented in this study along with a state-of-the-art algorithm to compare results improvement. The Haar cascade, Joint cascade and MTCNN representation of the state-of-the-art algorithms. Five algorithms have been proposed on each of the three original state-of-the-art algorithms and implemented to achieve better accuracy and processing time over the the original version. The proposed system has achieved better accuracy and speed on datasets including face detection and tracking videos-20 (FDTV-20) [11] and 300 Videos In-the-Wild (300-VW) [12], which contain videos with different resolutions and durations. The key contributions of this paper are as follows:

1. Proposed MROI based hybrid face detector and tracker comprises of a main and an escape routine to improve accuracy and processing speed for face detection and tracking systems.
2. Implemented the proposed hybrid models with variations in MROI on each of the three main routines, i.e. Haar cascade, Joint cascade and MTCNN.
3. Using TM as escape routine to detect faces when the main routine failed.
4. Developing FM and DM based MROI to significantly improve the processing speed by ignoring avoidable region in video frame.
5. Thoroughly evaluated five variants of each state-of-the-art algorithm used as the main routine on two datasets with real-life videos.

2. Related Work

Face detection is the primary stage to all vision-based human computer interaction (HCI) systems [13]. Several face detection algorithms [3] have been proposed from time to time. These automatic facial image analysis techniques consist of face recognition and face verification [14], face tracking for HCI [15], video and surveillance system [16], facial attribute analysis (e.g. gender, age and beauty) [17][18][19][20], behavior analysis for face [21], face morphing and relighting [22], shape reconstruction for faces [23], retrieval and organization of image and video within photo-albums [24]. In [13], a survey of different face analysis and synthesis techniques and models was presented including hand crafted, skeletal based, active contours, 3D features based, parameters based, statistical and manual models.

2.1 Simple Features Based Methods

Simple features based multi-view face detectors were presented in [25][26][27]. These techniques used separate training methods (i.e. decision trees and FloatBoost) under different head poses and viewpoints. A frontal face detection method using gradient energy representation for face detection in MPEG videos was presented in [28]. The Haar cascade based face detection algorithm [29][4] has become a convenient face detector in many software tools e.g. OpenCV [30]. Due to the simple nature of the features, it faces considerable problems with non-frontal face detection and processing speed [5]. These issues have been improved by combining face alignment on the training set with the simple features based method [5], which also improved the non-frontal faces detection.

2.2 Complex Features Based Methods

Tackling complex face variations is the main drawback of simple features based methods, which encourages researchers to look into more complex and sophisticated techniques like convolutional neural network (CNN) [7][31]. Deep CNN has been used [32] for facial feature detection to achieve high response in regions of face according to candidate windows of faces. However, the approach has been observed to be time costly for real-time systems due to the complex structure of CNN. In [28], Faster Recurrent-CNN (R-CNN) was proposed to improve both detection accuracy and processing time. Faster R-CNN [33] could be used as a good alternative of the CNN for future reference. However, the single task approach made it less effective from merely face detection [6][34]. In [6], multi-task approach was proposed for face detection and alignment using MTCNN. In this paper we have used both simple features based methods (i.e. Haar cascade and Joint cascade) as well as complex features based methods (i.e. MTCNN).

2.3 Template Matching

Template matching (TM) algorithm for face localization was presented in [35] and its variations were presented in [36]. Template matching based approach addressed issues regarding shape, color and motion was discussed in [37]. As template matching algorithm is fast by its nature [10], thus it can be used together with the main routine algorithms for effective face detection and tracking task. The major drawback of the template matching based method is that it cannot be used as sole face detection and tracking method, because it requires initial template to kicks in within a video analysis environment. Therefore, there is a need of another method to support it detecting a face and provide a template. Hence, it can be used as an escape routine rather than the main routine.

2.4 Tracking Algorithms

Since our system track a face within a video or HRI environment, relevant objects tracking algorithms were studied to use the face tracking algorithm [31] effectively. Considering different tracking algorithms, a centered correlation filters based tracking systems was discussed in [38]. A face tracking system was developed using tracking-learning-detection (TLD) algorithm for real time environment [16]. In order to track a particular person in a lecture delivering environment, informative random fern (IRF) was used in [39] and TLD was applied in [40]. Similarly, an active context learning model was applied for object tracking in video surveillance environment [41]. Haar wavelet and edge orientation based feature were used to ROI grouping and classification for the purpose of pedestrian detection were discussed in [42]. Re-registration and dynamic template based approach were developed for head motion recovery in [43].

2.5 Hybrid Approaches

A hybrid algorithm for face detection and tracking in a video and real-time environment was presented using Haar cascade and template matching [44]. The work presented in this paper is an extended version of the face detectors presented in [44] [45]. Another hybrid face recognition system was proposed using CNN as the main routine [46]. There are other hybrid approaches to improve the conventional Haar cascade method e.g. a cascade based Deep Neural Networks (DNNs) [47] used to obtain pose estimation results with high precision. Similarly, in [48] multi-task cascaded convolutional networks was adapted for face detecting using DNNs. The works mentioned above helped understand the various concepts to develop the hybrid system with improved accuracy and processing time.

3. System Components

The main idea of using a hybrid approach is that the main routine algorithms used sometimes fail to detect a face during face detection and tracking process. However, it leaves a template for the TM method to continue with the face detection and tracking process for the set number of frames. Using the TM method as the escape routine facilitates to continue face detection and tracking process until the next face detected by the main routine. Using the hybrid model provide more robust face detection and tracking system and without any major discontinuity. It also provides the facility to detect a face in any orientation as the template updates with the face rotation within a video or real-time environment. All the components used in this system are presented in the following sub-sections.

3.1 General Architecture

The hybrid algorithm is using main and escape routines. The escape routine comes into play when the main routine fails. Once a face is found, the face position is stored and MROI is calculated around it. In the next iteration it searches a face within the MROI calculated from the previous frame. The system uses this information to speed up detection and processing time significantly.

Figure 1 shows the main algorithm of the proposed approach for face detection and tracking. The original video resolution of the FDTV-20 [11] dataset is 640 x 480, while the video resolution of the 300-VW [12] dataset is not consistent. In order to get to a common point between both datasets, the downscaling technique has been used. The downscaling technique has been performed to reduce the frame size and to speed up the computation.

Algorithm 1: MROI based face detection and tracking

```
Data: Video frames
Result: Face position, Template and Rectangle, MROI
Position & Template & ROI & Rectangle = null
for each frame f in the video stream do
    Get the frame f and downscale
    Compute and extract MROI on frame f
    Apply main routine to detect face in MROI
    if Face detected with main routine then
        Go to next Section
    else if Face template is not NULL then
        Fall back to escape routine
        Apply TM to detect face
    end
    update
        Face position, area and template
        MROI with the new face detected (for NT and
        NTMT the MROI is taken as full frame)
end
```

Figure 1. Hybrid and MROI based algorithm for face detection and tracking

Both fixed width and height based downscaling is not a good technique to apply, because they squeeze the face and other object's shape within the video frame. Therefore, fixed width and dynamic height technique has been used for downscaling the video frame. The downscaled width has been fixed to 320 pixels, while the height of the video is downscaled by the relative original video ratio. For instance, a video frame of size 1280 x 720 will be downscaled to 320 x 180, while a video of size 640 x 480 will be downscaled to 320 x 240. In this way the desired downscaling has been achieved without affecting the object shapes inside the video frames, i.e. maintaining the aspect ratio of the original image frames. The downscaling is performed for run time performance only. For analysis, the video frame has been upscaled to its original size.

The general system architecture of the main algorithm is shown in Figure 2. Using MROI within single routine makes the algorithm fast but almost all the three main routines algorithms fail sometime to detect a face in non-frontal position. Therefore, TM is used to rescue the situation. If the main routine fails, the TM algorithm finds the maximum

likelihood face position based on the face template extracted from the previous detected face.

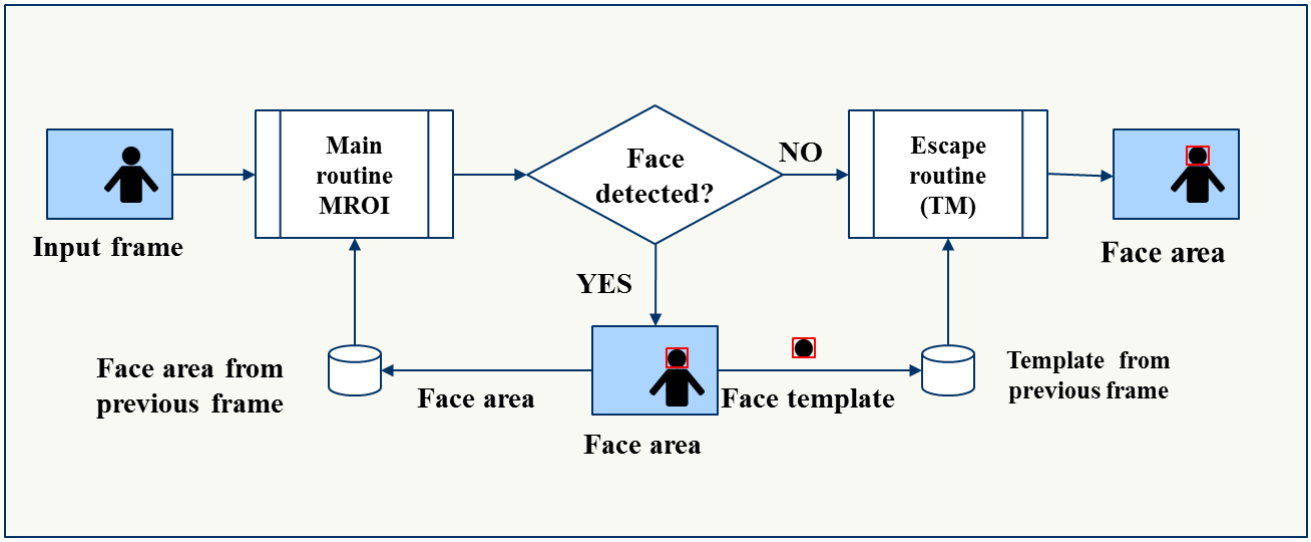


Figure 2: General system architecture

The TM detects a face based on the template of the latest face detected and does not rely on dataset that was used to train the main detector routine. TM continues until the main routine re-detects a face or a counter is reached.

3.2 The Hybrid Approach

The proposed system uses two routines as discussed in the general architecture (Figure 3). The algorithms included in both main and escape routines are described in this Section.

3.2.1 Main routine

The main routine consists of the algorithm responsible for face detection. For experimental purpose, three different algorithms have been used as main routine separately. The algorithms used as main routines are Haar cascade, Joint cascade and MTCNN.

The first algorithm used as the main routine in this work is Haar cascade [4]. It uses two basic principles for applied solutions, i.e. simple features and boosted cascade structure. It is a machine learning technique used for face detection capable to rapidly process images with high accuracy. This method used integral image technique, which allows fast computation for Haar like features. AdaBoost has been used as learning algorithm, which picks few important features from the larger set of visual features, making it very efficient classifier [49]. This method also combined various complex classifiers within a cascade. This approach enabled to quickly discard the background regions and focus more on favorable object-like regions. The cascade is a specific object focusing mechanism which focuses only on regions contain the object of interest. Many real-time face detection systems are based on these two principles.

The second algorithm used as the main routine is the joint cascade. The main idea behind the joint cascade [5] is to combine face alignment with face detection. This concept provides an observation that better features for face classifications can be achieved with aligned face shapes. The joint cascade learns both tasks i.e. detection and alignment, jointly in the same cascade framework to make the combination effective. The joint learning significantly improves detection while keeping its real-time performance.

The third algorithm used as the main routine is MTCNN [6]. CNN is a multi-layered neural network, in which all layers are partially connected. The input layer receives same size images. The convolutional kernel processes a set of units in a small neighborhood to form a single unit in the feature map of the convolutional layer. Each plane in the convolutional layer is presented in Figure 3 as the general architecture of CNN.

The MTCNN routine has been used as the image pyramid, which is the input of the three-stage cascaded framework [6]. The multi task CNN handles two tasks, i.e. detect face and do alignments. The CNN model used in this paper is inspired from the work presented in [6], which also revealed the complete detailed diagrams of all the three stages used.

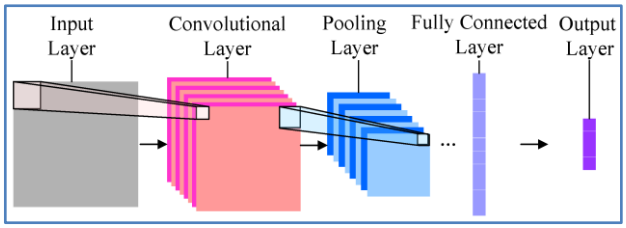


Figure 3: General CNN structure

The three stages of the MTCNN are called Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). Stage one of the MTCNN used in this work consists of a full CNN, called P-Net. The P-Net uses 12 network sizes, 3 convolutional layers of kernel size 3 x 3, a maximum pooling layer of kernel size 2 x 2. The output of the P-Net consist of a binary face classifier, face boxes of size 4 and facial landmarks of size 10. P-Net is used to obtain the candidate facial box regression vectors. These vectors are adjusted with the estimated bounding box vectors. The non-maximum suppression (NMS) has been applied to merge the overlapped candidates.

Stage two of the MTCNN used is called R-Net. The R-Net consists of 24 networks size. There are two convolutional layers of kernel size 3 x 3 which are followed by maximum pooling layers of the same kernel size. These four layers are followed by another convolutional layer of kernel size 2 x 2 and fully connected layer of size 128. The output of the R-Net consist of a binary face classifier, face boxes of size 4 and facial landmarks of size 10. The R-Net is responsible for further rejection of large number of false candidates.

Stage three of the MTCNN model used is called O-Net. The O-Net uses 48 network sizes in these experimental setups. Each of the three convolutional layers of size 3 x 3 was followed by maximum pooling layer of same kernel size. These size layers are followed by another convolutional layer of size 2 x 2 and further followed by a fully connected layer of size 256. The output layer consists of the facial classification, face bounding boxes and facial landmarks. The parametric rectified linear unit (PReLU) [50] was used as nonlinearity activation function after each convolutional and fully connected layer. The O-Net produces the final face classification, face bounding boxes, and facial landmarks position.

3.2.2 Escape routine

The template matching (TM) has been used as the escape routine to rescue the main routine. The sum of squared difference (SSD) [51] has been used for matching the template image within the input image. Mathematically, the normalized SSD method used for TM algorithm in this work is expressed in Equation (2). TM targets to detect a given template image in the input grayscale frame on the basis of best match procedure using sliding. As shown in Figure 5, the template image is defined by a matrix template $m \times n$. In the main full frame searching area the template can be matched at matrix (a, b) area. On the other hand, the source image is defined by a matrix test $M \times N$.

The template matching procedure can be described as locating the best location (a, b) for the template image template $m \times n$ so that the match between template matrix $(1 : m, 1 : n)$ and test matrix $(a : (a + m - 1), b : (b + n - 1))$ is maximized within the reasonable search area as shown in Figure 4.

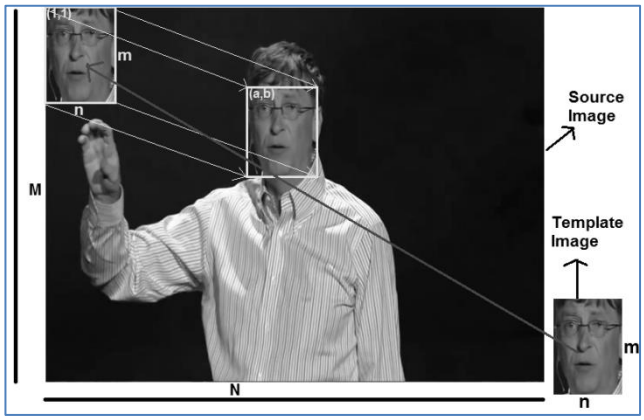


Figure 4: Template matching schema diagram

3.3 The MROI concept

Figure 5 illustrates the MROI approach used for this work. In the fixed margin approach the MROI is taken as x percentage extra pixels of the total face box area as the margin offset. While in dynamic

margin approach a fixed percentage *y* and the movement in face position between two consecutive video frames δy are taken into considerations.

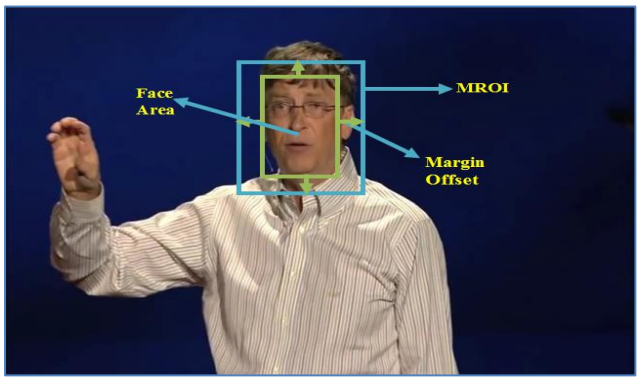


Figure 5: MROI illustration

On successful detection of face within the image frame, the face position, region and template are stored and the MROI is calculated around it. Several experiments have been performed to select a the size for the fixed and dynamic margin. The margin size was started from 5% to 35% of the face area during the margin size selection process. It can be observed from Figure 6 that after 25% margin size there is no effect on the face detection accuracy. The results shown in Figure 6 are taken for the Haar cascade method.

In view of the results shown in Figure 6, for the fixed margin (FM), extra pixels around the face area are taken at a fixed percentage *x* (i.e. 25%) on each side. In dynamic margin (DM) calculation, *y* percentage of FM is taken as compulsory margin with the added pixels proportional to the change in the face position δy between two consecutive image frames. The compulsory margin for the DM *y* is taken as 20% of the face region around each side of the face region. In addition, the extra dynamic pixels are taken as directly proportional to the distance between the previous frame face position and current frame face position.

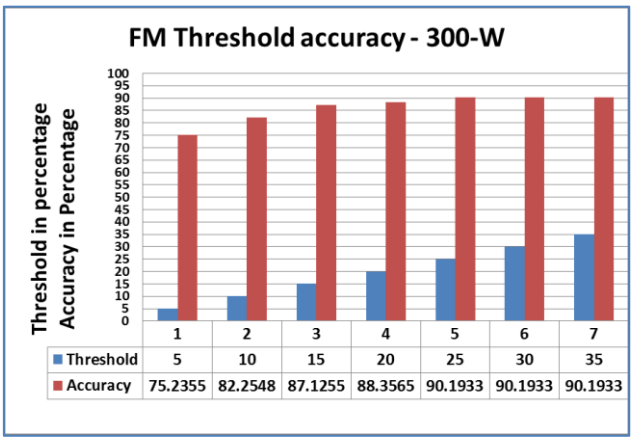


Figure 6: Threshold accuracy analysis for FMT

4. Proposed Algorithms

A total of six algorithms have been presented in this paper in which we proposed five and one is taken as state-of-the-art method. All the algorithms were implemented for each of the three main routines (i.e. Haar, joint-cascade and MTCNN). These algorithms are named as below:

1. Normal Face Tracking (NT)
2. Fixed Margin Face Tracking (FMT)
3. Dynamic Margin Face Tracking (DMT)
4. Normal Template Matching Face Tracking (NTMT)
5. Fixed Margin with Template Matching Face Tracking (FMTMT)
6. Dynamic Margin with Template Matching Face Tracking (DMTMT)

The NT algorithm is considered as the state-of-the-art algorithm. The rest of five algorithms are proposed in this work. We can express the application of main routine algorithm on a video (i.e. sequence of frames) as in Equation (1). The summation symbol in Equation (1) and the other Equations in this paper represents the iteration. Mathematically, the conventional full frame scanning main routine can be expressed by Equation (1).

$$F_n(z) = \begin{cases} \sum_{m=\frac{z}{10}}^{m=z} Main(m) & FB = 0 \\ \sum_{m=A_{FB}\times\frac{3}{4}}^{m=z} Main(m) & otherwise \end{cases} \quad \text{.. (1)}$$

In Equation (1), F_n is the main routine algorithm implemented in the conventional way where m is pixels scale window. The frame is represented by *z*.

There are two scenarios to calculate the minimum and maximum size of the scale windows which is dependent on whether the system has obtained a face box *FB* from the previous frame or not. If the face box *FB* is 0 then the minimum size of the window will be $\frac{z}{10}$ while the maximum is the size of the image frame, i.e. z. On the other hand, if the system has a face box *FB* then the minimum size of the window will be 75% of the area of the face box $A_{FB,}$ while the maximum size of the window will remain the same. Setting the minimum and maximum scaled windows size helps significantly in speeding up the detection process by avoiding the faces which are less than the minimum scaled window size. The notation used as *Main* refers to the main algorithm, *n* represents the normal tracking technique, which is the conventional implementation.

Equation (2) represents the TM algorithm using normalized sum of squared difference [51] as discussed in Section 3.2.2.

$$\boldsymbol{TM(x,y)} = \left(\frac{\sum_{x'y'} \left(\boldsymbol{T(x',y')} - \boldsymbol{I(x+x',y+y')} \right)^2}{\sqrt{\sum_{x'y'} \boldsymbol{T(x',y')^2} . \sum_{x'y'} \boldsymbol{I(x+x',y+y')^2}}} \right) \quad \text{.. (2)}$$

$TM(x,y)$ is the equation for the template matching algorithm. *T* is the template image while *I* denotes the input image. Match metric contains each location *(x,y)* in TM. The location within the template is represented by (x', y').

The mathematical representations of the proposed fixed and dynamic margin-based detectors are given by Equations (3) and (5) respectively.

$$\boldsymbol{F_{fm}(z)} = \sum_{\boldsymbol{m=A_{FB} \times \frac{3}{4}}}^{\boldsymbol{m=r}} \boldsymbol{Main(m)} \quad \text{.. (3)}$$

$$\textbf{where } \boldsymbol{r = (1 + \Delta b) b} \quad \text{.. (4)}$$

F_{fm} represents the fixed MROI approach for face detection. In Equation (3), r is the region of interest (ROI) extracted from the frame *z*. *FB* is the face box detected in the previous frame, while A_{FB} is the area of the face box *FB*. The scaled window m was started from a minimum size of 75% of the face box area A_{FB}, to a maximum size r. If A_{FB} is zero then the face detector will use the default scaled window. Extra pixels around the face area b are taken at a fixed percentage of Δb at all sides. For the fixed MROI, Equation (4) calculates the ROI. In this work, Δb is set at 25% for fixed MROI approach.

$$\boldsymbol{F_{dm}(z)} = \sum_{\boldsymbol{m=A_{FB} \times \frac{3}{4}}}^{\boldsymbol{m=r}} \boldsymbol{Main(m)} \quad \text{.. (5)}$$

$$\textbf{where } \boldsymbol{r = (1 + \Delta b) b + \Delta p} \quad \text{.. (6)}$$

F_{dm} represents the dynamic margin approach. It is the same as F_{fm} except the *r* is calculated with extra dynamic pixels Δp, which is directly proportional to the face position movement in the prior two frames. Equation (6) calculates the ROI *r*, where Δb is set to 20% for this work.

Equations (1) to (6) are used as the base models to derive and implement the proposed algorithms in this study. The base models are represented in Equations from (7) to (12) by the name of the Equation. For instance, $F_n(z)$ represents Equation (1), $TM(x,y)$ represents Equation (2), $F_{fm}(z)$ represents Equations (3) and (4), and $F_{dm}(z)$ represents Equations (5) and (6). The proposed algorithms are mathematically represented from Equations (7) to (12). The MROI approaches have been used in Equations (8), (9), (11), and (12). Equations (8) and (11) express fixed margin approaches, while Equations (9) and (12) present dynamic margin approaches.

$$\boldsymbol{F_{NT}(z)} = \boldsymbol{F_n(z)} \quad \text{..(7)}$$

$$\boldsymbol{F_{FMT}(z)} = \begin{cases} \boldsymbol{F_n(z)} & \boldsymbol{z = 1 \parallel F_n(z-1) = 0 \parallel F_{fm}(z-1) = 0} \\ \boldsymbol{F_{fm}(z)} & \boldsymbol{otherwise} \end{cases} \quad \text{..(8)}$$

$$\boldsymbol{F_{DMT}(z)} = \begin{cases} \boldsymbol{F_n(z)} & \boldsymbol{z = 1 \parallel F_n(z-1) = 0 \parallel F_{dm}(z-1) = 0} \\ \boldsymbol{F_{dm}(z)} & \boldsymbol{otherwise} \end{cases} \quad \text{..(9)}$$

$$\boldsymbol{F_{NTMT}(z)} = \begin{cases} \boldsymbol{F_n(z)} & \boldsymbol{z = 1 \parallel T = null \parallel cnt = 10} \\ \boldsymbol{TM(x,y)_{z,cnt}} & \boldsymbol{F_n(z) = 0} \end{cases} \quad \text{..(10)}$$

$$\boldsymbol{F_{FMTMT}(z)} = \begin{cases} \boldsymbol{F_n(z)} & \boldsymbol{z = 1 \parallel T = null \parallel cnt = 10} \\ \boldsymbol{F_{fm}(z)} & \boldsymbol{F_n(z-1) = 1 \parallel F_{fm}(z-1) = 1} \\ \boldsymbol{TM(x,y)_{z,cnt}} & \boldsymbol{F_n(z) = 0 \parallel F_{fm}(z) = 0} \end{cases} \quad \text{..(11)}$$

$$\boldsymbol{F_{DMTMT}(z)} = \begin{cases} \boldsymbol{F_n(z)} & \boldsymbol{z = 1 \parallel T = null \parallel cnt = 10} \\ \boldsymbol{F_{dm}(z)} & \boldsymbol{F_n(z-1) = 1 \parallel F_{dm}(z-1) = 1} \\ \boldsymbol{TM(x,y)_{z,cnt}} & \boldsymbol{F_n(z) = 0 \parallel F_{dm}(z) = 0} \end{cases} \quad \text{..(12)}$$

z=1 indicates the first frame of the video. The $cnt = 1\ to\ 10$ is the counter for the escape routine (i.e. TM) to process the subsequent *10* frames before transferring back to the main routine.

In addition, a distance variable is applied to measure the distance between the previous and current frame face position. If the distance exceeds a certain threshold (i.e. 30 pixels), it is considered as a wrong detection and the face tracking switches from main to escape routine for continuation of face detection and tracking. Main detector randomly has false detection; therefore such situations are minimized by introducing the distance filter.

5. Datasets

Two datasets comprised of RGB videos have been used for this work. These datasets include face detection and tracking videos-20 (FDTV-20) [11] and 300 Videos In-the-Wild (300-VW) [12]. The datasets were carefully selected in order to test the proposed algorithms to handle the video data captured under various conditions including indoor, outdoor, non-frontal faces orientation, over exposed faces, various cluttered backgrounds, and illuminations etc.

5.1 The FDTV-20 dataset

The FDTV-20 dataset comprised of 20 videos has been created for this work and made available for general public [11]. The videos were obtained from YouTube and resized to resolution of 640x480. Figure 7 shows some sample frames extracted from the dataset used for this work.

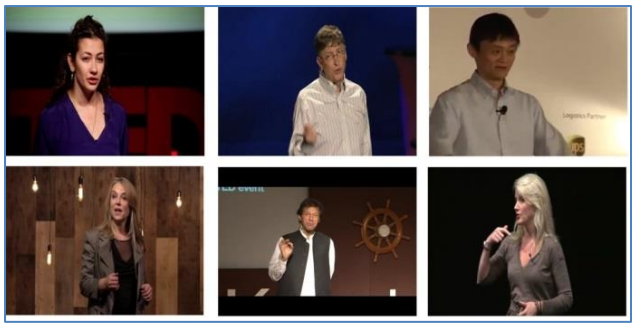


Figure 7: Screen shots of some of the videos in the dataset.

Details of all the videos in the dataset are given as below:

1. Each video contains a person in a lecture delivery environment
2. The face in each video changes orientation with scenes of both frontal and non-frontal face orientations.
3. Videos length is 15 seconds with resolution 640 x 480 containing roughly 450 frames.
4. In each video, both camera and the person are moving.
5. 13 videos of male and 7 videos of female.
6. Truth table is provided to give information regarding the face box and face position in the format as: frame no, x1, x2, y1, y2.

5.2 The 300-VW dataset

The 300 Videos in the Wild (300-VW) dataset [12] contains 114 videos of length varies from one to five minutes each. This dataset was mainly created for facial landmark detection, however, the diversity and variations exist in this dataset provided a perfect opportunity to test the proposed algorithms over this dataset for the purpose of face detection and tracking. This dataset provides the opportunity to test the ability of any algorithm to work on unseen subjects, robust for variations in face orientation, facial expression, various backgrounds, different video resolutions, different illuminations, dark rooms, overexposed shots, and occlusion.

A number of scenarios were considered during the creation of this datasets. The videos were recorded keeping in mind various unconstrained conditions including well-lit rooms, dark rooms, various head poses, people with glasses, children, beard, make-up, facial expressions, indoor and outdoor, more than one person but the focus should be one person, people with different ethnicities, and people performing various tasks. The tasks which the persons in the videos perform include delivering lectures, speeches, indoor talking in front of camera, outdoor shots, singing and etc. 68 facial landmarks were provided for each of the 218,597 video frames as the truth table. Figure 8 shows a selection of snapshots captured from a few of the videos.

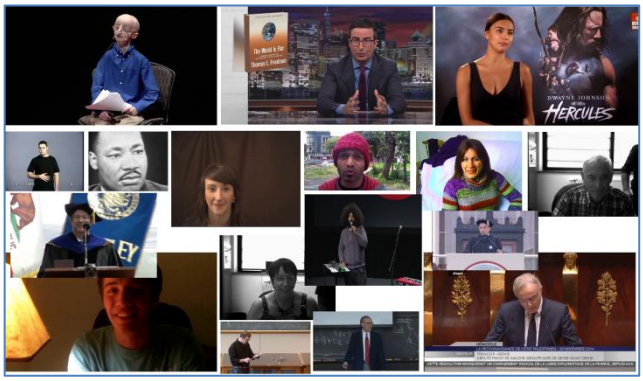


Figure 8: An overview of 300-VW videos

6. Experiment details

The six algorithms as described in Section 3 were implemented on each of the three main routines resulting in eighteen models have been tested on the FDTV-20 [11] and 300-VW [12] datasets to evaluate the performance of each algorithm in terms of accuracy (correct, incorrect, and not detected), average time taken per frame in milliseconds, and the ability of the whole system to process the number of frames per second. Each algorithm was executed ten times on each video file in each dataset. For instance, while processing the FDTV-20 dataset, each face detector and tracker algorithm was executed 200 times for all 20 videos. From the obtained results, the accuracy, execution time and frames per second (FPS) performance were calculated by taking the average of all obtained results.

Table 1 shows the specifications of the development environment. No graphics processing unit (GPU) has been used during these experiments and analysis.

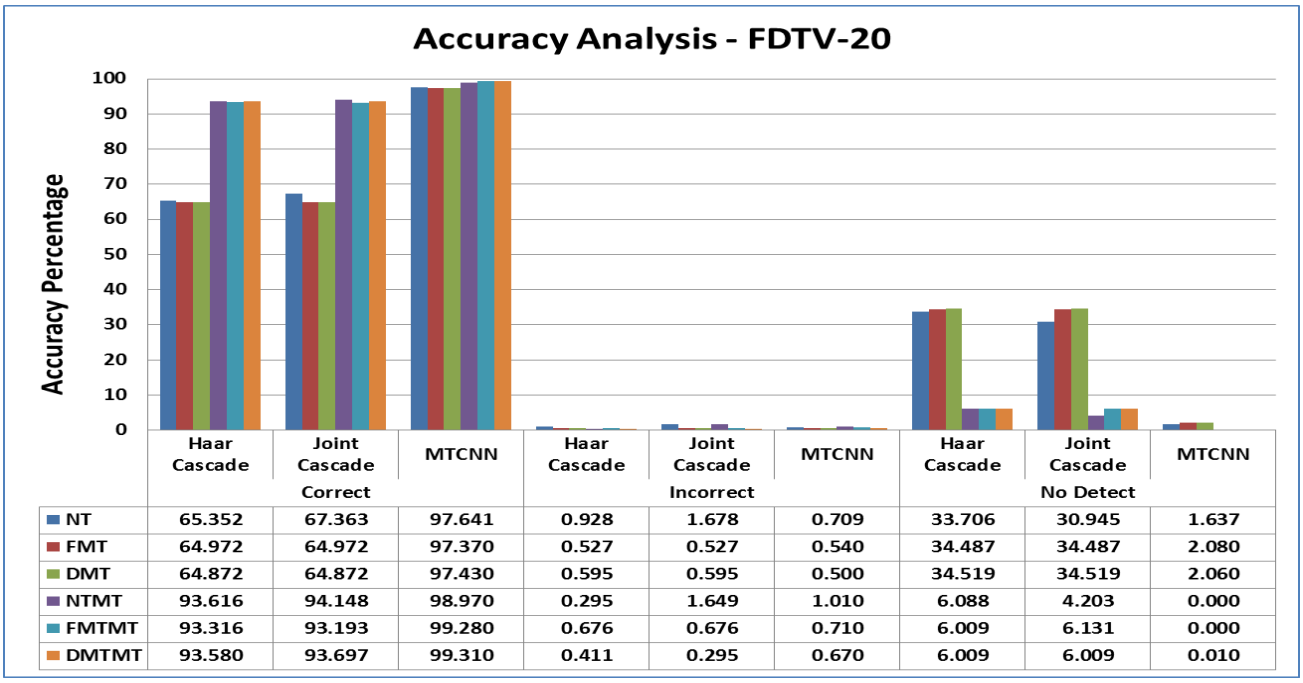
Table 1. Hardware and software used for the system development

Hardware	CPU	Intel® Core™ i5 CPU 650 @ 3.20 GHz
	RAM	8 GB
Software	OS	Widows 8.1 pro 64 bits
	Language	Python
	Tool	OPENCV, Tensorflow for MTCNN

7. Experimental Results

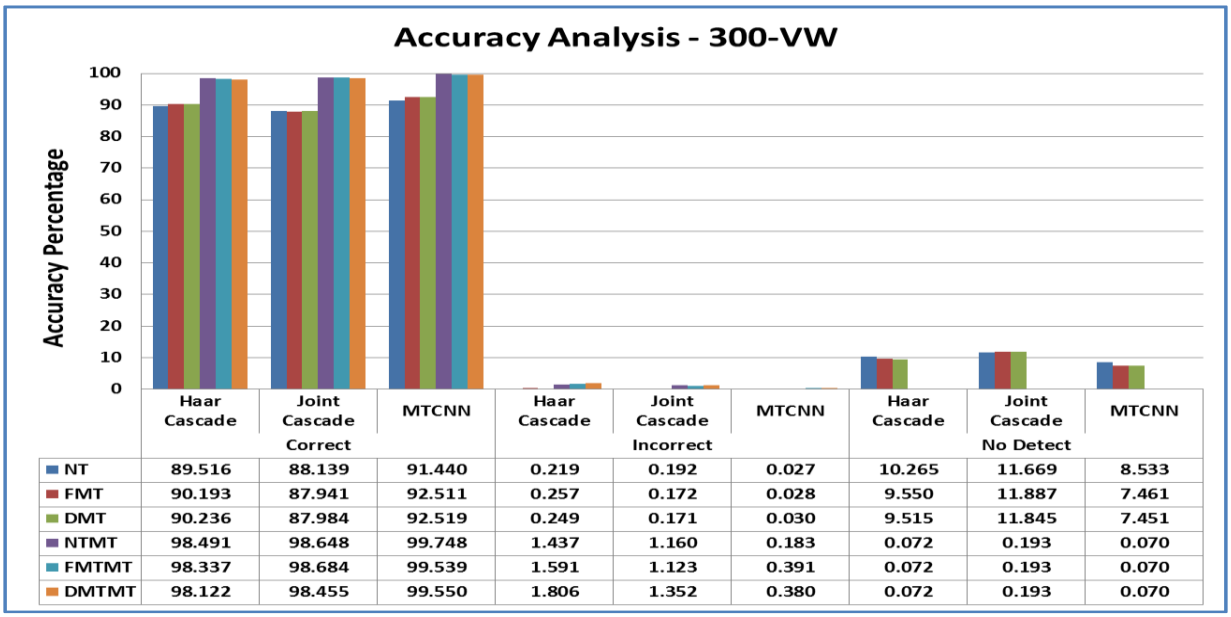
The accuracy has been calculated in a way if the face position in the truth table lies inside the detected face box or not. For each frame of the FDTV-20 dataset, we are provided with the x1, x2, y1, and y2. In order to calculate the face position from this truth table, the middle point of this truth table box was calculated. To calculate the accuracy of the 300-VW dataset, we have taken facial point number 28 as the face position. The facial point 28 is the start of the nose which is corresponds to the middle of any face box. If the truth table's face position lies inside the detected face box then it was considered as correct detection, otherwise it would incorrect detection. If face was not detected then it would be considered as Not Detected.

Figure 8 shows the complete accuracy analysis of all the algorithms for FDTV-20 dataset including correct, incorrect and not detected results. Considerable high accuracies have been achieved by the algorithms involved MTCNN. It can be also observed that by introducing the TM based hybrid model increased the accuracy quite significantly among all the Haar cascade, Joint cascade and MTCNN. The entire hybrid based models of MTCNN including NTMT, FMTMT and DMTMT achieved significantly improved accuracy. Overall, using escape sequence based hybrid model significantly improved the detection accuracy. Among all the algorithms used, the best accuracy results achieved are 99.31% for DMTMT and 99.28% FMTMT using FDTV-20 dataset.



	Correct			Incorrect			No Detect		
	Haar Cascade	Joint Cascade	MTCNN	Haar Cascade	Joint Cascade	MTCNN	Haar Cascade	Joint Cascade	MTCNN
NT	65.352	67.363	97.641	0.928	1.678	0.709	33.706	30.945	1.637
FMT	64.972	64.972	97.370	0.527	0.527	0.540	34.487	34.487	2.080
DMT	64.872	64.872	97.430	0.595	0.595	0.500	34.519	34.519	2.060
NTMT	93.616	94.148	98.970	0.295	1.649	1.010	6.088	4.203	0.000
FMTMT	93.316	93.193	99.280	0.676	0.676	0.710	6.009	6.131	0.000
DMTMT	93.580	93.697	99.310	0.411	0.295	0.670	6.009	6.009	0.010

Figure 8: Accuracy Analysis FDTV-20



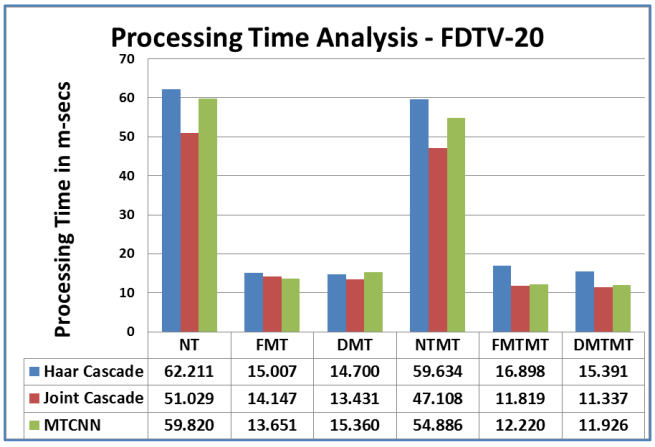
	Correct			Incorrect			No Detect		
	Haar Cascade	Joint Cascade	MTCNN	Haar Cascade	Joint Cascade	MTCNN	Haar Cascade	Joint Cascade	MTCNN
NT	89.516	88.139	91.440	0.219	0.192	0.027	10.265	11.669	8.533
FMT	90.193	87.941	92.511	0.257	0.172	0.028	9.550	11.887	7.461
DMT	90.236	87.984	92.519	0.249	0.171	0.030	9.515	11.845	7.451
NTMT	98.491	98.648	99.748	1.437	1.160	0.183	0.072	0.193	0.070
FMTMT	98.337	98.684	99.539	1.591	1.123	0.391	0.072	0.193	0.070
DMTMT	98.122	98.455	99.550	1.806	1.352	0.380	0.072	0.193	0.070

Figure 9: Accuracy Analysis 300-VW

Figure 9 shows the accuracy results for all the algorithms applied on 300-VW dataset. The results are based on correct, incorrect and not detected parameters. It can be observed from this figure that the inclusion of the escape sequence significantly improved the accuracy of the system and hence certify the results obtained over FDTV-20 datasets. Considering a dataset of this big size, achieving the accuracies of 99.75% for NTMT, 99.54% for FMTMT, and 99.55% for DMTMT by using MTCNN as main routine suggest that the algorithms using the proposed hybrid models perform robustly for data captured under various conditions.

As the algorithms were developed keeping in mind the real-time system with critical time constraints, therefore, the importance of processing time and the FPS processing ability is of great concerns. Figure 10 shows the processing time analysis of all the

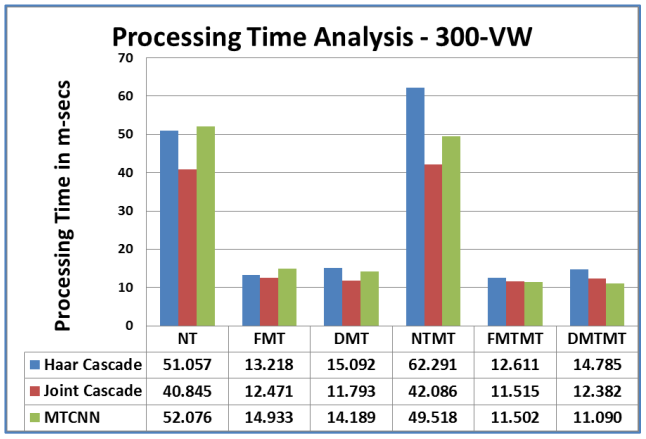
algorithms used in terms of how many milliseconds each algorithm took to process a single image frame for FDTV-20 dataset. Figure 10 also shows that the MTCNN based algorithms were consistent in achieving fast processing speed. Overall, the results show that the incorporation of the MROI has significantly improved the processing speed. It can be observed from Figure 10 that an improved processing speed can be achieved while using the FMTMT or DMTMT. In the whole process also shows that the Joint cascade and MTCNN works quite fast after applying the MROI techniques resulting into huge improvement in the processing speed. Figure 11 summarizes the processing time taken by each algorithm during the entire face detection and tracking process for 300-VW dataset. It is necessary to mention that the processing time was the average of the ten time executions of each algorithm.



	NT	FMT	DMT	NTMT	FMTMT	DMTMT
Haar Cascade	62.211	15.007	14.700	59.634	16.898	15.391
Joint Cascade	51.029	14.147	13.431	47.108	11.819	11.337
MTCNN	59.820	13.651	15.360	54.886	12.220	11.926

Figure 10: Processing time analysis – FDTV-20

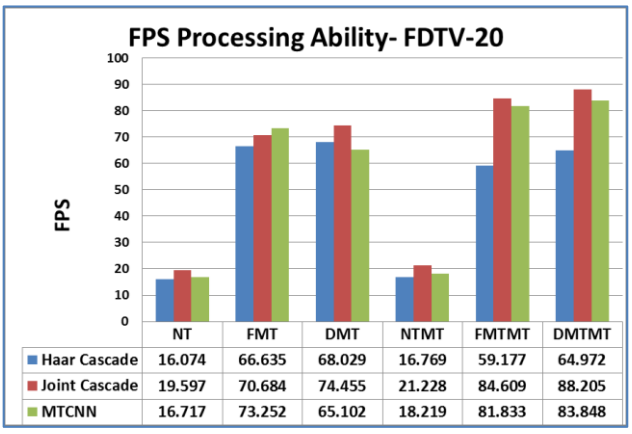
The calculation of processing time for each frame were used to calculate the FPS processing ability of the system. It can be seen that after applying the MROI techniques, the processing speed significantly increased. Using MTCNN, the FMTMT and DMTMT reduced the processing time from 52 to 11 milliseconds to process a single frame.



	NT	FMT	DMT	NTMT	FMTMT	DMTMT
Haar Cascade	51.057	13.218	15.092	62.291	12.611	14.785
Joint Cascade	40.845	12.471	11.793	42.086	11.515	12.382
MTCNN	52.076	14.933	14.189	49.518	11.502	11.090

Figure 11: Processing time analysis – 300-VW

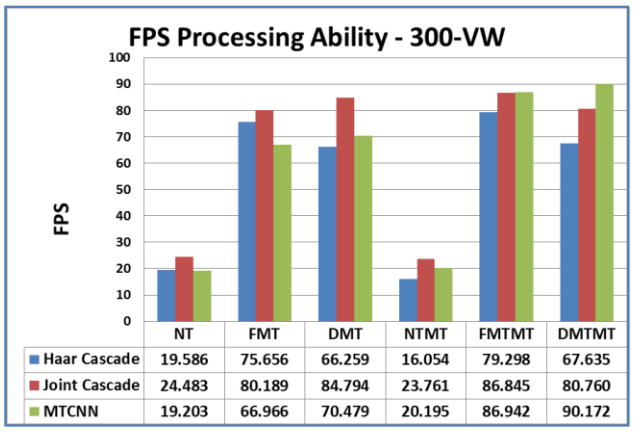
We calculated the FPS ability of all the algorithms used on the two datasets by using the processing time each algorithm took. Figure 12 shows the FPS processing ability for all the algorithms used over the FDTV-20 dataset. Significant FPS ability has been achieved with the FMTMT and DMTMT for Joint cascade and MTCNN. With these algorithms the system is able to process over 80 FPS.



	NT	FMT	DMT	NTMT	FMTMT	DMTMT
Haar Cascade	16.074	66.635	68.029	16.769	59.177	64.972
Joint Cascade	19.597	70.684	74.455	21.228	84.609	88.205
MTCNN	16.717	73.252	65.102	18.219	81.833	83.848

Figure 12: FPS processing ability – FDTV-20

Figure 13 shows the FPS ability for all the algorithms used over the 300-VW dataset. It can be seen that the FPS ability significantly improved in the algorithms involving the MROI techniques. The FPS ability further improved in the MROI based hybrid systems by taking advantage from the speedy face detection using TM algorithm. In view of the results presented in these figures, the proposed hybrid models have successfully improved the FPS ability of the system by at least 400%. For instance, the FPS processing ability has been increased in the MTCNN algorithm from 19 to 90 FPS.

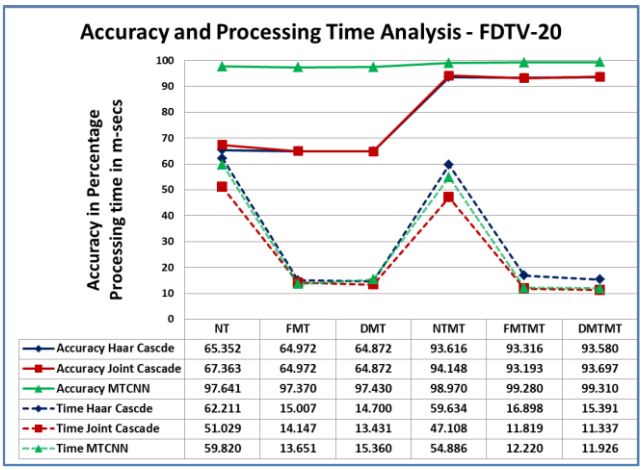


	NT	FMT	DMT	NTMT	FMTMT	DMTMT
Haar Cascade	19.586	75.656	66.259	16.054	79.298	67.635
Joint Cascade	24.483	80.189	84.794	23.761	86.845	80.760
MTCNN	19.203	66.966	70.479	20.195	86.942	90.172

Figure 13: FPS processing ability – 300-VW

Keeping in mind the average face detection accuracy, processing time and FPS processing ability for each of the algorithm, it can be seen from the results that the incorporation of template matching and MROI significantly improved all the parameters including detection accuracy, processing time and FPS ability. The results also reveal that the dynamic margin is better approach in terms of all the parameters for detecting faces. These analyses reveal that the dynamic margin approach shows quite robust results in all aspects.

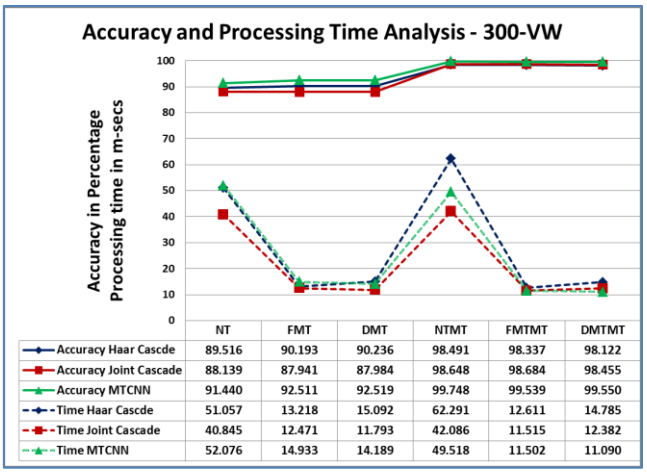
As the main research problem is to set a tradeoff between accuracy and processing time, therefore, we combined results with accuracy, processing time, and FPS processing ability. The selection of a relevant and desired algorithm has been done keeping in mind those aspects. Figure 14 summarizes the complete results regarding detection accuracy and processing time for the three main routines for the FDTV-20 dataset. Summarizing all the results using main algorithm parameters, the accuracy in Haar cascade based approach increased from 65.35% (NT) to 93.62% (NTMT). But considering the processing time parameter the NTMT does not contribute well, hence contradicting the desired goal to present a trade-off between accuracy and processing time. On the other hand, FMTMT and DMTMT satisfied the research goal of this project as both approaches have significantly improved processing speed as well as boosted detection accuracy. Considering DMTMT approach particularly during Haar cascade based system, it has increased the detection accuracy from 65.35% to 93.58% while decreasing the processing time from 62.21 to 15.39 milliseconds (ms) per frame. Similarly, FMTMT achieved accuracy of 93.32% with processing time as 16.90 ms.



	NT	FMT	DMT	NTMT	FMTMT	DMTMT
Accuracy Haar Cascde	65.352	64.972	64.872	93.616	93.316	93.580
Accuracy Joint Cascade	67.363	64.972	64.872	94.148	93.193	93.697
Accuracy MTCNN	97.641	97.370	97.430	98.970	99.280	99.310
Time Haar Cascde	62.211	15.007	14.700	59.634	16.898	15.391
Time Joint Cascade	51.029	14.147	13.431	47.108	11.819	11.337
Time MTCNN	59.820	13.651	15.360	54.886	12.220	11.926

Figure 14: Accuracy and Processing Time Analysis – FDTV-20

Considering joint cascade based approach, the DMTMT increased the detection accuracy from 67.36% to 93.70% while reducing the processing time from 51.02 ms to 11.34 ms. On the other hand, FMTMT achieved 93.19% detection accuracy with 11.82 ms processing time per frame. Considering the MTCNN as main routine algorithm, the DMTMT increased the detection accuracy from 97.64% to 99.31% while decreasing the processing time from 59.82 ms to 11.93 ms. Moreover, the FMTMT achieved detection accuracy of 99.28% with processing time of 12.22 ms.



	NT	FMT	DMT	NTMT	FMTMT	DMTMT
Accuracy Haar Cascde	89.516	90.193	90.236	98.491	98.337	98.122
Accuracy Joint Cascade	88.139	87.941	87.984	98.648	98.684	98.455
Accuracy MTCNN	91.440	92.511	92.519	99.748	99.539	99.550
Time Haar Cascde	51.057	13.218	15.092	62.291	12.611	14.785
Time Joint Cascade	40.845	12.471	11.793	42.086	11.515	12.382
Time MTCNN	52.076	14.933	14.189	49.518	11.502	11.090

Figure 15: Accuracy and Processing Time Analysis – 300-VW

Figure 15 shows the combined results regarding the accuracy and processing time for all algorithms used for 300-VW dataset. Based on the previous discussion we will focus and consider only the FMTMT and DMTMT based results. For Haar cascade, the DMTMT increased the accuracy from 89.52% to 98.12%, while reducing the processing

time from 51.05 ms to 14.78 ms. The FMTMT achieved 98.34% accuracy with 12.61 ms. For joint cascade, the DMTMT achieved 98.46% detection accuracy, while reducing the time from 40.84 ms to 12.38 ms. The FMTMT achieved 98.68% detection accuracy with 11.51 ms processing time. For the MTCNN based system, the DMTMT increased the detection accuracy from 91.44% to 99.55% while decreasing the processing time from 52.07 ms to 11.09 ms. Moreover, the FMTMT achieved detection accuracy of 99.54% with processing time of 11.50 ms.

All the results reveal that the margin-based hybrid approach significantly improves the detection accuracy, processing speed and FPS processing ability. For the face detectors, the FMTMT and DMTMT approaches make the processing time reducing by at least 400% when comparing with the non-margin based algorithm. Both the FMTMT and DMTMT were among the fastest approach. The DMTMT based method has got an advantage to use the dynamic margin, which is proportional to the movement in the face position, to cover the fast face moment in the video stream. For the MTCNN based approach, both margin-based approaches could easily handle a frame rate of more than 80fps in both the dataset used in this work. All the results show that the FMTMT and DMTMT significantly improved performance in terms of both accuracy and processing time.

8. Discussion

It has been observed that using complex features based methods significantly decrease the divergence from main routine to the escape routine. The reason for this significant decrease is the ability of complex features based method to tackle with rough situations i.e. non-frontal face detections.

The accuracy, processing time and FPS ability of Haar cascade are different from what we have achieved in our initial work [44] on the earlier dataset FDTV-10 [52] because we have used different scaled windows size and also the Haar cascade based method failed to detect faces in majority of frames in two videos in the FDTV-20 [11]. All the results in Section 7 reveal that using MTCNN as the main routine has achieved significantly improved performance in terms of accuracy and processing time. It was observed that due to the simple nature of features used in Haar and Joint cascade methods, these two methods gave quite poor detection rate in two videos in FDTV-20. On the other hand, the MTCNN algorithm has shown consistent performance in processing the entire dataset including those two videos. This consistency was achieved due to the complex nature of features used in MTCNN method which makes it better to deal with more rough situations.

This observation shows that the Haar and joint cascade failed to handle significant changes in face orientation, while the MTCNN has less issue handling variation in face orientation. This observation reveals that the MTCNN is quite ideal to be used as the main routine algorithm due to its advance nature of features and robustness to tackle several hard and real-world situations. Added with our proposed margin based hybrid approach, the resulting face detection and tracking system has achieved further improved performance.

9. Conclusion

This paper presents hybrid models using two routines and incorporating MROI techniques to improve processing time and face detection accuracy. Haar-cascade, join-cascade and MTCNN were used as the main routine. Two MROI applications were implemented, i.e. fixed and dynamic. Several experiments have been performed on eighteen combinations, which lead us to observe the effect of each component of the system on processing speed and accuracy. Using TM as escape routine helped boost accuracy to handle various variations in face orientation, background, image quality, facial expression, illuminations, lightening, overexposed shots, occlusion, and non-frontal face orientation in images sequence. The MTCNN based face detectors have achieved accuracy of 99.31% on FDTV-20 and 99.55% on 300-VW datasets. Incorporating MROI based techniques significantly improved processing speed without the use of GPU, i.e. by 400%, when compare to the state-of-the-art method used as NT. Both the hybrid FM and DM based approaches achieve similar performance. However, it is observed that the DM base setup achieved better performance with good processing speed in all applied combinations. The ability to detect fast face moment and the excellent performance of the DM based approach makes it a robust setup, hence a better

choice. Further research can be performed on the topic by adding more algorithms to each routine.

Compliance with Ethical Standards:

Conflict of Interest: All the authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Yang, M.-H., Kriegman, D.J., Ahuja, N.: Detecting Faces In Image : A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 24, 34–58 (2002).
2. Zhang, C., Zhang, Z.: A Survey of Recent Advances in Face Detection. Microsoft Res. 17 (2010).
3. Zafeiriou, S., Zhang, C., Zhang, Z.: A survey on face detection in the wild: Past, present and future. Comput. Vis. Image Underst. 138, 1–24 (2015).
4. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. I–I (2001).
5. Chen, D., Ren, S., Wei, Y., Cao, X., Sun, J.: Joint cascade face detection and alignment. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 109–122 (2014).
6. Zhang, K., Zhang, Z., Li, Z., Member, S., Qiao, Y., Member, S.: Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. IEEE Signal Process. Lett. 23, 1499–1503 (2016).
7. Li, Haoxiang and Lin, Zhe and Shen, Xiaohui and Brandt, Jonathan and Hua, G.: A Convolutional Neural Network Approach for Face Detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5325–5334 (2015).
8. Dai, D., Tan, W., Zhan, H.: Understanding the Feedforward Artificial Neural Network Model From the Perspective of Network Flow. arXiv Prepr. arXiv1704.08068. (2017).
9. Ruder, S.: An Overview of Multi-Task Learning in Deep Neural Networks. arXiv Prepr. arXiv1706.05098. (2017).
10. Wei, L.-Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00. pp. 479–488 (2000).
11. Data/Code Section, http://ailab.space/projects/multimodal-human-intention-perception/ -- Last accessed Jan-2019.
12. Shen, J., Zafeiriou, S., Chrysos, G.G., Kossaifi, J., Tzimiropoulos, G., Pantic, M.: The First Facial Landmark Tracking in-The-Wild Challenge: Benchmark and Results. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1003–1011 (2016).
13. Salam, H., Séguier, R.: A survey on face modeling: building a bridge between face analysis and synthesis. Vis. Comput. 34, 289–319 (2018).
14. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, a: Face recognition: A literature survey. Acm Comput. Surv. 35, 399–458 (2003).
15. Bulbul, A., Cipiloglu, Z., Capin, T.: A color-based face tracking algorithm for enhancing interaction with mobile devices. Vis. Comput. 26, 311–323 (2010).
16. Kalal, Z., Mikolajczyk, K., Matas, J.: Face-TLD: Tracking-learning-detection applied to faces. In: Proceedings - International Conference on Image Processing, ICIP. pp. 3789–3792 (2010).
17. Singh, C., Walia, E., Mittal, N.: Robust two-stage face recognition approach using global and local features. Vis. Comput. 28, 1085–1098 (2012).
18. Kumar, N., Peter, A.C.B., Belhumeur, P.N., Abstract, S.K.N.: Attribute and Simile Classifiers for Face Verification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 365–372 (2009).
19. Fu, Y., Guo, G., Member, S.: Age Synthesis and Estimation via Faces: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 32, 1955–1976 (2010).
20. Laurentini, A., Bottino, A.: Computer analysis of face beauty: A survey. Comput. Vis. Image Underst. 125, 184–199 (2014).
21. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: The state of the art. IEEE Trans. Pattern Anal. Mach. Intell. 22, 1424–1445 (2000).
22. Wang, Y., Zhang, L., Liu, Z., Hua, G., Wen, Z., Zhang, Z., Samaras, D.: Face relighting from a single image under arbitrary unknown lighting conditions. IEEE Trans. Pattern Anal. Mach. Intell. 31, 1968–1984 (2009).
23. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. Proc. 26th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '99. 187–194 (1999).
24. Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R., Seitz, S.M.: Exploring photobios. In: ACM SIGGRAPH 2011 papers on - SIGGRAPH '11. p. 1 (2011).
25. Wang, Z., Miao, Z., Jonathan Wu, Q.M., Wan, Y., Tang, Z.: Low-resolution face recognition: A review. Vis. Comput. 30, 359–386 (2014).
26. Li, Stan Z., Long Zhu, Z.Z.: Statistical Learning of Multi-view Face Detection. In: European Conference on Computer Vision. pp. 67–81 (2002).

27. Jones, M.J., Jones, M.: Fast multi-view face detection. Mitsubishi Electr. Res. Lab TR-20003-96. 3, 2 (2003).

28. Chua, T., Zhao, Y., Kankanhalli, M.S.: Detection of Human Faces in Compressed Domain for Video Strati cation 1 Introduction. Vis. Comput. 18, 121–133 (2002).

29. Viola, P., Jones, M.: Robust real-time face detection. Int. J. Comput. Vis. 57, 137–154 (2004).

30. Bradski, G.: The OpenCV Library. Dr. Dobb's J. Softw. Tools Prof. Program. 25, 120–123 (2000).

31. Wang, Y., Hu, S., Wu, S.: Object tracking based on Huber loss function. Vis. Comput. 1–14 (2018).

32. Yang, S., Luo, P., Loy, C.C., Tang, X.: From Facial Parts Responses to Face Detection: A Deep Learning Approach. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3676–3684 (2015).

33. Jiang, H., Learned-Miller, E.: Face Detection with the Faster R-CNN. In: 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). pp. 650–657 (2017).

34. Park, J., Kang, D.: Unified convolutional neural network for direct facial keypoints detection. Vis. Comput. (2018).

35. Dawoud, N.N., Samir, B.B., Janier, J.: Fast Template Matching Method Based Optimized Sum of Absolute Difference Algorithm for Face Localization. Int. J. Comput. Appl. 18, 975–8887 (2011).

36. Tan, T.K., Boon, C.S., Suzuki, Y.: Intra Prediction by Template Matching. In: International Conference on Image Processing. pp. 1–4 (2006).

37. Held, D., Levinson, J., Thrun, S., Savarese, S.: Robust real-time tracking combining 3D shape, color, and motion. Int. J. Rob. Res. 35, 1–28 (2015).

38. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-Speed Tracking with Kernelized Correlation Filters. IEEE Trans. Pattern Anal. Mach. Intell. 37, 583–596 (2015).

39. Wang, R., Dong, H., Han, T.X., Mei, L.: Robust tracking via monocular active vision for an intelligent teaching system. Vis. Comput. 32, 1379–1394 (2016).

40. Quan, W., Chen, J.X., Yu, N.: Robust object tracking using enhanced random ferns. Vis. Comput. 30, 351–358 (2014).

41. Quan, W., Jiang, Y., Zhang, J., Chen, J.X.: Robust object tracking with active context learning. Vis. Comput. 31, 1307–1318 (2015).

42. Gerónimo, D., Sappa, A.D., Ponsa, D., López, A.M.: 2D-3D-based on-board pedestrian detection system. Comput. Vis. Image Underst. 114, 583–595 (2010).

43. Xiao, J., Kanade, T., Cohn, J.F.: Robust full-motion recovery of head by dynamic templates and re-registration techniques. In: Proceedings - 5th IEEE International Conference on Automatic Face Gesture Recognition, FGR 2002. pp. 163–169 (2002).

44. Rehman, B., Hong, O.W., Tan, A., Hong, C.: Hybrid Model with Margin-Based Real-Time Face Detection and Tracking. In: The 11th Multi-disciplinary International Workshop on Artificial Intelligence (MIWAI). Lecture Notes in Computer Science. pp. 360–369. Springer, Cham (2017).

45. Rehman, B., Hong, O.W., Tan, A., Hong, C.: Using Margin-based Region of Interest Technique with Multi-Task Convolutional Neural Network and Template Matching for Robust Face Detection and Tracking System. In: Proceedings of 2nd International Conference on Imaging, Signal Processing and Communication (ICISPC) (2018).

46. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: A convolutional neural-network approach. IEEE Trans. Neural Networks. 8, 98–113 (1997).

47. Toshev, A., Szegedy, C.: DeepPose: Human pose estimation via deep neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1653--1660 (2014).

48. Ranjan, R., Sankaranarayanan, S., Castillo, C.D., Chellappa, R.: An All-In-One Convolutional Neural Network for Face Analysis. In: 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). pp. 17–24 (2017).

49. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. 55, 119–139 (1997).

50. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1026–1034 (2015).

51. Derpanis, K.G.: Relationship Between the Sum of Squared Difference (SSD) and Cross Correlation for Template Matching. York Univ. (2005).

52. http://ailab.space/wp-content/uploads/multimodal-human-intention-perception/FDTV10.zip.

Bacha Rehman is currently doing PhD in Computer Sciences in the Faculty of Science (FOS), Universiti Brunei Darussalam. His research interests are human robot interactions, computer vision, multimodal facial expression, machine learning, and deep neural network.

Ong Wee Hong received the B.Eng. in Communication and Control Engineering from the University of Manchester (1997), M.Sc. in Computing Science from the Imperial College London (2004) and PhD in Electrical and Information Systems from the University of Tokyo, Japan (2014). He is an Assistant Professor in Computer Science program in the Universiti Brunei Darussalam (UBD). He joined the UBD in 2007. His research interests are personal robots, ambient intelligence.

Abby Tan Chee Hong obtained his undergraduate degree (first class honours) in Mathematics at University of Manchester Institute of Science and Technology (UMIST) in 2002. He received PhD in Mathematics in 2006 from Manchester University. He joined Universiti Brunei Darussalam (UBD) and is currently senior lecturer in Mathematics in the Faculty of Science (FOS). He is an active researcher in financial mathematics and mathematics Education.

Trung Dung Ngo obtained his PhD in Robotics from Aalborg University, Denmark. He is an Associate Professor at University of Prince Edward Island, Canada. He is the founder and principal investigator of www.morelab.org. His research interests include Robotics and Intelligent Systems.