Driver Drowsiness Monitoring Based On Eye Map And Mouth Contour

Kamal Reddy, Amir Sikandar, Pranav Savant, Adityam Choudhary

Abstract: Number of accidents occurs now a days due to the drowsiness of driver. Therefore to assist the driver with the problem of drowsiness, the system must be design to carefully developed to provide an interface and interaction the make sense for the driver. This article introduces a new approach towards detection of drives' drowsiness based on yawning detection. The main aim is to reduce the number of accidents occurred due to drivers fatigue and hence increases the road safety. The special gestures of body and face are used as the sign of driver fatigue, including yawning, eye tiredness and mouth movement, and it indicate that the driver is no more in a proper driving condition. This involves several steps including the real time detection and tracking of driver's face detection, tracking of the mouth contour, eye and the detection of yawning based on measuring both the rate and the amount of changes in the mouth contour area, Eye detection using Eye Map.

Keywords: Active contour model, Canny Edge Detection, Eye Map, Yawn detection.

I. INTRODUCTION

According to the National Highway Traffic Safety Administration (NHTSA), driving while drowsy is a contributing factor to 22 to 24 percent of car crashes. The record shows that the car accidents caused by fatigue drivers is 4-6 times higher than near-crash/crash risk relative to alert drivers as fatigue drivers fail to take correct actions prior to a collision. This is mainly due to the fact that driver fatigue impacts the alertness and response time of the driver thus increases the chances of getting involved in car accidents. Drowsy drivers may fall asleep at the wheel or tend to make serious – sometimes fatal – driving errors. Drowsiness and fatigue can often affect drivers' ability long before they notice that they are getting tired. An important irony of driver drowsiness is that the driver may be too tired to realize his own level of inattention. Hence, driver monitoring systems which can detect lack of attention and alert the driver will play an important role in the automation system of future vehicles to prevent accidents and save lives. There are different signs and body gestures that can be monitored as indicators of driver fatigue. These include daydreaming while on the road, driving over the center line, yawning, feeling impatient, heavy eyes and slow reaction. A number of related work in the field have addressed the problem of detecting driver drowsiness based on the expressions of the driver's face (i.e. eye motion ,mouth position, yawning, etc.), that will be briefly reviewed in the related work section. In this paper we only focus on yawning as a sign of fatigue. Unlike existing systems that suffer from computational complexity, facial obstruction factors, and adverse lighting conditions. In our approach, the driver's face is continuously captured using a video camera that is installed under the front mirror inside the car.

- *Kamal Reddy, Amir Sikandar, Pranav Savant, Adityam Choudhary*
- *Department Of Computer Science Engg, Siddhant College Of Engineering, University Of Pune.*

Next, detecting drowsiness involves two main steps to properly measure changes in facial gestures that imply drowsiness. First, the driver's face is detected and tracked in the series of frame shots taken by the camera. After locating the driver's face, the next step is to detect and track the location of the mouth. We have chosen to detect and track the face prior to tracking the eye and mouth as this makes the eye and mouth tracking procedure more robust against false detections. After detection of the eye and mouth, the yawning state is detected based on measuring the rate of changes in the area of the mouth contour and the aspect ratio of mouth area. Figure 1 demonstrates several steps of the algorithm. The details of each step will be further explained in the following subsections. After discussing the related work in Section II, Section III describes our approach towards detection and tracking of the face. Section IV presents the eye tracking method and Section V presents the mouth contour tracking method section VI describes the yawning measurement process, before finishing the paper with summary and conclusions.

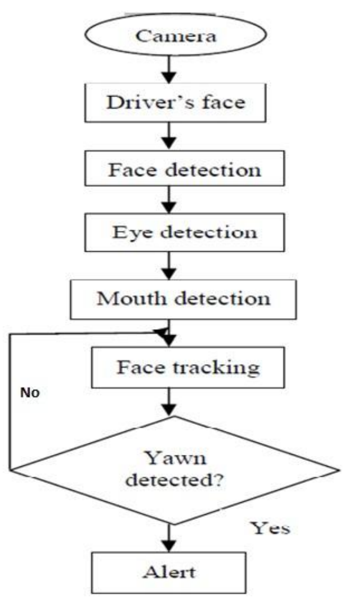

Figure 1

II. RELATED WORK

Driver drowsiness detection is well-studied and has been addressed in a number of related research works. [2] Proposes a yawning detection system based on the distance between the midpoint of nostrils and the chin. [3] Takes advantage of grey projection and Gabor wavelets to detect the mouth corners and uses LDA to find a linear combination of the those features to detect the yawning mouth. [4] Detects the face using Viola- Jones technique and extracts the mouth region, in which lips are searched for through spatial fuzzy c-means (s-FCM) clustering. The proposed system in [5] requires the use of two cameras: a low resolution camera for the face and a high resolution one for the mouth. It uses haar-like features to detect driver's mouth, and yawning is detected based on the aspect ratio of the mouth. [6] proposes the use of mouth geometrical features to detect yawning. [7] introduces a driver drowsiness monitoring system based on a combination of eyes and mouth gestures. The proposed system determines the state of mouth and eyes by analyzing their feature points using Back Propagation neural networks in order to checks for conditions that involve driver drowsiness. [8] uses the cascade of classifier as proposed by Viola-Jones face detection approach. It then uses a support vector machine to train the classifier with the mouth features in yawning condition. Another approach to facial movement analysis has been proposed in [9] by using Adaboost and multinomial ridge regression to train the classifier of different facial actions such as blinking and yawn motions. Despite considerable research in this area, today only a few expensive yawning monitoring systems exist in some luxury cars that still suffer from a high rate of false positive detection and do not have sufficient accuracy [10] . There are three main reasons why existing techniques, such as the ones described above, are not robust enough for a production-grade and serious consumer system: computational complexity, facial obstruction, and lighting conditions. To the best of our knowledge, most existing techniques have a high processing load and computational complexity and cannot satisfy the real-time requirements of resource-limited embedded smart camera systems. Most of the existing works are either lab reports without an actual car deployment, or use a high-end laptop/desktop platform to run their method, which is far from a practical and economical compact system that can be installed in a car. More research is therefore needed to develop a computer vision based platform for a realistic in-car smart camera system. Facial obstruction and lack of resiliency against it is another shortcoming of the related research. The presence of beards, mustache, glasses, and sunglasses create a great deal of challenge. Sunglasses in particular make it difficult as they are commonly used during driving and not only cover a considerable area of the face but also cause a shadow on it, rendering many of the existing detection technologies ineffective. Finally, most of the cited works suffer from requiring constant illumination and fixed lighting source, which doesn't necessarily exist in car driving at all times. Lighting conditions can change depending on time of the day, and face detection becomes most challenging when there is not enough lighting or the lighting source is un i-directional resulting in shadows on the face. Therefore, existing systems that operate perfectly under controlled lighting conditions fail to perform well in actual driving scenarios. Our proposed approach aims at eliminating yawning detection's dependency on the previously mentioned factors such as facial occlusion and lighting change. At the same time, we do not to use complex algorithms in order to have a realistic implementation with a reasonable computational footprint that can run on automotive smart camera systems such as APEXTM. We have also avoided using techniques that are based on classifiers in order to alleviate the need for a large training database. However, we have tried to main a high level of detection efficiency when optimizing other aspects of the system such as complexity and ease of implementation. The presented work is an extension of the research previously published by the authors [11]. The extension includes the use of more advanced algorithms for face detection and mouth tracking leading to a higher efficiency and robustness of the system, as well as the introduction of a new and more efficient algorithm for yawning measurement based on mouth contour area monitoring. The following sections will describe our proposed approach in detail. We start by the first stage of our method: face detection and tracking.

III. FACE DETECTION AND FACE TRACKING

The first steps towards yawning detection is the detection and tracking of the driver's face. Given a single image, the goal of face detection is to identify all image regions that contain a face regardless of its position, orientation, and lighting conditions. Such a problem is challenging because faces are non rigid and have a high degree of variability in size, shape, color, and texture [12]. The orientation of the face can also be a challenge in the detection process. However, we have assumed that the monitoring camera is installed inside the vehicle under the front mirror facing the driver at a fixed angle. Therefore the problem of relative camera-face pose is less challenging in our application while head position might still vary from driver to driver. There is also a great deal of variability among faces including shape, color, and size. Presence of facial features such as beards, moustaches, and glasses can also make a great deal of difference. The other important factor is lighting conditions. This is mainly affected by the environment light that can change depending on the time of the day and weather conditions. [13] addresses the problem of face detection in the presence of illumination variations and proposes a solution based on the use of an adaptive illumination normalization procedure. [14] takes advantage of infrared images in proxy with visible range images in order to solve the problem of illumination variation while having access the more detailed information present in visible images. While in our system we do not use infrared cameras, due to economic reasons, similar to [13] we also address the problem of illumination variation, as will be described in details later. For face detection itself, several approaches have been used in the related literature. Knowledge based methods [15] try to encode human knowledge about the characteristics of a typical face, such as the relationships between facial features, and use them as a way to detect faces in an image. Feature invariant approaches [16][17] aim to find structural face features, such as eyebrows, eyes, nose, mouth, and hairline, which persist under various poses, viewpoints, or lighting and use those features to detect faces. Such features are mostly extracted using Canny edge detection[37][38]. The texture of human faces [18] or human skin color [19][20] has also proven to be an effective feature that can be used towards face detection. Such methods can take advantage of Neural Networks or

Support Vector Machines (SVM)[23], Naïve Bayes Classifiers [24], or Hidden Markov Models (HMM) [25] as tools to evaluate the matching of the pattern to the training database. Current state of the art face detection systems are mostly based on the use of classifiers. The most famous and commonly used face detection scheme in this category is the viola-Jones face detection algorithm [26] as implemented in OpenCV. It is able to efficiently detect neutral faces as it has been trained with a large database of faces. However, when it comes to a yawning face, it sometime fails to detect the whole face down to the bottom of the chin especially when the mouth is wide open. As we don't have access to a database of thousands of yawning faces, we have decided to adapt a face detection approach that is not based on the use of classifiers. Our approach can be summarized in the following main steps: In order to locate a face in the image, the possible locations of the face are first estimated based on skin color segmentation. The use of human skin characteristics in face location will be further discussed in subsection A. It should be noted that the results from skin segmentation is subject to high percentage of error resulting in regions in the background that might have been falsely segmented as skin regions. Therefore, we will do a number of post processing steps in order to find the correct face location. The post processing phase stars by searching for possible face candidates around the location that has been segmented as human skin regions. To verify the possible matching of a candidate region to a face, we first apply the Integration Projection to the region . The wavelet transform will highlight the edges and helps removing the background noise. We then calculate the vertical projection and the horizontal projection of the upper half of the candidate region and measure its projection similarity with template projections of a typical face. The face location is the region where the similarity degree is at its maximum. The details of each step will be further explained in the following subsections.

A. Skin Segmentation

Given a single image, the goal of face detection is to identify all image regions, which contain a face regardless of its position, orientation, and lighting conditions. Such a problem is challenging because faces are non rigid and have a high degree of variability in size, shape, color, and texture [7]. It is basically assumed that the camera is installed inside the vehicle facing the driver at a fixed angle. Therefore the problem of relative camera-face pose is less challenging in our case while head position might still vary from driver to driver. There is also a great deal of variability among faces including shape, color, and size. Presence of facial features such as beards, mustaches, and glasses can also make a great deal of difference. The other important factor is the lighting conditions. This is mainly affected by the environment light that can change depending on the time and weather conditions. Keeping all the above considerations in mind, the most functional way to detect face is by detecting the skin color and texture. However, it should be noted that the detections scheme should be invariant to skin type and change in lighting conditions. Therefore we take advantage of a set of bounding rules for different color space (RGB, YCbCr and HSV) in order to improve the detection efficiency [8]. RGB color space is used to detect skin color at uniform or lateral daylight illumination and under flashlight illumination:

(1)

(RED>95) AND (GREEN > 40) AND (BLUE > 20)
AND(max{RED,GREEN,BLUE}-
min{RED,GREEN,BLUE} >15)
AND (|RED - GREEN| > 15) AND (RED > GREEN)
AND (RED > BLUE) AND (RED > 220) AND (GREEN > 210) AND (BLUE > 170) AND (RED > BLUE) AND (GREEN > BLUE)

Cb-Cr color space is a strong determination of skin color. The following rules apply to this color space:

(2)

(Cr <= 1.5862 * Cb + 20) AND
(Cr >= 0.3448 * Cb + 76.2069) AND
(Cr >= -4.5652 * Cb + 234.5652) AND
(Cr <= -1.15 * Cb + 301.75) AND
(Cr <= -2.2857 * Cb + 432.85)

The last space to be used is the HSV space. Hue values exhibit the most noticeable separation between skin and nonskin regions.

(3)

H < 25 and H > 230

The result of the skin location technique is a black and white image which highlights the skin location by converting the face to white and the background and the areas around the driver to black. This background elimination reduces the subsequent errors due to false object detection in the background. The face is detected by finding the biggest white connected component and will cut that area.

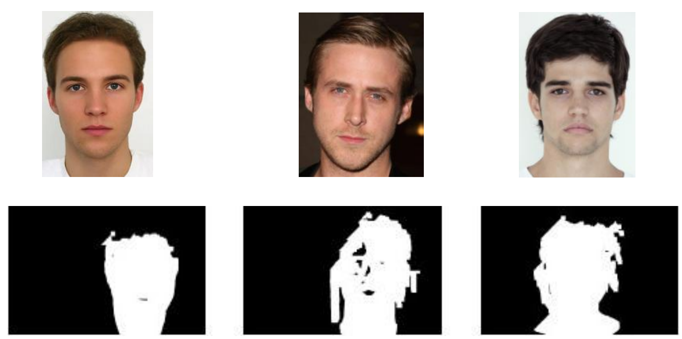

Figure 2. Face feature detection

B. Face Detection Based On Template Matching.

Once the possible face regions are detected based on skin segmentation, template matching can be applied to the area around each region in order to find the exact location of the face. In order for the system to have a better performance, template matching is performed on the image itself. We will later describe the details of each step and explain the reasons behind their choice. But before that, let us briefly review the theory of projection that will be later used in the proposed algorithm:

1. Integration Projection.

Integration projection is one of the most commonly used methods to extract the features of an object in image processing. The basic idea behind it is to map a two dimensional function into a one single dimensional vector by scanning it in the horizontal or the vertical direction. The resulting vector is calculated by summing up the function values along the specified direction (vertical or horizontal). Assuming a 2D function, f(i,j), the horizontal and vertical projections can calculated as shown in equations (6) and (7).

$$U[i]=\sum_{K=1}^{N} f(i,k), l<i<M \qquad (6)$$

$$V[i]=\sum_{K=1}^{M} f(i,k), l<j<N \qquad (7)$$

Our proposed approach takes advantage of the projection operation to transform the two dimensional images into a one dimensional function known as the *image profile*. The reason is that the profile of a typical face has special characteristics that can be used in the face detection process, and the matching of profiles is easier compared to the matching of images themselves. We will later demonstrate that detection efficiency is higher in our approach as the detection dependency on variations of the faces is lower. Through the rest of the paper, we will discuss the use of projection operation in our face detection algorithm.

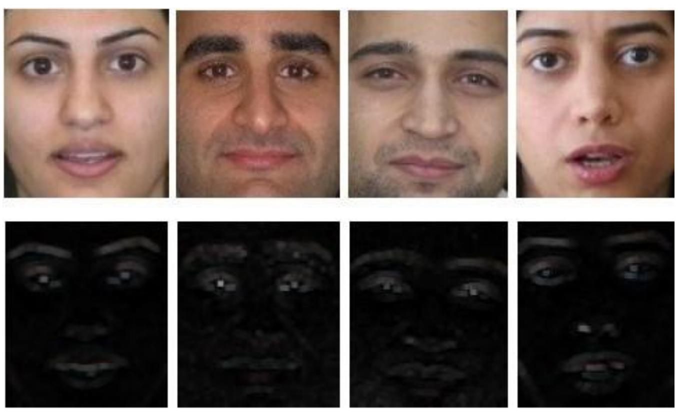

Figure 3. Level 1 Daubechies Wavelet Transform

2. Profile matching

The profile matching process involves the final detection of the face location among the approximate results from skin segmentation. First, we apply a level1 Daubechies wavelet to the image and use the horizontal component of the wavelet for further processing. The reason behind the use of the horizontal component of the wavelet is that the face is better characterized with horizontal edges rather than vertical ones. The DTW is applied to the intensity channel of a color image resulting in enhancing the edges of the image. The intensity of the pixels in the gray image is proportional to frequency components in their neighborhoods. Therefore, the skin region of the face will turn black as there is no significant high frequency component in that area. On the edge of the face where facial features happen, the pixels will get the value close to 1 since there will be a color change, i.e. from skin to eyebrow, skin to eyes or skin to mouth on the face. The original image typically includes a high amount of undesired details that affect the detection efficiency. The human face has a specific vertical and horizontal projection profile that can be used in the face detection process. Face detection basically involves the matching of the profile of different image regions to a template face profile in order to find the region with the highest similarity as the face region. The projection profiles that we will use in the matching process are the Y-profile and the X-profile of the upper part of the face as the shape and the location and maxima and minima of these profiles pretty much define the geometrical characteristics of the human face such as the relative location of the eyes, mouth and nose. Prior to the start of the profile matching process, we need template X and Y profiles that best represent a face profile. The template profile image is found by applying the wavelet transform and integration projection to the images of a face database and averaging over the whole normalized set. The resulting projection will then be used as the template in the detection process. We have also applied the approximate wavelet to the template profiles to make it smoother and eliminate the undesired noise resulting from the variance among different faces. Figure 4. shows the Y-profile and Xprofile of the template face. As can be seen in the figure, the Y-profile has peaks that relate to the location of eye-brows, eyes, nose and mouth and upper face, while the X-profile has peaks that relate to the location of the eyes [30].

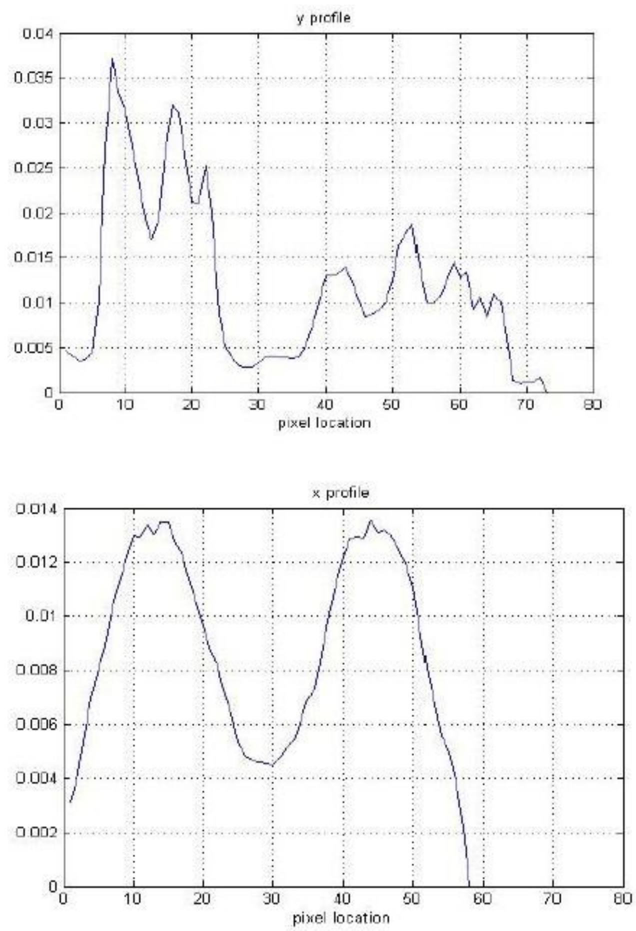

Figure 4. Template Y-Profile and X-Profile used in the matching process

As mentioned before, the matching of the template face profiles to the profile of a region can be a measure of similarity of that region to a face. However, as faces have different geometrical ratios, the careful choice of a similarity measure is essential in the matching process. The similarity measure that we propose to use here is Dynamic Time Warping (DTW). DTW is an effective technique for measuring the similarity between two sequences independent of the shift and the scaling. Therefore, it is a good similarity measurement technique for the face as it can reveal the similarity of two face profiles even if the faces have different sizes or the location of facial features slightly shift; e.g., one has a longer forehead than the other. The face detection procedure starts by sliding a window over the image around the centroid points from the skin segmentation step. The wavelet transform and Y-projection and upper part X-projection of the region located inside the window is then calculated resulting in two profiles. The last step would be measuring the DTW distance between the two profiles and the template Y-profile and X-profile. The sum of DTW distance between the two profiles can be assumed to be inversely proportional to the similarity score of the region in the window to a face. At the end of the template matching process, the location of the face is defined to be the region whose profile has the shortest distance to the template profile according to the DTW measure. It should be noted that the above face detection process is performed very infrequently; i.e., only once at the beginning, or when a face is completely lost and needs to be re-detected. Once the face is detected, it is then continuously tracked with a much simpler and less time consuming process, as described next.

C. Face Tracking Using Canny Edge Detector.

Having found the face in one frame, we can then use the detected face as a template and start tracking it in the subsequent frames. Tracking the face involves much simpler and less time consuming operations than face detection, allowing the monitoring system to operate in real time with a reasonable amount of processing power. To track the face, we use a canny edge detector. The purpose of edge detection is to reduce the amount of data in an image, while preserving the structural properties of image to be used for further image processing. The algorithm runs in 5 separate steps:

1. ***Smoothing Process*****:** Blurring of the image to remove noise.

2. ***Gradients Detection*****:** The edges is marked where the gradients of the image has large magnitudes.

3. ***Non-maximum suppression*****:** Only local maxima is marked as edges.

4. ***Double thresholding Mechanism*****:** The Potential edges can be found easily by thresholding.

5. ***Edge tracking*****:** Then Final edges are determined by combining all the edges that are not connected.

Each step is described in detail clearly below:

Figure 5: Test Image used for Canny edge detection.

2.1 Smoothing

It is not necessary that all images contain noise which has taken from camera. To prevent that noise which is mistake for the edges, noise must be reduced at any cost. Therefore the image is smoothed by using a Gaussian filter. The Gaussian filter kernel with a standard deviation of σ = 1.4 is shown in Equation (7). The effect of smoothing the original image with this filter is shown in Figure 6.

$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (7)$$

Figure 6: The original image is smoothed with a Gaussian filter to eliminate noise.

2.2 Gradients Detection

The Canny algorithm is used to find the edges in the image where the grayscale intensity of the original image changes the mostly. These areas can be evaluted by determining the gradients of image. Gradients of each pixel can be determined by using sobel operator in the smoothed image. First we approximate the gradient in the x-direction and y-direction by applying the kernels shown in Equation (8),(9).

$$KGX = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad (8)$$

$$KGY = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \qquad (9)$$

The gradient magnitudes can then be measure as an Euclidean distance measure by using the law of Pythagoras in Equation below. It can be sometimes simplified by applying Manhattan distance measure formula is used to reduce the computational complexity as given below in Equation. Euclidean distance measure is applied to the original image. The computed edge strengths are compared with the smoothed image.

$$|G|=\sqrt{Gx^2+Gy^2} \qquad (10)$$

$$|G| =|Gx|+|Gy| \qquad (11)$$

where: *Gx*= gradients in the x-direction and *Gy* = gradients in the y-directions. An image of the gradient magnitudes oftenly indicate the edges quite clear. However, the edges are typically broad enough and thus do not indicate exactly where the edges are present. To make it possible to determine , the direction of the edges must be determined & stored as shown in the Equation below.

$$\theta=\arctan(|Gy| / |Gx|) \qquad (12)$$

Figure 7: The gradient magnitudes present in the smoothed image using Sobel-operator.

2.3 Non-maximum suppression

The main purpose of non-maximum suppression is to convert the “blurred” edges of the image of the gradient magnitudes to“sharp” edges. This is done to preserve all the local maxima present in the gradient image, and delete other things. This algorithm is for every pixel in the gradient image:

1. Round up the gradient direction Ө to the nearest 45◦, corresponding to the use of an 8-connected neighbourhood.
2. Compare all the edge strength of current pixel with edge strength of the pixel present in the Positive gradient & negative gradient direction. , compare it with the pixels to the north & south direction.
3. When edge strength of current pixel is greater; preserve thevalue of the edge strength else suppress remove the value.

Figure 8: Non-maximum suppression of Original Image and the Edge-pixels are preserved where the gradient has local maxima mostly.

2.4 Double thresholding

After the non-maximum suppression step the edge pixels remain and marked with the strength pixel by pixel. Many edges are true edges in the image, but some are caused due to noise or color variation for an instance due to the rough surfaces. The easy way would be to utilize a threshold, to find the only edges stronger having appropriate value would be preserved. The Canny edge detector algorithm uses double thresholding mechanism. Edge pixels having strong threshold are marked as strong; edge pixels having weaker than the low threshold are suppressed & edge pixels between the two thresholds value are marked as weak.

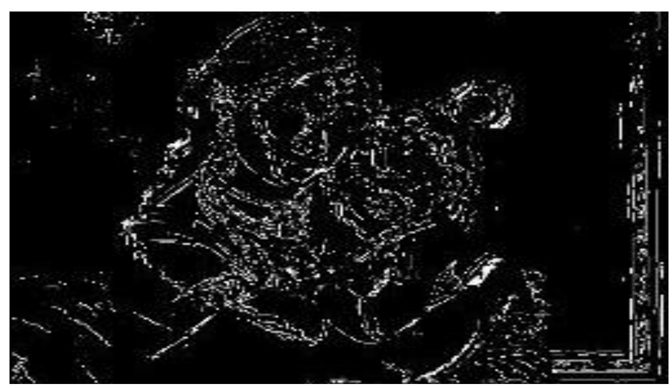

Figure 9: Thresholding of the edges. In the image stronger edges are white, while weaker edges are grey.

2.5 Edge tracking by hysteresis

Strong edges are consider as certain edges, & immediately included in the final edge image. Weak edges are consider when they are connected to strong edges. The other logic such as noise & other small variations are unlikely to give result in a strong edge. Thus strong edges are present due to the true edges in the original image. Weak edges can be present due to true edges or noise or color variations. It uniformly distributed on the entire image, & only a small amount will be located at the strong edges. Due to true edges the weak edges are connected directly to the strong edges. Edge tracking can be also implemented by using BLOB-analysis (Binary Large Object). The connected edge pixels are divided into BLOB's using 8-connected neighborhood. BLOB's analysis should containing at least one strong edge pixel which are then preserved, while other BLOB's are suppressed.

Figure 10: Edge tracking and final output.

The last step in the algorithm is edge tracking can be implemented in two way either iterative or recursive BLOB analysis [40]. A recursive implementation use the grass-fire algorithm for edge tracking. Our implementation goes with the iterative approach. First of all weak edges are scanned for the neighbour edges and joined them into a groups. At the same time adjacent group are marked. Then all of these markings are analyse to determine which groups of weak edges are connected to strong edges directly or indirectly. All the weak edges which is connected to strong edges are marked as strong edges. The rest of the weak edges are unavoidable. This can be interpreted as BLOB analysis where only BLOB's containing strong edges which are preserved.

IV. EYE DETECTION & TRACKING

After detecting the face, the location of the eyes will be detected. The main reason behind locating the eyes is to use them as a verification method in order to make sure that the location of the mouth in face is correctly detected (using the geometrical relation between eyes and mouth in human face). In order to detect the eyes, the eye maps based on chrominance components are built [9] according to the following equation:

$$\text{Eye\_location} = \frac{1}{3}\left\{(C_b)^2 + (C_r)^2 + \left(\frac{C_b}{C_r}\right)\right\}$$

The eye map highlights the eyes regions. We can then convert the eye map image to a black and white image using proper thresholding. This new image is supposed to include the eyes in white while the rest is all black. However, several pre-processing steps including erosion, dilation and finding the biggest connected components as eyes are required. Moreover, we use some geometrical features of the eyes in the final step to reject the false detections. Therefore we do not use the geometrical features for detection and we rather use them only for verification purpose.

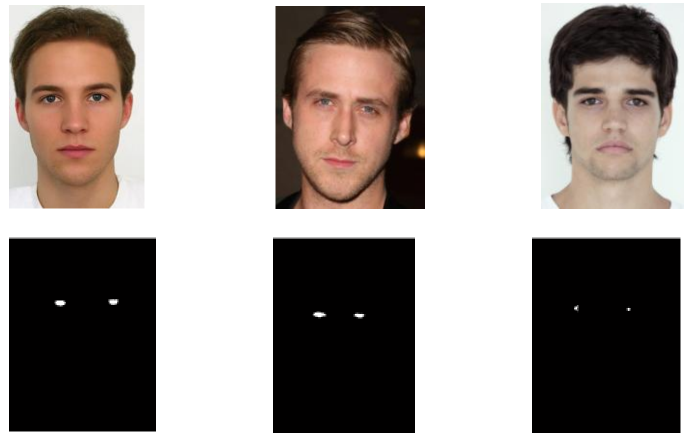

Figure 11. Eye detection using Eye Map.

V. MOUTH DETECTION & TRACKING

A. Color segmentation of the mouth region

After the detection of the face, the first step towards mouth segmentation is the usage of color properties of the mouth in order to highlight the mouth region in the bottom half of the face area. The red color is the strongest component in the mouth area while the blue color is the weakest [28]. Therefore, the Cr component is bigger than the Cb component in the mouth area. The following equations can be applied to the face in order to highlight the mouth area in the face region:

$$\text{Mouth\_map} = (C_r)^2 \times \left((C_r)^2 - \frac{\eta \times C_r}{C_b}\right)^2$$

$$\eta = 0.95 \frac{\frac{1}{n}\sum_{(x,y)} C_r(x,y)^2}{\frac{1}{n}\sum_{(x,y)}\left(C_r(x,y)/C_b(x,y)\right)}$$

The output of applying the mouth map to the face is a gray scale image where brighter pixels represent the mouth area. Even though the mouth area is highlighted in the output, there might be some other regions that have been falsely classified as mouth according to the mouth map criteria. Therefore, we perform a post processing step to find the mouth contour with a better precision. The post processing step involves the use of the active snake contour technique to extract the mouth contour around the area highlighted by the mouth map. The use of active contours for lip contour detection helps removing the noise that is present due to the inaccurate color segmentation of the mouth by the use of the mouth map. The following subsection presents further details about the use of active snake contours in mouth tracking.

B. Active snake contours and application to mouth tracking

The snake method [32][32] aims at spline interpolation of lines around object boundaries. It starts from a set of initial points in an image and deforms in several iteration to reach the object boundaries in the image. The change in the shape of the snake is made in a way to minimize three sets of energies known as internal, external, and constraint energies. Internal energies determine the internal energy of the spline due to stretching and bending. External energies are defined by image boundaries and cause the snake to be attracted towards lines, edges, and termination points in the image. Therefore, the value of the external energy can be computed as the sum of the energies from lines, edges and termination points in the image as defined in equation (11):

$$E_{\text{Image}=}W_{line}E_{line}+W_{edge}E_{edge}+W_{term}E_{term} \quad (11)$$

Finally, the constraint energies are related to external constraint forces. The total image energy can be expressed as a weighted combination of the three previously mentioned energy functions. The overall energy function minimized by the snake is defined in equation (12):

$$E_{snake}=\int_0^1(E_{\text{internal}}v(i)+E_{external}v(i)+E_{constraint}v(i))di \quad 12)$$

where v is the set of points on the snake. Once the snake reaches the boundary of the object, it starts to settle down in the boundary and moves slower. If the object slightly moves, snake is able to track the motion and resides in the new boundary. The dynamic update feature of the snake contour makes it an interesting choice for lip contour tracking. The use of snake contours in lip reading has been previously discussed in [33][34][35]. We have used the snake in the application of yawning detection by finding a snake on the external lips, and updating the snake as the shape of the lips change over time. Figure 12. shows the result of finding the mouth snake contour for a sample image.

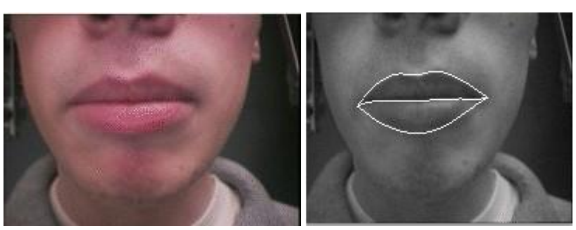

Figure 12. An example of mouth contour detection

VI. Yawning Detection

In the final stage and after having reliably found the mouth, we will detect yawning by measuring certain parameters as described next. The yawn is assumed to be modelled with a large vertical mouth opening. When the mouth starts to open, the mouth contour area starts to increase in subsequent frames[36]. The rate of such increase can be used as an indication of yawning. The number of states to the point where the mouth is wide open is larger in yawning compared to speaking and can be used in differentiating these two states We then applied the yawning detection algorithm to a large number of videos (100+) that have been recorded by the authors. The videos have been recorded in various conditions such as different light reflection and directional lightings. The features of the subjects have also been chosen to vary in terms of skin color, haircuts, having beard, eye glasses. The algorithm has been able to detect the yawn with a high precision detection rate. shows one such video and the use of the snake contour for tracking the mouth in a sequence of frames in the video. Figure 13. demonstrates the changes in the mouth contour area in the 38 frames when a yawn is happening. It should be noted that the original video has been recorded in 30 FPS. A complete yawn takes about 100 frames to happen. However; we have down sampled the video frames before processing them as the frames include a large amount of redundancy and all of them need not to be processed for the task of yawning detection.

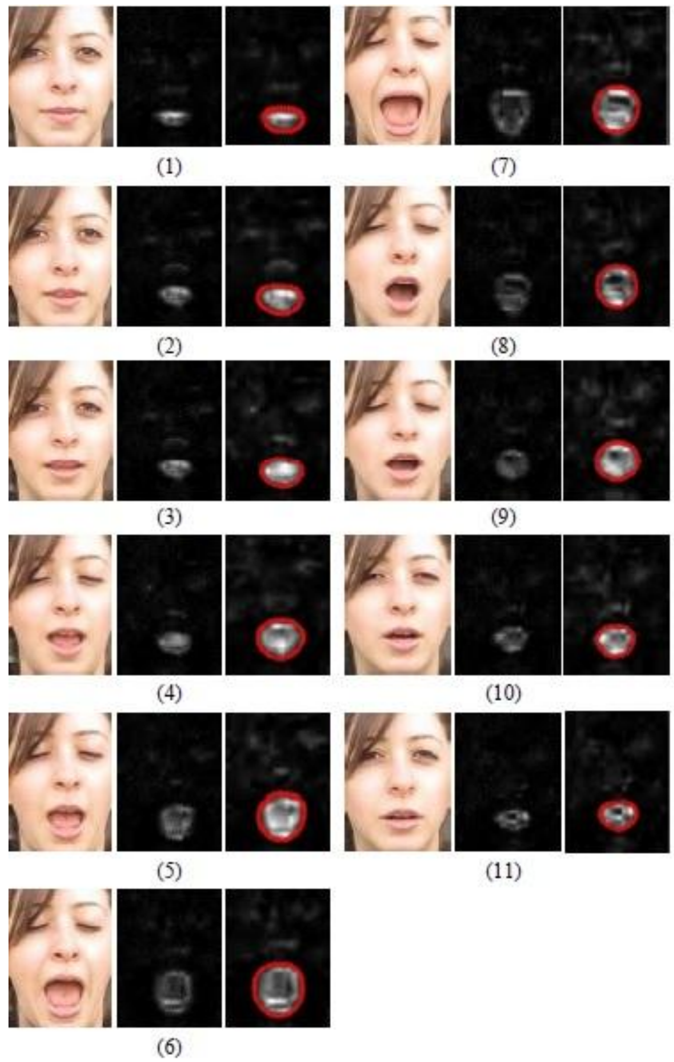

Figure 13. Sequence of mouth contours in yawning

VII. Conclusion

This article introduced a new measurement method to detection yawning of a driver. The system includes several blocks that perform face detection, tracking, eye tracking and mouth contour tracking. The yawning state is determined based on measuring the rate and amount of changes in the mouth contour area and eye movements. The proposed face detection and tracking approach is based on both properties of human skin as well as face projection profiles in terms of location of eyebrows, eyes, nose and mouth. The algorithm has been designed in a way that it is not much sensitive to changes in lighting conditions and skin types as it does not only rely on skin color properties. Moreover, we have chosen to avoid the use of complex algorithms and large training sets in order to make feasible implementation of the system in actual industry products.

REFERENCES

[1]. S.G. Klauer , T. A. Dingus, Neale, V. L., Sudweeks, J.D., and Ramsey, DJ,"The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data," Virginia Tech Transportation Institute, Technical Report # DOT HS 810 594

[2]. U. Yufeng, W. Zengcai, "Detecting driver yawning in successive images." In: Proc. 1st International Conf. on Bioinformatics and Biomedical Engineering, 2007 , pp. 581-583.

[3]. X. Fan, B. Yin, Y. Fun. "Yawning Detection For Monitoring Driver Fatigue." In: Proc. Sixth International Conf. on Machine Learning and Cybernetics, Hong Kong, 2007, pp. 664-668.

[4]. T. Azim, M.A. Jaffar, A.M. Mirza. "Automatic Fatigue Detection of Drivers through Pupil Detection and Yawning Analysis." In: Proc. Fourth International Conf. on Innovative Computing, Information and Control, 2009, pp. 441-445.

[5]. L. Li, Y. Chen , Z. Li. "Yawning Detection for Monitoring Driver Fatigue Based on Two Cameras." In: Proc. 12th International IEEE Conf. on Intelligent Transportation Systems, St. Louis, MO, USA, 2009, pp. 12-17.

[6]. T.Wang, P. Shi. "Yawning Detection For Determining Driver Drowsiness." IEEE International Workshop VLSI Design & Video Tech. Suzhou,China, 2005, pp. 373-376.

[7]. Y. Ying , S. Jing, Z. Wei, "The Monitoring Method of Driver's Fatigue Based on Neural Network", In: Proc. International Conf. on Mechatronics and Automation, China, 2007 , pp. 3555-3559.

[8]. M. Saradadevi , P. Bajaj, "Driver Fatigue Detection Using Mouth and Yawning Analysis" , IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6, June 2008- pp. 183-188

[9]. E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett and J. Movellan, "Drowsy Driver Detection Through Facial Movement Analysis"

[10]. K. Barry, "Yawn If You Dare. Your Car Is Watching You", Wired magazine, Autopia section, July 30, 2009, http://www.wired.com/autopia/2009/07/sleep-detection/

[11]. S. Abtahi, B. Hariri, S. Shirmohammadi, "Driver Drowsiness Monitoring Based on Yawning Detection", Proc. IEEE International Instrumentation and Measurement Technology Conferencet, Binjiang (Hangzhou), China, May 10-12, 2011

[12]. M.H. Yang, D.J. Kriegman, N. Ahuja,: "Detecting faces in images: A survey." IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, pp. 34-58, 2002

[13]. H. Sellahewa, S.A. Jassim, "Image-Quality-Based Adaptive Face Recognition," Instrumentation and Measurement, IEEE Transactions on , vol.59, no.4, pp.805-813, April 2010.

[14]. W. Jian-Gang , E. Sung , "Facial Feature Extraction in an Infrared Image by Proxy With a Visible Face Image," Instrumentation and Measurement, IEEE Transactions on , vol.56, no.5, pp.2057-2066, Oct. 2007

[15]. G. Yang and T. S. Huang, "Human Face Detection in Complex Background," Pattern Recognition, vol. 27, no. 1, pp. 53-63, 1994.

[16]. K.C. Yow and R. Cipolla, "Feature-Based Human Face Detection," Image and Vision Computing, vol. 15, no. 9, pp. 713-735, 1997.

[17]. T.K. Leung, M.C. Burl, and P. Perona, "Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching," Proc. Fifth IEEE Int'l Conf. Computer Vision, pp. 637-644, 1995.

[18]. Y. Dai and Y. Nakano, "Face-Texture Model Based on SGLD and Its Application in Face Detection in a Color Scene", Pattern Recognition, vol. 29, no. 6, pp. 1007-1017, 1996.

[19]. R. Kjeldsen and J. Kender, "Finding Skin in Color Images," Proc. Second Int'l Conf. Automatic Face and Gesture Recognition, pp. 312-317, 1996.

[20]. S. McKenna, S. Gong, and Y. Raja, "Modelling Facial Colour and Identity with Gaussian Mixtures," Pattern Recognition, vol. 31, no. 12, pp. 1883-1892, 1998.

[21]. I. Craw, D. Tock, and A. Bennett, "Finding Face Features," Proc. Second European Conf. Computer Vision, pp. 92-96, 1992.

[22]. A. Lanitis, C.J. Taylor, and T.F. Cootes, "An Automatic Face Identification System Using Flexible Appearance Models," Image and Vision Computing, vol. 13, no. 5, pp. 393-401, 1995.

[23]. E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 130-136, 1997.

[24]. H. Schneiderman , T. Kanade, “Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition,” IEEE Conf. Computer Vision and Pattern Recognition, pp. 45-51, 1998.

[25]. A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri, “Finding Faces in Photographs,” IEEE Int’l Conf. Computer Vision, pp. 640- 645, 1998.

[26]. P. Viola and M. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), ISSN: 1063-6919, Vol.1, pp. 511-518, December 2001.

[27]. N. A. A.Rahman, K.C. Wei and J. See. “RGB-H-CbCr Skin Colour Model for Human Face Detection.” In Proceedings of The MMU International Symposium on Information & Communications Technologies, 2006

[28]. Hsu Rein-Lien, M. Abdel-Mottaleb, and. A. K. Jain. “Face detection in color images.” IEEE Trans. Pattern Analysis and Mechine Intelligence, Vol. 24, issue 5. 2002

[29]. C. Chen, “Information Fusion of wavelet projection features for face recognition” In: Proc. International Joint Conf. on Neaural Network, Hong Kong, September 2008 , pp. 3881-3887.

[30]. L.Zhen-yu, W. Miao-miao, “ Eyes detection based on One-Dimensional Wavelet Transform”, International Symposium on Computational Intelligence and Design, Hangzhou, China 2010, pp. 123-127

[31]. Z. Yan, M.B Yeary., S. Cheng, N. Kehtarnavaz, , "An Object-Tracking Algorithm Based on Multiple-Model Particle Filtering With State Partitioning," Instrumentation and Measurement, IEEE Transactions on, vol.58, no.5, pp.1797-1809, May 2009

[32]. M. Kass, A. Witkin, D. Terzopoulos,” Snakes: Active contour models”, In’l Journal of Computer Vistion, vol. 1, no. 4, pp. 321-331, 1987

[33]. M. Barnard, E.Holden, R. Owens,”Lip tracking using pattern matching snakes”, The 5th Asian conference on Computer Vision, Melbourne, Australia, January 2002, pp. 1-6

[34]. G.I. Chiou, J.Hwang,“Lipreading from color video” IEEE Transactions on Image Processing, vol. 6, no. 8, pp. 1192-1195, August 1997

[35]. R. Séguier and Nicolas Cladel, Genetic Snakes. Application on Lipreading, International Conference on Artificial Neural Networks and zGenetic Algorithms (ICANNGA), 2003.

[36]. A. Benoit, A. Caplier, “ Hypovigilence analysis: open or closed eyes or mouth? Blinking or yawning frequency?”, Proc. IEEE Conference on advanced video and signal based surveillance, 2005, pp.207-212

[37]. Sergei Azernikov. Sweeping solids on manifolds. In Symposium on Solid and Physical Modeling, pages 249–255, 2008.

[38]. John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, Nov. 1986.

[39]. F. Mai, Y. Hung, H. Zhong, and W. Sze. A hierarchical approach for fast and robust ellipse extraction. Pattern Recognition, 41(8):2512–2524, August 2008.

[40]. Thomas B. Moeslund. Image and Video Processing. August 2008.