Vision-based patient wellness monitoring using facial cues

Sathyanarayana, Supriya

2017

Sathyanarayana, S. (2017). Vision-based patient wellness monitoring using facial cues. Doctoral thesis, Nanyang Technological University, Singapore.

http://hdl.handle.net/10356/72179

https://doi.org/10.32657/10356/72179

NANYANG TECHNOLOGICAL UNIVERSITY

Vision-based Patient Wellness Monitoring using Facial Cues

Supriya Sathyanarayana

School of Computer Science and Engineering

A thesis submitted in fulfillment of the requirements

for the degree of Doctor of Philosophy

May 2017

“Dedicated to my family, friends and

above all my Guru

H.H. Paramahamsa Nithyananda”

Acknowledgements

I would like to express my humble gratitude to my supervisor, Professor Thambipillai Srikanthan for his constant guidance, motivation, time and support. I am grateful to him for his constructive suggestions and ideas that have played a key role in the unfolding of this thesis. Words fail to express my gratitude to him.

I would like to express my gratitude to Dr. Ravi Kumar Satzoda and Dr. Suchitra Sathyanarayana for the brainstorming sessions and for their invaluable time, inputs and support throughout. I would like to offer my special thanks to my husband, who has been there with me through the period of the thesis, as a huge pillar of strength and support.

My gratitude to all my friends at Hardware and Embedded Systems Lab, especially Dr. Nirmala Ramakrishnan, Kratika Garg, Dr. Alok Prakash, Dr. Meiqing Wu, Dr. Abhishek Ambede and Dr. Lam Siew Kei. I would like to thank all my friends at NTU, who have been there for me, in all ways possible. I would like to thank Dr. Aruna Sathyanarayana and Dr. Kshipra Nandakumar for their valuable inputs as medical doctors.

I would like to thank Mr. Jeremiah Chua for all the technical support throughout. I would like to acknowledge School of Computer Science and Engineering, Nanyang Technological University for providing this opportunity to me for carrying out this research work.

I would like to express my gratitude to my parents, my parents-in law and my extended family in Singapore, for their inexplicable support in this journey. My deepest gratitude to my spiritual master, Guru Paramahamsa Sri Nithyananda, who has been the guiding light from within, and the source of strength.

Abstract

Patient monitoring systems (PMS) are gaining importance considering the increasing demand for wellness monitoring at affordable costs. Vision-based PMS using CMOS cameras are being increasingly explored due to their low-cost and passive sensing capabilities. In this thesis, computationally efficient techniques for extracting facial features such as eyes, mouth and brow furrows as well as an integrated framework for a vision based PMS have been proposed.

Noting that eyebrow is a stable facial feature, a compute-efficient technique for eyebrow detection has been proposed to help localize other facial features. An iterative thresholding method was proposed for the efficient extraction of the eyebrow edges. Evaluation on standard databases shows a high detection rate of 96% and robustness to variation in ethnicities, expressions, illumination and presence of partial occlusions. Computational savings of 35% was achieved compared to a state-of-the-art method.

Next, eyebrows were relied upon as anchor points for the efficient detection of eyeball and eye-state. The proposed technique for eyeball detection relies on the local intensity variations within the eye while the eyeball position together with inter-feature distances were used to classify eye state as - open, closed or partially-open. The proposed eye state detection method achieves a recall and precision of 91% and 96% respectively, when evaluated on the proposed WellCam database and results in significant cost savings compared to existing techniques. The temporal analysis of eye state and eyeball position was then performed for the efficient extraction of the wellness indicators: eye blink, eye state over time and eyeball movement. The average mean accuracy of blink detection when evaluated on two standard databases was 98.8%. Eye-state-over-time and eyeball movement were evaluated using the proposed WellCam database. The eye-state-over-time achieves a recall and precision of 98.2% and 97% respectively.

The technique proposed for the robust detection of the mouth state as open and closed relies on the efficient detection of the upper and lower lip positions and the analysis of the region between them. In addition, computations were restricted to vertical cross-sections of the mouth for drastically collapsing the complexity, leading to significant computational savings compared to existing techniques. The technique achieves an accuracy as high as 95% when evaluated on standard databases and is shown to be invariant to changes in facial expressions and ethnicities. Temporal analysis of mouth state has enabled the extraction of the wellness indicators such as mouth kept open, yawning and talking. Evaluation of the yawning detection method on the YawDD database confirms a perfect detection rate (i.e. 100%).

Noting that the existing techniques for detecting brow furrow features typically rely on a compute-intensive detection of brow lowering actions, a technique to directly extract and analyse brow furrows has been proposed in this thesis. A selective method for extracting brow furrow edges against surrounding noise was also incorporated to improve the average precision and recall to 92% and 91% respectively.

In order to lower the compute complexity of the conventional face detection technique deployed in this work, a search space reduction technique has been proposed to limit the regions being investigated. The proposed method relies on the head and shoulder curves, which were extracted using a block based analysis for improved compute efficiency. An average reduction in search space of 70% is achieved. The computational complexity is reduced by 34%, if a worst case of 50% reduction in search space is considered, compared to the entire image.

The wellness indicators such as eye state, eyeball movement, blink, mouth kept open, yawning, talking and brow furrows were finally integrated into a framework to determine the wellness state to be one of - asleep, awake, drowsy, inactive and discomfort. Past history of the wellness state over a period of time was also incorporated to generate a wellness profile, which is then used to assess the condition of the patient relative to an initial state. A confidence measure, based on the intensity and persistence of the wellness indicators, has been incorporated to accommodate situations in which not all relevant wellness indicators were present. The system is also capable of generating triggers in the event the patient's state drifts towards a threshold set by the doctor for notifications of improvement or worsening of patients condition.

Finally, the major contributions in this thesis have paved way for propelling further research in this emerging area of interest, and for realizing an affordable vision-based patient wellness monitoring that can eventually lead to mass volume adoption.

Contents

List of Figures

List of Tables

Abbreviations

3D	3-dimesional
AAM	Active Appearance Model
ACC	Accuracy
AU	Action Units
DL	Dark to light transition
FACS	Facial Action Encoding System
FDR	False Detection Rate
FN	False Negatives
FP	False Positives
FPPF	False Positives Per Frame
FPR	False Positive Rate
FPS	Frames per second
GAH	Gradient Angle Histogram
GMM	Guassian Mixture Models
ICU	Intensive Care Unit
IR	Infrared
LD	Light to dark transition
PMS	Patient Monitoring Systems

PSPI Prkachin and Solomon Pain Intensity

PVP Peak-valley-peak

PV Peak-valley

RFID Radio-frequency Identification

ROI Region of Interest

STIP Space-time interest points

SVM Support Vector Machine

TN True Negatives

TP True Positives

TPR True Positive Rate

VP Valley-peak

Symbols

α	average thickness of eyebrows
β	eyeball movement
Δ_T^{max} and Δ_T^{min}	percentages of face height h used to set threshold in the eyebrow detection technique
μ_s, σ_s^2, μ_r and σ_r^2	Mean and variance of distance between lips s and amount of dark pixels r respectively
σ_B^2	variance of the blob extracted from eye ROI
$\mathbf{\Theta}^B$	set of angles in the GAH for block B
$\mathbf{b_D}$	blink duration
b	band
B	blink rate
$C(i,k)$	eyebrow candidate pair
d_{eu}, d_{el}	eyebrow-eyelid distances
$\|\mathbf{d_j}\|$	Euclidean distance between windows P_{ref} and P_{W_j}
D_e	eyeball diameter
e_d	percentage of edge content In the image
$\mathbf{E}^B$	Set of edge maps for block B
E_{y+},E_{y-}	signed edge maps

F	eyebrow region
g_t	gap threshold used in the verification step of eyebrow detection
G_x+, G_x-	gradient maps along x-axis
$G_{x_P}^W$	weighted gradient magnitudes
G_y	gradients in y-direction
G_{y+}, G_{y-}	corresponding gradient maps along y-axis
G_y+^{max}	maximum gradient value in E_{y+}
h	face height
h_b	band-size used in eyeball detection process
$\mathbf{h}^B$	GAH for block B
I	detected face
I_i	initial eye ROI
I_R, I_L	average intensity within the right and left eyebrows respectively
I_d	threshold for the intensity difference between left and right eyebrows
J	horizontal projection of pairs of points in the eyebrow segment
$\mathbb{K}_L$	kernels used to capture number of right and left curves
$\mathbf{M}$	intensity-weighted accumulation map of mouth ROI
$\mathbf{MB}_L$, $\mathbf{MB}_R$	macroblock arrays used in face localization technique
n_B	number of bands that satisfy a certain condition in the eyebrow detection process
n_l	number of iterations to extract upper eyebrow edge
n_d	number of iterations to extract lower eyebrow edge
N_b	width of the eye ROI
p_{UL}, p_{LL}	Upper and lower lip positions respectively
$P1 - P7$	properties of eyebrows
$\mathbf{P}$	mean intensity profile of the mouth ROI

P_R, P_L	right and left eyebrow location
$P_x \times P_y$	dimensions of association window in face localization technique
R_1, R_2, R_2	Mouth ROIs
$\mathbf{S}_{b_j}$	Intensity-weighted accumulation map for band b_j used in eyeball detection
s	distance between upper and lower lips
S_f	intensity-weighted accumulation map of eyebrow region
S_W	eye state within each window of frames in the temporal extraction
S_y	Sobel kernel to compute gradients along y-direction
t_b	frame count threshold to classify as blink
T_u, T_l	thresholds to extract signed edge map in eyebrow detection
w	face width
w_n	window size for non-maximal suppression

CHAPTER 1

Introduction

1.1 Motivation

Patient monitoring systems (PMS) and assisted living systems are gaining increasing importance considering the scenario of the world's ageing population and the increasing costs of nursing facilities [1]. The major functionalities of a PMS are monitoring the physiological parameters of the patient, activity monitoring, detecting any unusual activity and fall, monitoring sleep, facial expression, emotion detection and wellness monitoring [1]. Among other sensors, cameras are a vital component of PMS, they are affordable, easily available, contact-less sensors and hence aid in unobtrusive monitoring of the patient. Vision based solutions have a huge potential and play a major role in PMS. In vision-based sensing in PMS, existing literature primarily focuses on fall detection, abnormal event and abnormal behavior detection, sleep apnea detection, respiration monitoring, daily activity monitoring, epilepsy monitoring, posture detection and patient monitoring based on facial expressions [2], [3], [4]. Face is a very important indicator of well-being. Existing methods on assessing wellness based on facial analysis involve detecting a certain medical condition, a specific emotional state or expression [5], [6], [7]. The *wellness* of a patient assessed relative to a reference state, and the relative improvement or deterioration from that state is useful information for the

doctor. A patient appearing drowsy or tired, staring without blinking for long durations, closing his eyes for excessively long periods, shaking his head all of a sudden or showing signs of discomfort are all possible indicators of *non-wellness*, that need to draw the attention of the nurse or doctor. Patient wellness monitoring is extremely important in monitoring patients under examination or post-surgical care, etc. So far, less work is done in this regard, limited to detecting the wakefulness of patients in an intensive care unit (ICU) based on facial changes and detecting emotions of the patient. There is a need to develop a versatile, comprehensive and configurable wellness assessment framework that monitors wellness relative to a reference state, based on facial analysis. Also, existing techniques include a combination of feature extraction, machine learning and inference, which can be computationally intensive. Hence, there is a need to develop robust, yet compute-efficient algorithms which are suited for embedded vision systems that can be deployed in large-scale.

1.2 Research Objectives

This research aims at proposing compute-efficient techniques for vision-based patient wellness assessment based on the analysis of a set of facial features - eyes, mouth and brow furrows. It is envisaged to devise image processing techniques to extract the facial features, followed by the temporal extraction of wellness indicators based on these features. This work also aims to thoroughly evaluate the techniques on existing datasets, and generate new datasets wherever required. Finally, this research also aims to propose a framework that integrates the various modules that extract the facial features and wellness indicators to infer patient wellness.

1.3 Research Contributions

1. A detailed literature survey of existing techniques in vision-based patient monitoring based on facial analysis is carried out.

2. In order to localize the facial features, a computationally-efficient technique for eyebrow detection is proposed. The technique involves extraction of partial-gradient maps to selectively extract the upper and lower edges of the eyebrow and verifying them against certain properties unique to eyebrows. The evaluation of the technique on standard databases with variations in ethnicities and lighting conditions shows that a high detection rate is achieved. Computational cost analysis of the proposed method is conducted and shows a savings of 35% compared to a state-of-art method is achieved.

3. A novel technique to detect the eyeball position is proposed. The technique is based on extracting the local intensity within the eye using band-wise intensity and gradient weighted accumulation maps. One of the key features of the technique is that, it is iterative in nature and hence enables extraction of eyeball in partially open eye and eyes that appear smaller in size. Further, an eye state detection technique is proposed, that uses the eyeball detection technique along with blob analysis and inter-feature distances to detect eye state - open, closed and partially-open. A temporal analysis of eye state and eyeball position is done to extract the eye-related wellness indicators - eyeball movement, blink and eye state over time. Evaluation of blink detection on three standard databases shows that detection rates similar to the state-of-art are achieved.

4. A technique to detect mouth state i.e., open or closed is proposed. The novelty of the technique lies in the efficient processing of a reduced set of pixels to capture the positions of the upper and lower lip and the analysis of the region between them to detect mouth state. The technique is extensively evaluated on three standard databases for its accuracy. A temporal analysis of mouth state is done to extract the wellness indicators - yawning, mouth kept open and talking.

5. A technique to extract brow furrows caused due to the eyebrow lowering action, is proposed. Partial gradient maps along with magnitude-based weight-assignment are used to extract the brow furrows. In order to address the cases where a person would already have furrows as natural intransient features, the technique analyzes the change relative to the natural state to detect the occurrence of furrows.

6. A computational cost analysis is performed to establish the cost efficiency of the proposed techniques.

7. In order to overcome the computational challenges in conventional face detection, a technique to reduce search space for face detection is proposed. The technique takes advantage of the controlled setting of a patient and is aimed at detecting the head and shoulder curves of a front-facing human. As part of this, a novel block-based curve detection technique based on extracting gradient angle histograms (GAH) is proposed. The proposed method is evaluated on two standard databases and is shown to reduce search space by up to 80%. The computational cost analysis is carried to show the reduction in computational cost when the proposed search space reduction method is applied as a pre-processing step to face detection.

8. A dataset *WellCam* is proposed in this work to capture the wellness indicators discussed in this work. The dataset has image sequences of ten subjects simulating different wellness indicators. It has image sequences of ten subjects simulating different wellness indicators. To the best of our knowledge, this is the first such patient wellness monitoring dataset with the various face related wellness indicators. The video frames were captured at 25 frames per second.

9. A wellness assessment framework that combines all the wellness indicators proposed in this work in order to assess wellness, is proposed. The novelty lies in the definition of the term *wellness* and the approach in realizing the objectives of the framework.

10. Finally, critical conclusions are drawn and the future research directions are proposed.

1.3.1 Research Publications from this Work

Following is the list of publications by the author of this report that were published in various international conferences and journals:

Journal

1. S Supriya, R K Satzoda, S Suchitra, T Srikanthan, "Vision-based patient monitoring: a comprehensive review of algorithms and technologies, *Journal of Ambient Intelligence and Humanized Computing*, Springer, pp. 1-27, 2015.

Conferences

1. S Supriya, R K Satzoda, S Suchitra, T Srikanthan, "Compute-efficient eye state detection: algorithm, dataset and evaluations," *Proceedings of the 9th International Conference on Distributed Smart Cameras*, ACM, 2015.

2. S Supriya, R. K. Satzoda, S. Suchitra, and T. Srikanthan, "WellCam: Dataset for Vision-based Patient Wellness Monitoring, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Future of Datasets in Vision*, 2015.

3. S Supriya, R K Satzoda, S Suchitra, T Srikanthan, "Identifying Epileptic Seizures based on a Template-based Eyeball Detection Technique", *Intl. Conf. on Image Processing (ICIP)*, Sep 2015

4. S Supriya, R K Satzoda, S Suchitra, T Srikanthan, "A Compute-Efficient Algorithm for Robust Eyebrow Detection," *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol., no., pp.664-669, 23-28 June 2014

5. S Supriya, R K Satzoda, S Suchitra, T Srikanthan, "Reducing computational complexity for face detection," *2014 14th International Symposium on Integrated Circuits (ISIC)*, 10-12 Dec. 2014

6. S Supriya, R K Satzoda, S Suchitra, T Srikanthan, "Block-Based Search Space Reduction Technique for Face Detection using Shoulder and Head Curves, *Proceedings of the 6th Pacific-Rim Symposium, PSIVT 2013*, Guanajuato, Mexico, 2013, pp. 385-396

To be submitted

1. S Supriya, R K Satzoda, S Suchitra, T Srikanthan, "Wellness Assessment based on Analysis of Facial Features", *IEEE Transactions on Affective Computing*

2. S Supriya, R K Satzoda, T Srikanthan, "Compute-efficient Mouth State Detection", *2017 IEEE International Symposium on Circuits and Systems*

The journal publication (1) is linked to Chapter 2, conference publications (1), (2) and (3) is linked to Chapter 4, (4) is linked to Chapter 3, (5) and (6) are linked to Chapter 6. To be submitted papers (1) and (2) are linked to Chapters 7 and 5 respectively.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

- **Chapter 2:** In this chapter, a review of the existing technologies and techniques in the area of vision-based patient monitoring systems is provided. An introduction to patient monitoring systems (PMS) is presented first. Then, an overview of the sensing technologies in patient monitoring systems is provided, where the scope of the various sensors in patient monitoring is reviewed. The role of vision sensors in PMS is discussed next, following which a comprehensive review of patient monitoring applications where vision sensors have been explored is undertaken. The state-of-art in vision-based monitoring based on facial analysis is reviewed next, with their strengths and challenges outlined. The chapter is concluded with a summary of the observations based on this study, leading to the chapters that follow.

- **Chapter 3:** A compute-efficient technique to detect eyebrows as anchor points for the facial feature detection has been proposed. The unique properties of eyebrows that are retained in spite of changes in expressions on the face are first stated. A candidate extraction step using signed edge maps and iterative thresholding is proposed. Then, the properties of eyebrows are used in the candidate verification step to eventually detect the eyebrow position. The robustness of the technique is

evaluated on three standard databases. The computational complexity analysis of the proposed method is computed and compared with the state-of-art.

- **Chapter 4:** In this chapter, techniques for extracting eye features, such as the eye state and eyeball location are proposed. Techniques to extract the eye related wellness indicators such as eye state over time, eye blink and eyeball movement based on the temporal analysis of the eye state and eyeball position are presented. The WellCam dataset is proposed for the purpose of evaluating face-related wellness indicators. The eyeball detection, eye state detection techniques and the extraction of wellness indicators are evaluated on standard datasets and the WellCam dataset. The computational cost of the eye state detection technique is computed and compared with a state of art method.

- **Chapter 5:** In this chapter, techniques for extraction of feature states and wellness indicators from mouth and brow furrows are presented. Firstly, a compute-efficient technique to robustly extract the upper and lower bounds of the mouth is proposed, which is used to detect the mouth openness or mouth state. The method for mouth state detection is evaluated on standard datasets. Temporal analysis of the detected mouth state is performed to extract the wellness indicators such as: mouth kept closed, mouth kept open, talking and yawning. The wellness indicators are then evaluated for detection accuracy. The computational cost of the mouth state detection technique is computed and compared with a state-of-art method. Then, a simple technique based on partial gradient maps is proposed for brow furrow detection, followed by the temporal analysis to extract wellness indicators related to brow furrows. The method was evaluated on standard databases.

- **Chapter 6:** Additionally, to further improve computational efficiency, an effective strategy for search space reduction for face detection is proposed. The technique takes advantage of the controlled setting of an on-bed patient monitoring scenario. Evaluation of the technique on standard datasets is carried out. The computational cost savings that result by applying the search space reduction technique as a pre-processing step for face detection are computed.

- **Chapter 7:** In this chapter, an integrated framework for wellness assessment is proposed. The framework brings together the techniques proposed in the previous chapters for the extraction of wellness indicators from facial features, for the purpose of wellness assessment. The steps in wellness assessment are presented and illustrated through an example.

- **Chapter 8:** This chapter presents the conclusions of the research in relation to the objectives set in Chapter 1. Finally, recommendations for future work are presented.

CHAPTER 2

Literature Survey

2.1 Introduction

In this chapter, a review of existing technologies and techniques in the area of vision-based patient monitoring systems is provided. An introduction to patient monitoring systems (PMS) is presented first. Then, an overview of the sensing technologies in PMS is provided and the scope of the various sensors in patient monitoring is reviewed. The role of vision sensors in PMS is discussed next, following which a comprehensive review of patient monitoring applications where vision sensors have been explored is undertaken, with their strengths and challenges outlined. The chapter is concluded with a summary of observations based on this study, leading to the chapters that follow. In literature, the techniques in the field of patient monitoring are proposed either in the context of the elderly or in the specific context of monitoring patients. However, the word patient monitoring is eventually aimed at addressing the needs of both the elderly and patients. Therefore, in the rest of the chapter and the thesis, the word patient is used interchangeably with user, subject or elderly.

2.2 Overview of Patient Monitoring Systems

In the recent past, particularly over the past two decades patient monitoring systems (PMS) and ambient assisted living (AAL) systems have been gaining increasing importance [1], [8], [9], [10], [11], [12]. The main factors contributing to this trend are: (a) shortage of nursing staff for continuous monitoring of patients and the elderly, (b) increase in age-related diseases [1] (c) increasing rate of growth of the elderly population [13], [8], [14]. Around 89% of the elderly prefer to live independently and age in their homes [9]. Studies on age-related diseases reinforce the need for developing technologies to assist the elderly and enable remote monitoring [1]. As stated in [10], "ambient assisted living systems are aimed at empowering people's capabalities by means of digital environments that are sensitive, adaptive and responsive to human needs, habits, gestures and emotions".

Patient monitoring systems or patient health monitoring systems are an important component of ambient assisted living systems [8], [11], [10]. The recent advances in sensors, communication and technology has made remote monitoring and management of patient health a possibility, not only in hospitals, but also in homes [15].

Some of the primary applications of patient monitoring systems (PMS) include measuring and monitoring the vital parameters such as blood pressure, heart rate, temperature etc. [1], [16], [17], [11], monitoring of patients' posture, activities, falls, inactivity, behavior, breathing, sleep, apnea [1], monitoring medication intake, monitoring emotional well-being, detecting and managing emergency situations.

Technologies used in patient monitoring include (a) smart home technology where a regular home is augmented with various types of sensors and actuators to assist the patients, or elderly aging in place [18], [19], [20]. Certain smart home projects have been designed to provide non-invasive assistive environment specifically to assist individuals with certain medical conditions, for such as dementia [19] and diabetes [20], (b) wearable and ambient sensors - wearable sensors are used to monitor the body's vital parameters [16], [17], [11]; (b) ambient sensors - that are sensors embedded in the environment and used to monitor patient activity, sleep, behavior etc. [1], [21], [22],

[23], [11], [24], [25], [26]; and (c) assistive robotics [27], that allow the older adults to overcome their physical limitations by helping them in their daily activities.

Wearable devices and the ambient sensors are built with wireless sensing technology, by which communication with a central server or the doctor is enabled. This has also increased the efficiency of emergency response systems [28], [29]. The increased performance, miniaturization and availability of wireless communication systems have a major role to play in enabling increased deployment of health care services [30]. The recent advancements in remote health monitoring applications has made it possible for health professionals (doctors, nurses, etc.) to constantly monitor patients fitted with wearable sensors capable of collecting vital health information in real-time [28], [29], [30], [31]. These patients could be in their own homes, in an aged care facility or in a hospital [32].

2.3 Sensing Technologies in Patient Monitoring Systems

The sensors used for patient monitoring can be classified broadly under the following categories: ambient sensors and wearable sensors. An overview of the sensors and their associated patient monitoring applications are provided in Table. 2.1. Wearable sensor-based devices can either be body-worn or implantable devices [17]. Wearable sensors are contact-based sensors, and need to be in contact with the body during measurement. Ambient sensors are sensors that are embedded into the living environments [10].

(a) Wearable sensors

The physiological and biomechanical parameters extracted from the measurement and processing of biosignals can be used to estimate the health condition of the user [17], [33]. Some examples of biosignals, and the sensors used to monitor them are - heart rate (the frequency of the cardiac cycle) is measured using pulse oximeter or skin electrodes; blood pressure (the force exerted by the circulating blood on the walls of blood vessels, especially the arteries) is measured using an arm-cuff based monitor; respiration rate (breathing rate) is measured using a piezoelectric/piezoresistive sensor [17]. Examples of biomechanical signals are: body movements, physical activity frequency and motion

TABLE 2.1: Overview of sensing technologies in patient monitoring and assisted living systems [1]

Type of sensor	Sensor	Measurement
Wearable sensors	skin/chest electrodes	Electrocardiogram [17], [1]
	Piezoelectric/piezoresistive sensor	Respiration rate [17], [1]
	Pulse oximeter/skin electrodes	Heart rate [17], [1]
	Arm cuff-based monitor	Blood pressure [17], [1]
	strip-base glucose meters	Blood glucose [1]
	Pulse oximeter	Oxygen saturation [17], [1]
	Temperature probe/skin patch	Body/skin temperature [17], [1]
	Skin electrodes	Electromyogram [1]
	Scalp-placed electrodes	Electroencephalogram [1]
	Galvanic skin response	Perspiration/skin conductivity [1], [33]
Ambient contact-based sensors	Accelerometer	Body movements [33], [34], [35]
	RFID	Object information [11], [36]
	Magnetic switches	Door opening/closing [22]
	Pressure sensors	Pressure on mat/chair etc. [21], [37]
Ambient contactless sensors	microphone	Activity [24], [34]
	Camera	Activity, sleep, breathing, vital signs, facial expressions, behavior, falls [38], [39], [40], [41], [42], [43], [44]
	Infrared	Motion [45],[46],[47]

intensity, which can be measured through accelerometers [48]. They have been explored for fall detection and daily activity monitoring [49]. A detailed description of the sensors used in wearable health monitoring is provided in [33]. An integrated system of biomechanical and physiological sensors can help to interpret the physiological signals more holistically, given the physical activities detected. For example, an accelerometer and a wearable heart rate sensor have been combinedly used to accurately classify human activity [17]. For instance, an increase in heart rate can be because of some altered cardiac condition or the user might be running.

There are many research prototypes as well as commercially available wearable health monitoring systems [17]. Examples of research prototypes of wearable health monitoring systems are as follows - AMON is a wrist-worn system that is capable of measuring blood pressure, SpO_2 and ECG, and can perform activity recognition using an acceleration sensor [49]. The system is based on wired transmission of biosignals. Smart Vest is a vest that has various sensors embedded into the garment fabric that can be used to collect the biosignals in a non-invasive and unobtrusive manner; CodeBlue is a

mote-based body area network (BAN), where wirelessly enabled tiny nodes that form a body area network collect the biosignals and transmit to a central node. Some examples of commercially available wearable health monitoring systems are: small, wearable, low cost and lightweight finger tip pulse oximeters by manufacturers such as Philips, Agilent and others that can display heart rate and blood oxygen saturation in real-time; the chest-worn Polar and wrist-worn Omron that measure heart rate; SenseWear Armband from BodyMedia that measures activity, temperature and galvanic skin response; LifeShirt is a washable lightweight vest that can be used to monitor respiration rate, heart rate and activity [17]. The aforementioned examples are discussed systematically in [17].

Wearable sensor networks have tremendous potential in health monitoring, but challenges such as limited processing power, storage, energy-efficient use and user comfort need to be addressed [50], [33]. The energy efficiency problems can potentially be addressed by using energy harvesting or short-range wireless energy submission [1]. An extensive review on wearable health monitoring systems can be found in [17].

(b) Ambient contact-based sensors

Ambient sensor-based systems have been used to monitor patient health, vital signs, motion, activity, detect falls etc. [1], [21], [22], [23], [11], [24], [25], [26]. State-of-art ambient sensors being used are discussed as follows:

Pressure sensors [21], [37], [51]. have been used to detect movement. In [21], pressure sensors are used to detect presence of the individual at a particular place in the house. Change in pressure is detected and used to monitor the mobility of the user. In [37], pressure sensors have been used to monitor the body movements during sleep and the statistical analysis of the pressure data is used to detect events other than breathing, viz. apnea or body movement. Pressure sensors are used as a component of pressure mats, which are used to identify movement of the user from the bed to the chair, etc. [51]. Apart from being useful as a fall detection system, pressure mats can also be used to create a behavoir profile of the night-time wandering of the user, or even raise an alarm if the user is away from the bed beyond a certain duration.

Radio-frequency identification devices (RFID) has been deployed for person and object identification, and tracking user actions, where the user or the object needs to wear an RFID badge [11], [36], [52]. The challenges when using RFID based systems is the limited range they have and that the user needs to always have the badge on [1], [52] and several of these sensors also need to be placed throughout the user's environment. Magnetic switches have been used to detect opening and closing of doors [22].

(c) Ambient contactless sensors

From the above survey in section 2.3, it is observed that many of the sensors used in patient monitoring systems are either wearable or intrusive [16], [17], [11] or contact-based sensors such as pressure sensors, magnetic switches and RFID tags [21], [22], [37], [35]. They are mainly designed for monitoring the physiological parameters [17] or activities [1]. While such monitoring systems are necessary because they directly monitor the physiological parameters or certain activities, one of the main challenges faced such sensors is that they have to be in contact with the individual - else, it can give inaccurate results [52], [8]. This calls for the need for a complementary system based on contactless sensors. Examples of contactless sensors are: audio sensors, vision-based sensors such as cameras and infrared sensors.

Audio sensors have been used to detect sound in order to monitor sleep [24]. Audio sensors have also been used to capture environment sounds, that are transmitted to a monitoring unit. They are then used to distinguish the sound of a human fall from other events such as walk, run etc. [34]. Infrared (IR) sensors have been used for various applications [23], [25], [26]. They have been used for detecting body movement as part of a fall prevention system [23]. IR sensors have been used to monitor breathing by either the thermal imaging of the region around the nostrils or the chest region [25], [53] and sleep [54], [55], [56], [57]. IR sensors have also been used to monitor vital signs based on the thermal imaging of the face and neck areas [58], [26].

Vision-based Sensing in Patient Monitoring: Among contactless ambient sensors, vision sensors are gaining increasing importance in the recent past. With decreasing costs and increasing miniaturization of cameras, vision-based PMS are being explored to assist medical staff to monitor patients and their activities. This is motivated further

because of the increasing pervasiveness of smart phones and tablets that are equipped with high resolution cameras and high speed computing systems. Therefore, such systems can be deployed to monitor the patients remotely as well [52]. Examples of vision sensors include video cameras, RGB-D cameras (RGB-depth cameras), infrared sensors and time-of-flight cameras. Data collected by any (or a combination) of these sensors are processed aiding in the various patient monitoring functions, which is discussed in the next section in detail. Vision-sensors can play a significant role in the development of intelligent environments to support individuals and older adults [52].

2.4 Existing Vision-based Patient Monitoring Applications

A brief review of the most common applications in patient monitoring where vision sensors have been explored in literature is presented in this section.

1. Fall detection and accident monitoring have been of great interest among remote patient monitoring applications. This is because they are a major cause for fatal injury, especially for the elderly [59], [60]. Surveys on vision-based fall detection can be found in [59], [60], [61], [62], [63] that review the technologies, techniques and challenges associated with vision-based fall detection.

   Monitoring the activities of patients and the elderly [2], [38], [64], [38], [65], [66]; detecting unusual activities such as such as faint, fall, vomiting, headache etc. have also been of interest [67], [68].

2. Visual indications of epileptic seizure onset include changes in posture, eye, mouth and head movements [69]. A systematic review of vision-based detection of human motion in epileptic seizures is provided by Pediaditis et.al [70].

3. Breath or respiration monitoring are ususally based on either tracking movement of chest and abdomen using infrared sensors [53, 71–73], or thermal imaging of nostril region [74], [25]. Sleep monitoring methods also deploy infrared sensors,

considering that sleep monitoring includes night time monitoring. In [57], motion information is used to infer the wake/sleep status using near infrared (NIR) cameras. Multi-modal sensing is employed in [75] and [24] to monitor sleep patterns. In [37], pressure maps are used, in which the periodic changes due to breathing are induced, that are then represented as image patterns. Sleep apnea is characterized by repetitive interruption or cessation of ventilation during sleep [76]. Sleep apnea detection involves approaches similar to breathing monitoring, viz. tracking movement of chest and abdomen, and thermal imaging of nostril region. [42], [4], [77], [78].

4. Face and facial expressions are an important source of information while monitoring the health and wellness of a patient. Face is an indicator of emotions, mood, well-being. It is considered as the base of medical semiotics [3]. It is indicative of the health status of an individual through a combination of facial expressions and physical signs [3]. Examples of applications based on facial analysis include monitoring vital signs, detecting epileptic seizures, detecting pain, depression etc. Visual indications of epileptic seizure onset include changes in posture, eye, mouth and head movements [69]. In [69], an optical flow based method is used to detect absence seizure by monitoring the flow vectors in the eyes and mouth regions of the face.

   Methods that are based on the thermal imaging of the face and neck areas to detect breathing and heart rate [79], [58]. Recent works on color-based methods to detect vital parameters include [80], [81]. They are based on the idea that the color of the skin/face changes when there are movements of blood in particular areas. [39] propose a technique for detecting heart rate from subtle head movements that are caused by cardiovascular pulse. A review of thermal imaging based and RGB imaging based methods for heart rate measurement is provided in [82]. A more detailed review of patient monitoring applications based on facial analysis is provided in the next section.

2.5 Patient Monitoring based on Facial Analysis

As discussed in section 2.4, important cues that aid in monitoring the patient's health condition can be extracted by facial analysis. In this section, the patient monitoring applications that involve facial analysis, including facial expressions are comprehensively reviewed.

2.5.1 Pain Detection and Monitoring

Pain detection is important in post-operative care [83], regulating medications, long-term monitoring and guaging the effectiveness of a treatment and hence it is considered critical to detect pain in patients [5]. Pain detection from facial images is done by capturing the pain-related facial actions [84]. A common facial response to pain is observed across age groups of people [85]. Facial actions such as orbit tightening, cheeks raising, eyebrow lowering are indicative of pain [85]. The main challenge in pain detection methods is the frame-by-frame labeling of the FACS actions [86] in terms of action units in the ground truth data, which is very time-consuming. Methods have been proposed to address this challenge. Other challenges include the variation in the way individuals express pain, and hence the adaptability of the pain detection methods to individuals. The approaches for pain detection can be categorized under two classes viz., appearance based methods and appearance & geometry based methods. Appearance based methods involve detecting the features representing the facial texture, such as wrinkles, bulges and furrows and geometry based methods involve detecting geometric features such as shapes or corners of the facial features such as eyes and mouth [87].

One of the first attempts to automatically detect pain was carried out by Littlewort et.al [88]. Action units are recognized using gabor filters, adaboost and SVM (Support Vector Machine) and were used to differentiate between genuine pain and fake pain. Methods for neonatal pain detection have been proposed [89, 90]. Images of the COPE database are classified as pain or non-pain. [89] uses the column stacking of intensity values as feature extractors and RVM as a classification technique, whereas,

elongated ternary pattern (ELTP) are used as feature descriptors and SVM is used in [90]. In [5], a multiple-segment multiple instance learning framework for detecting spontaneous expressions of pain in videos was proposed. This method also includes the temporal dynamics in the proposed method of detecting pain.

Geometry based features such as shape and distance along with appearance based features are used to add robustness to changes in head pose. AAM-based shape and appearance features are extracted in [91]. A rigid representation of face appearance and frame-level labels to train classifiers is used to provide a sequence-level detection of pain. In [92], distance and gradient features are extracted and an SVM classifier to distinguish pain from non-pain. A comparative labeling and learning model is proposed for assessing pain expression intensity. Further, pain detection at frame-level in a video was explored in [93, 94].

While most of the methods consider head pose as a source of registration error, it is a potential source of information for pain and pain intensity indication [95]. Levels of pain are classified based on the Prkachin and Solomon Pain Intensity Metric (PSPI) metric. Other methods that use head pose information for classification in addition to facial expression was proposed in [96, 97]. In [96], 3D parameters derived from the AAM are used along with facial expressions to detect pain and assess the level of pain based on the PSPI scale. In [97], a method for fully automatic detection of pain based on facial expressions and head pose information is proposed. Depth and color features are extracted at the frame level and a time window descriptor is calculated from them. [97] also contribute a new database, the BioVid Heat Pain database.

All of the above discussed methods except [5] are based on coding every frame of the video based on FACS. Considering real-time applications, this could be a time-consuming process and hence be a possible bottleneck. Also, most of these methods are based on AAM-based features, which require face alignment as a necessary step. Recently, [98] proposed a framework which is not based on FACS, nor is face alignment required. Shape information is extracted using pyramid histogram of orientation gradients (PHOG) and appearance information using pyramid local binary pattern (PLBP).

Studies related to pain assessment in patients who could verbally not communicate the pain due to dementia show that facial expressions among other indications such as vocalizations are accurate measures of assessing the presence of pain, but not its intensity [99, 100]. In [101], the FACS is applied for assessing pain from the recorded video of the patients with dementia.

2.5.2 Depression Monitoring

Facial expressions have been used to detect depression, based on the FACS action units (AUs) [40], [102], [103], [104]. In [40], the approach used is that positive emotions are less expressed in depressed subjects and that recognizing depression by detecting positive emotions is more accurate than detecting negative emotions. In [102], the number of occurrences of positive (such as happiness) and negative prototypical expressions (such as anger, fear) and non-prototypical expressions within a certain duration of time are noted, based on which depression is detected. Local shape and texture features from the facial image are measured using AAM, which are then used to classify regions using a multiboost classifier. In [103], the facial behavior of the subject in response to certain questions were noted. The facial actions through both manual coding and by using AAM were measured and compared.

Head pose and movement provide effective cues while recognizing depression as a binary classification task (depressed versus non-depressed) [40]. The head pose and movement features are extracted by projecting the 2D points of face AAM into a 3D face model. A hybrid classifier based on Guassian Mixture Models (GMM) for each subject and SVM is used for classification. A recognition rate of 71.2% was achieved just based on the head pose and movement cues, while a combination of these cues along with the frame-by-frame features results in an average recall rate of 76.8%.

Alghowinem et al. also propose a technique for detecting depression by analyzing eye movement and blinking [6]. Eye movement features extracted using AAM models and hybrid classifier based on GMM and SVM is used which results in 70% accuracy in

detecting depression. Eye movement by themselves can be used as a complementary cue along with other cues such as speech, for recognizing depression.

[102],[40], [6], [103] use person-specific AAM models, and hence a new AAM model needs to be trained for every new subject, and this can be time-consuming and complex. Joshi et al. propose a framework that is subject-independent [104], [105]. In [104], facial dynamics are analyzed using features based on Local binary pattern (LBP) and upper body movements are analyzed using features based on space-time interest points (STIP). A maximum accuracy of 88.6% was obtained. An extension of [104] is proposed [105], where upper body expressions and gestures are used for automatic depression analysis, and the effect of upper body parts versus face versus head movements are compared. STIP and BoW based framework is proposed; bag of body expressions and bag of facial dynamics are created. A non-linear SVM was used for classification and 71% accuracy was seen in both the cases of using only head movements for analysis and using facial dynamics alone. Intra-facial muscle movements and the head and shoulder movements are combined with audio features and then fused to result in a multi-modal framework [106]. The authors of this work conclude that only a multi-modal system will be suitable for achieving the robustness needed for real-world applications.

While the methods proposed have been tested on real-world clinical data, based on subjects with depression, a major challenge faced is the limited size of the dataset on which the methods are tested, based on a few subjects. This indicates the need to create larger datasets [40].

2.5.3 Monitoring Psychological Health Conditions

Expression, emotion analysis and changes in indicators such as for e.g. blink rate have been used for detecting psychological conditions such as for e.g. stress, anxiety, schizophrenia and dementia. Dai et al. propose a technique to monitor patients on bed based on analyzing expressions such as happiness, easiness, uneasiness, disgust, suffering, and surprise [107]. Optical flow and associate model are used for classification of the facial expressions.

Liao et al. proposed a multi-modal stress recognition system [108], that includes information from facial expressions, eye movements and head movements extracted using vision sensors. A dynamic Bayesian network (DBN) framework is used to model the stress. Eye detection and tracking, gaze estimation, facial feature tracking and face pose estimation are the main components of the visual tracking system. In [3], semiotic face signs related to cardio-metabolic risk are monitored. For e.g., stress, anxiety and fatigue detection based on changes in indicators such as blink rate, percentage eye closure, yawning, eye blink closure duration, lip deformations, along with other features such as heart rate is proposed.

Facial expression recognition has been used to study deficits in emotional expression and social cognition [109], and automated vision-based expression detection has been explored in the context of neuropsychiatric disorders, primarily schizophrenia and dementia [110], [111], [112], [113]. Automated computational framework for analyzing facial expressions from video data of Asperger's and schizophrenia patients was proposed by Wang et al. [112] and it was shown that healthy subjects show the intended emotion better than the subjects with Aspergers and schizophrenia. Regional volumetric difference (RVD) functions of the expression changes are used to train classifiers and pattern classification techniques are applied. The flatness and inappropriateness in the expressions of subjects with schizophrenia was also studied in [111]. Neurological distress expresses as loss of expressiveness in the face, and facial expression recognition based on the FACS is used to diagnose neurological disorders such as Alzheimer's disease or depression [110].

Initial work has been done on monitoring patients in coma or patients under post-surgical care in an intensive care unit (ICU), based on facial changes such as eyes and mouth movements [114] and [115]. Features of interest (distance between eyelids and lips, areas of eyes and mouth) are extracted and passed through a fuzzy classifier and eventually the 'awake' state of the patient is detected.

2.5.4 Monitoring the After-effect of Drug

Monitoring the after-effects of drug is vital in patient monitoring and can help prevent the occurrence of medical emergencies due to unexpected reactions of a patient to certain drugs. Vision-based monitoring of after-effects of drug based on facial changes of a patient have gained interest lately. The effectiveness of certain drugs or clinical procedures are monitored by monitoring the temporary evolution of facial edemas. A vision-based method to quantify the edema was proposed by Brusco et al. [116]. 3D (3-dimensional) model of the patient's face is constructed and the difference in facial surfaces indicate any facial volume changes due to edema. A challenge faced by this method is the alignment of the 3D models of the same person taken at different times. After giving Botox injections to patients with facial nerve disorders, the subtle changes in their facial motion was detected using automated facial image analysis [117]. The types of abnormal facial activity following facial paralysis have been listed in [117], and feature points of interest were marked in the first frame and then tracked based on the optical flow algorithm.

2.5.5 Wellness Monitoring

Wellness has been defined in many ways in literature [118]. A variety of indicators of wellness have been considered in the works on wellness monitoring. Some works consider monitoring how well the daily activities are performed as indicators of wellness [119], [120], [21], [121]. In [120] and [21], how well the elderly person is performing his activities in terms of usage of the house-hold appliances are considered as indicators of wellness. Other works consider voice and linguistic parameters [122], [7]. Some works are based on face as an indicator of wellness. Face is an indicator of emotions, mood, well-being. It is considered as the base of medical semiotics [3]. It is indicative of the health status of an individual through a combination of facial expressions and physical signs such as skin color, subcutaneous fat, etc. [3]. Wellness monitoring based on facial analysis has been proposed in [3], [123], [7] and [124]. In [7], the basic standard emotions of the person are considered while assessing the person's wellness state,

where the emotion is detected and classified as happy, sad, angry. In [124], classification into relaxed and non-relaxed states is done, and the pleasure-arousal-dominance (PAD) emotional model is used.

Discussion: As can be seen from this section, with respect to patient monitoring based on facial analysis, pain detection has gained most importance in literature. The techniques for pain detection involve extraction of action units that define pain, which in turn involve complex feature extractions [91], [90], [88]. Other than pain, research on other health conditions such as depression, stress, anxiety, dementia, etc. have started to gain interest in the recent past. Further, it can be inferred from the literature survey that existing works on wellness monitoring based on facial analysis are limited/restrictive, aimed at detecting specific medical conditions or emotional states, and not designed for detecting relative changes of a medical condition. A unified framework which is comprehensive, versatile, configurable and affordable for monitoring wellness relative to a reference state based on facial analysis is lacking.

2.6 Extraction of Wellness Indicators from Facial Features

Eyes and mouth are important indicators in facial analysis [125], and they have been chosen along with brow furrows, such that a framework involving these facial cues efficiently (both computationally and robustly) detects the wellness of a patient. Hence, wellness indicators from three facial features have been of interest in this research. In the following paragraphs, a comprehensive review of the techniques for the extraction of the wellness indicators from these facial features is presented.

2.6.1 Wellness Indicators related to Eyes

Eyes of the patient carry important information indicative of wellness [126]. Various indicators of wellness are extracted from the eyes, for e.g. how open are the eyes, is

the eyeball movement normal or very slow, is the blink rate slow or very rapid. These indicators are extracted based on the frame-level detections of eye-related features such as eye state, eyelids and eyeball position. The review presented here includes eye state detection, eyeball detection and tracking and eye blink detection.

Eye state detection has been of significant interest in applications such as driver drowsiness and fatigue monitoring [127]. Some authors detect eye state based on a dual state-model: *open* and *closed* [128] [129], while some others classify eye state as *open*, *closed* and *half-closed* [130]. While most of the eye state detection techniques in literature are proposed for driver attention monitoring, eye state detection in the context of patient monitoring is relatively less explored. Conditions such as drowsiness, fatigue, lack of sufficient eye movement or blank stares that indicate inactivity, or restless movements of the eye are among the many health conditions which can be detected based on eye state and eyeball movement. [131], an eye closeness detection technique has been proposed which uses gabor wavelets (to extract global shape features), LTP (local texture feature), MultiHPOG (local shape features) for robust detection of eye closeness in uncontrolled real-world scenarios.

Eye center and eye gaze detection has been an active area of research in computer vision. [132] and [133] propose techniques aimed at accurate detection of eye center in low resolution images. [132] is based on extracting the isophote properties of the pupil and [133] uses Hough transform filters for the coarse iris detection step. Appearance based techniques such as active appearance models (AAM) [134] and classifiers have been used to detect eye gaze efficiently. While such methods are able to achieve high robustness, they also incur high computational costs [135]. This can become a bottleneck in the embedded realization of these algorithms. In [6] subject-specific AAMs to model and track the eyes use a hybrid classifier based on a Gaussian Mixture Model for each subject and a Support Vector Machine for classification.

A blink consists of rapid closing and opening of the eyelid [136]. Eye blink detection has different applications, for e.g, monitoring a user for dry eye syndrome prevention [137], helping physically challenged people to interact with a computer [138] or face liveness detection [139]. Some methods detect open and closed states of the eye [140],

[141], [136], [138], [142], while some methods track the movement of the eyelids to detect eye closure during the blink [128], [6]. In [128], a technique to detect eye blink by tracking iris and eyelids is proposed. In [138], the extent of eye openness is extracted based on the correlation with the open eye template and the complementary motion information obtained from the eye areas.

Real Time Eye Tracking and Blink Detection with USB Cameras was proposed in [140]. In [140] and [143], a frame differencing and thresholding is used to distinguish the open and closed eye states in order to detect the eye blinks. In [144], a simplified head model is used and facial feature points are tracked using particle filtering. Eyelid motions are detected as the deformation from the reference facial image. In [137], a real-time method for blink detection that combines the motion of the eye and its appearance is proposed. In [136], the eye region is divided into 3×3 cells and an average *cell* motion is calculated for each cell. The variances for each eye are analyzed using simple state machines. In [142], horizontal symmetry feature of the eyes are used to detect visual changes in eye locations. The system uses a standard webcam and detects the pattern of long duration eyelid closures.

[141] proposes a pose-invariant blink detection. The height to width ratio of the eye region in a still image, and the cumulative difference of the number of black pixels in the eye region with an adaptive threshold for successive images, are used to distinguish between closed and open states of the eye. Finally, the SVM classifier for determining the eye state is adaptively selected according to the facial rotation.

Discussion: Based on the literature survey conducted, it is observed that although many existing driver drowsiness monitoring systems involve analysis of the eyes and mouth, we would like to highlight the following observations: (i) the environment and challenges in a driver monitoring scenario are different from a patient monitoring scenario in a hospital set up, (ii) existing driver drowsiness monitoring systems mostly involve complex techniques involving multiple sensors other than vision sensors, (iii) drowsiness is only one of the various wellness states that are detectable by the wellness monitoring system, and not the sole problem of interest.

It is found that techniques proposed in [130], [131], [6], [133], [132] use gabor wavelets, subject-specific AAM models, etc. for eye state and eye movement analysis. Likewise, blink detection methods [144], [137], [141], [143] and [140] involve techniques such as particle filtering, extraction of motion and appearance features, adaptive thresholding and frame differencing and thresholding. While such methods are able to achieve high robustness, they also incur high computational costs [135]. This forms the motivation to develop robust, yet compute-efficient eye state and eyeball detection techniques.

2.6.2 Wellness Indicators related to Mouth state

Wellness indicators extracted from mouth state considered in this research include yawning, mouth openness over a period of time and talking. These indicators are extracted based on the temporal analysis of mouth features such as lip corners color in order to detect mouth state. The following is a review of techniques in mouth state detection, majority of which have been proposed for yawning detection.

The primary features used for mouth state detection are lip color [145], [146], edges [145], [147], [145], lip corners [148], lip contour [149] and the intensity changes [150].

Detection of yawning based on mouth state has been proposed in [150], [149], [151], [151], [152], [145], [146], [148], [153], [154], [155], [156], [157], [147], [158]. The degree of mouth openness over a certain duration is used to classify as yawning. Some methods classify mouth state into two [159], or three or four states [155], [156]. [155] and [156] classify the mouth state into close, talking and yawning.

In [160] where the mouth state detection is proposed for a mobile computing environment, mouth state is classified into four states, viz. closed, widely open, smile and a certain expression “i”. In [155], classification of mouth state into closed, yawning and talking is done using two cameras. A low resolution camera is used to detect the face and a high resolution camera is used and haar like features are applied to detect the mouth; yawning is detected based on the ratio of mouth width to height. [147] uses nose tracking in order to extract the mouth ROI. The yawning detection algorithm is

based on the observation that vertical mouth edges increase during the yawn. A Neural network is used for the mouth state classification.

In [153], an improved version of the CERT (Computer Expression Recognition Toool-box) is used to detect the facial action units of yawning. In [154], Fuzzy C-means clustering is used to segment the lips and the mouth aspect ratio is used to detect yawning. In [156], Fischer classifier is used to segment the lips and skin pixels, following which connected component analysis is applied to extract the lips and a neural network is used to classify the mouth state into closed, open and wide open. In [151], lip corners are tracked based on the dark line in between the lips and color detection is applied to detect yawning. In [146], color detection is followed by conversion to black and white and eventually connected component analysis to detect the mouth state for yawning detection. In [148], mouth corners are detected using supervised descent method, and the distance between upper and lower lips gives the height of the mouth. [145], edge extraction and extraction of the red pigments are used in detecting yawning. In [152], the degree of mouth openness is detected based on the integral projection of vertical difference of mouth region along the direction of mouth rotation to detect the two lines running through upper and lower lip boundaries. [159] proposes a mouth state detection technique based on the luminance information of static images and classification into open and closed states. The log polar spectrum corresponding to a bio-inspired signature of the images are computed, which are then used for training and classification. [149] uses color processing followed by active snake contour to detect the mouth shape. The rate of increase in mouth area is used to detect yawning.

Given that many application scenarios consider real-time requirements of a system and limited processing power and computing abilities of an embedded platform, some works have been proposed to address this challenge [150], [149], [160]. In [150], a modified implementation of the Viola-Jones algorithm for face and mouth detection is proposed and back-projection theory is used to compute the rate and amount of change in the grayscale image of the mouth, to detect yawning. [149] use color based processing followed by extraction of an active snake contour to extract the mouth contour. The rate of increase in the mouth contour area is used to detect yawning. Face that is detected in one frame is tracked in the subsequent frames using mean-shift algorithm using the

hue histogram of template face in order to increase the computational efficiency. An optimized implementation of the face detection itself is also done. In [160], a technique for mouth state detection for a mobile computing environment is proposed, and the mouth state is classified into four states including closed, widely open and smiling.

Discussion: A review of techniques for mouth state detection has been carried out. Based on this, it is observed that the techniques are either intended for precise detection of the lip contour and mouth corner points and involve extraction of features such as log polar signature, Fuzzy C-means clustering, Gabor wavelets, connected component analysis, supervised descent method etc. [159] [160], [151], [156], [154], [148] or yawning detection techniques [149], [150], where the drastic change in mouth area during yawning is of interest. This forms the motivation to develop robust, yet compute-efficient mouth state detection techniques.

2.6.3 Brow furrow as a wellness indicator

Brow furrows are casued by the facial action of brow lowering [161]. In literature, the facial action of brow lowering action is detected, rather than the appearance of brow furrows itself. Appearance based methods are mostly used for detecting the brow lowering action, which is an important indicator in pain detection [88], [89], [90]. [5]. The techniques proposed in these works are aimed at robust detection and involve complex feature computations.

2.7 Summary

Following a thorough review of the existing literature on vision-based patient monitoring, the key observations made are summarized below:

1. Patient monitoring systems (PMS) are gaining increasing importance considering the scenario of the world's aging population and the increasing costs of nursing facilities [1], [13], [162]. The major functionalities of a PMS are monitoring the physiological parameters of the patient, activity monitoring, detecting any unusual activity and fall,

monitoring sleep, facial expressions, emotions and wellness. The sensors used in the PMS are primarily the wearable sensors [33] and ambient sensors such as pressure sensors, motion sensors, cameras etc. [163], [1]. Cameras are a vital component of PMS [1], they are affordable, easily available, contactless sensors and hence aid in unobtrusive monitoring of the patient.

2. Vision based solutions have a huge potential and play a major role in PMS [52]. In vision based sensing in PMS, existing works in literature have been primarily focused on fall detection [60], [164], abnormal event and abnormal behavior detection [165], sleep apnea detection [166], respiration monitoring [167], daily activity monitoring [168], epilepsy monitoring [69], posture detection [169] and monitoring based on facial expressions [170], [6], [171].

3. Face is a very important indicator of well-being. Existing methods on assessing wellness based on facial analysis involve detecting a certain medical condition or a specific emotional state or expression. [3], [123], [7], [124], [6]. Moreover, monitoring wellness based on facial analysis for patients on-bed is extremely important in monitoring patients under examination, post-surgical care, or in an Intensive Care Unit (ICU). Initial work has been carried out to detect the wakefulness of patients in an intensive care unit (ICU) based on facial changes [114], [115], and detecting emotions of the patient on bed [107]. We find that the existing works on wellness monitoring based on facial analysis are limited/restrictive, aimed at detecting specific medical conditions or emotional states, and not designed for detecting relative changes of a medical condition. Hence, there is a need to develop a unified wellness assessment framework which is comprehensive, versatile, configurable and affordable.

4. Computation cost is an important consideration for the feasibility of large-scale deployment of patient monitoring systems. Therefore, there is a need to develop robust, yet compute-efficient algorithms which are suited for embedded vision. Existing techniques related to patient monitoring based on facial analysis are mostly aimed at accurate detection of specific emotions and expressions, and involve computationally

intensive feature extractions and techniques [111], [6], [103], [98]. For instance, it is observed that most techniques for eye state detection involve complex computations [128], [6], [130], [131] and do not address the computational constraints posed by embedded platforms. Similarly, most of the methods proposed for mouth state detection are based on either color detection of the lips [145], [146], or edge detection [145], [147], [145], or lip corners [148], or lip contour [149]. Hence, they do not address the computational constraints posed by embedded platforms, and their direct implementation is not suited for mass volume production. So, rather than using the existing algorithms, computationally-efficient algorithms are proposed in this thesis.

In this work, analysis of individual facial features is eventually combined in an informed manner for the purpose of wellness assessment. Given that eyes and mouth are important indicators in facial analysis [125], they have been chosen along with brow furrows, such that a framework involving these facial cues efficiently (both computationally and robustly) detects the wellness of a patient.

In the following chapters of the thesis, compute-efficient techniques to localize the patient's face and facial features, namely eyes, mouth and brow furrows, and extraction of wellness indicators from these facial features are presented. Eventually, an integrated framework for wellness assessment is proposed, which combines the wellness indicators for the assessment of wellness.

The first step in the proposed research is to detect eyebrows that serve as anchor points for the landmark facial feature detection. In the following chapter, a computationally-efficient eyebrow detection technique is proposed.

CHAPTER 3

Detecting Eyebrows for Facial Feature Localization

3.1 Introduction

Among the intransient facial features, i.e. eyebrow, eyes, nose and mouth, eyebrow is the most stable and salient feature on the human face [172]. For instance, eyes and mouth appear different when open and closed, but eyebrows remain relatively more stable in appearance. Even under changing expressions, eyebrows are observed to show lesser variation in appearance compared to eyes and mouth [173], [174] as shown in Fig.3.1. Eyebrows being distinctive features, are used as reference or anchor points for the localization of the rest of the facial features [173]. In this work, eyebrows are detected first, to serve as anchor points for the localization of the other facial features: eyes, mouth and brow furrows. Eyebrow detection algorithms have been proposed under various contexts such as face alignment, face recognition in biometrics, recognition of landmark facial features and facial expression recognition. In [175], a rough estimation of the eyebrows are first obtained using a spatial constrained sub-area K-means clustering algorithm, followed by a precise tracing of the eyebrows using the Snake method. In [174], active shape models are used to obtain the facial features including eyebrows. A skin color model and a Laplacian operator are used in [176], where

FIGURE 3.1: Eyebrow as a relatively stable feature across different facial expressions

the non-skin color regions above the eyes are detected as potential eyebrow candidates which are further processed. In [177], a template matching technique is used within a defined area relative to location of the eyes. In [178], eyebrow segmentation is performed based on a modified level set method.

It is observed that the techniques used in the eyebrow detection algorithms proposed in literature [175], [172], [176], [174], [178] are intended for achieving high precision and robustness, but involve complex computations. Given that eyebrow detection is used for anchoring the other key facial features such as eyes, mouth etc. in this work, a computationally-less complex method to locate the eyebrows will aid in increasing the efficiency while integrating all the feature detections into a single system on an embedded platform. This motivates the need for a computationally efficient algorithm for eyebrow detection. A compute-efficient and robust method to detect the eyebrow is proposed in this chapter. The concept of signed-edge maps [179] are used to capture the distinct properties of eyebrows followed by a systematic evaluation of the other static and symmetry properties unique to eyebrows. The proposed method is evaluated on standard databases for robustness and is also shown to be computationally efficient compared to existing methods.

3.2 Properties of Eyebrows

To aid the detection of eyebrows, properties unique to them are observed and listed below from P1 to P7. Fig. 3.2 provides an illustration of the properties *P1* to *P7*.

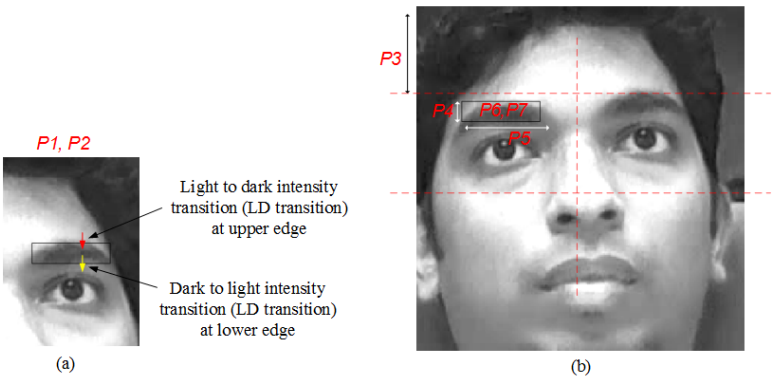


FIGURE 3.2: Unique (static) properties of eyebrows

- *P1 - Prominent intensity transition*: On scanning the face from top, eyebrows are most often the features that show the first prominent transition from light → dark intensity at the upper edge of the eyebrow.
- *P2 - Pair of intensity transition*: The light → dark intensity transition at the upper edge of the eyebrow is followed by a transition from dark → light intensity at the lower edge of the eyebrow.
- *P3 - Symmetry in position*: The eyebrows are symmetrically located on the face with a certain aspect ratio relative to the edges of the face.
- *P4 - Symmetry in thickness*: The right and left eyebrows will be of similar thickness that is within a certain range defined with respect to the width and height of the face.
- *P5 - Symmetry in Length*: The right and left eyebrows will be of similar length for an individual and within a range of length defined with respect to the width of the face.
- *P6 - Continuity*: An eyebrow will have continuity along its length. It is a continuous segment showing a nearly consistent gray level intensity along its length.
- *P7 - Gray level intensity*: The two eyebrows will show nearly similar gray level intensity along their length.

3.3 Proposed Method for Eyebrow Detection

The proposed algorithm is based on the unique properties of the eyebrows P1 to P7 listed in section 3.2 above, and is comprised of the following steps. Face detection using conventional methods is applied on the grayscale input image as a pre-processing step. Following the face detection, the next step is the ROI estimation for eyebrow detection, which is based on the property P3 listed above. The next step involves extraction of possible eyebrow candidates by taking advantage of properties P1, P2 and P5 listed above. This is followed by the candidate verification step that uses properties P3, P4, P6 and P7 to detect the correct eyebrow candidates from the set of eyebrow candidates obtained from the previous step. The latter two steps are performed in an iterative manner so that the algorithm is robust to varying lighting conditions, wrinkles on the skin and changes in facial expressions.

3.3.1 Face Detection as a Pre-processing Step

Given an input image, the technique proposed by Paul Viola and Michael Jones in [180] is used to detect the patient's face as a pre-processing step, and is briefly described here. The key features of this technique are - the integral image for feature computation, Adaboost for feature selection and an attentional cascade for efficient computational resource allocation.

The technique uses Haar-like features [180]. Three kinds of features are used (as shown in Fig. 3.3). The value of a two-rectangle feature is the difference between the sum of the pixels within two rectangular regions. Similarly, the three and four-rectangle features are extracted. These rectangle features are computed very quickly with an intermediate image representation called the *integral image*. In every image sub-window, 160,000 such rectanglular features are associated. However, a very small number of them are combined to form an effective classifier. A variant of Adaboost is used to

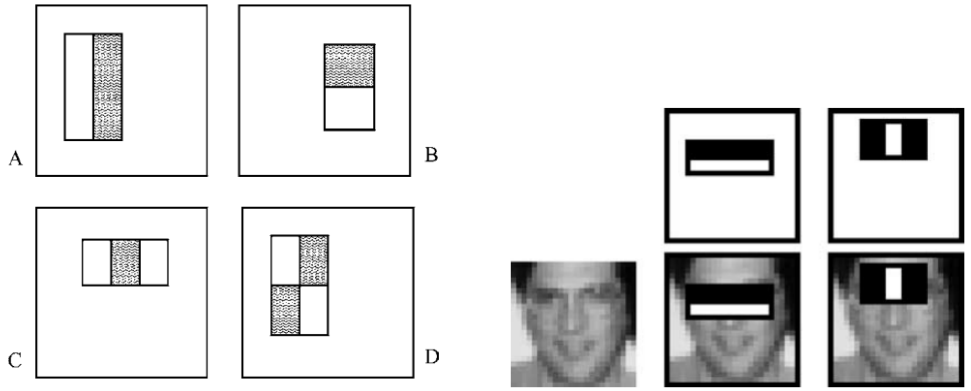


FIGURE 3.3: Example rectangle features (on the left), the first and second features selected by Adaboost (on the right) [180]

select the best features and to train the classifier. The algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier.

In the detection phase, the concept of cascade of classifiers is introduced, in which, instead of applying all the features (around 6000) on an image sub-window, the features are grouped into different stages of classifiers and applied one-by-one. For example, in the first stage, a two-feature classifier is applied. In this manner, a large number of non-face sub-windows are discarded in the initial stages of the classifer, and the sub-window that passes all the stages is detected as a face region.

3.3.2 Estimation of Eyebrow ROI

Let I be the detected face with face width and height w and h respectively. Based on property $P3$, the eyebrows are located in the upper half of the face and at a certain distance from a vertical line drawn at $C(x, y)$ cutting the face into two halves. We consider the section bounded by $0.2h$ to $0.6h$ along the y-axis and $0.1w$ to $0.9w$ along the x-axis with respect to the origin as shown in Fig. 3.4 to be the initial ROI I. So, the initial ROI I_i can be expressed as:

$$I_i = I(0.2h : 0.6h, 0.1w : 0.9w) \tag{3.1}$$

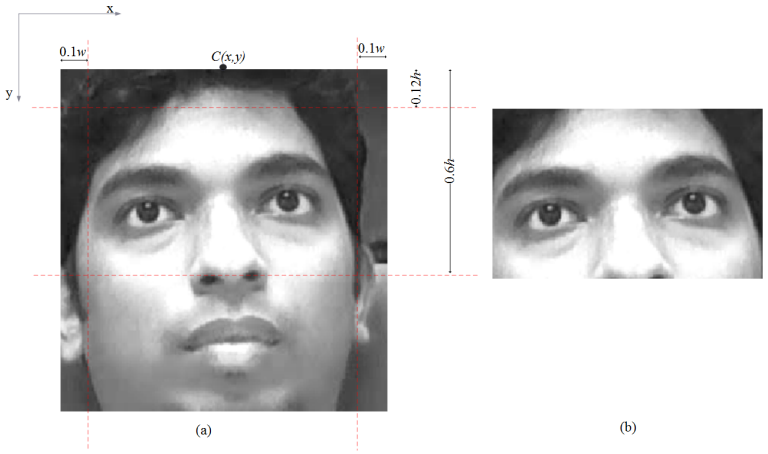


FIGURE 3.4: Initial ROI estimation for eyebrow detection based on property *P3*, w and h are width and height of the face respectively

A contrast stretch is applied to I to enhance the image contrast between the eyebrows and surrounding pixels. Next, I_i is divided into two halves with a vertical line drawn at $0.5\times$ width of I_i, the extraction of left and right eyebrows will be done on the left and right halves of I_i respectively, as will be described later. The right and left eyebrows are separable in the two halves respectively even in cases of $\pm 15^{\circ}$ variation in yaw and roll.

3.3.3 Signed Edge Maps for Eyebrow Candidate Extraction

For generating the eyebrow candidates, pairs of upper and lower edges of the eyebrows need to be extracted. In order to detect the edges of the eyebrow, we initially tried applying standard edge detection techniques with varying thresholds. The challenges faced when using the standard edge detection techniques were: (a) they required an image with good contrast, (b) lower threshold resulted in a lot of noise being extracted, and (c) higher thresholds resulted in extracting only parts of the eyebrow in most of the cases (as shown in Fig. 3.5).

In order to promote the eyebrow edge features and eliminate noise, we use the first and second properties (P1 and P2), i.e., the upper edge of the eyebrow has the first prominent transition while scanning the face from top (unless there are occlusions, which will be discussed later in this chapter), and a light $\rightarrow$ dark intensity transition at the eyebrow's upper edge is followed by a dark $\rightarrow$ light intensity transition at the lower edge. In order

FIGURE 3.5: (a) Right half of initial ROI, standard edge detection techniques applied on (a) with varying thresholds (b) Sobel 0.1 (c) Sobel 0.01 (d) Canny 0.1 (e) Canny 0.5

to effectively extract the eyebrow's edges, we use signed edge maps [179], through which the light to dark and dark to light transitions are separated.

In order to generate the signed edge maps, the Sobel kernel [181] is applied to every pixel of I. Through this, the gradients of each pixel in I in the y-direction G_y is computed first, as shown in (3.2 below:

$$G_y = I * S_y \tag{3.2}$$

where S_y is the Sobel kernel that is used to compute gradients along y-direction. Then, the edge pixels in G_y are split into two groups as per equations 3.3, which will be referred to as the two *signed edge maps*.

$$\begin{aligned} E_{y+}(x,y) &= 1 \; if \; G_y(x,y) \geq T_u \wedge E_y(x,y) = 1 \\ E_{y-}(x,y) &= 1 \; if \; G_y(x,y) < T_l \wedge E_y(x,y) = 1 \end{aligned} \tag{3.3}$$

where E_{y+} has the edge pixels that show light $\rightarrow$ dark intensity transition and E_{y-} has the edge pixels that show dark $\rightarrow$ light intensity transition while scanning from the top. The upper and lower edges of the eyebrow are extracted in the two signed edge maps E_{y+} and E_{y-} respectively, as shown in Fig. 3.6(b) and (c).

The thresholds T_u and T_l are computed as a fraction of the highest gradient magnitude of the edge pixels in E_{y+} and E_{y-}. T_u is initialized to a certain value at the start of the algorithm. A starting value of T_u that worked well across databases was heuristically derived to be $0.1 * G_{y+}^{max}$ where G_{y+}^{max} is the maximum gradient value of the pixels in the partial gradient map E_{y+}.

T_l is set to a value lesser than T_u, since the dark $\rightarrow$ light transition across the lower edge of the eyebrow is usually not as distinct as the light $\rightarrow$ dark transition across the upper edge of the eyebrow under ambient lighting conditions. The importance of setting a lower threshold for T_l is illustrated in Fig. 3.7. In Fig. 3.7.(b), the same threshold is applied for extracting both the upper and lower edges, due to which the eyebrow's lower edge is only partially extracted, and a lower threshold is applied for extracting the lower edge, and it is extracted Fig. 3.7.(c). In the proposed algorithm, the value of T_l is set to $(0.5 \times T_u) * |G_{y-}^{max}|$.

E_{y+} and E_{y-} are then filtered such that every think edge is reduced to a thin edge by retaining the uppermost row of edge pixels of the think edges on the E_{y+} and E_{y-}. The resulting filtered E_{y+} and E_{y-} are as shown in Fig. 3.6(d) and (e). Hereafter, the *filtered* E_{y+} and E_{y-} will be referred to as the signed edge maps E_{y+} and E_{y-} respectively. The corresponding gradient maps will be referred to as G_{y+} and G_{y-} respectively.

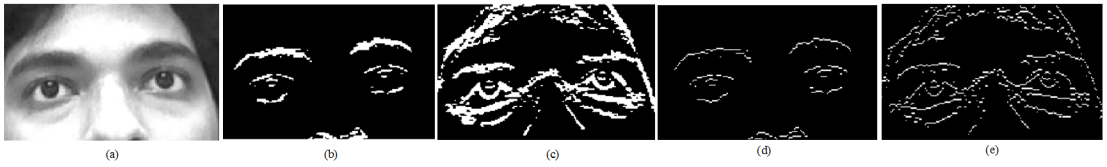


FIGURE 3.6: (a) I_i (b) E_{y+} (c) E_{y-} (d) filtered E_{y+} (e) filtered E_{y-}^R

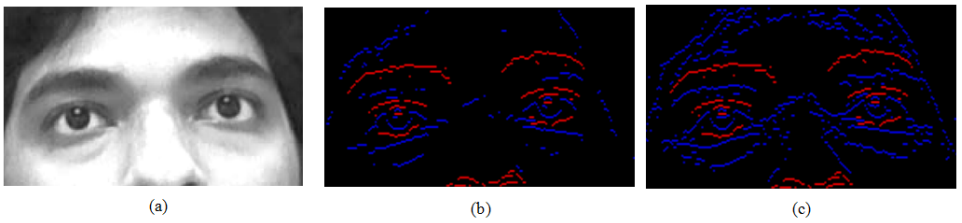


FIGURE 3.7: (a) Initial ROI I_i, (b) combined map of E_{y+} and E_{y-} when the same threshold is applied for extracting both the upper and lower edges, resulting in the eyebrow lower edge being partially extracted, (c) combined map of E_{y+} and E_{y-} when a lower value of threshold T_l is set for extracting the lower edge, resulting in a better extraction of the lower edge.

The following paragraphs explain the further steps performed using the signed edge maps for the right half of the face, and the same steps are carried out for the left half of the face as well. The right and left halves of E_{y+} will be addressed as E_{y+}^R and E_{y+}^R respectively. Similar notation is used for the right and left halves of E_{y-}. E_{y+}^R and E_{y-}^R are shown in Fig. 3.8.

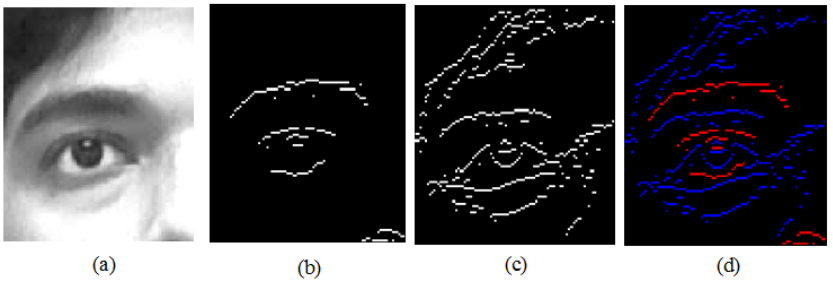


FIGURE 3.8: (a) Right half of I_i (b) E^R_{y+} and (c) E^R_{y-} (d) E^R_{y-} and E^R_{y+} combined, where red and blue pixels are the edge pixels in E^R_{y+} and E^R_{y-} respectively

As shown in Fig. 3.9, E^R_{y-} and E^R_{y+} are divided into overlapping horizontal bands b_1, b_2,... b_n with a bandwidth $\Delta_T^{max} \times h$. $\Delta_T^{max} \times h$ is the estimated maximum thickness of eyebrow as shown in Fig. 3.10.(b), and Δ_T^{max} and Δ_T^{min} are percentages of face height h derived heuristically as shown in Fig. 3.10.(d). Edge pixels in E^R_{y-} and E^R_{y+} are accumulated in each band to find the most prominent edges.

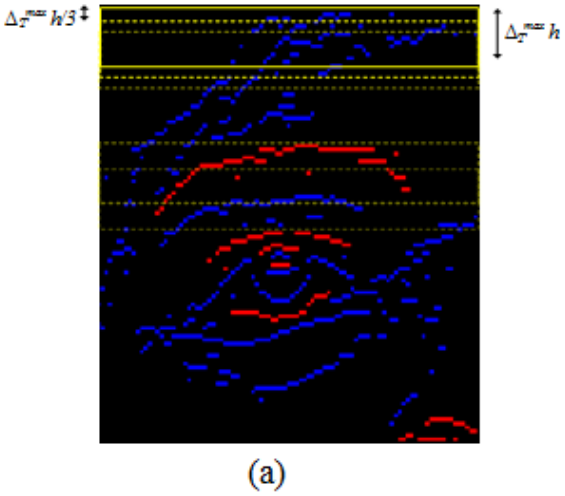


FIGURE 3.9: Summation of edge pixels in E^R_{y+} and E^R_{y-} within overlapping bands of band size $\Delta_T^{max} \times h$ and overlap of $\Delta_T^{max} \times h/3$

The bands in E^R_{y-} with the summation of edge pixels greater than the eyebrow length threshold $\Delta_L \times w$ are identified. The summation computed for every band b_j is checked for the following condition:

$$\Sigma E^R_{y-}(b_j) \geq \Delta_L \times w \tag{3.4}$$

where Δ_L is set to 50% of the difference between Δ_L^{max} and Δ_L^{min}, which are derived as shown in Fig. 3.10.(c). Let n_B such bands that satisfy equation (3.4) be obtained. Since

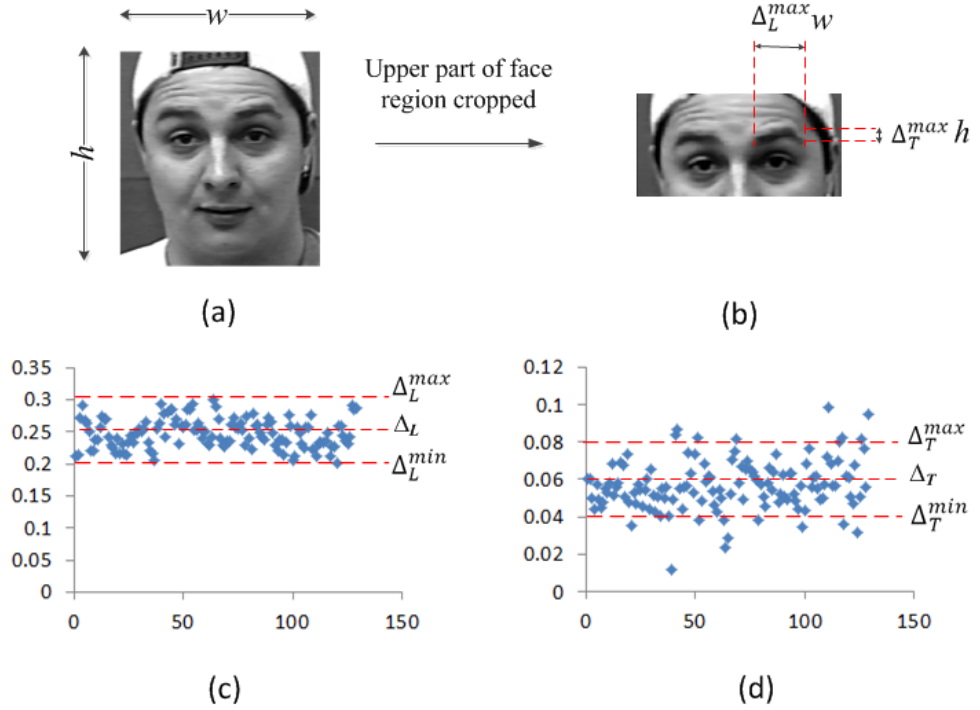


FIGURE 3.10: (a) Width and height of face w and h respectively (b) Maximum length and thickness of eyebrow relative to w and h respectively (c) Plot showing ratio of eyebrow length to width of face over 140 images (d) Plot of ratio of eyebrow thickness to height of face over 140 images

the bands are overlapping, a non-maximal suppression function is applied to extract the most prominent bands.

Among the n_B bands in E_{y-}^R, the topmost band $b_{j'}$ is considered first. Then, the signed edge map E_{y+}^R is scanned to look for the lowermost band $b_{j''}$, which satisfies the condition in 3.5 as shown below:

$$\Sigma E_{y+}^R(b_{j''}) \geq \Delta_L \times w \tag{3.5}$$

Band $b_{j''}$ marks the detected approximate lowest bound of the eye region. We narrow down the search space to extract eyebrow candidates in E_{y+}^R and E_{y-}^R to the region between $b_{j'}$ and $b_{j''}$.

Then, the signed edge map E_{y+}^R is scanned to look for the lower edge of the eyebrow. For every band b_j that satisfies (3.5), a band b_l which satisfies the condition in 3.6 as

shown below is identified:

$$\begin{aligned} \Sigma E^R_{y+}(b_l) &\geq \Delta_L \times w \\ where\ j + \Delta^{min}_T \times h &\leq l \leq j + \Delta^{max}_T \times h\ and \\ l &\leq j'' \end{aligned} \tag{3.6}$$

The band b_l is obtained such that b_i is located within a distance ranging from the minimum eyebrow thickness $\Delta^{min}_T \times h$ to the maximum eyebrow thickness $\Delta^{max}_T \times h$, and is above j''. Such a pair of b_j and b_l is called a segment pair, representing an eyebrow *candidate*. All possible b_j and b_l pairs are extracted as eyebrow candidates.

The following cases may be encountered while extracting the eyebrow candidates:

- At least one band that satisfies (3.4) as the upper edge of the eyebrow is not found, then the gradient threshold T_u is reduced by 15% and E_y is extracted again, and the above process is repeated iteratively till at least one such band b_j is found or till the stopping condition $T_u \geq 0.25$ is reached. An example illustrating this case is shown in Fig.3.11, where a significant part of the eyebrow's upper edge was detected only in the third iteration.

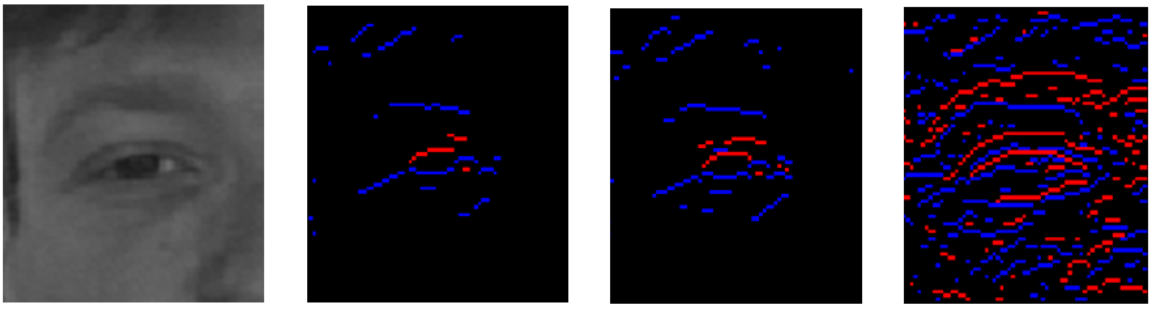


FIGURE 3.11: (a) Eyebrow ROI, (b)-(d) Combined map of $E(y-)$ and $E(y+)$ in each iteration where T_u is reduced; the upper eyebrow edge in $E(y+)$ was captured in the third iteration shown in (d)

- If an upper eyebrow edge is found, but there is no lower edge of the eyebrow that satisfies (3.6) that is present, then T_l is reduced by 25% and E^R_{y+} is extracted again and scanned as described above, unless a stopping condition $T_l \geq 0.1$ is reached.

The above process of extracting the upper and lower edges of the eyebrow as eyebrow candidates is repeated for the left half of the face as well. A candidate from the left and a candidate from the right halves of the ROI are called a *candidate pair*.

3.3.4 Eyebrow Candidate Verification

Each eyebrow candidate pair $C(i, k)$ extracted needs to pass the candidate verification step to qualify as the detected eyebrows. Properties P4, P6 and P7 described in Sec. 3.2 are used in this process. The candidate verification is done for each of the n candidate pairs in each of the m iterations until a candidate pair that passes the verification step is found.

Then, the following verification checks are applied on the candidate pairs:

- *Check #1: Continuity*: First, the edge pixels captured in bands b_j and b_i are paired columnwise along the x-axis, i.e an edge pixel $E^R_{y-}(x, j)$ above an edge pixel $E^R_{y+}(x, i)$ along the same x co-ordinate are grouped as a pair (as shown in figure 3.12).

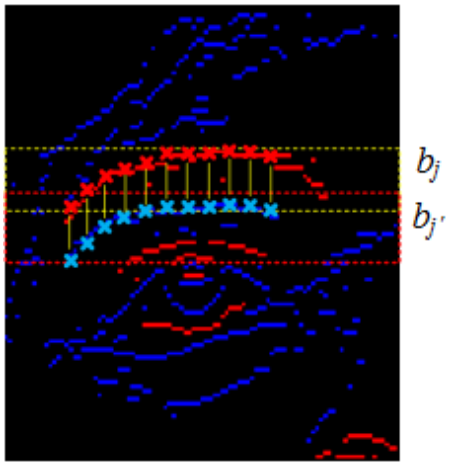


FIGURE 3.12: Edge pixels in $E^R_{y-}j$ and an edge pixels in $E^R_{y+}j'$ grouped as pairs columnwise

Once the pairs of edge pixels along the upper and lower edge of the eyebrow are obtained, a continuity check is done in order to check if they represent a continuous segment of the eyebrow. The horizontal projection J of the pairs is

taken and the gap between any pair of consecutive points along the segment must be within a 'gap threshold' g_t. If each point on J is represented by $J(p_i, p_j)$, then,

$$J_{p_i+1} - J_{p_i} \leq g_t \tag{3.7}$$

If the above check if true for an eyebrow candidate pair, it has passed this check. This check ensures stray pixels are not being picked up as eyebrow segments.

- *Check #2: Symmetry in width*: The thickness of the right and left eyebrow segments are compared to check if they are of similar thickness. The average thickness of the eyebrow segment along the length of the segment is computed on both the right and left sides and the difference in their average thickness α is computed. The value of α is checked if it is not greater than one-third of the estimated eyebrow thickness:

$$\alpha \leq 0.3\Delta_T^{max} \times h \tag{3.8}$$

- *Check #3: Average intensity*: The average intensity of the pixels in the grayscale image I enclosed within the upper and lower eyebrow edges is computed and is denoted by I_R and I_L for the right and left eyebrows respectively. The difference between these two average intensities must lie within a threshold I_d to satisfy the symmetry property with respect to the gray level intensities of the eyebrow.

$$|I_R - I_L| \leq I_d \tag{3.9}$$

The above properties are checked and if the right and left candidates (also referred to as a *candidate pair*) pass all the verification checks, the eyebrow candidate pair are the eyebrows detected on the face image. If any of the above properties are not satisfied, the next candidate pair is considered and the above steps of candidate verification are repeated.

Overall Algorithm

In sections 3.3.3 and 3.3.4, the eyebrow candidates are extracted in an iterative manner and the candidate pair that satisfies the properties of eyebrows is detected as the eyebrows. In this subsection, the overall flow of the algorithm is summarized. Referring to

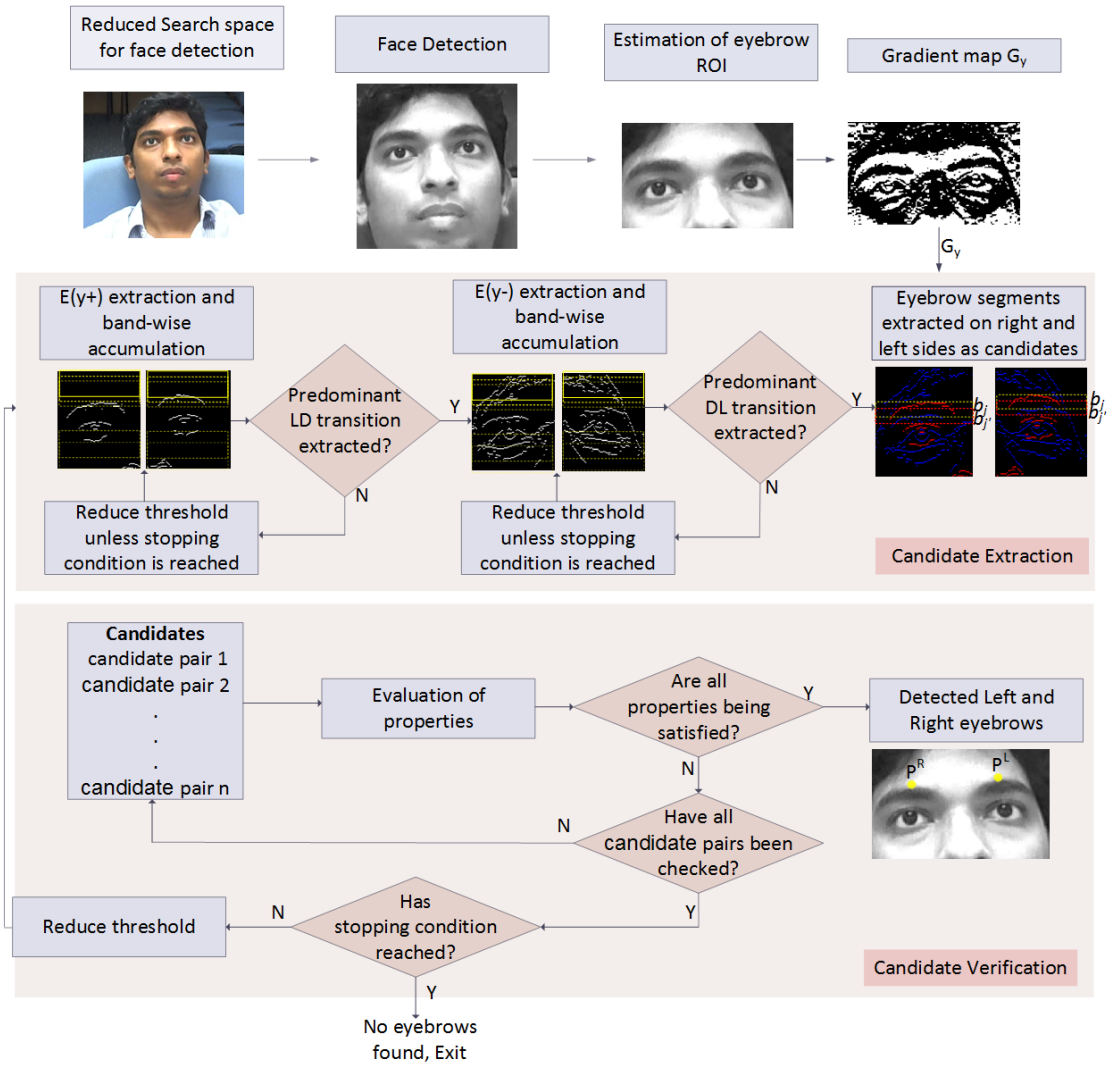


FIGURE 3.13: The proposed eyebrow detection algorithm (LD and DL refer to light $\rightarrow$ dark and dark $\rightarrow$ light transitions respectively)

Fig. 3.13, following the face detection step, the ROI for eyebrow detection is estimated. In step 1, the potential eyebrow candidates are extracted in an iterative manner in order to extract the lower edges (with reduction of thresholds if necessary) of the left and right eyebrows in the partial gradient maps (as explained in Sec. 3.3.3). In step 2, the right and left eyebrow candidates are grouped as candidate pairs and are tested for the properties of eyebrows as explained in Sec. 3.3.4.

In an i^{th} iteration, if none of the eyebrow candidates qualify as eyebrows, then step 1 is repeated with a lower gradient threshold for extracting the upper edge. The lower edge is extracted, followed by step 2 until the eyebrows are found or a stopping condition is reached. The output of the system is the eyebrow co-ordinates $P_R(x_R, y_R)$ and

$P_L(x_L, y_L)$. If there are n candidate pairs in the i^{th} iteration of eyebrow candidate extraction, a candidate pair k is represented by $C(i, k)$, k ranges from 1 to n and i ranges from 1 to m iterations).

3.4 Performance Evaluation

In this section, the accuracy results of evaluating the proposed eyebrow detection technique on standard databases are presented first. Then, the computational cost of the technique is computed and compared with the state-of-art. The term *database* will be used interchangeably with *dataset* in the rest of the thesis.

3.4.1 Evaluation Method

In order to evaluate the proposed techniques throughout the thesis, the following evaluation method is used. The techniques are evaluated for two metrics: (1) Accuracy, and (2) Computational cost efficiency. In order to evaluate the accuracy of the proposed techniques, the techniques proposed in *all* the chapters have been developed using the MATLAB 2013A programming platform. The accuracy evaluation was performed on a PC-based platform with Intel Xeon 3.5 GHz processor, under Windows OS. The toolboxes used were Computer Vision System Toolbox and Image Processing Toolbox.In addition to accuracy, we also evaluate the computational complexity of the proposed methods against existing methods. For each of the proposed techniques, we derive the number of computations and compare them with equivalent metrics of existing techniques. Considering that most existing techniques for facial feature extraction have not addressed the computational complexity aspect of the algorithms, we derive the number of computations for existing techniques also, with which the proposed methods will be compared.

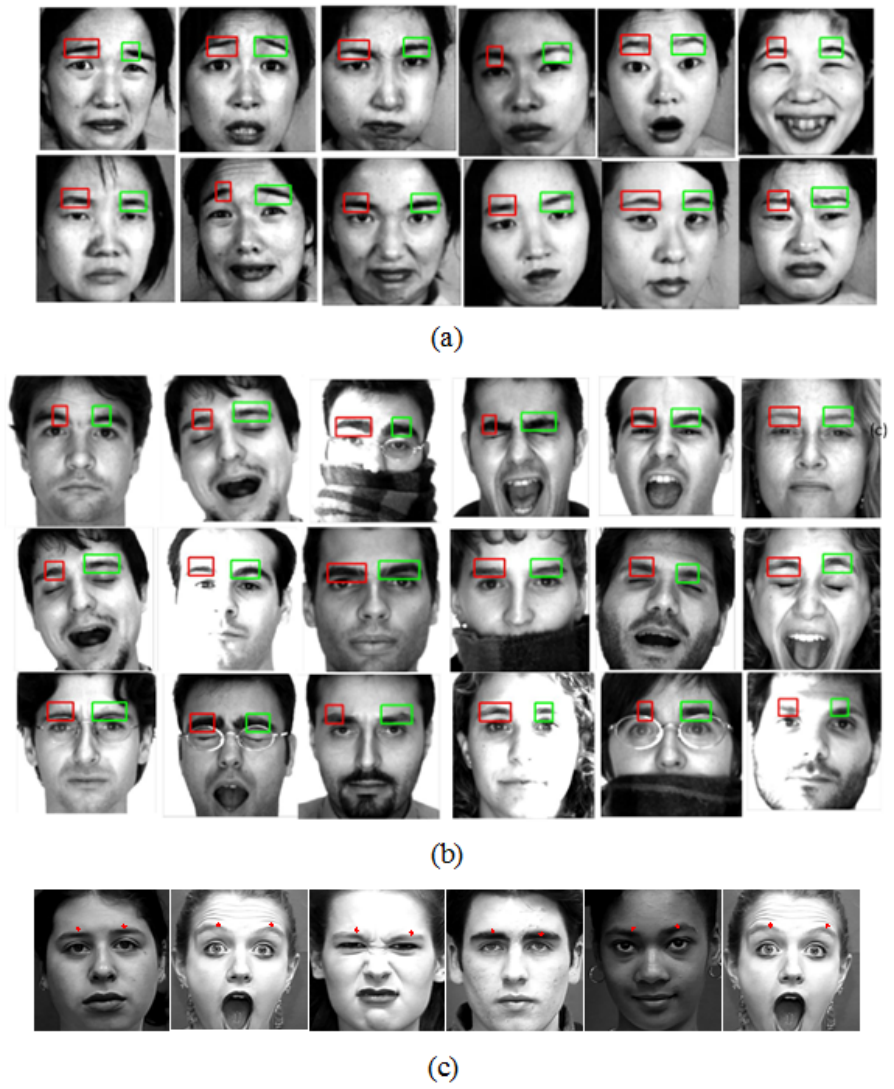


FIGURE 3.14: Eyebrow detection results for (a) Jaffe database (b) AR database (c) CK database (the mid-point of the bounding box has been marked by a red dot in case of CK database)

3.4.2 Accuracy Evaluation

The proposed algorithm was tested on a subset of the Cohn-Kanade face database consisting of 126 images (3 frontal face images each of 42 subjects from various ethnicities with different facial expressions and changes in illumination) [182], 213 images of the JAFFE database (21 images of 10 Japanese subjects with different facial expressions) [183] and 310 images of the AR database (10 images each of 31 subjects with variations in expressions and illumination) [184]. A description of all the databases used in this thesis has been provided in Appendix B.

The ground truth data is generated by manually going through each image and marking a bounding box around each eyebrow. The overlap of the output bounding box with the bounding box marked during ground truth generation is used for evaluation. If the overlap exceeds 50%, it is considered as a true positive (TP). If the algorithm is unable to detect the eyebrow, it is defined as a false negative (FN) and a mis-detection is defined as False positive (FP).

The detection rate (TP/(TP+FN)) for the three databases is tabulated in Table. 3.1. The reason for the lower detection rate in AR database is due to the images with subjects wearing dark colored glasses completely covering the eyebrows and eyes, and cases where the eyebrows were barely visible. Fig. 3.14 shows the eyebrows detected in the images of the three databases respectively. The detected eyebrows are marked with a red and green bounding box respectively on the right and left eyebrows. The figure shows the successful detection of eyebrows in images with variation in facial expressions with eyebrows lowered, raised, contracted in different measures, and variation in lighting conditions. Examples of challenging cases that were detected include - partial

TABLE 3.1: Detection rates for the Cohn-Kanade, Jaffe and AR databases

	Cohn-Kanade	*Jaffe*	*AR*
Number of images	126	213	310
Detection Rate	97.7%	99.5%	92%

occlusion of one eyebrow, poor contrast between eyebrows and skin intensities, graying eyebrows, variation of head pose upto 15^o, as shown in Fig. 3.15. Although examples to illustrate such detections existed in the Cohn-Kanade database, in order to comply with the database release agreement, we are unable to present them. However, sample images from other databases to illustrate the above are presented. The technique was unable to detect eyebrows in cases of complete occlusion of one or both eyebrows, drastic non-uniform lighting conditions and large variationsin head pose (greater than $\pm 15^{\circ}$), examples of which are shown in Fig. 3.16.

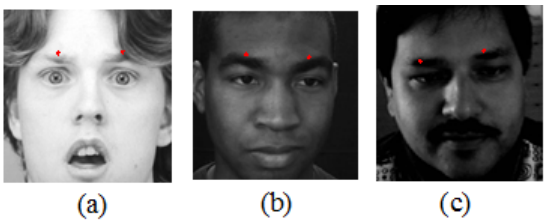


FIGURE 3.15: Examples of challenging cases that were successfully detected (a) partial occlusion of one eyebrow (b) and (c) low contrast between skin and eyebrow; the mid-point of the bounding box detected has been marked by a red dot (images shown here are taken from BioID database, KDEF database [185], DISFA database [186])

FIGURE 3.16: Examples of challenging cases that were not detected by the technique: (a) drastic non-uniform lighting, (b) complete occlusion of eyebrows

3.4.3 Computational Cost Analysis

The computational cost of the proposed eyebrow detection algorithm is evaluated and compared against an existing eyebrow detection algorithm [175].

The cost of the proposed detection technique is computed as follows:
Given the grayscale input image of the upper part of the face $M \times N$, the Sobel kernel is applied to every pixel of the image. The computational cost for computing the gradient G_y for all pixels in the image is summarized in equation (3.10).

$$C_{ADD} = 5MN \tag{3.10}$$

The computations involved in extracting the upper edge of the eyebrow are summarized in (3.12), where n_l is the number of iterations needed, with the gradient threshold reduced in every iteration and w_n is the window size for the non-maximal suppression.

$$\begin{aligned} C_{COMP} &= n_l(2N(M + (1 + w_n)/3w)) \\ C_{ADD} &= n_l MN + n_l \\ C_{MUL} &= n_l \end{aligned} \tag{3.11}$$

Now, having obtained the bands with the most prominent light to dark transition, the cost of finding the lower edge of the eyebrow is computed next. Considering a window size of w_n for the non-maximal suppression and the eyebrow thickness to be Δ, and n_d is the number of iterations till the lower edge of the eyebrow is extracted, the computations can be summarized as follows:

$$\begin{aligned} C_{COMP} &= n_l n_d B(2MN + (2/3) \times (\frac{N}{w_n} + 2\Delta)) \\ C_{ADD} &= n_l n_d BMN \\ C_{MUL} &= n_l n_d \end{aligned} \tag{3.12}$$

(3.13)

The cost of performing a continuity check is computed, assuming the eyebrow segments extend to half-the width of the face. Then, the cost of computing the average intensity of the eyebrow segments and their difference is computed considering a worst case condition that the thickness is equal to the maximum thickness. The computational cost for computing thickness or width, vertical separation between the eyebrows and position of eyebrows with respect to the face are computed next. The cost of computing the different threshold settings for performing these checks are also computed, given that the thresholds are reduced by a small fixed amount in each of the n_e iterations performed. If n_B number of right and left candidate eyebrow segment pairs need to be considered till the segment pair that satisfies the properties of eyebrows is found, the above computations and those in (3.13) will have to be repeated n_B number of times.

TABLE 3.2: Summary of computations in proposed method and [175]

Operations	Method in [175]	Proposed
Comparisons	$2*[3(wh+^{n_c}C_2+^{n_{cl}}C_2)+2n_p$ $+n_c+4*0.1X^2]+m-1+2*MN/5$	$[n_l(2N(M+(1+w)/3w))+B\times n_l n_d(2MN$ $+(2/3)\times(N/w+2\Delta))*(2MN+N+2*\Delta/2)+8Bn_l]$
Additions	$+2*MN*(15m)$	$[5MN+n_lMN+Bn_ln_dMN$ $+n_lB(M(1+2*\Delta/2)+N/2+2)]$
Multiplications	$2*[4*n_p+2n_c+4*0.1X^2+8*0.1X]$ $+2*5*MN/5*(16m)$	$n_l6B+5n_lBn_e+8$ $+n_ln_d+n_l$

TABLE 3.3: Comparison of computational complexity between proposed method and [175]

Operations	Proposed Method	*method in [175]*		
		m=3	m=5	m=10
Additions	1087780	185340	222840	447840
Multiplications	335	192620	272620	472620
Comparisons	1921867	102240	112240	137240
Total computations (Equivalent additions)	3015007	3369500	4697000	8147000
% Savings	-	10.52	35.80	62.99

Summarizing these computations as follows:

$$C_{COMP} = n_l 8B \tag{3.14}$$

$$C_{ADD} = n_l(B\times M(1+2\times\frac{\Delta}{2})+\frac{N}{2}+2) \tag{3.15}$$

$$C_{MUL} = n_l 6B+5n_l\times n_e+8 \tag{3.16}$$

The computational cost involved in [175] is computed as follows: In the eyebrow contour extraction method in [175], the rough estimate of the eyebrow region is first obtained and then the exact eyebrow contour is extracted. First, the face contour is estimated by the Snake method, following which the eye corners are detected. With the eye

corners as reference, the eyebrow position is estimated. Let $w_e \times h_e$ be the size of the window in which the eye corners will be detected (for each eye). Multi-binarization of the image within this window is performed, to extract the eye boundary. Considering that 10% of the edge map is the edge content, within the $w_e \times h_e$ window, intersecting lines forming a corner are detected within a 7×7 window through the 48 line combinations that are defined within the 7×7 window. Then, once the corners are detected, they are grouped into clusters (n_{cl} clusters) if the distance between every 2 points of the cluster is less than a threshold. Computational cost is computed assuming n_c corners are detected. Region dissimilarity is computed for every corner point, which is the measure of difference in gray level averages in the two regions bounded by the edges forming the corner, the cost of which is computed. n_c comparisons are carried out to find out the corner with the largest D for every cluster. Then, pairs of corners from among the n_{cl} corner points are formed based on a distance measure, the cost of which is computed. Cost function for the resulting n_p such pairs is computed and the pair of points which gives minimum cost is found to be two of the eye corners. Computational cost of this step is computed. The next two corners are detected by evaluating a cost function for every edge pixel in the window of a certain size. All of the above computational cost will have to be multiplied by a factor of 2, since the above computational cost analysis was for finding corners of one eye. Summarizing the cost of the above computations in the following equations:

$$\begin{aligned} C_{COMP} &= C2[3(w_e h_e +^{n_c} C_2 +^{n_{cl}} C_2) + 2n_p + n_c + 4 * 0.1X^2] \\ C_{ADD} &= 2[3^{n_c} C_2 + 48n_c + n_p + 4 * 0.1X^2 + 4 * 0.1X + 2 *^{n_{cl}} C_2] \\ C_{MUL} &= 2[4n_p + 2n_c + 4 * 0.1X^2 + 8 * 0.1X] \\ & \quad \text{where } X \text{ is } 2(x_1 - x_0)x1.5(x_1 - x_0) \end{aligned} \tag{3.17}$$

Based on the eye corners obtained, the eyebrow location is approximately found. Then, a spatial constrained sub-area K-means clustering is performed in the region just above the eye based on the estimated eye corner positions. The computations for the spatial constrained sub-area K-means clustering is summarized as follows (m is the number of iterations in the spatial constrained sub-area K-means clustering).

$$C_{ADD} = 2 \times 5\frac{MN}{5}(3m + 6m + 3m + 3m) \quad (3.18)$$

$$C_{MUL} = 2 \times 5\frac{MN}{5}(6m + 6m + 4m) \quad (3.19)$$

$$C_{COMP} = m - 1 + 2\frac{MN}{5} \quad (3.20)$$

$$(3.21)$$

The above equations represent the computations for the spatial constrained sub-area K-means clustering considering 1 iteration. The equations are summarized in Table 3.2.

We now consider an image patch with the upper half of the face of size 100×200 for the sake of computational costs comparison between the proposed method and [175]. In the proposed method, the number of iterations and the number of candidates in each iteration have been set to reasonable values as follows: n_l = 3, n_d = 3, n_e = 3, n_B = 5, w_n = 3, Δ = 8. With respect to [175], 45×55 is the area considered in each eye region for eye corner detection and 25×50 is the estimated eyebrow region for K-means clustering and the value of m, the number of iterations for K-means clustering is set to three different values and the computations are tabulated in Table 3.3. From Table 3.3, it is seen that the total number of computations in the proposed method is 35% lesser when compared to [175], even when m is set to a reasonably low value of 5 (where m is the number of iterations in the spatial constrained sub-area K-means clustering used in [175]).

3.5 Summary

A compute-efficient technique to detect eyebrows as anchor points for the facial feature detection has been proposed. The method takes advantage of the unique properties of eyebrows that are retained in spite of changes in expressions on the face. Firstly, a candidate extraction step using signed edge maps and iterative thresholding was proposed. Then, the properties of eyebrows were used in the candidate verification step to eventually detect the eyebrow position. The technique was evaluated on three standard

databases - Cohn Kanade database, JAFFE database and AR database, with the images being those of front facing subjects. An average detection rate of 96% was achieved upon evaluation. Other challenging cases such as complete occlusion of one eyebrow, large head pose variation (more than $\pm 15^{\circ}$) and drastic lighting conditions can possibly be addressed by including additional features and symmetry. Cases such as painted eyebrows may also be detected, given that the method is based on detecting the light$\rightarrow$dark and dark$\rightarrow$light transitions, as long as the eyebrows are of minimum thickness. The computational complexity analysis shows that the proposed method achieves a computational savings of 35% compared to [175], while the detection rate is comparable to existing methods in literature. The eyebrows will be used as reference points in the localization of the facial features - eyes, mouth and brow furrows, as will be presented in the following 2 chapters.

CHAPTER 4

Extracting Wellness Indicators from Eyes

4.1 Introduction

In the previous chapter, a compute-efficient eyebrow detection technique was proposed. In this chapter, techniques to extract wellness indicators from the eyes are proposed. Eyes carry important information indicative of health and wellness. How open are the eyes; is the eyeball movement normal or very slow; is the rate of blinking very high or very slow - these are some of the important indicators of a patient's well-being that can be extracted from the eyes.

The wellness indicators from the eyes that are of interest in this work are - blink rate, blink duration, eye openness (eye state) and eyeball movement. They are extracted using the temporal analysis of eye related features - eyeball position and eye state. The eye related features and wellness indicators of interest in this work are summarized in Fig.4.1. In literature, techniques have been proposed for eye state and blink detection in the context of driver drowsiness monitoring. As seen from 2.6, the eye related features, such as the eyeball and eye state detection techniques involve complex computations, which may be unsuitable for running on embedded platforms which have resource constraints.

In order to address this challenge, computationally efficient techniques are proposed in this work, to extract the eye related features and hence the wellness indicators related to the eye.

In this chapter, techniques to extract eye related features are proposed first, followed by the techniques to extract eye-related wellness indicators based on the temporal analysis of eye features.

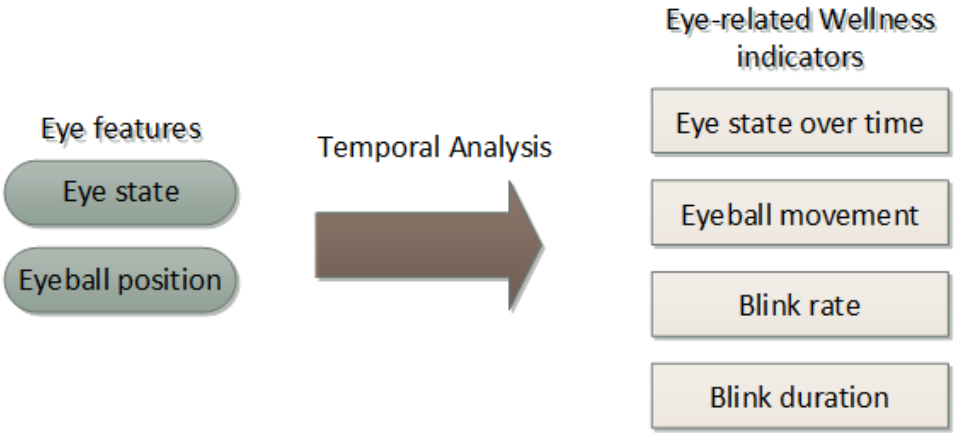


FIGURE 4.1: Eye-related features and wellness indicators of interest in this work

4.2 Proposed Method to Extract Eye Features

In this section, the method proposed to extract the basic eye features namely, the eyeball position and eye state are presented, starting with the steps involved in the estimation of the eye region of Interest(ROI). These features are then analysed temporally to extract the wellness indicators, which will be presented in the next section.

4.2.1 Estimation of Eye Region of Interest (ROI)

The ROI is initially estimated using the detected eyebrow position as reference. It is then fine-tuned to the region between the upper and lower eyelids.

4.2.1.1 Eye ROI Estimation

The eyebrows are located using the technique presented in Chapter 3. We have $P_R(x_R, y_R)$ and $P_L(x_L, y_L)$ - the co-ordinates of the mid-points of the detected eyebrow positions. Then, a rectangular window $w_e \times h_e$ centered at $P'_R(x_R, y'_R)$ and $P'_L(x_L, y'_L)$ is marked under the right and left eyebrows respectively as the initial region of interest (ROI) as shown below, where y'_R and y'_L are given by:

$$y'_R = y_R + 1.3 * e_t + 0.15 * w \tag{4.1}$$

$$y'_L = y_L + 1.3 * e_t + 0.15 * w \tag{4.2}$$

$$\tag{4.3}$$

and h_e, w_e and eyebrow thickness e_t are estimated based on anthropometric measures relative to the face width w [188], [189]:

$$h_e = 0.15 \times w \tag{4.4}$$

$$w_e = 0.22 \times w$$

This rectangular bounding window of size $w_e \times h_e$ marked under each eyebrow is the initial estimation of the eye ROI. The iniital ROI extraction from the eyebrow positions is illustrated in Fig. 4.2. At this step, the ROI is likely to be bigger than the actual eye region. The ROI is fine-tuned further to eliminate noise attributed by drastic changes in intensity in the regions surrounding the eye.

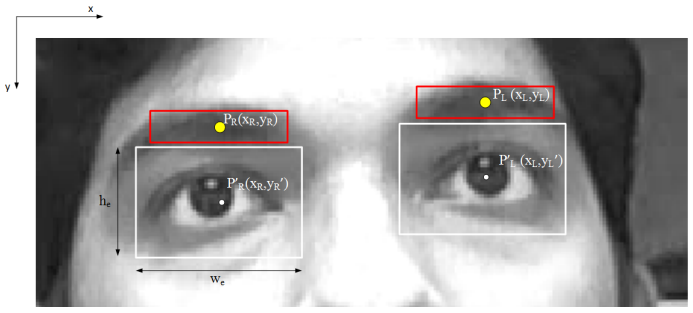


FIGURE 4.2: Initial eye ROI extraction with respect to the position of eyebrows, the red windows showing the position of eyebrows and the white windows showing the initial eye ROI

The initial ROI is fine-tuned by localizing the region between the upper and lower eyelids. In order to extract the position of upper and lower eyelids, we do not use edge detection directly, due to similar challenges that are encountered in eyebrow detection, as mentioned in 3.3.3.

The signed edge maps E_{y+} and E_{y-}, and the gradient magnitudes G_y that were extracted for the eyebrow detection 3.3.3 are reused. Fig. 4.3 shows the initial ROI and final ROI in Fig. 4.3. (a) and (e) respectively, and Fig. 4.3. (b) to (d) show the partial gradient maps that are used in the process of fine tuning the eye ROI.

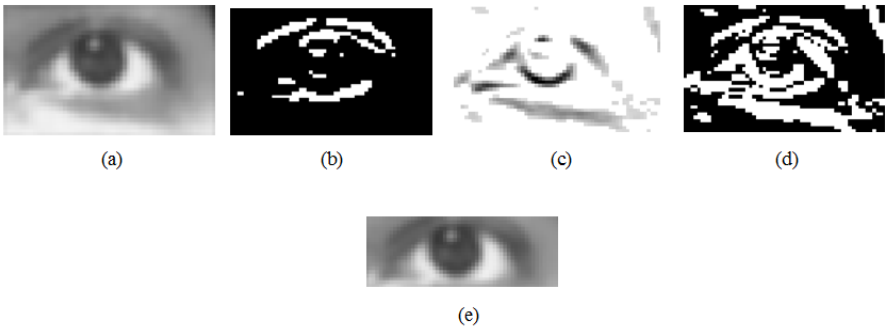


FIGURE 4.3: (a) Initial eye ROI within window marked under eyebrow (b) Light to Dark transition map (c) Dark to Light transition map (d) Combined gradient map showing both the LD and DL transitions (e) The final eye ROI that is sent for further steps of eye feature extraction

4.2.2 Eyeball Detection

We now present the technique for eyeball detection. Firstly, the distinct properties that are characteristic of the eyeball are observed and listed below:

- P1: The eyeball is surrounded by the sclera, the white portion in the eye. The eyeball is relatively darker than sclera and hence, they have a local intensity variation between them.
- P2: The eyeball $\rightarrow$ sclera and sclera $\rightarrow$ eyeball transition is featured by a distinct change in intensity
- P3: The eyeball lies within the bounds of the eyelids.

- P4: The eyeball and eyelashes on the upper eyelid are among the darkest parts in the eye region relative to their surrounding local region.
- P5: The width of the eyeball lies within defined bounds computed using anthropometric measures relative to the face width.

Eyeball detection is applied within the eye ROI as follows. The technique is based on extracting the characteristic local variation in intensity and gradients within the eye based on the properties P1 to P5 listed above.

4.2.2.1 Weighted Accumulation Maps

In the first step of the eyeball detection, two kinds of weighted accumulation maps are generated - intensity-weighted and gradient-weighted accumulation maps. In order to generate the intensity-weighted accumulation map, the fine-tuned ROI is divided into horizontal bands in order to capture the local intensity variation between the eyeball and the sclera (See Fig. 4.4 (a)). The fine-tuned eye ROI is divided into n overlapping bands $(b_1, b_2, ..b_j, ..b_n)$ with 50% overlap between them. For each band, an intensity weighted accumulation map $\mathbf{S}_{b_j} = \{S_i\}$ is computed such that

$$\begin{aligned} \mathbf{S}_{b_j} &= [S_0, S_1, \ldots, S_i, \ldots, S_{N_b}] \\ \mathbf{S}_{b_j} &= [\sum_{k=1}^{h_b} I_{(k,0)}, \ldots, \sum_{k=1}^{h_b} I_{(k,i)}, \ldots, \sum_{k=1}^{h_b} I_{(k,N_b)}] \end{aligned} \tag{4.5}$$

where $I_{(k,i)}$ is the gray level intensity value of a pixel in the k-th row of the i^{th} column within the band and h_b is the height of the band. The summation values are normalized with respect to the highest summation value in the band i.e, $N_i = S_i/S_{i_{max}}$. N_b in the above equations refer to the width of the ROI (or the band).

Similarly, the gradient magnitude weighted accumulation maps are generated within each band, as follows. The gradients G_{x_+} and G_{x_-} of the signed edge maps are computed along the x-direction in a manner similar to that explained in Sec. 3.3.3, but along the x-axis. G_{x_+} is used to detect the eyeball edge from iris to sclera and G_{y_-}

is used to detect the eyeball edge from sclera to eyeball. G_{x_+} and G_{x_-} are considered separately and summed for every column along the x-axis within each band. So, $\mathbf{G}_{x+_{(b_j)}} = \sum_{k=1}^{h_b} G_{x_+(k,i)}$, where $G_{x+_{(i)}}$ is the summation of the gradient magnitude G_{x_+} along i^{th} column within the band b_j. The gradient magnitude is high at the points of transition from eyeball to sclera, and hence, their accumulation will reflect as a peak in $\mathbf{G}_{x+_{(b_j)}}$. Similarly, the transition from sclera to eyeball will reflect as a peak in $\mathbf{G}_{x-_{(b_j)}}$. The most prominent peaks in $\mathbf{G}_{x+_{(b_j)}}$ and $\mathbf{G}_{x-_{(b_j)}}$ are considered and are used to ascertain the presence of the eyeball, as explained in the paragraphs to follow.

Now, the steps in eyeball detection, after having extracted $\mathbf{S}_{b_j}$, $\mathbf{G}_{x+_{(b_j)}}$ and $\mathbf{G}_{x-_{(b_j)}}$ are explained. consider $\mathbf{S}_{b_j}$. When the intensities of the eyeball and sclera are accumulated within the bands, they reflect as a peak and valley respectively in $\mathbf{S}_{b_j}$.

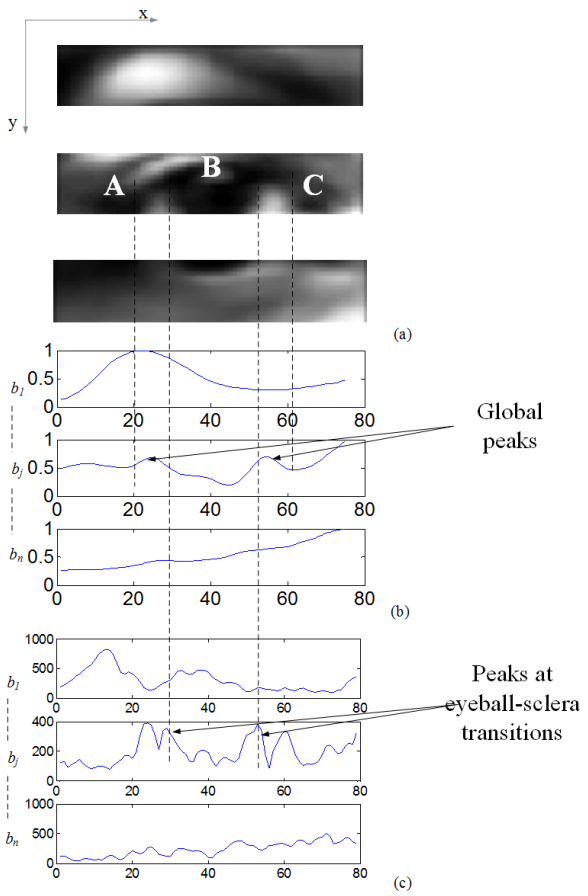


FIGURE 4.4: (a) Eye region (ROI) divided into bands (b) Normalized band-wise intensity weighted accumulation map of eye region (c) Band-wise gradient magnitude weighted accumulation map of eye region

So, for a center-looking eye, the eyeball is reflected as a valley bounded by a peak on either side in the $\mathbf{S}_{b_j}$, or in other words, as a peak-valley-peak (PVP) pattern, as shown in Fig. 4.6. In the case of a right or left-looking eye (with respect to the subject himself), the eyeball-sclera will reflect as a valley followed by a peak, or a peak followed by a valley respectively in $\mathbf{S}_{b_j}$. In other words, they are reflected as a VP or a PV pattern, as shown in Fig. 4.5. Note the increase in peak width in the case of side-looking eye at the transition from eyeball to sclera.

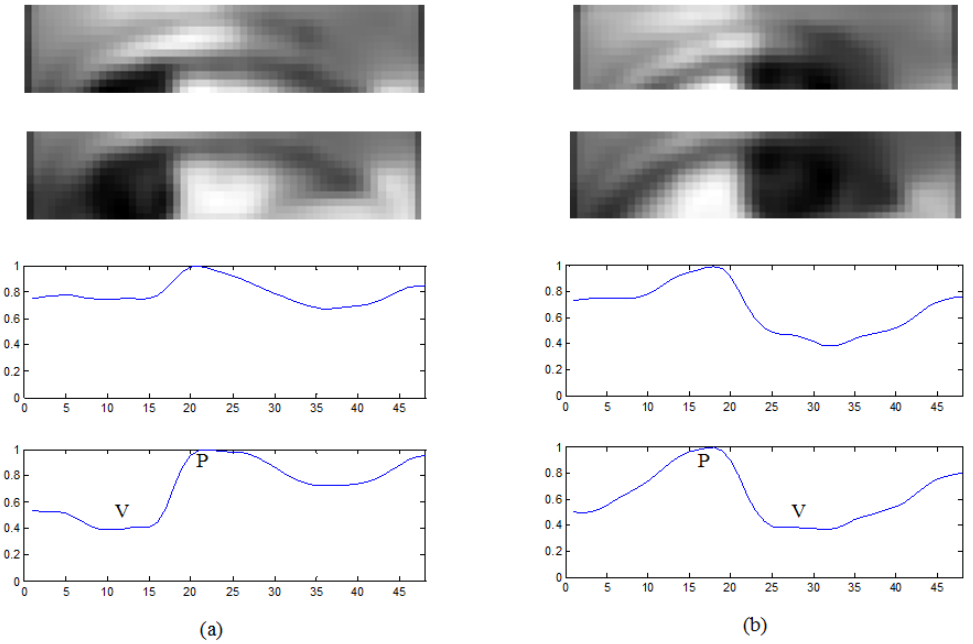


FIGURE 4.5: Fine-tuned eye ROI divided into bands shown along with the respective Normalized band-wise intensity weighted accumulation maps showing the VP and PV combination respectively for a (a) right looking eye and (b) left looking eye

The band-wise gradient magnitude-weighted accumulation maps $\mathbf{G}_{x+(b_j)}$ and $\mathbf{G}_{x-(b_j)}$ are used to eliminate peak-valley patterns extracted from noise. This is done by checking for the presence of a peak in $\mathbf{G}_{x+(b_j)}$ at the region of transition from the valley to peak along the x-axis in the corresponding band in $\mathbf{S}_{b_j}$ and for a peak in $\mathbf{G}_{x-(b_j)}$ at the region of transition from the peak to valley (Fig. 4.4).

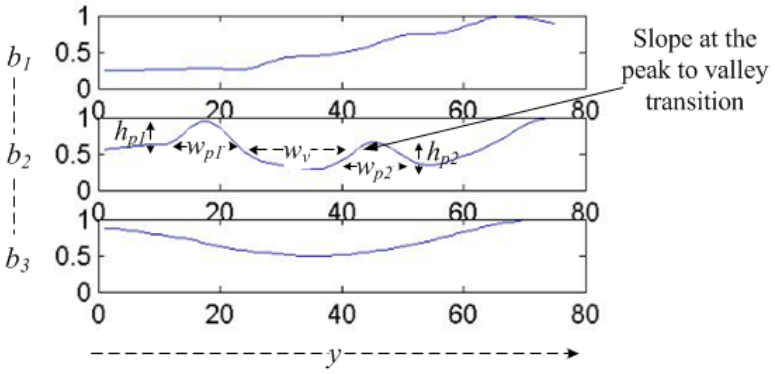


FIGURE 4.6: Peak-valley analysis of the normalized band-wise intensity weighted accumulation map

4.2.2.2 Peak-Valley Analysis

(a) Candidate extraction: After the potential peaks and valleys in $\mathbf{S}_{b_j}$ in each band are extracted, the bands are prioritized based on the height and number of potential PVP, PV and VP candidates and the presence of peaks in the band-wise gradient-weighted accumulation maps $\mathbf{G}_{x-(b_j)}$ and $\mathbf{G}_{x+(b_j)}$. The peak height is computed in the following manner: the height of the peak with respect to the neighboring valley in the accumulation map is computed, and since this value is normalized, it is multiplied by the band size w to obtain the actual height of peak. Fig. 4.7 shows the eye in fully open (in (a)) and partially open states (in (b) and (c)) and the respective peak heights computed based on the band size. The mean height of the inner edges of the peak bounding a valley are considered for computing the peak height.

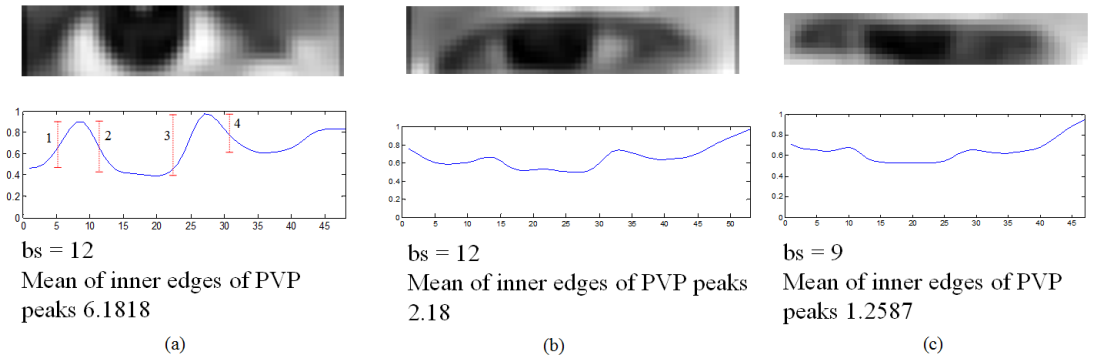


FIGURE 4.7: Computation of peak height considering the width of the band

In this work, the further steps were performed only on the top two bands with highest priority.

(b) Candidate verification: In the candidate verification step, the PVP, PV and VP candidates from the highest priority band are checked for the following properties that are distinct to an open eye - peak height h_p, peak width w_p, valley width w_v, ratio of gray-level intensities between the regions corresponding to the peak and valley, slopes of the peaks in $\mathbf{S}_{b_j}$ (Fig. 4.6); height of the peaks at the eyeball-sclera transition in $\mathbf{G}$ (Fig. 4.4).

If a candidate does not satisfy any of the properties, the next candidate is considered and this step is repeated until the candidate that satisfies the properties corresponding to the eyeball-sclera region is found. If no eyeball is detected at this step, the band size h_b is decreased and the aforesaid steps, starting from generating the accumulation maps are repeated.

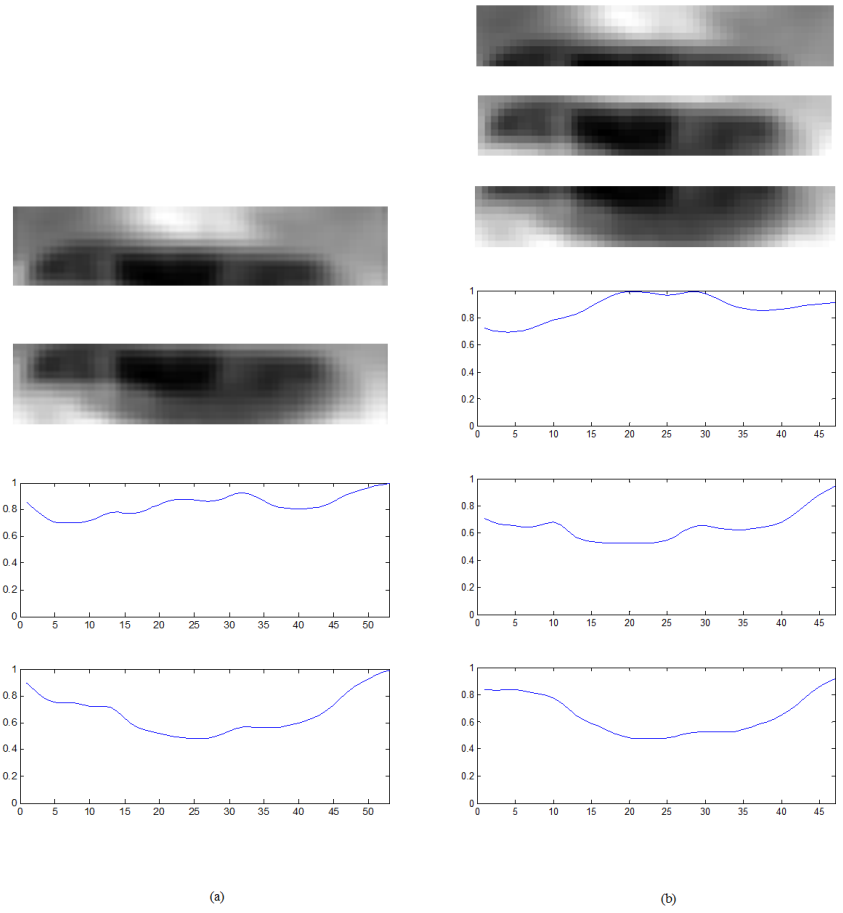


FIGURE 4.8: Detection of eyeball in partially open eye in the second iteration with a smaller band size, band sizes in pixels in (a) h_b = 12, (b) h_b = 9

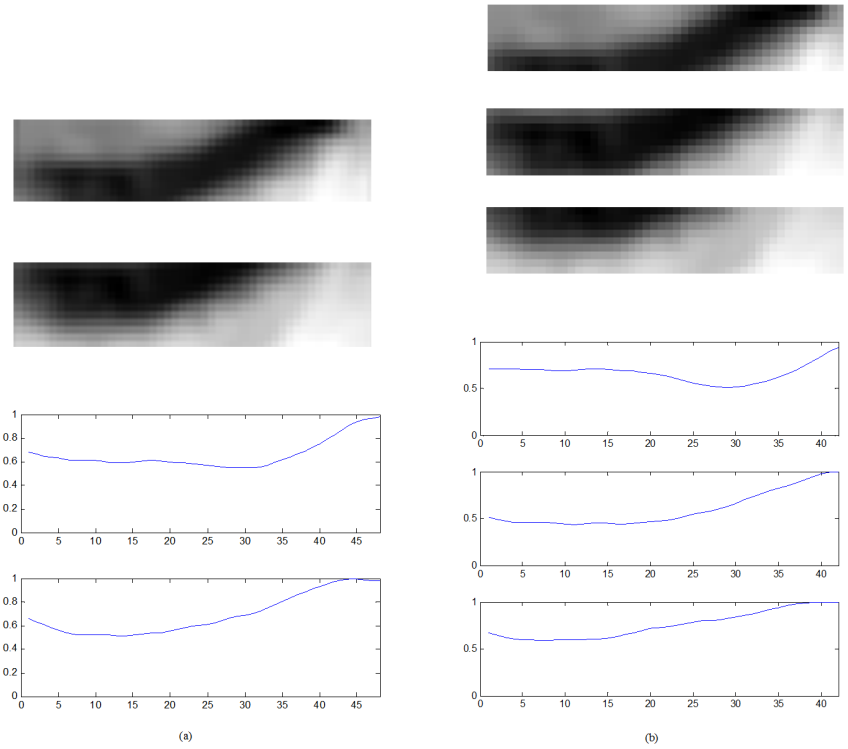


FIGURE 4.9: Eye ROI divided into bands shown along with the respective normalized bandwise intensity weighted accumulation maps for closed eye in two iterations shown in (a) and (b) respectively

The iterative process is repeated until the eyeball is found or the stopping condition is reached i.e $h_b < 0.5 * D_e$, where D_e is the eyeball diameter. The iterative approach has many benefits. First of all, if the eyeball nearly matches the starting band size, it is detected in the first iteration. If the eyes appear relatively smaller in size, as in the case of certain ethnic backgrounds, then, it is likely that the eyeball is detected in the further iterations. The other possibility is when the eye is partially open. As shown in Fig. 4.8 where the eyeball is detectable in the second iteration. In the experiments conducted, the algorithm was restricted to two iterations. Since we perform a normalization of the accumulation values within each band, the smaller the band size, the more we are enhancing lesser information. For example, in the case of closed eye, a valid peak and valley combination is not observed (as shown in Fig. 4.9). However, if the band size is reduced and the image has high contrast, then it is possible for the accumulation map of the closed eye to appear as a weak peak valley combination.

The steps in the eyeball detection technique are summarized in Fig. 4.10.

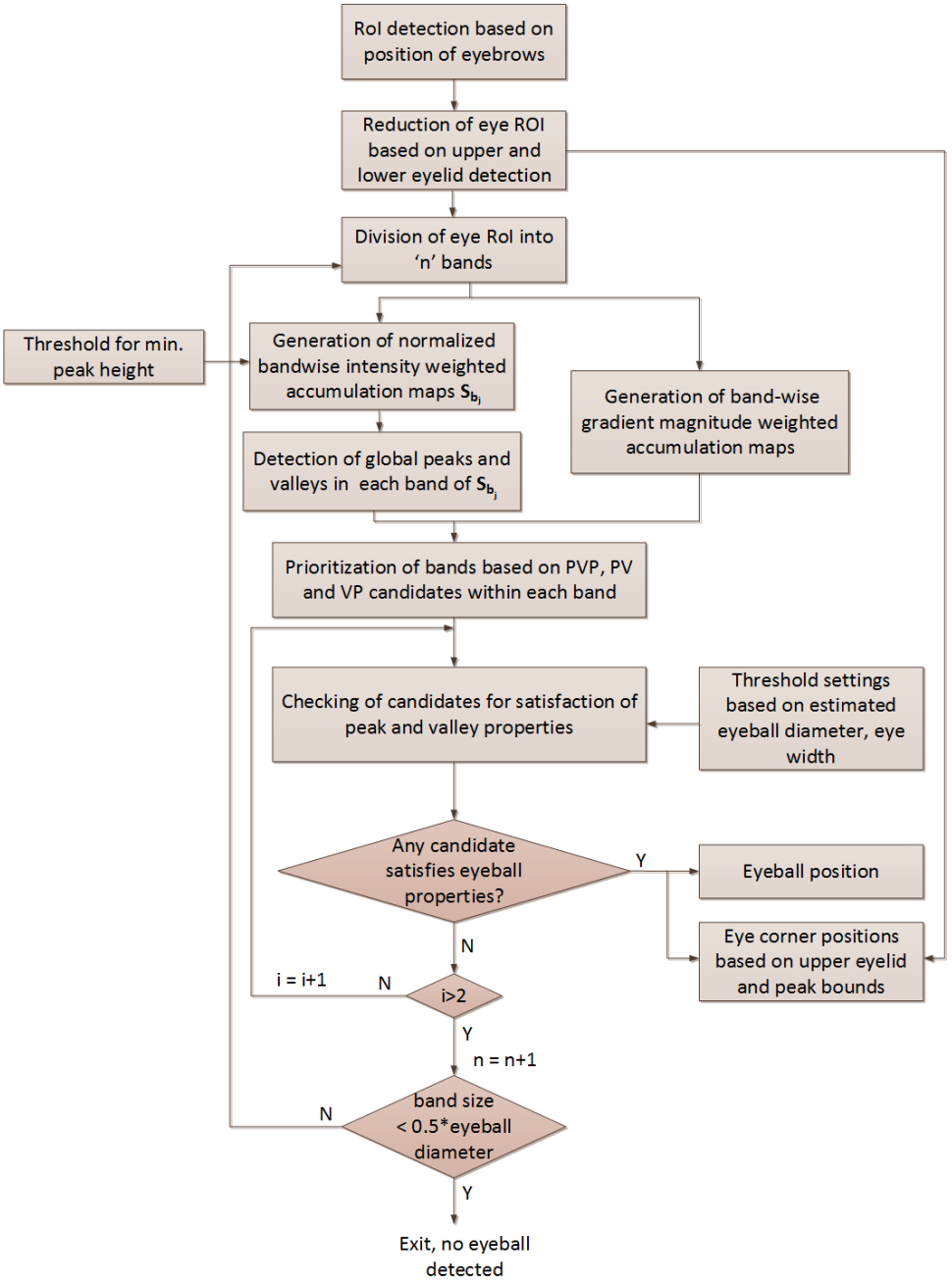


FIGURE 4.10: Steps in the eyeball detection technique

Experiments were also conducted to find out the starting band size and the band size at stopping condition. The band sizes were set with respect to the estimated eyeball diameter D_e using anthropometric measures relative to the face size (we set it to $0.057 \times f_w$). The detection rate for a fully open eye is not affected if the band size is reduced. Based on an empirical study, it was found that a starting band size of 0.75*eyeball diameter gave the best results.

4.2.3 Eye State Detection

In this section, a technique is proposed to detect the eye states namely, open, closed and partially-open. The partially open state is when the eye is neither fully open as in the *open* state nor fully closed as the *closed* state. The three states are shown in Fig. 4.11.

FIGURE 4.11: Three states of the eye (a) Open, (b) closed and (c) partially open

The proposed eye state detection system consists of the following main parts: (1) feature extraction, (2) initialization module and the (3) detection module.

The basic feature extraction step for eye state detection is discussed first. This step includes the peak-valley analysis in eyeball detection (Section 4.2.2), blob analysis in grayscale and computation of the eyebrow-eyelid distance. Then, the initialization module is discussed, in which the values of the parameters that are used to distinguish the states are initialized. The decision module is then discussed, whose output is the eye state decision based on the parameters learned during initialization. The individual modules are discussed in the following paragraphs.

4.2.3.1 Basic Feature extraction

Given an input image, the eye ROI is detected based on the steps described in Section 4.2.1. Then, the following steps for extracting the basic features needed to detect the eye state are performed.

- **Peak-valley analysis for eyeball detection**: The peak valley analysis step as explained in Sec. 4.2.2.2 is performed to extract the valid PVP, PV or VP combinations indicative of the eyeball. The iteration and the corresponding band size

in which the valid combinations were detected is noted and provided as input to the eye state decision module, which is discussed next.

- **Blob analyis in grayscale:**

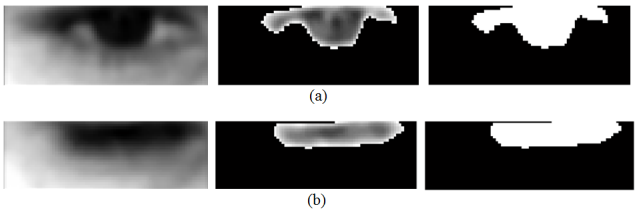


FIGURE 4.12: Blob of the darkest pixels extracted from grayscale images of (a) open and (b) closed eye

$$\begin{aligned} \mathbf{s_B} &= [s_0, s_1, \ldots, s_i, \ldots, s_{B_l}] \\ {\sigma^2}_B &= \sigma^2[\mathbf{s_B}] \end{aligned} \tag{4.6}$$

The darkest pixels are extracted in the eye ROI (which most often includes at least the pupil along with eyelid and eye lashes in case of open eye and the eyelid with eye lashes in case of closed eye. Fig. 4.12 shows examples of blobs extracted from open and closed eyes. The blob of the closed eye will appear more uniform in thickness along the y-axis, whereas the blob of open eye is more likely to appear non-uniform in thickness, due to the presence of eyeball. In order to quantify this difference, the number of pixels in the blob along every column along the x-axis $\mathbf{s_B}$ is computed and the variance of $\mathbf{s_B}$ about the x-axis is computed (4.6, where B_l is the length of the blob). The variance is higher in case of open eye compared to closed eye.

- **Eyebrow-eyelid distance**: The distance between the eyebrow and upper eyelid d_{eu}, and the distance between the eyelids $d_u l$ are computed as follows:

$$\begin{aligned} d_{eu} &= y'' - y' \\ d_{ul} &= y''' - y'' \end{aligned} \tag{4.7}$$

where y', y'' and y''' are the y co-ordinates of the eyebrow, upper eyelid and eyebrow respectively. d_{eu} and d_{ul} combinedly give a measure of the openness of the eye. As shown in the Fig. 4.13, d_{eu} is the least and d_{ul} is the highest when the eye is in open state, and d_{eu} is the highest and d_{ul} is least when the eye is in closed state.

FIGURE 4.13: Images showing the combination of the change in eyebrow-upper eyelid distance and the distance between upper and lower eyelids being used to in eye state extraction

Having discussed the method to extract the basic features, the overall approach in eye state detection is explained next - it consists of two steps: *initialization* and *eye state decision*.

4.2.3.2 Initialization step

In the initialization step, threshold values for the unique features - height and width of the peaks, width of the valley, d_{eu}, d_{ul} and the value of ${\sigma^2}_B$, that distinguish closed and open eye are learned from images of the subject with open and closed eyes. The closed and open states being the extreme states, the thresholds are set such that the two states are clearly distinguishable. Since the partially open state is more fuzzy to determine, the thresholds for this state are derived based on those set for open and closed states.

In the context of patient monitoring, an additional input is taken from the doctor during the initialization step. The doctor is required to look at the patient and determine if the patient's eyes are fully open or partially open. In some cases, for e.g., the patient is drowsy or extremely tired, his eyes may not be open to the full extent possible. The algorithm would have by default considered the openness of the eye to be the maximum, although the eye was only partially open. But, the doctor can specify the % openness

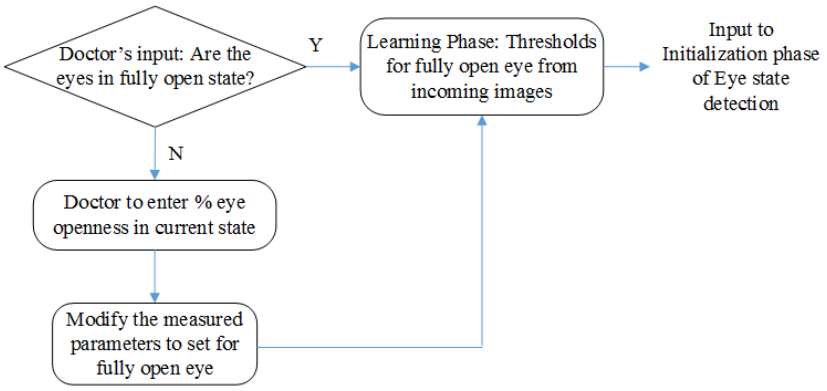


FIGURE 4.14: Doctor's input as a feedback while initializing the eye related thresholds

of the eye with respect to the maximum openness for that patient. Hence, the doctor's input is important at the initialization step. The above steps are summarized in Fig. 4.14.

The thresholds set in the initialization step are then used as reference in the detection step.

4.2.3.3 Eye state decision module

During the detection step, the peak-valley analysis for eyeball detection, blob detection and the eyebrow-eyelid distances are extracted for an incoming image, which are then combined to classify the eye state into open, closed or partially-open. Scores are assigned based on how far the extracted parameter values lie with respect to the thresholds that distinguish closed and open eye.

TABLE 4.1: Possible cases in the eye state detection module

Peak-valley analysis	Blob analysis	Eyebrow-eyelid distances	State
Valid peak-valley candidate	$\sigma_B^2 >= {\sigma^2}_{B_o}$	$d_{eu} <= d_{eu}^o$ $d_{ul} >= d_{ul}^o$	Open
No valid peak-valley candidate	$\sigma_B^2 <= {\sigma^2}_{B_c}$	$d_{eu} >= d_{eu}^c$ $d_{ul} <= d_{ul}^c$	Closed
A weak peak-valley candidate	$\sigma_B^2 <= {\sigma^2}_{B_o}$ or $\sigma^2 >= {\sigma^2}_{B_c}$	$d_{eu}^c > d_{eu} > d_{eu}^o$ $d_{ul}^c < d_{ul} < d_{ul}^o$	Partially-open

The different possible cases and the eye state classified are summarized in Table. 4.1. A valid peak-valley candidate refers to a candidate with peak height, peak width and valley width that cross the thresholds set for open state. A weak peak-valley candidate refers to the peak height being $\leq$50% of the peak height of open state, while peak width and valley width cross the thresholds of open state. The eyebrow-upper eyelid distance in the case of a fully open eye d_{eu}^{o} and that of a fully closed eye d_{eu}^{c} are those that were noted in the initialization step. So, $d_{eu} >= {d_{eu}}^{c}$ indicates that the eye is nearly closed or closed. $d_{eu} <= {d_{eu}}^{o} + e'$ is indicative of the eye being in open state. Similarly, the distance between the eyelids $d_{ul} >= d_{ul}^{o} + e''$ indicates eye is open and $d_{ul} <= d_{ul}^{c}$ indicates eye is closed. The thresholds ${\sigma^2}_{c}$ and ${\sigma^2}_{B_c}$ are set based on the parameter values in the closed and open states.

Scores are computed based on the conditions discussed above. A combined score for the left and right eye state is computed, and the state with the highest score is considered as the eye state S_{f_i} in a frame f_i. In case the closed and open states have equal score, the output state is determined to be partially-open. In case the partially-open and the open states have the same score, the output state is determined to be open, similarly for closed and partially-open states. The steps in the proposed eye state detection technique are summarized in Fig. 4.15.

4.3 Temporal Analysis to Extract Eye-based Wellness Indicators

At the frame level, the eye features namely, eye state and eyeball position are extracted. Then, the eye related wellness indicators are extracted through the temporal analysis of these eye features. The wellness indicators that are extracted are - eye state over time, blink rate, blink duration (derived from eye blink) and eyeball movement. Methods to extract these wellness indicators are presented in the following paragraphs.

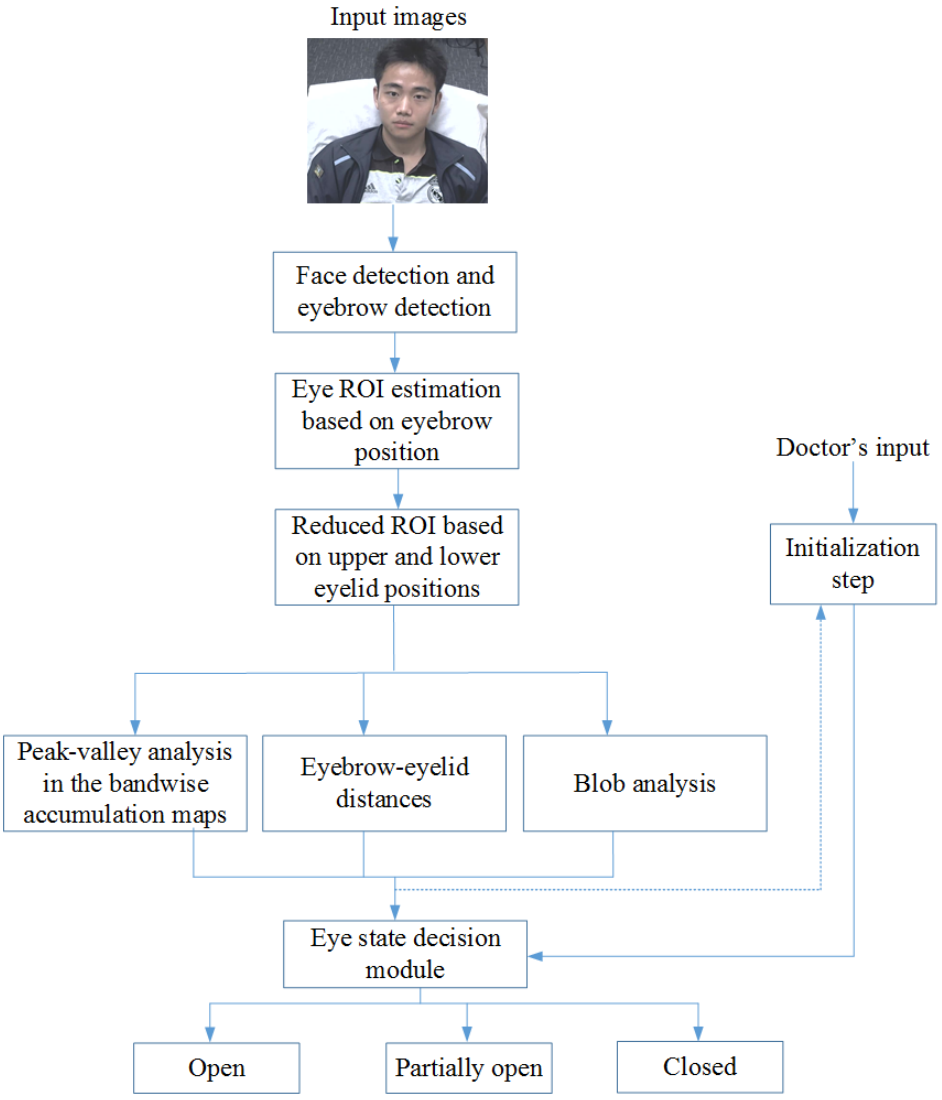


FIGURE 4.15: The proposed eye state detection system

4.3.1 Eye State over Time

Tracking the eye state over time provides information on whether the patient was awake with eyes open, or asleep with eyes closed or half-awake with eyes partially open. In order to extract the eye state over time, the percentage of frames with eyes open, closed and partially open is computed within overlapping windows of width W frames and the state with the highest percentage is voted as the predominant state in that window. Let S_{f_i} be the state detected in a frame f_i. Over a window W, the most predominant state is found. Then, in each such window W_i, the predominant state found will be S_{W_i}. Over time, $\mathbf{S_W}$ will be the array of values of eye state S_{W_i}. In other words, the history over

the last N number of W windows is stored as $\mathbf{S_W}$.

$$\mathbf{S_W} = [S_{W_1}, S_{W_2}, \ldots, S_{W_i} \ldots, S_{W_N}] \tag{4.8}$$

The next level of inferencing is done by computing the predominant state over overlapping windows W' where W' will combine m number of W windows. Let $S_{W'_j}$ denote the predominant state within W'_j. Then, the eye state over P number of W' windows is given by:

$$\begin{aligned} W' &\approx 3 \times W \\ \mathbf{S'_W} &= [S_{W'_1}, S_{W'_2}, \ldots, S_{W'_j} \ldots, S_{W'_P}] \end{aligned} \tag{4.9}$$

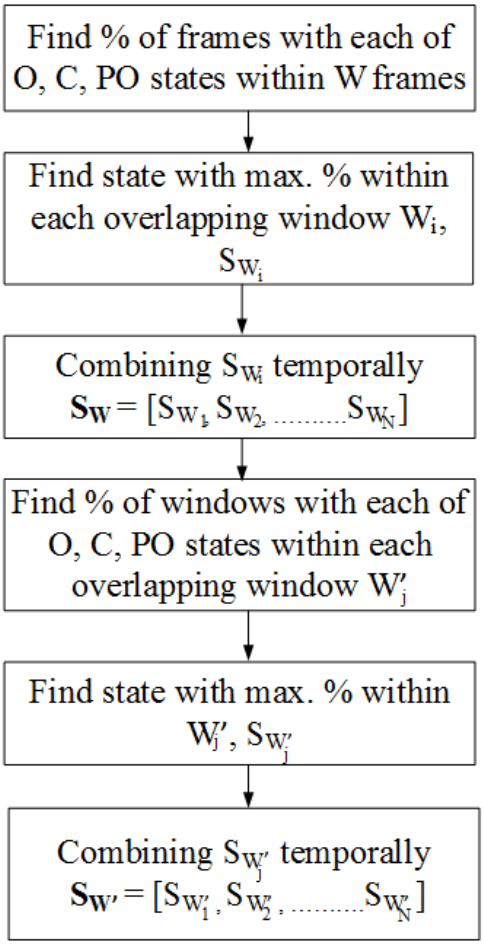


FIGURE 4.16: Steps in the extraction of eye state over time, where O, C, PO refer to open, closed, and partially open states. The local eye state extraction is done within window W and global extraction is done within window W'.

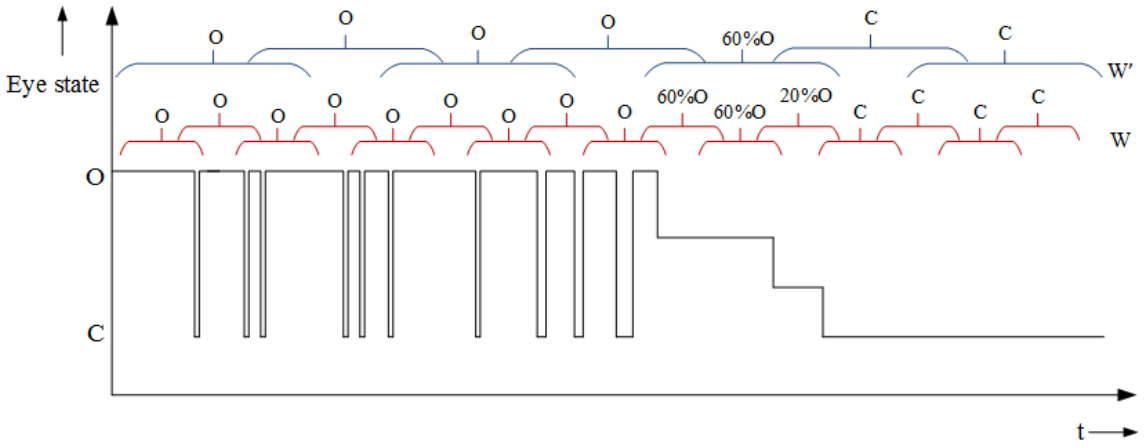


FIGURE 4.17: An example illustrating the local and global temporal extraction of eye state, W and W' are the overlapping windows. O and C refer to open and closed states respectively. Eye state extraction at global level is done using W'.

So, the windows W and W' are of different sizes, and enable the local and global extraction of the eye state respectively. The above steps in temporal extraction of eye state are summarized in Fig. 4.16. An example of temporal eye state extraction is shown in Fig. 4.17. In the example shown in this figure, the eye state transitions from fully open, to partially open, to fully closed.

4.3.2 Eye Blink

An eye blink is detected when the eye state transitions from open to closed for a brief period and transitions back to open state. Similar to the state machine proposed in [136], a state machine based on the proposed method is shown in Fig. 4.18. The initial state is denoted by state 0. When the eye transitions from open to closed, the state transitions to state 1. Now, the frame count is incremented for every frame that the eye is in closed state. If the frame count exceeds a threshold t_b which denotes the duration of blink, that is set based on the frame rate, then the state machine returns to state 0, since it is no more classified as a blink. When the eye transitions back to open state before the frame count is exceeded, the state transitions to state 2, which denotes a blink. A sequence of images showing the transition from open state to closed, and then back to open, that occurs in a blink is shown in Fig. 4.19.

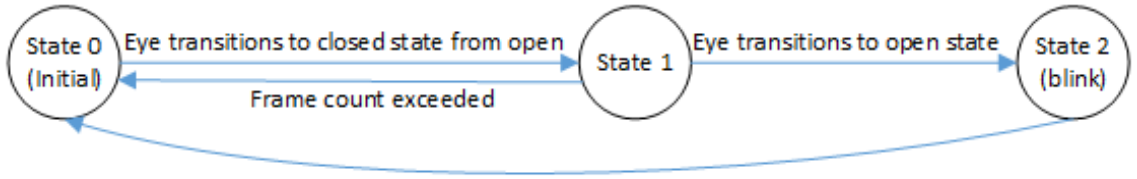


FIGURE 4.18: State machine showing the three states in a blink

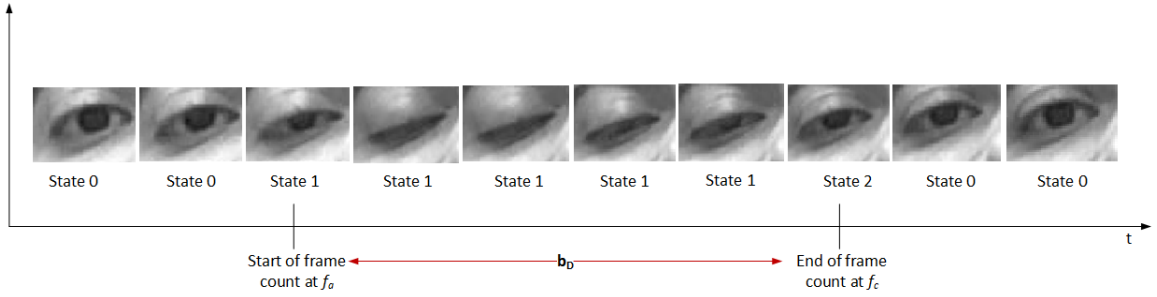


FIGURE 4.19: Sequence of images showing the transition from open state to closed, and then again to open, that occurs in a blink

The steps in blink detection are shown in Fig. 4.20. Note that, the computation of frame count for blink duration starts when the eye begins its transition into closed state via a partially open state of less than 50% open. Similarly, when the eye transitions to open state via a partially open state of more than 50% open. By detecting the blink event, two wellness indicators can be extracted, viz., blink duration and blink rate. Usually, the average duration of a single blink is about 150 to 300ms (5 to 10 frames while 30fps) [136]. In order to compute the blink duration, we use the definition in [138], where the blink duration is defined as the number of consecutive frames of closure in a blink. The blink duration in seconds will then be computed as a product of the blink duration in frames and frame rate, which is used to set the reference blink duration. If the frame at which eye state transitions from open to close is denoted by f_{OC} and f_{CO} is the frame at which the eye state transitions back to open state and a blink has been detected, then, the blink duration $\mathbf{b_D}$ is given by (4.10). In the context of patient monitoring, threshold t_b is set such that the detection of blinks with longer blink duration are accommodated.

$$\mathbf{b_D} = (f_{CO} - f_{OC} + 1)/f \tag{4.10}$$

Blink duration is an important indicator of the states of a patient such as drowsiness and inactive. During these states, it is observed that the blink duration increases than what

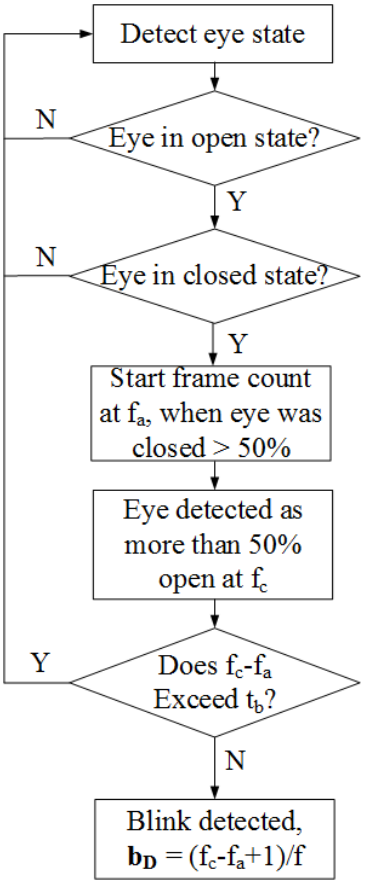


FIGURE 4.20: Sequence of steps in blink detection

is observed during normal states. The frequency of blinking or in other words, blink rate (number of blinks per minute) is extracted as follows: the number of blinks within overlapping windows that span across $f * 60$ frames are computed, where f is the frame rate. If the number of blinks b_{N_i} within a local window w_i of f' frames are known, then the blink rate within w_i, R_{w_i} is computed using the following equation. $\mathbf{R}$ is the array where R_{w_i} values are stored across N number of such windows.

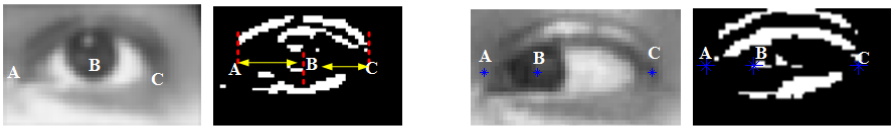
$$\begin{aligned} R_{w_i} &= b_{N_i}/f' \times f * 60 \\ \mathbf{R} &= [R_{w_1}, R_{w_2}, \ldots, R_{w_i} \ldots, R_{w_N}] \end{aligned} \tag{4.11}$$

The higher level of inferencing is done by computing the mean of the blink rate within overlapping windows w' that combine n number of local windows, as shown in the

equation below. $\mathbf{R'_w}$ is the array of mean blink rate values across windows of width w'.

$$\begin{aligned} \mu_{w'_k} &= \sum_{j=1}^{n}(R_{w_i})/n \\ \mu'_{\mathbf{w}} &= [\mu_{w'_1}, \mu_{w'_2}, \ldots, \mu_{w'_k} \ldots, \mu_{w'_{N'}}] \end{aligned} \tag{4.12}$$

4.3.3 Eyeball Movement



(a) (b)

FIGURE 4.21: Distance from eyeball center to the left and right corners of the eye as features for eyeball movement detection (a) center looking eye, (b) side looking eye

We extract eyeball movement by computing the variance of the distance of the eyeball from the eye corners, over time. Referring to Fig. 4.21, the distances AB and BC are measured along the x-axis, and are shown for a center-looking and side-looking eye. The distance of the eyeball from the left and right corners of the eye are used for the right and left eyes respectively. This is because - based on the experiments conducted using the proposed eyeball detection technique, it was found that the inner eye corners are more impacted by shadow than the outer corners, and hence the outer corners of the eye (i.e., the left and right corners for the right and left eyes respectively) were more accurately detected.

Now, in order to extract the eyeball movement temporally, the images in the sequence are first resized to the same size. The distances AB_R and BC_L computed in each image are tracked over time (AB_R and BC_L refer to AB with respect to right eye and BC with respect to left eye respectively). The variance of AB_R and BC_L are computed across frames to compute the eyeball movement. Low variance is indicative of low eyeball movement and high variance is indicative of high eyeball movement. In Fig. 4.22.(a), an image sequence showing very low movement of eyeball is shown, sampled at every 100^{th} frame, from the video of 25 fps. The distances, AB_R and BC_L, were plotted (Fig. 4.22.(b) and (c) respectively). The variances computed for AB_R and BC_L were 3.17

and 0.6 respectively. A low variance indicates low eyeball movement and the position of the eyeball tells where the patient has been looking at.

Similarly, an example of a sequence showing high or restless movement of the eyeball is shown in Fig. 4.23.(a), sampled at every 10^{th} frame. The distances, AB_R and BC_L, are plotted in 4.23.(b) and (c) respectively. The variances computed for AB_R and BC_L over the frames were 17.22 and 19.94 respectively. Thus, the high variance is indicative of high or restless movement of the eyeball. The eyeball movement is tracked both

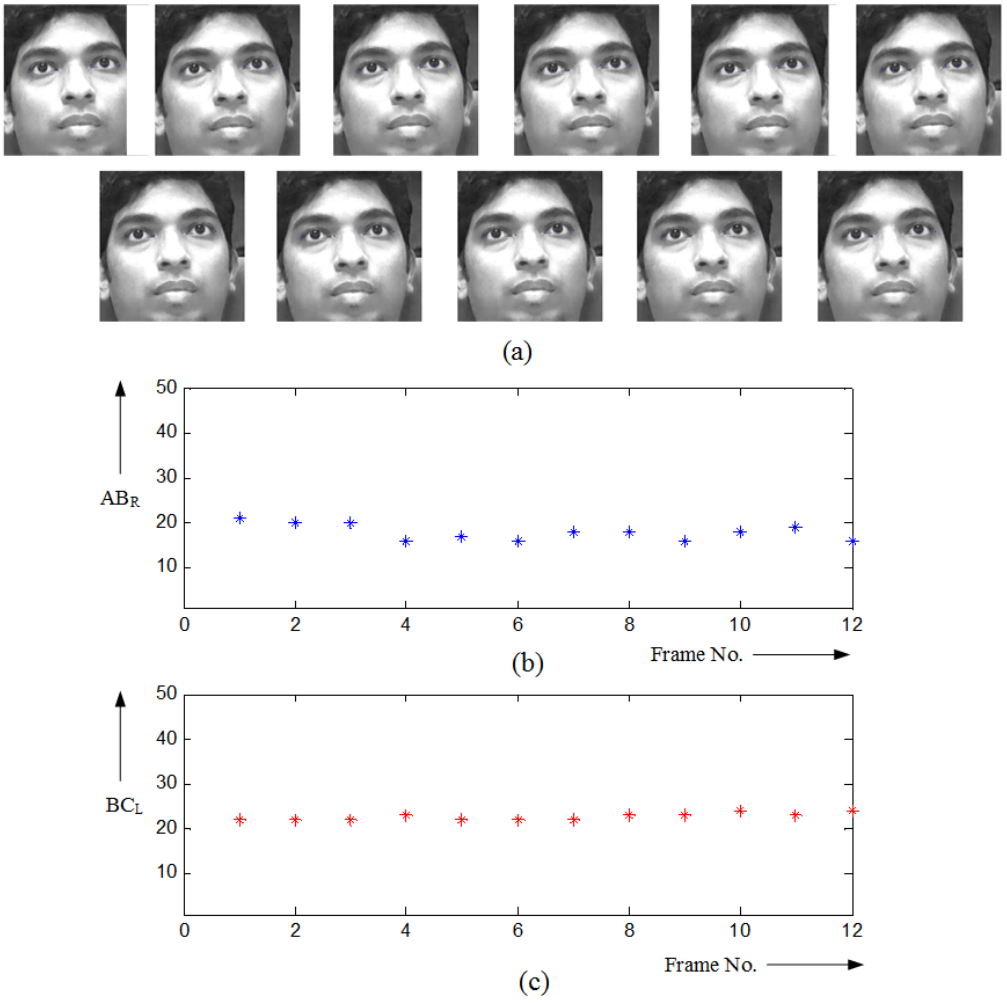


FIGURE 4.22: (a) Image sequence showing low eyeball movement across frames, (b) Plot of AB_R and (c) plot of BC_L

locally and globally. First, within overlapping windows, the variances $\sigma^2_{AB_R}$ and $\sigma^2_{BC_R}$ of AB_R and BC_L are computed respectively. Then, the mean β_M of $\sigma^2_{AB_R}$ and $\sigma^2_{BC_R}$ is computed. So, within a window M_i, the above computations are computed as follows,

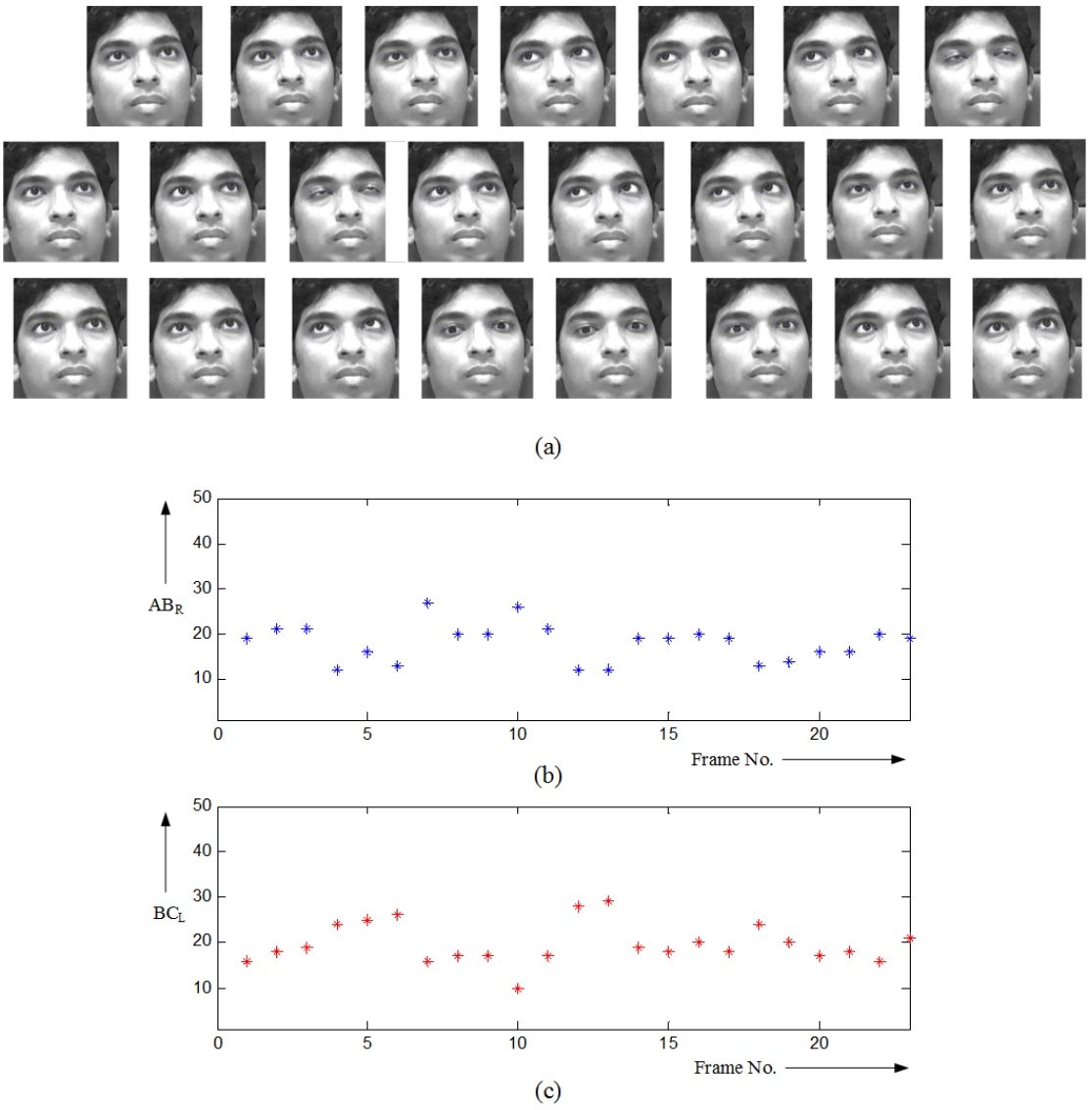


FIGURE 4.23: (a) Image sequence showing high or restless eyeball movement across frames, (b) plot of AB_R and (c) plot of BC_L

and $\beta_{\mathbf{M}}$ is the array storing β_{M_i} from each window M_i (as shown in the equation below):

$$\begin{aligned} \sigma^2_{AB_{R_i}} &= \sigma^2(AB_{R_j}) \; | \forall j \in M_i \\ \sigma^2_{BC_{L_i}} &= \sigma^2(BC_{L_j}) \; | \forall j \in M_i \\ \beta_{M_i} &= mean(\sigma^2_{AB_{R_i}}, \sigma^2_{BC_{L_i}}) \\ \beta_{\mathbf{M}} &= [\beta_{M_1}, \beta_{M_2}, \ldots, \beta_{M_i}, \ldots \beta_{M_P}] \end{aligned} \tag{4.13}$$

The higher level inferencing is done by computing the mean of $\beta_{\mathbf{M}}$ within L overlap-

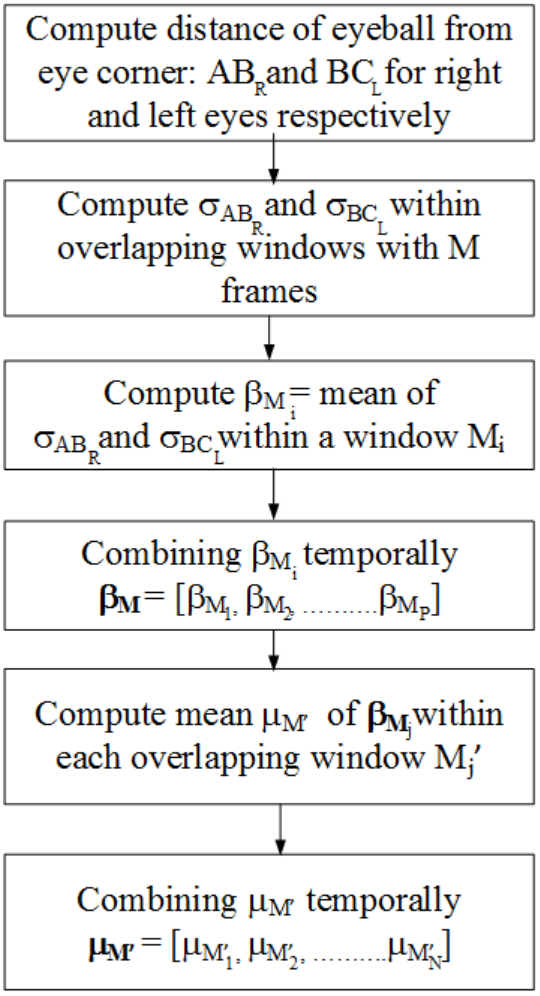


FIGURE 4.24: Steps in the temporal detection of eyeball movement

ping local windows, as shown below:

$$\begin{aligned} \mu_{M'_k} &= mean(\beta_{M_k}) | k = 1 \; to \; L \\ \mu_{\mathbf{M'}} &= [\mu_{M'_1}, \mu_{M'_2}, \ldots, \mu_{M'_i}, \ldots \mu_{M'_P}] \end{aligned} \tag{4.14}$$

The steps in the temporal extraction of eyeball movement starting with the distances AB and BC computed at frame level are summarized in 4.24.

Apart from detecting if the eyeball movement was low or high, we can also detect the direction of sight, based on the eyeball position with respect to the eye corners. If AC is the total eye length and AB_R and BC_L are the distances AB and BC computed for the right and left eyes respectively, we can determine the direction of sight as 'right', 'left'

and ‘center’ (the direction is with respect to the subject) based on the following rules:

$$\begin{aligned} Right \quad if \quad & AB_R \leq AC/3 \ , \ BC_L > 2AC/3 \\ Center \quad if \quad & AC/3 < AB_R \leq 2AC/3, \ AC/3 < BC_L \leq 2AC/3 \\ Left \quad if \quad & AB_R > 2AC/3 \ , BC_L \leq AC/3 \end{aligned} \tag{4.15}$$

4.4 Performance Evaluation

The evaluation of the proposed techniques in this chapter will be discussed as follows: (a) the proposed *WellCam* database for evaluating the wellness indicators is described first, (b) evaluation of techniques for extraction of eye features is presented next, (c) then, the evaluation of techniques to extract eye-related wellness indicators is presented, (d) the section is concluded with a computational complexity analysis of the proposed eye feature extraction techniques.

4.4.1 WellCam Dataset

FIGURE 4.25: Snapshots from the proposed WellCam database

With regards to patient monitoring based on facial analysis, publicly available datasets for the detection of certain specific emotions such as pain exist. However, pain is one of the many indicators of wellness. There is a lack of datasets for the evaluation of

the various facial wellness indicators. Next, procuring such a dataset from hospitals involves administrative and privacy-related challenges.

This motivated us to propose the *WellCam* dataset, which consists of image sequences of ten subjects from different ethnicities simulating the wellness indicators. The subjects were seated in an inclined posture and a Logitech web camera was placed facing the subject. The camera was placed at a distance of 4 feet from the subjects face for 5 subjects, and 3 feet for the other 5 subjects, within a vertical distance of 1.5 feet from the subject's face. Head rotation upto $\pm 15^{\circ}$ was allowed and the images were captured under standard lighting conditions. The video was captured at the rate of 25 fps, with a frame resolution of 720×1280. Snapshots of subjects from WellCam database are shown in Fig. 4.25.

TABLE 4.2: Details of the WellCam dataset listing the various wellness indicators and their average duration in the dataset

Wellness Indicators		*Average duration per subject (No. of frames)*
Reference/Normal state		1500
Eyes closed		750
Eyes open		1500
Eyes partially open		750
Blank stare with eyeball direction	Center	600
	Extreme right	600
	Extreme left	600
Blinking rate	6-8 times/min	750
	30 times/min	750
Restless eyeball movement		1250
Mouth open		375
Head tilt with eyes closed	Left	250
	Right	250
Expression of discomfort with head tilt	Left	250
	Right	250

The subjects were asked to simulate the wellness indicators as follows:

1. The subject was asked to remain normal for the first few minutes.
2. Then, the subject was asked to make blank stares in three different directions (extreme right, left and center), hence simulating a low eyeball movement.
3. The subject was then asked to simulate restless eyeball movement, by looking at different directions, moving the eyeball continuously.
4. Then, the subject was asked to remain as if he was in a drowsy state, simulating partially-open eye state, and some subjects were asked to keep the head slightly lowered or mouth open partially.
5. Then, the subject was asked to enact the feeling of suffering or discomfort, which involved lowering the eyebrows, in some cases shaking the head and opening the mouth partially or widely.

The details of the simulated wellness indicators and the duration for which the indicator is simulated are summarized in Table. 4.2. To the best of our knowledge, this is the first such patient wellness monitoring dataset with the above face related wellness indicators.

4.4.2 Accuracy Evaluation

In order to evaluate the performance of the proposed techniques, they were evaluated on standard databases and compared with existing works. The evaluation metric used will be the same as that used in the existing works. However, for certain wellness indicators and features, due to lack of publicly available databases, the techniques are evaluated on the proposed WellCam database.

4.4.2.1 Eyeball detection

The eyeball detection technique 4.2.2 was evaluated on the BioID database [187]. The size of each image is 384x288. The ground truth of the left and right eye centers (eyeball) is provided along with the BioID database. Percentage accuracy is obtained based

on the normalized error $e = max(d_{left}, d_{right})/w$ as defined in [190], where d_{left} and d_{right} is the Euclidean distance between the detected left and right eye centers and the ones in the ground truth, and w is the Euclidean distance between the eyes in the ground truth. A detection accuracy of 95% was achieved for e = 0.12. The results are comparabe with [132], where a detection accuracy of 96.53% is acheived for an e = 0.1. Sample results of evaluating the eyeball detection technique on the BioID database are shown in Fig. 4.26.

FIGURE 4.26: Sample eyeball detection results on the BioID database, yellow dots indicate the eyeball position detected

4.4.2.2 Eye state detection

The proposed eye state detection technique was evaluated on the proposed "WellCam" database.

A subset consisting of 500 images each of 5 subjects with open, closed and partially-open eye states from the proposed WellCam eye state dataset were manually annotated for eye state. Sample results showing the correct detection of open, closed and partially-open states are shown in Fig. 4.27.

$$\begin{aligned} Recall &= TP/(TP+FN) \\ Precision &= TP/(FP+TP) \end{aligned} \tag{4.16}$$

The recall and precision are computed for each eye state as follows, where TP = true positives, FP = false positives and TN = true negatives. The recall and precision for the dataset are tabulated in Table. 4.3, giving an average recall and precision of 91.3% and 96% respectively. The results show that the proposed method is effective in detecting

the eye states. The eyeball is detected even in nearly closed eye state (refer to Fig. 4.27 (a) and (c)), distinguished from a closed eye state. Sample detections of partially open and open state from images of the WellCam database are shown in Fig. 4.28.

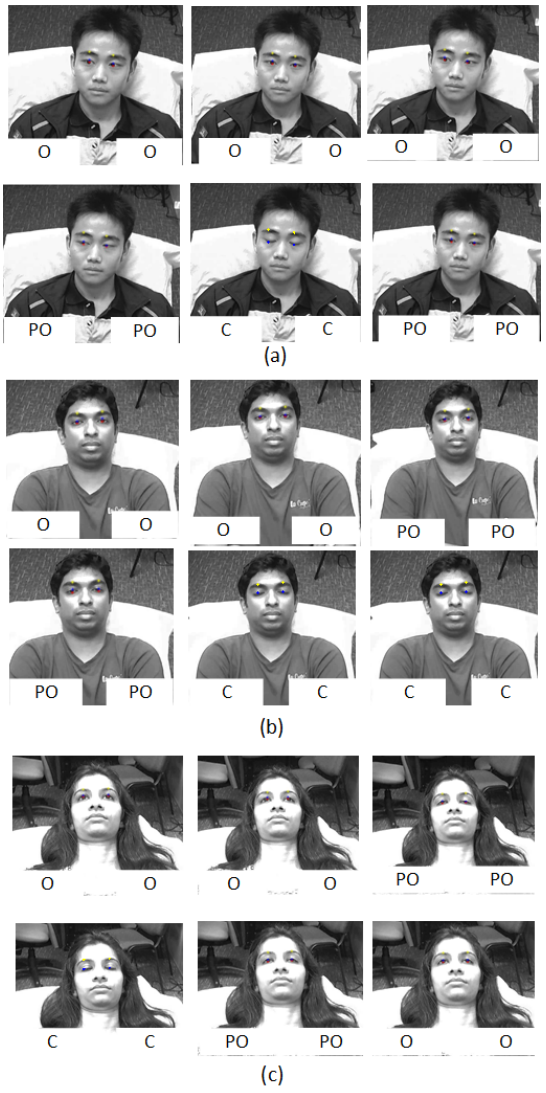


FIGURE 4.27: Sample eye state detection results on the proposed WellCam dataset, 'O', 'C', 'PO' stand or open, closed and partially-open (half-closed) states respectively, which are indicated for the left and right eyes for three subjects (a red dot represents the eyeball detected)

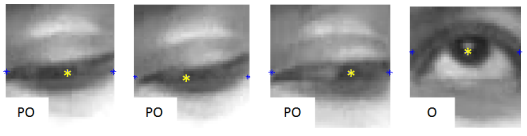


FIGURE 4.28: Sample detections of eyeball in challenging cases of partially open and open state from images of the WellCam database (yellow dots are marked at the position of the detected eyeball)

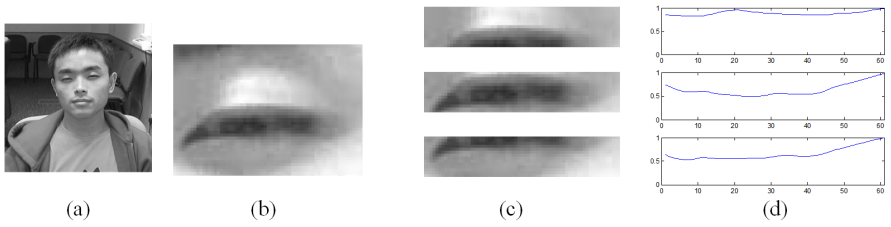
TABLE 4.3: Average recall and percision for open, closed and half-closed states on the proposed WellCam dataset

	Open	Closed	Half-closed
Recall	98%	93%	83%
Precision	98%	99%	91%

Next, the eye state over time was evaluated on a subset of the proposed WellCam dataset consisting of a total of 40,000 images of 6 subjects, as per the steps discussed in 4.3.1. The images were annotated for eye state at window level, i.e., every sequence of images within a window were annotated for the predominant eye state. The size of the window w_1 considered was six times the frame rate 25 fps. The evaluation results for eye state over time have been tabulated in Table. 4.4, where recall and precision are defined by 4.17. The relative lower precision for closed state is contributed by the false detections of partially-open state as closed state for one of the subjects, as shown in Fig. 4.29. The average recall and precision for eye state detection were 98.2% and 97% respectively. .

TABLE 4.4: Average recall and precision for detection of eye state over time on the WellCam dataset

	Open	Closed
Recall	98.4%	98.1%
Precision	99.6%	94%



(a) (b) (c) (d)

FIGURE 4.29: An example of misdetection of eye state, (a) subject (b) eye ROI (c) eye ROI divided into bands and (d) corresponding weighted bandwise accumulation map

The proposed technique was unable to distinguish a partially open state from closed state, such as the case shown in Fig. 4.29.(a). which is difficult to distinguish even by the human eye. The basic features - the peaks, blob variance and eyebrow eyelid distances were of similar values for both partially open and closed states, and hence were not clearly distinguishable by the algorithm as shown in Fig. 4.29.(c).

4.4.2.3 Eye Blink Detection

The proposed blink detection technique was evaluated on three standard databases - ZJU eye blink database [191], Talking face video database [192] and DISFA database [186]. Sample images from ZJU database, Talking face and DISFA database are shown in Figures. 4.30, 4.31 and 4.32 respectively. An example of blink detected in a sequence from the ZJU database is shown in Fig. 4.33.

FIGURE 4.30: Sample images from the ZJU database

FIGURE 4.31: Sample images from the Talking Face video database

The value of the maximum blink duration threshold t_b was set to 10 frames, and a minimum threshold $t_{min} = 3$ was also set in order to filter false positives. In order to calculate the blink detection algorithm performance, the evaluation metrics used in [136] have been employed. If the total number of blinks in the ground truth are P (number of positives), then N (number of negatives) is computed by dividing the number of non-eye blink frames by the average blink duration. Then, the recall, false positive rate FPR, precision and mean accuracy are computed as follows, where TP = true positives,

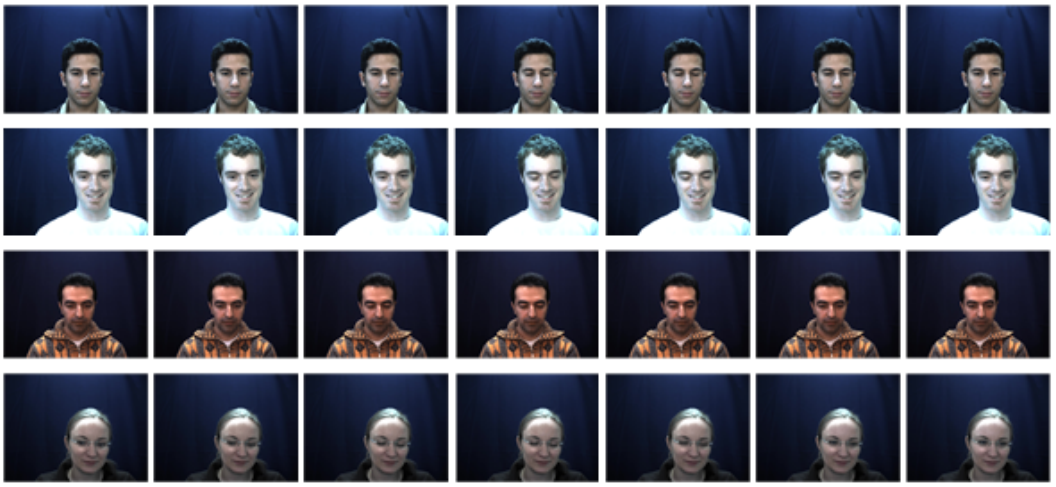


FIGURE 4.32: Sample images from the DISFA database

FP = false positives and TN = true negatives:

$$\begin{aligned} Recall &= TP/(TP+FN) \\ FPR &= FP/N \\ Precision &= TP/(TP+FP) \\ Mean\ accuracy &= (TP+TN)/(P+N) \end{aligned} \tag{4.17}$$

ZJU database has a total of 255 blinks. The total number of images in the database is 10876, with average blink duration of 7 frames, hence, N = 1199. The subset of DISFA database considered for evaluation consisted of sequences of 10 subjects and a total of 48450 frames. The database was manually annotated for a total of 521 blinks. Talking Face database has a total of 5000 images with 61 eye blinks. We computed N to be 479 based on the ground truth of the start and end of blinks and average blink duration of 9 frames.

Table. 4.5 summarizes the evaluation on the three standard databases, and the evaluation on the Wellcam database. An average recall and precision of 94.5% and 91% across the three databases is achieved. The technique is shown to be robust to wide range of skin color, ethinicities and lighting conditions. The technique is able to detect eye blink with a high accuracy even in cases where the subjects are looking slightly downwards, and not directly looking in to the camera in front, as in the case of DISFA database.

TABLE 4.5: Evaluation of blink detection on three standard databases (in percentages)

Dataset	Recall	FPR	Precision
ZJU	96.6	1.4	93.1
Talking face	93.3	1.2	90.3
DISFA	94.5	-	94
WellCam	93.5	-	86.4

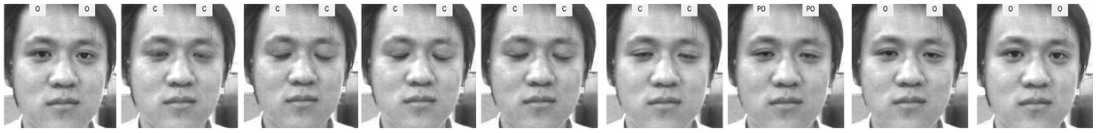


FIGURE 4.33: Example of blink being detected in a sequence from ZJU database, transition from open to closed and closed to open via a partially open state is shown here, t_b was set to 5 given the frame rate of ZJU database was 30 fps

The challenging cases where the proposed method failed to detect a blink are as follows. With respect to the ZJU database - blinks that start off in the beginning of the sequence with the eye in closed state, those where the eye does not fully close during the blink and blinks that are seen for only one frame. High reflection of the glasses also contributed to the challenges faced. In Talking Face database, the challenging cases were contributed by the instances the subject smiles, due to which the eyelids are drawn very close making it appear that the eyes are closed. Sample images of false negatives and misdetections from the Talking Face and ZJU databases, as discussed above are shown in Fig. 4.34. The misdetections in WellCam database were contributed by the proposed method being unable to detect the eyeball in cases such as Fig. 4.29, i.e., when the partially open and closed states were not clearly differentiable, due to smaller eye size.

Further, a comparison of the proposed method on the ZJU and Talking dataset, with [136] and [193] was also performed, and the comparisons are tabulated in Tab. 4.6. The table shows that the proposed method achieves accuracies comparable to the state-of-art.

FIGURE 4.34: (a) Examples of misdetections in Talking Face database - when the subject smiles and (b) ZJU database - reflections due to glasses

TABLE 4.6: Comparison of the proposed method on the ZJU and Talking dataset, with [136] and [193] (in percentages)

	Dataset	Precision	Recall	FP rate	Mean acc.
Divjak & Bischof [193]	ZJU	-	95%	19%	88%
Drutarovsky & Fogelton [136]	ZJU	91%	73.1%	1.58%	93.45%
Proposed method	**ZJU**	**93.1%**	**96.6%**	**1.41%**	**98.26%**
Divjak & Bischof [193]	Talking face	-	95%	2%	97%
Drutarovsky & Fogelton [136]	Talking face	92.2%	96.7%	0.7%	99%
Proposed method	**Talking face**	**90.32%**	**93%**	**0.3%**	**99.39%**

4.4.2.4 Eyeball Movement Analysis

To the best of the author's knowledge, a standard database to evaluate eyeball movement is not available. Hence, the proposed WellCam dataset is used to evaluate the eyeball movement detection. As discussed in Sec. 4.4.1, the WellCam dataset contains sequences of the subjects in their normal state, low eyeball movement simulated through blank stares to the extreme left, right and center, and restless eyeball movement. The eyeball movement was tested on a subset of the proposed WellCam dataset. Fig. 4.35 shows an example detection of high and low eyeball movement in (a) and (b) respectively. The mean variance β_M was plotted across 1250 frames in (a) and 600 frames in (b), with the plot shown for every 10th frame. β_M was computed to be 8.34 and 1.68 respectively for (a) and (b) respectively.

Discussion

The proposed method to analyze eyeball movement relies on the distances AB and BC, along x-axis, and were found to serve the purpose considering changes in yaw up to $\pm 15°$. In the Fig. 4.36, the AB for the right eye is 5,6 and 4 pixels respectively and BC is 25, 26 and 22 respectively. Although BC for (c) has reduced, since AB has also reduced, it shows that the eyeball is pointing to the right. For changes beyond $\pm 15°$, the head pose may need to be additionally considered while analyzing the eyeball movement. The eyeball movement detection technique can be used to classify a patient's eyeball movement as high, normal or low. Inferences over longer periods of time are useful in

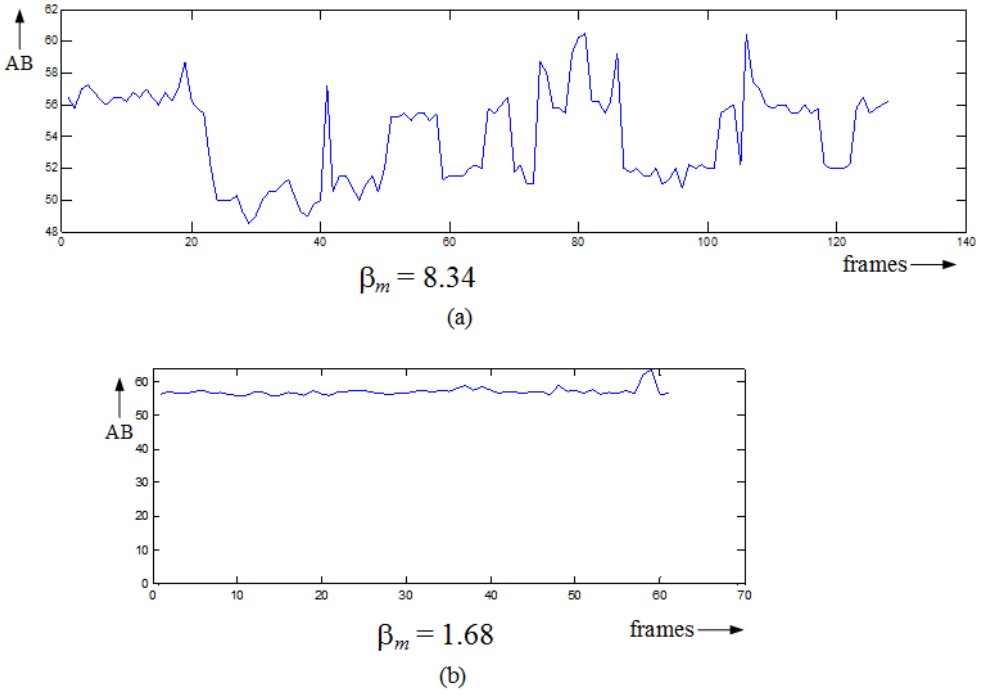


FIGURE 4.35: An example of high and low eyeball movement shown in (a) and (b) respectively, where the σ_{em} is 8.34 and 1.68 respectively.

FIGURE 4.36: Images to show the distances AB and BC for slight changes in head pose

analyzing the activity level of the patient and hence important in determining the patient state.

4.4.3 Computational Complexity Analysis

In this section, we present an analysis of the computational complexity of the proposed method for the eye feature extraction. The computational cost for eye state detection, which also involves the eyeball detection is computed and compared with the method

in [132]. In order to avoid any discrepancies in the comparison due to implementation platform, the computational complexity is presented in terms of number of basic operations that are incurred in the two methods. Although the analysis presented in this manner does not fully measure the computational cost till the gate level, it will be shown that it however gives an acceptable estimate of the orders of computational cost reduction that is possible using the proposed method.

TABLE 4.7: Computation of operations in the proposed method

Operations	Additions/Subtractions	Multiplications /Divisions	Comparisons
Band-wise weighted intensity accumulation maps	$nN_b(h_b - 1)$	-	$n(N_b - 1)$
G_y map generation and gradient accumulation map	$5bN_b + nN_b(h_b - 1)$	-	bN_b
Peak-valley analysis	$n(4(N_b - 1) + n_v) + n_v({}^{n_p}C_2 + n_p)$ $+{}^{n_p}C_2 + (A_{p_1} + A_{p_2} + A_v)^{n_p}C_2 n_v$ $+(A_{p_1} + A_v)n_p n_v$	$2nn_p + 5({}^{n_p}C_2 \times n_v)$ $+3(n_p \times n_v)$	$3n(N_b-1)+2n(n_p-1) + 4nn_p$ $+2(2n_v-1)+2(n-1) + 3n_{pvp}$
Blob analysis	$3B_l - 2$	$B_l + 1$	-

We will now summarize the number of operations in the proposed method, wherein all the operations are converted to additions (subtractions), multiplications, divisions and comparisons. We first compute the cost for detecting the state of a single eye, which is eventually multiplied by 2 to consider both the eyes.

The computational cost for a single eye is summarized in Table. 4.7. The computation cost is computed for an $N_b \times b$ sized ROI around the eye. The number of overlapping bands is denoted by n. In the peak valley analysis section of the proposed method, n_p and n_v denote the average number of peaks and valleys in each band. Also, A_{p_1}, A_{p_2} and A_v are the areas under the two peaks and valley that constitute the PVP candidate (as described in Section 4.2.2). The variables that are used for the blob analysis (Section 4.2.3) are denoted by the following: B_l is the blob width.

Similarly, in Table. 4.8, we formulate the number of operations for the method in [132]. In order to show the ascendancy of the proposed method in terms of computational complexity, we consider only the first part of Valenti et al.'s method in [132]. We compute the number of operations for the isocenter candidates extraction step only, and no cost (or zero cost) is added for the the mean shift algorithm and machine learning step in [132]. Also, for the sake of comparison, the same RoIs are taken for both methods.

TABLE 4.8: Computation of operations in the method proposed in [132] (upto the isocenter candidate extraction step)

Operations	Adds/Subs	Muls/Divs	Comp.
Gradient magnitude	$10bN_b$	-	-
Second derivative	$15bN_b$	-	-
Curvedness	$2bN_b$	$4bN_b$	-
Displacement vectors	$4bN_b$	$14bN_b$	-
Shortlisting max. isocenter candidates	-	-	$0.1bN_b$

TABLE 4.9: Comparison of number of computations between proposed method and [132]

Operations	Proposed Method	Method proposed in [132]
Additions	52992	226800
Multiplications	22292	57600
Comparisons	16926	720
Total operations	92210	285120
% Savings	67.65%	-

In order to see the computational cost savings in the proposed method, we consider the worst case scenario and assign the following values to the variables: N_b = 60, b = 30, n = 5, $n_p = n_v$ = 2, n_{pvp} = 4, B_l = 40, $A_{p_1} = A_v$ = 100, A_{p_2} = 50. In the case of [132], as mentioned earlier, we consider a part of the method only, excluding a computationally complex mean-shift algorithm and machine learning block. The number of operations are compared in Table. 4.9. It can be seen that the proposed method when compared to only a part of the method in [132], achieves 67% savings in the total operations.

As mentioned earlier, although this is not an accurate computational complexity measure (such as gate level analysis), the total operations show the amount of savings the proposed method can achieve as compared to [132].

4.5 Summary

In this chapter, techniques for extracting the eye features namely, eye state and eyeball location have been proposed, which were then used to extract the eye related wellness indicators for patient wellness monitoring. The proposed methods leveraged spatial properties of the human eye to achieve a selective processing of pixels, and also uses simple accumulations as the underlying operations. The technique for eyeball detection was shown to give an accuracy of 95%. The frame-level extraction of eye state resulted in a recall and precision of 91.3% and 96% respectively. These techniques were extended through temporal analysis to extract the wellness indicators such as eye state over time, blink rate and duration, and eyeball movement. The temporal extraction of eye state resulted in a recall and precision of 98.2% and 97% respectively. The blink detection mean accuracy of 98.8% was found to be comparable to the state-of-art when evaluated on standard databases. The blink detection was also evaluated on the DISFA database, with manual annotations done on a subset of the database. The WellCam dataset was proposed, which consists of image sequences of subjects simulating the various wellness indicators. The proposed techniques for eye state over time and eyeball movement were evaluated on WellCam database against the manual annotations done on this dataset. To demonstrate the computational efficiency of the proposed techniques, the computational cost of the eye state detection technique was computed and was shown to achieve 67% savings compared to a state of art method. In the next chapter, techniques to extract the wellness indicators related to mouth and brow furrows are proposed.

CHAPTER 5

Wellness Indicators from Mouth and Brow Furrows

5.1 Introduction

In the previous chapter, techniques for extraction of wellness indicators from the eyes were proposed. In this chapter, the extraction of wellness indicators from other facial features, such as mouth and brow furrows are proposed. Wellness indicators extracted from these features complement the indicators extracted from eyes, eventually enabling a more holistic extraction of wellness indicators from the face. For instance, if the eyes are open, and the mouth is also kept open - this is a possible sign of unwellness. Then, eye state alone will be incomplete as a wellness indicator unless it is combined with the mouth state. The wellness indicators are extracted based on the temporal analysis of the frame-level detection of states of these features. Examples to illustrate mouth state and brow furrows as wellness indicators are shown in Fig. 5.1.

Mouth state: The wellness indicators of interest in this study are - mouth kept open, yawning and talking; and these can be extracted based on the temporal analysis of mouth state over time. Most of the methods proposed for mouth state detection are aimed at yawning detection, and are based on color detection of the lips [145], [146], or edge detection [145], [147], [145], or lip corners [148], or lip contour [149] or the

FIGURE 5.1: Mouth state and brow furrows as wellness indicators

intensity changes [150]. As seen from Sec. 2.6, most of these methods involve complex computations. This can become a bottleneck in realization of the algorithms on an embedded platform, where resource constraints have to be met.

This motivates the need for further reduction in computational cost in mouth state detection. In this chapter, a computationally-efficient mouth state detection technique is proposed. The method is based on accumulation map and mean intensity profile of grayscale intensities of the mouth ROI. The temporal analysis of mouth state to extract wellness indicators is then presented.

Brow furrows: Brow furrows are *transient features* that are caused by the facial action of brow lowering [161]. They appear due to the movement of the muscle "corrugator supercilii", which draws the eyebrows together. This muscle is regarded as the principle muscle in the expression of suffering, and hence the brow furrows are often indicative of such a state.

In the Facial Action Recognition System (FACS) [86], each movement of a muscle or groups of muscles on the face are taxonomized, and each such action is called an action unit (AU). The brow lowering action is named as action unit 4 (AU 4). As seen from Sec. 2.6, most of the existing methods for detecting states of suffering such as pain involve detecting the eyebrow lowering action that cause the brow furrows. Such methods are mostly involve complex feature extractions. In this chapter, a computationally efficient method to extract brow furrows as appearance-based features is proposed.

The proposed method of extracting the mouth state and wellness indicators based on mouth state is presented first, followed by the method to extract the wellness indicators from brow furrows.

5.2 Frame-level Detection of Mouth State

The method is based on detecting the transitions between skin and lip, since they remain relatively consistent across the changes in mouth state. So, the upper and lower lips are detected based on the intensity changes across the skin to the upper lip and the lower lip to skin respectively. The detected lower and upper lips and the region between the lips are used to detect mouth state as - open or closed.

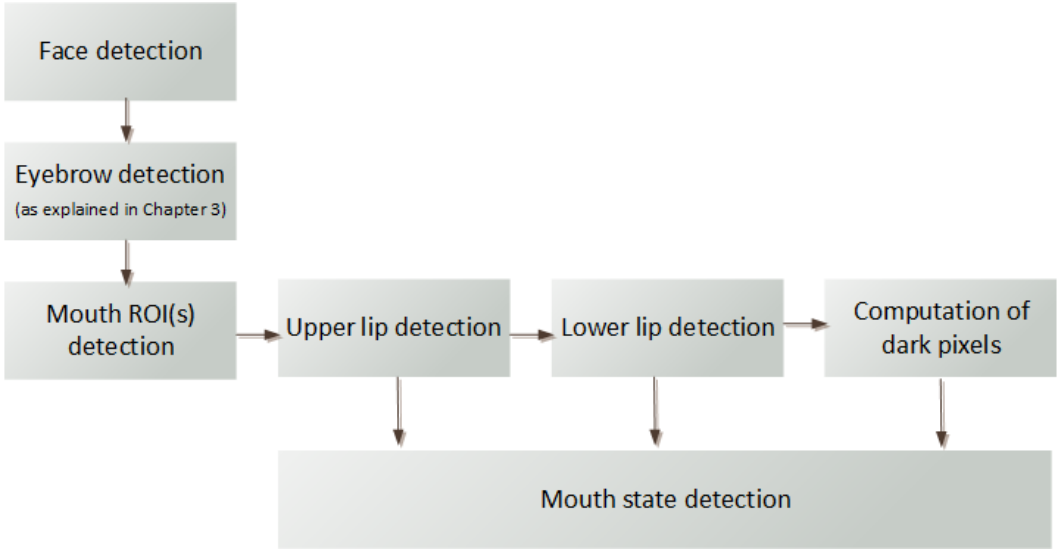


FIGURE 5.2: Overview of the proposed mouth state detection method

The overview of the proposed method for mouth state detection is shown in Fig. 5.2. The method comprises of the following steps: (a) localization of mouth ROI(s), followed by the detection of the mouth feature points: (b) upper lip and (c) lower lip, (d) mouth state detection.

5.2.1 Localization of Mouth ROI

Three ROIs are used to detect the upper lip and the lower lip and eventually the mouth state. In order to extract the ROIs for mouth state detection, the eyebrows and nostril positions are used as reference. We use the observation that the mouth occurs beneath the nostrils, and the x-coordinates of the eyebrow's inner ends occur within the bounds of the mouth, if yaw and roll of the head rotation of $\pm 15^{\circ}$ is considered.

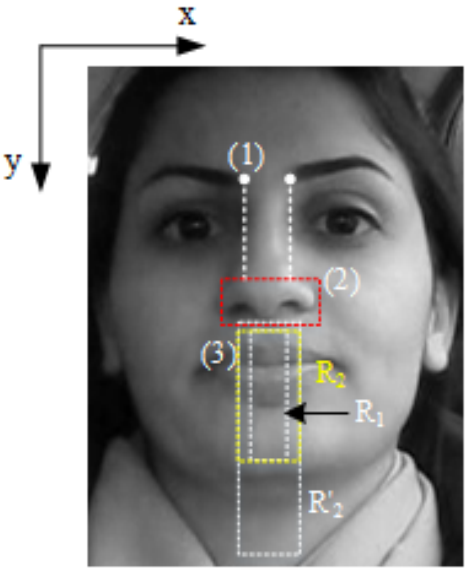


FIGURE 5.3: Steps in Mouth ROI extraction (1) Locating eyebrow inner ends (2) Detecting nostril region and nostril position (3) Estimating the mouth ROIs R_1, R_2 and R_2'

The three main steps in the mouth ROI(s) detection as shown in Fig. 5.3 are:

1. Locating the inner ends of the eyebrows (x-coordinates)
2. Detecting the position of nostrils (y-coordinates)
3. Estimating the mouth ROI(s) R_1, R_2 and R_2' under the nostril region.

(a) Locating the inner ends of eyebrows

To begin with, the eyebrow positions are detected using the method proposed in Chapter 3. Then, the eyebrow inner ends are detected as follows:

The eyebrow inner ends are extracted based on the observation that if we were to consider a window W_f across the eyebrows (as shown in Fig. 5.4.(a)), the intensity within W_f is such that it is relatively higher in the region between the eyebrows compared to the regions where eyebrows are present. We start with the position of eyebrows detected based on the technique proposed in Chapter. 3. A rectangular window F centered at the y-coordinate of the eyebrow position and extending from 50% of one eyebrow to 50% of the other eyebrow along the x-axis as shown in Fig. 5.4.(b). The intensities along every vertical column of pixels along the x-axis in F are accumulated resulting in the

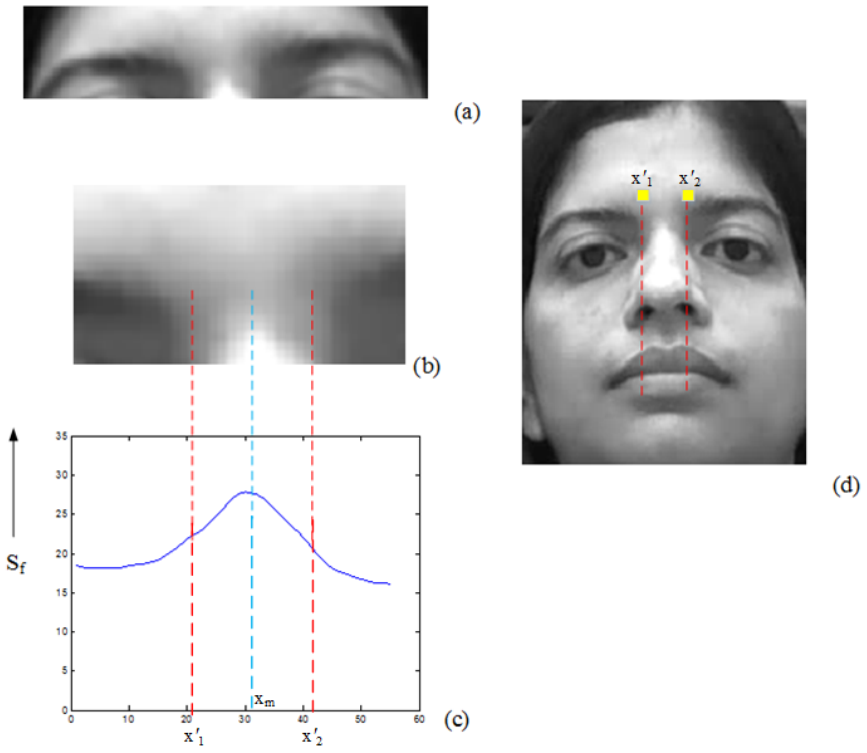


FIGURE 5.4: (a) Eyebrow region (b) window W_f extending across the eyebrow region (c) accumulation map of pixels S_f (d) inner ends of the eyebrows x'_1 and x'_2

accumulation map S_f (shown in Fig. 5.4.(c)). If w_f and h_f are the width and height of W_f, then S_f is given by:

$$\begin{aligned} \mathbf{S}_f &= [S_{f_0}, S_{f_1}, \ldots, S_{f_i}, \ldots, S_{f_{w_f}}] \\ \mathbf{S}_f &= [\sum_{k=1}^{h_f} F_{(k,0)}, \ldots, \sum_{k=1}^{h_f} F_{(k,i)}, \ldots, \sum_{k=1}^{h} F_{(k,w)}] \end{aligned} \tag{5.1}$$

We look for the point x_m in $\mathbf{S_f}$ along x-axis towards the center of $\mathbf{S_f}$ with the highest intensity accumulation value - in other words, a peak. In the event that a single distinct peak is not found but rather, a set of peaks are found, then the mid point x_m of the set of peaks across the x-axis is considered. The approximate x-coordinates of the eyebrow inner ends are computed based on anthropometric measures as:

$$\begin{aligned} x'_1 &= x_m - 0.083 \times w \\ x'_2 &= x_m + 0.083 \times w \end{aligned} \tag{5.2}$$

where w is the width of the face.

(b) Locating the position of nostrils

Next, the approximate y-coordinate of the nostril position is located. The ROI for locating nostrils is estimated using the y-coordinates of the eyebrows along with anthropometric measures. The nostril region is approximately located at around $2/3$ times the face height as shown in Fig. 5.5.(a). Binarization (Fig. 5.5.(b))followed by median filtering is applied to extract the nostril blobs. In the event the nostril blobs are not detected, the threshold for binarization is reduced marginally and the ROI is binarized again for nostril detection. The mean of the y-coordinates of the two nostrils is taken as the approximate y-coordinate of the nostrils y_N.

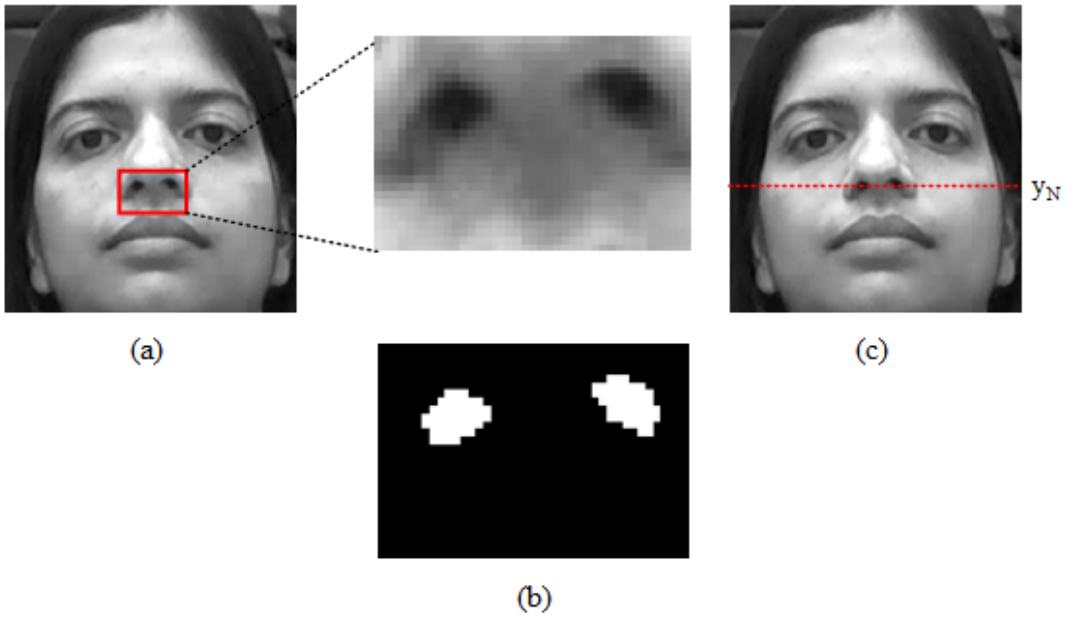


FIGURE 5.5: (a) ROI for detecting nostrils (b) binarization of the ROI (c) y-coordinate of nostril y_N

(c) Estimating the mouth ROI(s)

Once the approximate y-coordinate of the nostrils y_N is detected, the mouth ROIs R_1, R_2 and R_2' under the nostrils (as shown in Fig. 5.3) are generated. The ROI R_1 is a thin strip under the nostrils. R_2 is wider than R_1. ROI R_2' is an elongated version of R_2, will be used in the detection of lower lip. This is to ensure the lower lip is captured in the ROI even when the mouth is widely opened as in the case of yawning.

5.2.2 Detection of Mouth Feature Points

Once the ROIs have been extracted, the mouth feature points - namely, the upper and lower lip, are detected. The sequence of steps in upper and lower lip detection are summarized in figure Fig.5.8. The detection of the upper lip position is explained first, followed by the detection of the lower lip position.

5.2.2.1 Detection of Upper Lip

Upper lip detection involves detecting the transition to the upper lip from the skin under the nostril region, which is characterized by a change in intensity from a relatively lighter to darker intensity. In order to extract this transition, accumulation map $\mathbf{M_1}$ of the intensities in the mouth ROIs R_1 is generated. The accumulation map $\mathbf{M_1}$ is obtained by the summation of the intensities of pixels in all columns along each row i of pixels, (i $\in [1 \quad h_{R_1}]$). The computations in the generation of accumulation map are summarized in equation 5.3, where w_{R_1} is the width of the mouth ROI and h_{R_1} is the height of R_1.

$$\mathbf{M_1} = [\sum_{k=1}^{w_{R_1}} I_{(k,1)}, \ldots, \sum_{k=1}^{w_{R_1}} I_{(k,i)}, \ldots, \sum_{k=1}^{w_{R_1}} I_{(k,h_{R_1})}] \tag{5.3}$$

As shown in Fig. 5.6.(b), in $\mathbf{M_1}$, a valley follows a peak at the point of transition to upper lip from skin. Hence, the algorithm looks for a valley v following a peak p_1 starting from the top of $\mathbf{M_1}$. The point of transition from p_1 to v, in other words the point in between them, denotes the upper lip position p_{UL}. In the event that R_1 does not capture the transition into mouth from the skin above as a valley following a peak, the second mouth ROI R_2 is used. Then, the accumulation map $\mathbf{M_2}$ for the ROI R_2 is computed and analyzed for the presence of peak-valley in a similar manner as explained for $\mathbf{M_1}$.

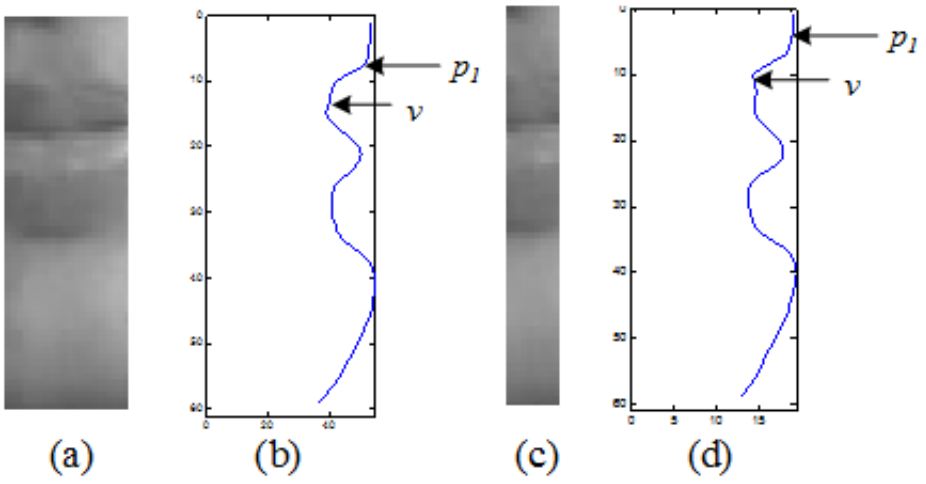


FIGURE 5.6: (a) Mouth ROI R_1 (b) Intensity accumulation map of R_1 (c) Mouth ROI R_2 (d) Intensity accumulation map of R_2

5.2.2.2 Detection of lower lip using Mean Intensity Profile

In order to detect the lower lip, the following observation is used - the region of the face starting from the lower lip down to the chin remains relatively unchanging in appearance compared to the region between the lips across changes in mouth state. The steps in the extraction of the lower lip profile are explained as follows, starting with the processing done on the reference image, followed by that on the incoming images.

(a) Reference image: First, consider the reference image of a subject, where mouth is in *closed* state. The upper lip position is extracted as explained in Sec. 5.2.2.1. Since the reference image is in closed state, the lower lip immediately follows the upper lip. In closed state, the transition from the mid region of the mouth to the lower lip appears as a relative increase in intensity, and is extracted as a peak in $\mathbf{M_1}$ that occurs immediately after the position of v. Let this peak be called p_2. Then, the intensity transition from the lower lip until the chin is captured using a *mean intensity profile*. Referring to Fig. 5.7(a) and (b), the signature or intensity profile $\mathbf{P_{ref}}$ of the region R_{ref} is captured, which will then be used to identify the lower lip in the incoming images.

So, the bandwise mean intensity profile $\mathbf{P}$ of the reference image (mouth in closed state) is extracted as follows:

1. The ROI R'_2 is divided into overlapping bands of size b, starting from p_2 (as shown in Fig. 5.7).(b). The mean of the intensities within each band is computed. The array of the mean intensities of the overlapping bands forms the mean intensity profile $\mathbf{P}$:

$$\mathbf{P} = [P_1, P_2, \ldots, P_j, \ldots, P_n] \tag{5.4}$$

   where P_j is the band-wise mean intensity for the band j.

2. Then, the mean intensity profile within a window W_{ref} combining h_P bands starting at the lower lip position p_2 is saved as reference, which will be called as P_{ref}. The corresponding region in R'_2 is called R_{ref}. P_{ref} is normalized with respect to its highest value (each value in P_{ref} is divided by the highest value in P_{ref}).

$$P_{ref} = \mathbf{P}(p_2 : p_2 + h_P) \tag{5.5}$$

where h_P was computed based on anthropometric measurements, and set to 5 times the thickness of the lip in this work.

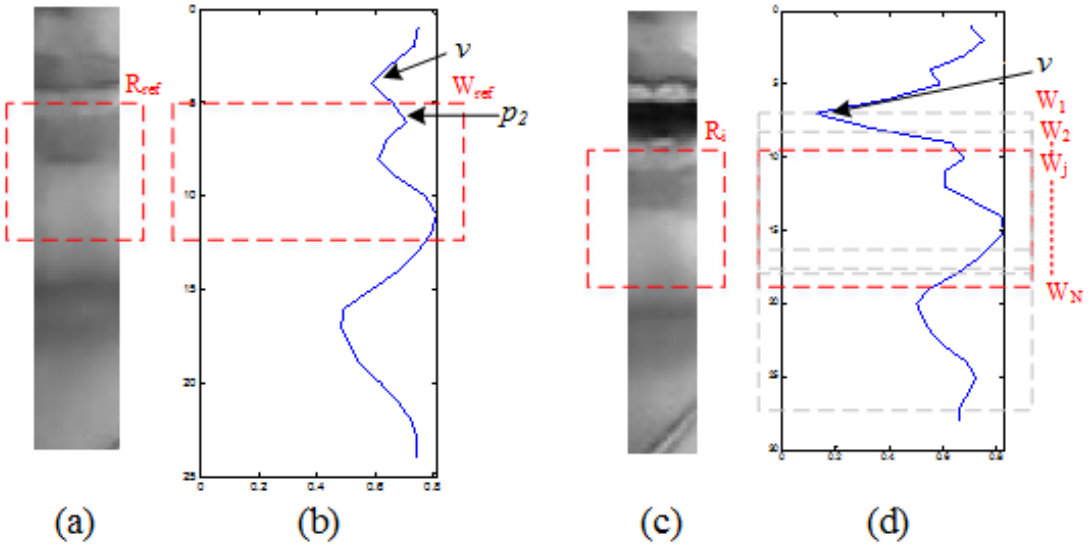


FIGURE 5.7: (a) Region R_{ref} from lower lip and skin underneath in R'_2 of the reference image (b) Mean intensity profile of R_{ref} shown within W_{ref} (c) Region R_j from lower lip and skin underneath in an incoming image (d) Mean intensity profile of R_j shown within W_j

(b) Incoming image: Now, consider an incoming image of the same subject. First, the image is intensity-normalized with respect to the reference image. Following the ROIs extraction, the accumulation map $\mathbf{M_1}$ is computed and the position of v is detected as explained in 5.2.2.1.

Then, R'_2 of the image is considered and the band-wise mean intensity profile $\mathbf{P}$ is computed in a manner similar to that explained for the reference image. Let P_{W_j} be $\mathbf{P}$ within a window combining h_P bands. As shown in Fig. 5.7.(d), P_{W_j} is extracted for every position j starting from v to $h_{R'_2} - h_P$, where $h_{R'_2}$ is the height of R'_2. P_{W_j} is then normalized with respect to its highest value. Then, the similarity of each such window R_j with R_{ref} is measured by computing the Euclidean distance $\|\mathbf{d_j}\|$ between P_{ref} and P_{W_j} as shown below:

$$\begin{aligned} \|\mathbf{d_j}\| &= d(P_{W_{ref}}, P_{W_j}), \forall j \in [v \;\; h_{R'_2} - h_P] \\ \|\mathbf{d_j}\| &= \sum_{i=1}^{h_P} (P_{ref}(i) - P_{W_j}(i))^2 \end{aligned} \tag{5.6}$$

where $P_{W_{ref}}(i)$ and $P_{W_j}(i)$ are the mean intensity in the i^{th} band, and i ranges from 1 to w_P. The window W_j which has the least Euclidean distance is taken as the best match W_M. Then, the y co-ordinate j_{W_M} of the top-most band of W_M is taken as the lower lip position p_{LL}.

$$p_{LL} = min(\|\mathbf{d_j}\|) \tag{5.7}$$

The distance between the upper and lower lips s is computed. It is also termed as degree of mouth openness and is used for mouth state detection as will be explained next.

$$s = p_{LL} - p_{UL} \tag{5.8}$$

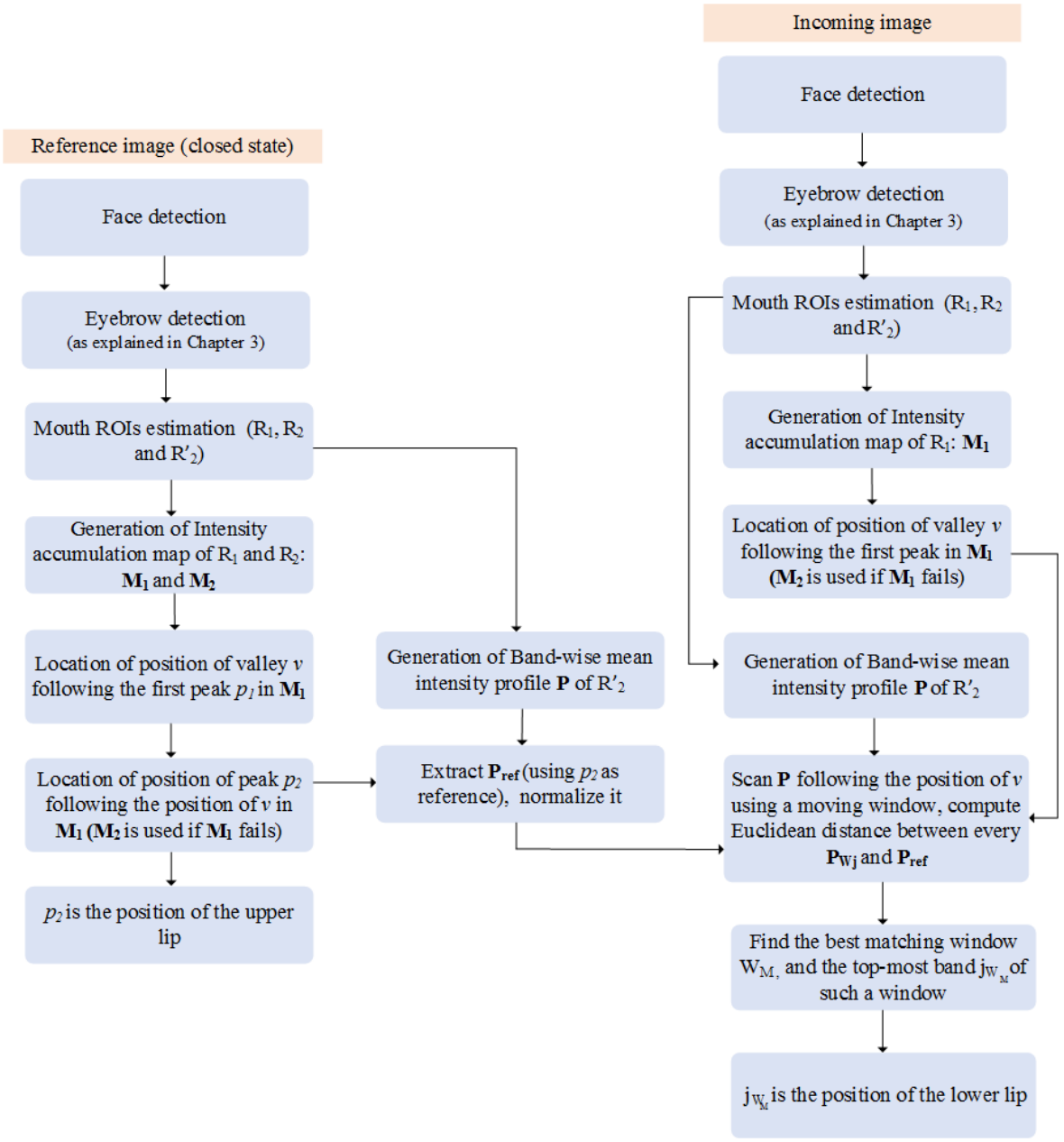


FIGURE 5.8: Flow of the algorithm for detection of upper and lower lips

5.2.3 Mouth State Detection

In order to extract the wellness indicators for patient monitoring, we would like to detect mouth in closed state and open state (shown in Fig. 5.9(a) and (c) respectively). The closed state for a subject is taken as reference while detecting the mouth state, and in this work, the open state is defined by that caused by the *lips parting* action. Other than the two states of the mouth namely, closed and open, as defined above, we have cases where the lips are drawn apart, and only teeth are seen, but the lumen is not seen, for

example in the case of smiling with teeth shown. Such cases are classified under *other* state. When the mouth is in closed state, the lips are drawn together and when in open

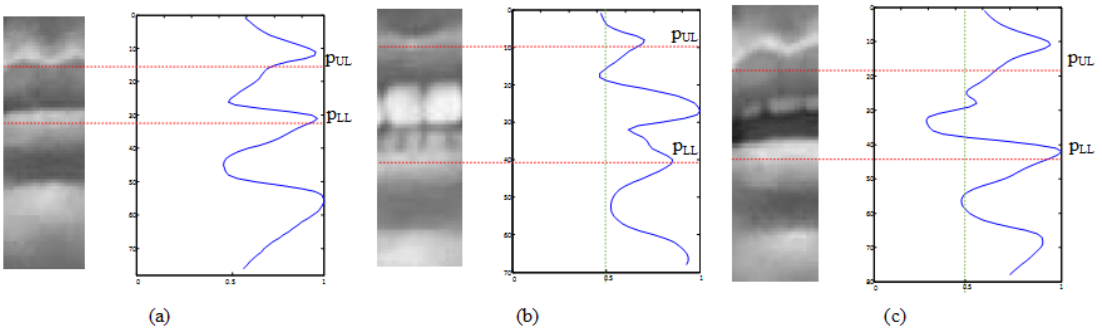


FIGURE 5.9: R_2 and the corresponding normalized accumulation map $\mathbf{M_2}$ of reference image (a), and incoming images (b) and (c). The dotted vertical line in (b) and (c) denotes the threshold set at 0.5

state, the lips are drawn apart. Also, since the lumen inside the mouth that is seen when the mouth is open, the region between the lips will appear relatively darker.

So, in order to classify mouth state, three parameters are used:

- the distance between the lips s, which is computed based on (5.8) and
- the amount of dark pixels present between the lips r.
- the presence of at least one peak in $\mathbf{M_2}$ between p_{UL} and p_{LL}

The amount of dark pixels between the lips r is quantified as follows: Consider the reference image, which is that of the closed state. Once the upper and lower lip positions are detected, the accumulation map $\mathbf{M_2}$ is used. Each value in $\mathbf{M_2}$ is the accumulated value of the intensity in the corresponding row of pixels. $\mathbf{M_2}$ is first normalized with respect to its highest value. Then, the number of rows r_C in $\mathbf{M_2}$ between p_{UL} and p_{LL} with the normalized accumulated intensity value less than 0.5 is computed. Next, consider an incoming image of the same subject. r - the number of bands in the normalized $\mathbf{M_2}$ between p_{UL} and p_{LL} with value less than 0.5 is computed.

Having extracted s and r for an incoming image, the mouth state is detected as follows. Let s_C be the distance between the lips in closed state, and r_C be the number of rows of dark pixels between the upper and lower lips in closed state. Then, thresholds s_O and

r_O are set to values greater than s_C and r_C respectively. If r exceeds a threshold r_O, then the mouth state is classified to be *open*, i.e. $r \geq r_O$. Else, the distance between the lips s is compared with threshold s_O and checked if $s \geq s_O$. Then, $\mathbf{M_2}$ is examined for the presence of a peak between p_{UL} and p_{LL}, which represents the presence of teeth. This is because, the intensity of teeth are relatively higher, and hence they will appear as a peak between p_{UL} and p_{LL} in $\mathbf{M_2}$ as shown in Fig. 5.9.(b). If these two conditions are not satisfied, the mouth state is classified as *closed*.

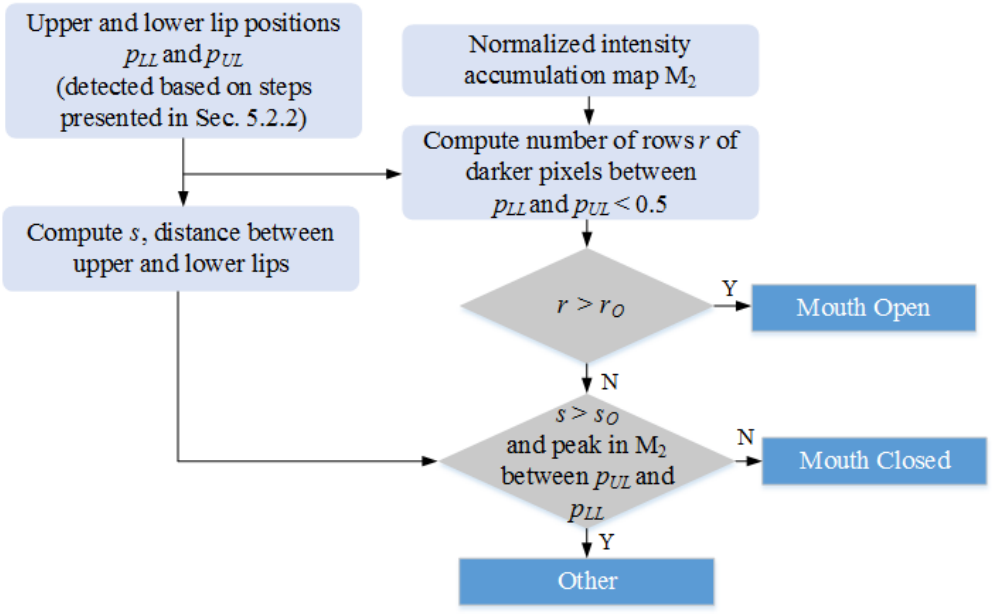


FIGURE 5.10: Flow of the algorithm for detection of mouth state

Fig. 5.9 provides an illustration of the three mouth states. Fig. 5.9.(a) represents closed state, Fig. 5.9.(c) represents open state and Fig. 5.9.(b) represents other state. The number of rows in $\mathbf{M_2}$ between the upper and lower lips that are less than 0.5, r will exceed the threshold r_O in Fig. 5.9(c), while in Fig. 5.9(b), r will not exceed r_O.

Fig. 5.10 summarizes the steps in mouth state detection after having detected the upper and lower lip positions. The temporal analysis of mouth state to extract wellness indicators is presented in the next section.

5.3 Temporal Analysis to Extract Wellness Indicators from Mouth State

In this research, the focus is on the following wellness indicators related to the mouth state - mouth being kept open, yawning and talking. These wellness indicators are extracted based on the temporal analysis of the mouth state. The distance between the lips s and amount of dark pixels r are analyzed temporally to extract the wellness indicators.

In this process, the parameters s and r are initialized using closed state as reference. So, the initial few images are required to be that of the subject with mouth kept closed, and the mean of the parameters across these images is computed to get s_C and r_C respectively.

Then, for every incoming image, the distance between the upper and lower lips s and the amount of dark pixels r are computed as explained in Sec. 5.2. The mean and variance of both s and r are computed across overlapping windows of the incoming image sequence: μ_s, σ_s^2, μ_r and σ_r^2 respectively. They are then used to extract the wellness indicators as described below.

(a) Yawning

The occurrence of yawning is detected as follows: If the distance between the lips s increases above a threshold s_Y for f_C frames and resumes closed state such that $T_Y^{min} \leq f_C \leq T_Y^{max}$, yawning has been detected. An illustration of this is provided in Fig. 5.11. Alternatively, if the mean of the number of dark pixels between the upper and lower lips r exceeds a threshold r_Y for f_C frames, an event of yawning is detected. A state machine for yawning detection is shown in Fig. 5.12.

$$r > r_Y \text{ or } s > s_Y, \; and \; T_Y^{min} \leq f_C \leq T_Y^{max} \tag{5.9}$$

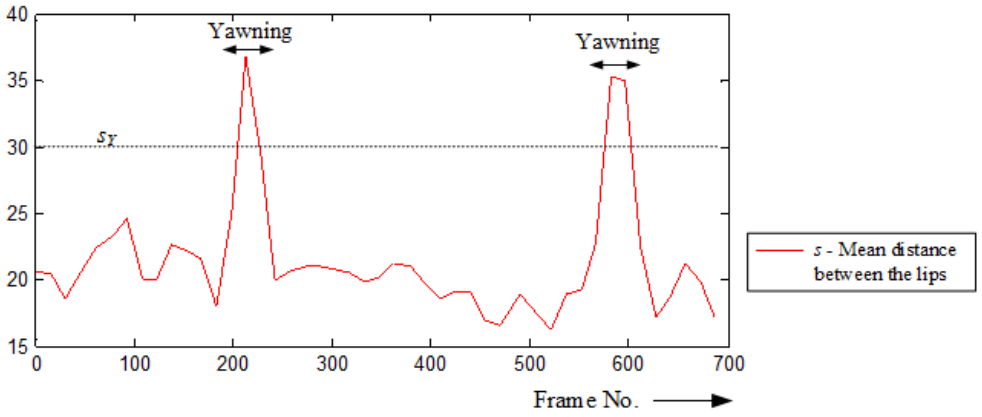


FIGURE 5.11: An illustration of temporal analysis of the mouth state to extract occurrences of yawning

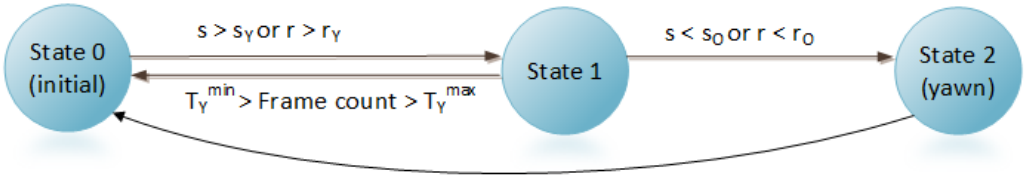


FIGURE 5.12: The temporal analysis of the mouth state to extract occurrence of yawning

(b) Talking

In order to detect the event of talking, the variance and mean of either the dark pixels or the distance between the upper and lower lip positions are used. μ_s, σ_s^2, μ_r and σ_r^2 are computed within overlapping windows across time.

If the variance σ_r^2 increases above the threshold $\sigma_{r_O}^2$, while the mean μ_r is greater than the threshold r_O consecutively over a period of T_A frames, then it is inferred that an event of talking has been detected.

$$\mu_r > r_O \text{ and } \sigma_r^2 > \sigma_{r_O}^2 \; over \; T_A \; frames \tag{5.10}$$

Alternatively, if the variance σ_s^2 increases above the threshold $\sigma_{s_O}^2$, while the mean μ_s is such that it is greater than the threshold s_O consecutively over T_A frames, it is inferred that an event of talking has been detected.

$$\mu_s > s_O \text{ and } \sigma_s^2 > \sigma_{s_O}^2 \; over \; T_A \; frames \tag{5.11}$$

(c) Mouth kept open

In order to detect the event of the mouth kept open, μ_r and σ_r^2 are computed within overlapping windows. Then, if the mean of the number of dark pixels μ_r is high (above a threshold r_O), and the variance of the number of dark pixels σ_r^2 is low (less than a threshold $\sigma_{r_O}^2$) consecutively across T_O frames, it is inferred that the event of *mouth kept open* is detected.

$$\mu_r > r_O \text{ and } \sigma_r^2 < \sigma_{r_O}^2 \; over \; T_O \; frames \tag{5.12}$$

In the experiments conducted, $\sigma_{s_O}^2$ and $\sigma_{r_O}^2$ were set to 1, and r_O and s_O were set to values greater than r_C and s_C respectively.

TABLE 5.1: Parameters used to extract the different mouth-based wellness indicators

	μ_s	σ_s^2	Duration	μ_r	σ_r^2
Mouth open	-	-	$f_C > T_O$	$\mu_r > r_O$	$\sigma_r^2 < \sigma_{r_O}^2$
Yawning	$s > s_Y$	-	$T_Y^{min} < f_C < T_Y^{max}$	$r > r_Y$	-
talking	$\mu_s > s_O$	$\sigma_s^2 > \sigma_{s_O}^2$	$f_C > T_A$	$\mu_r > r_O$	$\sigma_r^2 > \sigma_{r_O}^2$

The conditions that are checked for the parameters μ_s, σ_s^2, μ_r and σ_r^2, based on which the mouth based wellness indicators are detected, are summarized in Table. 5.1.

5.4 Performance Evaluation for Mouth related Wellness Indicators

In this section, the evaluation of the mouth feature points, i.e., the upper and lower lips is presented first, followed by the evaluation of mouth state detection. Next, the evaluation of mouth-related wellness indicators is presented. The section is concluded with a discussion on the computational complexity of the proposed mouth state detection technique.

5.4.1 Accuracy Evaluation

5.4.1.1 Mouth Feature Point Detection

The proposed mouth state detection technique was evaluated on two standard databases - Cohn Kanade facial expression (CK) database [182] and KDEF database [185]. A subset of 184 images of 40 subjects from the Cohn Kanade facial expression database and 137 images of 30 subjects from the KDEF database were considered for evaluation. The subset was chosen to specifically include cases with mouth in closed, partially open and fully open (as in the case of yawning) states, and required manual annotation.

FIGURE 5.13: Upper and lower lips detected in the Cohn Kanade database, denoted by the white dots

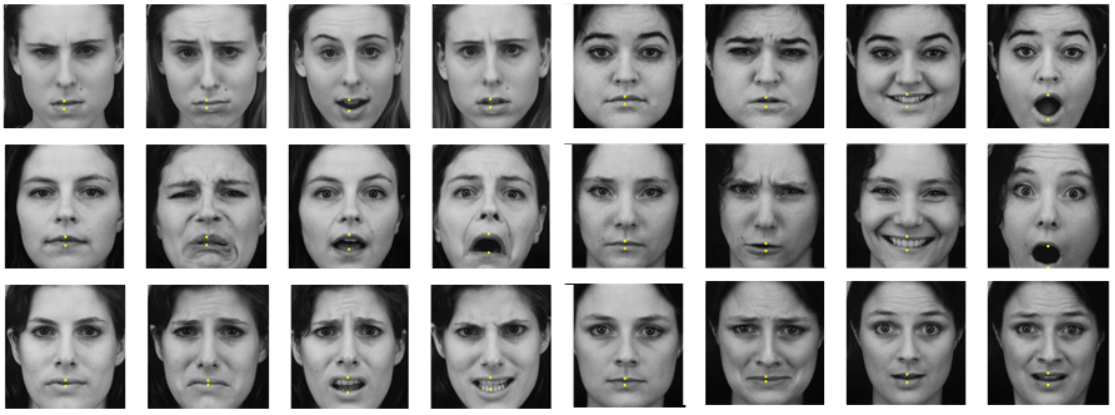


FIGURE 5.14: Upper and lower lips detected in the DISFA database, denoted by the white dots

(a) Upper and lower lip detection

The images were manually annotated for the upper and lower lip positions. The evaluation results of the upper and lower lip detection are summarized in Table. 5.2. The detection accuracy of the upper and lower lips within a tolerance of 10 pixels for Cohn-Kanade database and 15 pixels for KDEF database with respect to the ground truth is computed, given that the average size of face is 277×277 and 380×380 pixels and the average lip thickness is 13.2 pixels and 17.2 pixels for images in CK and KDEF databases respectively.

The upper lip and lower lip detection accuracy on Cohn Kanade database are 98.9% and 90.4% respectively, and 97% and 96% on the KDEF database respectively. The Cohn Kanade database has images with a wide variation in lighting conditions and subjects from different racial backgrounds and skin tone. The subset of images considered from the KDEF database cover a wide range of actions made with the mouth. During the evaluation process, the first image for every subject was that of closed mouth state, which was used as the reference image for that subject. Sample detections from CK and KDEF databases are shown in Fig. 5.13 and 5.14 respectively.

TABLE 5.2: Detection accuracy of upper and lower lips in Cohn Kanade (CK) and DISFA databases (in percentages). The number within brackets represents the tolerance in pixels

	CK (<10)	KDEF (<15)
Upper lip	98.9	97
Lower lip	90.4	96
Distance between lips	89.3	93.5

Misdetections were caused due to wrong norstil detection, especially when the subject was making a frowning expression (as shown in Fig. 5.15), by which the nostrils were not detected and hence the mouth ROI was incorrectly estimated and the upper lip was detected wrongly. Other reasons for misdetections were contributed by the presence of strong shadows as in the case of some images of the CK database, due to which the lower lip was detected incorrectly.

(b) mouth state detection

The images from the CK and KDEF databases were manually annotated for mouth

FIGURE 5.15: Example of misdetection of upper lip due to wrong nostril detection

state as - open or closed, since annotations for mouth state is not provided with these databases. The images with lumen visible were marked as open state. The mouth state detection technique was evaluated on the two databases. The values of r_O was set to the average lip thickness during evaluation. The detection accuracy is computed based on the following definition:

$$ACC = (TP + TN)/(P + N) \tag{5.13}$$

where if TP is the number of correct detections of closed state, TN is the number of correct detections of open state, and $P+N$ gives the total number of images in groundtruth. In other words, the total number of correctly detected images with respect to the total number of images in ground truth is computed. Tables. 5.3 and 5.4 are the confusion

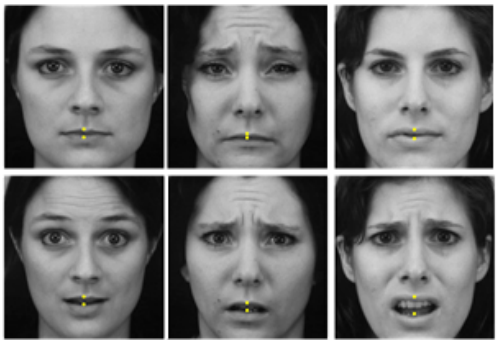


FIGURE 5.16: Example results of mouth state classification on the KDEF database, first row showing closed state and second row showing open state

matrices for the the mouth state detection evaluated on KDEF and CK databases respectively. The detection accuracy of the mouth state detection for the KDEF and CK databases is summarized in Table. 5.5. The average detection accuracy across the two databases is 94.8%. Sample detections of closed and open state on the KDEF database are shown in Fig. 5.16.

TABLE 5.3: Confusion matrix for mouth state classification for KDEF database, the labels on the left and top represent the actual and predicted classes respectively

	closed	open
closed	81	4
open	2	50

TABLE 5.4: Confusion matrix for mouth state classification for Cohn-Kanade database, the labels on the left and top represent the actual and predicted classes respectively

	closed	open
closed	141	6
open	5	32

TABLE 5.5: Evaluation of mouth state detection on KDEF and Cohn-Kanade databases (precision and recall indicated in percentages)

	No. of images	ACC
KDEF	137	95.6
Cohn-Kanade	184	94

Although the technique successfully detects most of the challenging cases, misdetections are contributed by cases where strong shadows were present near the lower lip (mainly in CK database), and due to incorrect ROI resulting from a misplaced nostril detection (mainly in KDEF database).

5.4.1.2 Mouth State based Wellness Indicators

The evaluation of the mouth state based wellness indicators - *yawning, talking* and *mouth kept open* are discussed in this section.

(a) Yawning

The detection of yawning was evaluated on a subset of the YawDD database [194]. The YawDD consists of two yawning databases, one of which is taken from a camera installed on the driver's dash. It consists of 29 video sequences with instances of yawning, captured at 30 FPS and frame size equal to 640 $\times$ 480. A subset of the database taken from across 10 subjects (5 male and 5 female), with head pose within $\pm 15^{\circ}$ was considered for evaluation. A total of 18 events of yawning were present in the subset

considered. The first few images in the sequence were that of the subject with mouth closed, which were used to set the thresholds in the reference state (closed state). The sequences were manually annotated for the events of yawning. The value of the thresholds T_Y^{min}, T_Y^{max} and r_Y were set to 20, 150 and 10% $\times$ face height respectively.

FIGURE 5.17: Sample detections from the YawDD database

TABLE 5.6: Evaluation of yawning detection on YawDD database

	No. of instances	% accuracy
Upper and lower lip	3518	97.7
Yawning	18	100

As seen from Table. 5.6, the average detection accuracy of upper and lower lips is 97.7%, and yawning is 100%. The proposed method was able to successfully detect even in cases where beard was present (some examples of this are shown in the second row of images in Fig. 5.17).

Discussion: There are very few other works that report quantitative evaluation results on the YawDD database namely - [150] and [195]. [150] reports a detection rate of 75% and [195] reports (qualitative) results after segregating the database into frontal facing, non-frontal facing, etc.). In our work, the results show a 100% yawning detection rate on the subset it was evaluated on. We have presented the comparison with respect to computational complexity with [150] in Section 5.4.2. In our work, a subset was chosen because the YawDD dataset is only accompanied with a sequence level annotation (of talking/yawning/normal), but we needed a frame level annotation (marking the start and end of a yawn, as well as the mouth feature points) within each sequence for evaluation, which is more detailed than the sequence level annotation considered in [150], and hence manually did the frame-level annotation on a subset of the database. Moreover,

the YawDD database has head movements much greater than what we have considered in this research in many parts of the video sequence, which is the other reason for choosing a subset of the database in which the head rotation is $\pm$ 15^o.

(b) Talking

The method of extracting the event of talking was tested on the Talking face video database [192], which consists of 5000 images taken from a video of a person engaged in conversation. The average size of the face of the person in this database is 333$\times$ 333 pixels. The database was manually annotated for the beginning and end of talking events. The thresholds T_A, r_O and $\sigma^2_{r_O}$ were set to 120, 0.7% $\times$ face height and 1 during evaluation. The size of the overlapping windows was set to 3$\times$frame rate of 25 FPS, with an overlap equal to half the window size.

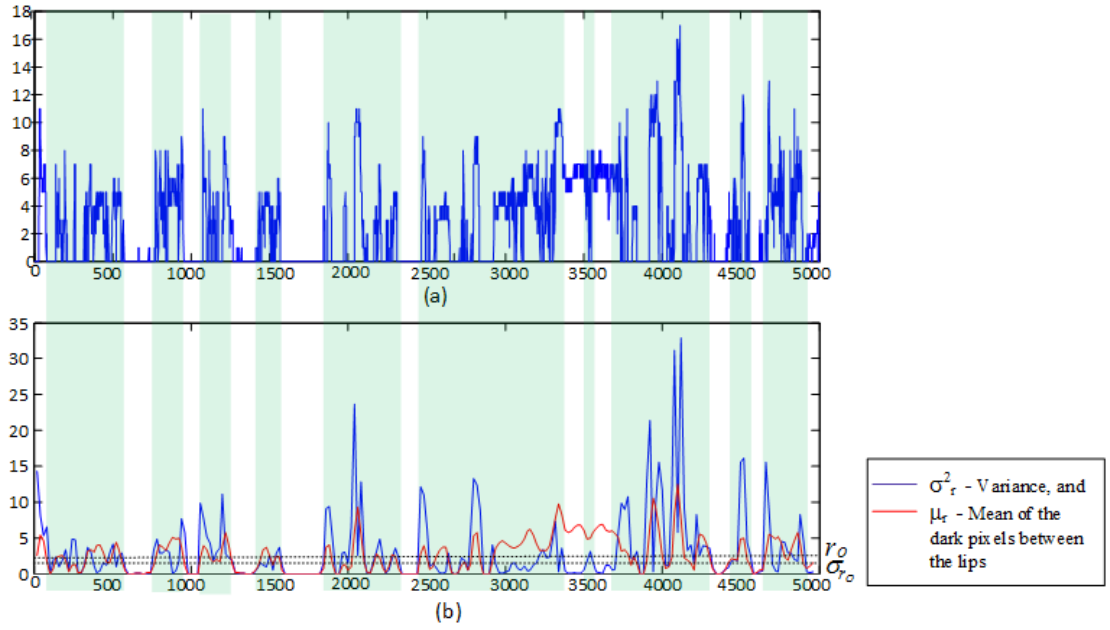


FIGURE 5.18: (a) r plotted for the 5000 frames of the Talking Face database, (b) mean and variance μ_r and variance σ_r plotted for the frames. The events of talking are shown by the regions shaded in green

Fig. 5.18 (a) shows the dark pixels between the lips r plotted for the 5000 frames, and Fig. 5.18 (b) shows the mean μ_r and variance σ_r plotted for the frames. The regions of high mean and variance are shaded in green, illustrating the time durations when talking events were detected. In the initial few frames of the Talking Face database (frames 30 to 86), the subject laughs. But, the proposed technique detects the same as an event

of talking. Thus, the method faces the limitation of differentiating between an event of talking, and other events that involve movements of the mouth, such as laughing.

The detection of talking was also demonstrated on a subset of the WellCam database, using the distance between the lips s. Figure. 5.19 shows the variance and mean of s plotted for a subset of images, showing a mean and variance above threshold over a period of time, indicative of talking. The thresholds T_A, s_O and $\sigma^2_{s_O}$ were set to 100, 6% $\times$ face height and 1 during evaluation. The size of the overlapping windows was set to 3$\times$frame rate of 25 FPS, with an overlap equal to half the window size.

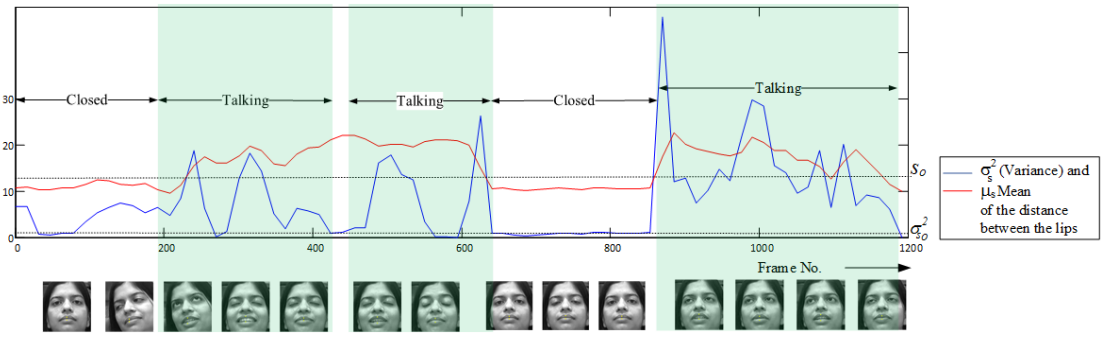


FIGURE 5.19: An illustration of temporal analysis of the mouth state on a subset of WellCam database showing detection of *talking*, the events of talking are shown by the regions shaded in green. The thumbnails visually illustrate the groundtruth

(c) Mouth kept open

Due to lack of databases to evaluate the detection of mouth kept open over a period of time, it was tested on a subset of the WellCam database consisting of 225 images of a subject. The thresholds T_O, r_O and $\sigma^2_{r_O}$ were set to 75, 3% $\times$ face height and 1 during evaluation. The size of the overlapping windows was set to 3$\times$frame rate of 25 FPS, with an overlap equal to half the window size. The plot of μ_r and σ_r is plotted for a sequence showing mouth kept open. mouth kept open from mouth kept closed and talking are plotted against time in Fig. 5.20. The event of mouth being kept open was detected successfully, based on the low variance ($< r_O$) and high mean values ($> \sigma^2_{r_O}$), as shown in Fig. 5.20.

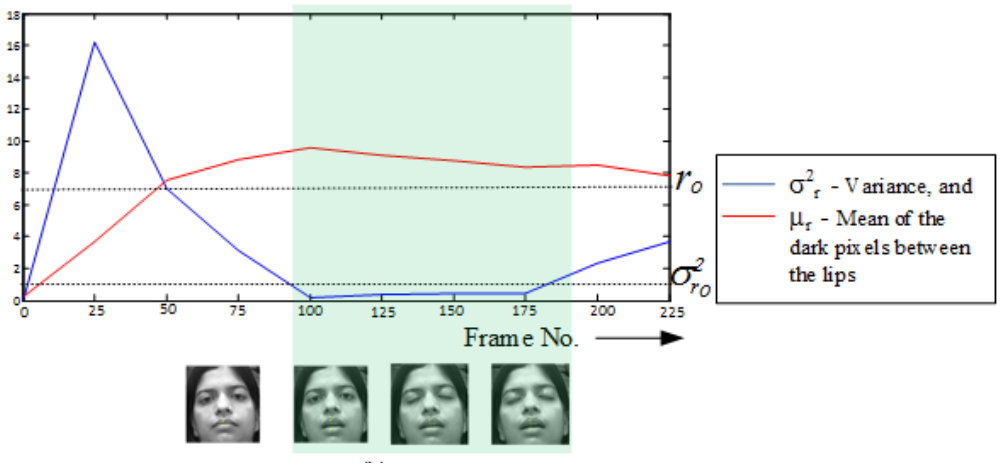


FIGURE 5.20: An illustration of temporal analysis of the mouth state to extract *mouth kept open*; the event of mouth kept open is shaded in green. Thumbnails visually illustrate the ground truth

5.4.2 Computational Complexity Analysis

In this section, computational cost of the proposed mouth state detection technique is computed and compared with a state-of-art method proposed in [150]. [150] method uses the Viola-Jones algorithm for face and mouth detection, with modifications made for fast and memory-efficient detection. Let us assume we use the face detection proposed in [150] in our method to detect the face. So, we consider the computational complexity of this step as provided in [150]. Next, we compare the computational complexity of the mouth and mouth state detection. In [150], the Viola-Jones method with modifications is applied in the lower half of the face, which is the search region. The order of complexity of this step provided in [150] is $O(SM'S')$ where S is the size of the face block, S' is the number of pixels in the mouth block and M' is the number of selected mouth features.

The order of complexity of the proposed method is computed as follows: The mouth ROI detection involves eyebrow detection, followed by the detection of eyebrow inner ends and nostril detection. The order of complexity is $O(S/2 + S/4)$ considering the eyebrow detection is run on the upper half the face image and the eyebrow inner ends and nostril detection is run on quarter the face image.

The detection of mouth feature points involves computations within the 3 mouth ROIs R_1, R_2 and R_2'. The order of computational complexity of the upper lip detection is $O(R_1+R_2)$ and that of the lower lip is $O(nR_2'/2)$ considering the mean intensity profile $\mathbf{P}$ was computed within n overlapping windows. Based on the experiments conducted, the value of n is mostly in the range of 20 to 25. R_1, R_2 and R_2' combinedly can be approximated to $S'/2$ (based on observation of the size of the mouth block shown in the figures in [150]). The overall order of complexity of the proposed method is $O(3S/4)+O(S'/3)+O(nS'/12)$. We note that the order of complexity of the proposed method is linearly dependent on S and S', whereas the order of complexity in the method proposed in [150] is dependent on the multiplication of S and S'.

Thus, drawing a comparison between the proposed method and [150], it can be seen that the order of complexity of the proposed method is significantly lesser than [150].

5.5 Extraction of Brow Furrows

So far, we discussed the technique for detection of mouth state and the extraction of mouth-based wellness indicators. Next, the extraction of brow furrows and the wellness indicators from brow furrows is presented.

The proposed method for brow furrow detection is intended to detect the presence or absence of brow furrows caused due to the eyebrow lowering action. The steps in the proposed brow furrow detection technique are discussed in this section.

(a) Estimation of ROI

The proposed method is aimed at detecting the brow furrows that appear as nearly vertical or vertical lines most predominantly in the region between the eyebrows. Thus, the ROI for detecting the brow furrows is the region between the eyebrows' inner ends. The inner ends of the eyebrows detected based on Sec. 5.2.1 are reused and the ROI for brow furrows is a rectangular window drawn in between the eyebrow's inner ends as shown in Fig. 5.21(b).

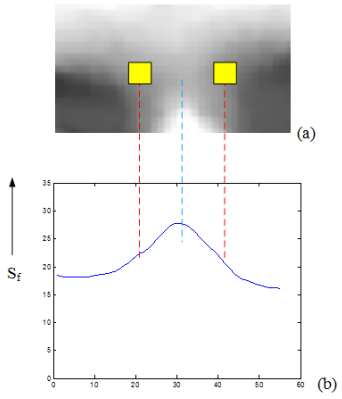


FIGURE 5.21: (a) detected inner ends of the eyebrow as discussed in Sec.5.2.1, marked by the yellow boxes (b) ROI for detecting brow furrows

(b) Vertical Signed Edge Maps to Extract Brow Furrows

Since brow furrows appear as lines in the ROI, initially, we tried edge detection techniques to extract brow furrows. However, the challenges in using standard edge detection techniques are similar to those discussed in Section 3.3.3 of chapter 3. Lower thresholds in edge detection increase noise and higher thresholds resulted in loss of information. Some examples of Canny and Sobel edge detectors with varying thresholds applied on the ROI are shown in Fig. 5.22. In order to eliminate the edges detected due to subtle noise contributed by the image acquisition process, the ROI is filtered. A 2-dimensional Gaussian filter was applied. It was hence observed that furrows are subtler

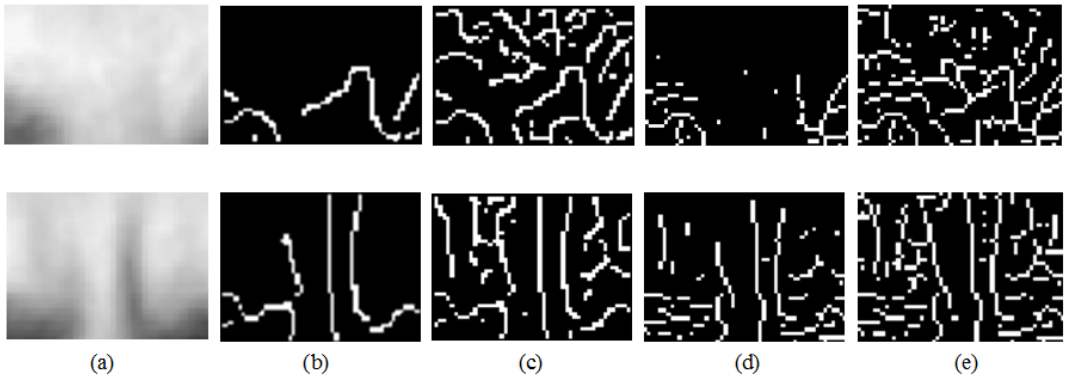


FIGURE 5.22: (a)Brow furrow ROI (b) Canny edge detector - threshold 0.5, (c) Canny - threshold 0.05, (d) Sobel edge detector - threshold 0.02, (e) Sobel - threshold 0.01

features compared to features such as the eyebrows or eyes. This led to the use of partial gradient maps, where the furrow edges can be extracted in a controlled manner. The extraction of partial gradient maps for brow furrow detection are described as follows.

A brow furrow is characterized by a transition of intensity from light to dark followed

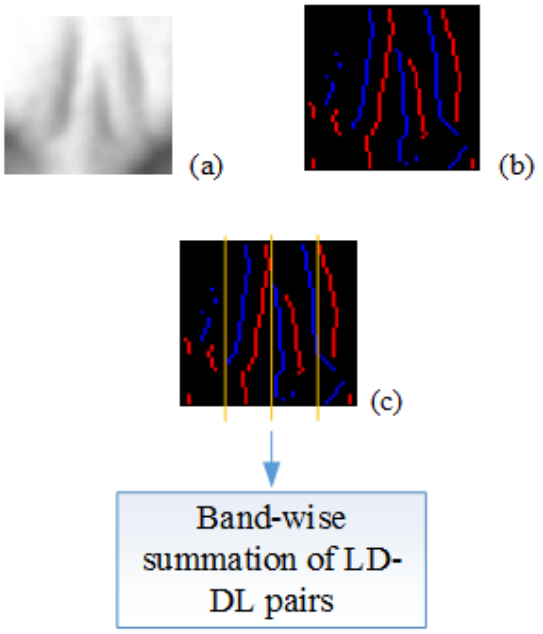


FIGURE 5.23: (a) Extracted ROI for detecting brow furrows (b) partial gradient maps of the ROI (blue pixels indicate light-dark transitions and red pixels indicate dark-light transitions) (c) band-wise pairing or light-dark and dark-light transitions

by a dark to light transition. The gradient map G_x is first generated by applying the Sobel kernel within the ROI. Then, the partial gradient maps $F_{x\_}$ and F_{x_+} of the ROI are generated (5.14).

$$\begin{aligned} F_{x-}(x,y) &= 1\ if\ G_x(x,y) \geq T_{ld} \wedge F_x(x,y) = 1 \\ F_{x+}(x,y) &= 1\ if\ G_x(x,y) < T_{dl} \wedge F_x(x,y) = 1 \\ F_x &= F_{x+} \cup F_{x-} \end{aligned} \tag{5.14}$$

In order to set the thresholds T_{ld} and T_{dl} to generate the partial gradient maps, the thresholds that were used for eyebrow detection are used as reference. We recall that the the thresholds were set in an iterative manner during the eyebrow detection process. Hence, the skin tone and lighting conditions have been taken into consideration while detecting the eyebrows. The thresholds for brow furrow detection are set as a fraction of

the thresholds with which eyebrows were extracted. If the threshold at which eyebrow's upper edge was extracted was T_u, then T_{ld} and T_{dl} are set to $0.07 * T_u$.

Then, the ROI is divided into 'm' overlapping narrow vertical bands. The width of the band is set to $0.3\times$ the band-width set for eyebrow detection. Within each band b_i, points from F_{x_-} and F_{x_+} along the same x-axis are paired (as illustrated in Fig. 5.23). If we were to retain only such *pairs* of pixels in F_x, let such a map be called F_{x_P}. Next, in order to ensure the pixels with higher magnitude are given higher weightage than the pixels with lower magnitude, the pixel magnitudes are multiplied by a gradient magnitude-based weight which is computed as follows. The gradient magnitudes of the pixels in F_{x_P} are first extracted. So, we have G_{x_P} extracted from G_x. The maximum value of the gradients in G_{x_P}, η is then computed as shown below:

$$\begin{aligned} G_{x_P} &= |G_x| \,.* F_{x_P} \\ \eta &= max(G_{x_P}) \end{aligned} \tag{5.15}$$

Then, each non-zero pixel in G_{x_P} is divided by η and multipled again with its original value to get the weighted gradient magnitudes $G^W_{x_P}$.

$$G^W_{x_P} = G_{x_P}/\eta \times G_{x_P} \tag{5.16}$$

The sum of all the weighted gradient magnitude values in $G^W_{x_P}$ and the number of pairs in $G^W_{x_P}$ is computed. Thus, for an image i, the weighted gradient magnitude sum g_i and total number of pairs n_i forming the furrows are computed:

$$g_i = \Sigma G^W_{x_P} \tag{5.17}$$

(c) Overall technique for extraction of brow furrows

The above steps are used to extract the brow furrows in an image. However, in reality, some subjects may have intransient furrows in natural state. In such cases, even if the algorithm detects furrow lines, they may be the naturally present intransient furrows. But, the furrows we wish to detect are those that are formed due to the brow lowering

action. Hence, the overall process of detection of brow furrows has two phases: (a) initialization phase and (b) detection phase.

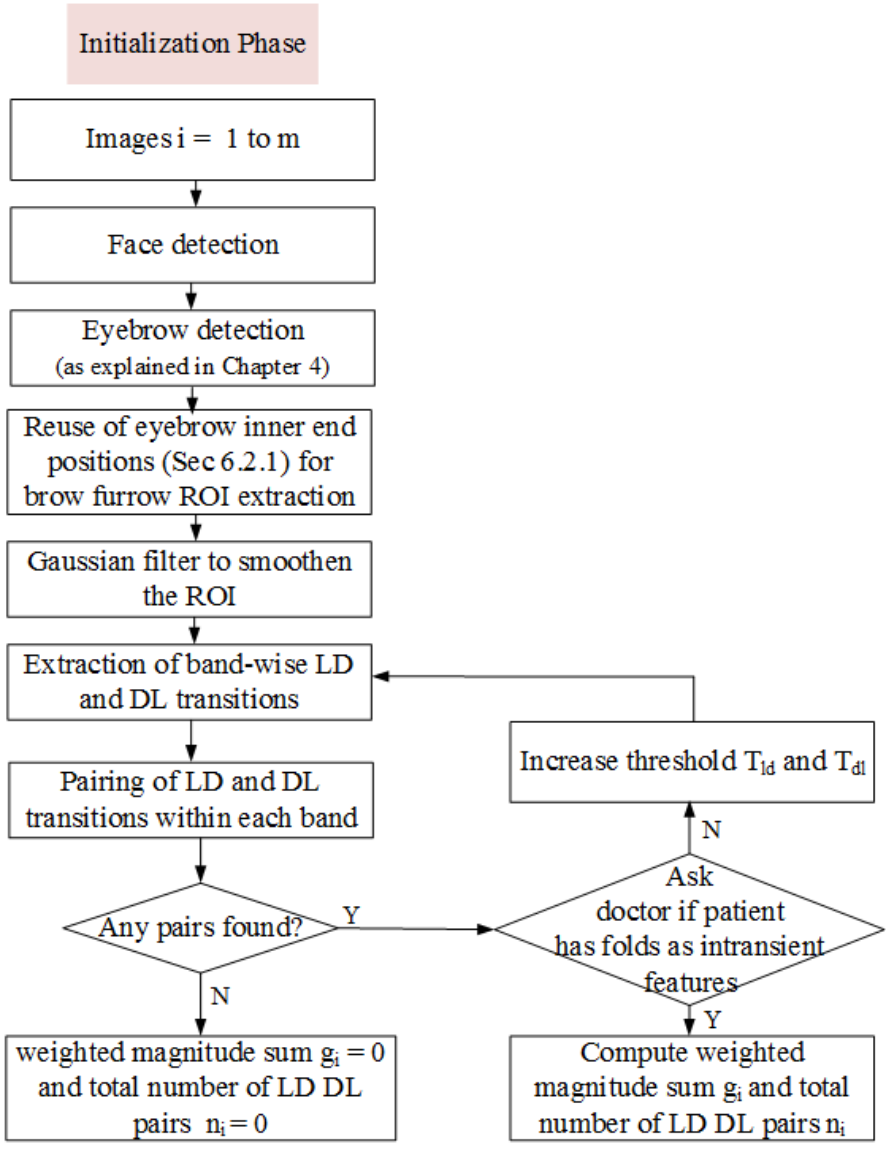


FIGURE 5.24: Steps in the brow furrow extraction technique during the initialization phase

- Initialization phase: During the initialization phase, the algorithm learns from a set of m reference images, which are images of the subject in natural state. Through this, the algorithm learns if the subject has brow furrows as intransient features in their natural state. In order to ensure the robustness of the threshold setting in this step, the doctor's input is used.

Let us consider the following case: When the thresholds T_u and T_l that are applied to extract the partial gradient maps, let us say a certain number of LD-DL pairs were extracted. Now, the doctor is asked to provide input as to whether the subject has furrows as intransient features in his natural state. If the doctor inputs a *yes*, then, the weighted gradient magnitude sum g_i and the number of LD-DL pairs n_i is computed. This is repeated for the first m images and the total mean g_I and mean count n_I are computed and stored as reference. Next, consider the case when LD-DL pairs were extracted from the partial gradient maps, but the doctor provides the input that the patient does not have natural brow furrows. In such a situation, the thresholds T_u and T_l are increased iteratively until no such LD-DL pairs are found. In such a case, g_I = 0 and n_I = 0.

$$g_I = mean(g_i) \forall i \in [1, N] \tag{5.18}$$

$$n_I = mean(n_i) \tag{5.19}$$

The steps in brow furrow extraction during initialization phase are summarized in Fig. 5.24.

For such cases, although the natural furrow lines will be detected during initialization, the relative change in gradient magnitude and count are considered to detection.

- Detection phase: During detection phase, for an incoming image 'k', the weighted gradient magnitude sum g_k and the number of LD-DL pairs n_k are computed. Two cases are possible: (1) Subject does not have natural intransient furrows, n_I =0 and g_I = 0, and (2) subject has natural intransient furrows, n_I and g_I are non-zero. For case 1, the presence of furrows is computed based on the count n_k. If n_k exceeds a threshold t_{n_1}, or if n_k exceeds $t_{n_1'}$ and g_k exceeds t_{g_1}, then furrows are detected. For case 2, the relative change with respect to the reference values g_I and n_I is first computed:

$$g_d = (g_k - g_I)/g_I \tag{5.20}$$

$$r = (n_k - n_I)/n_I$$

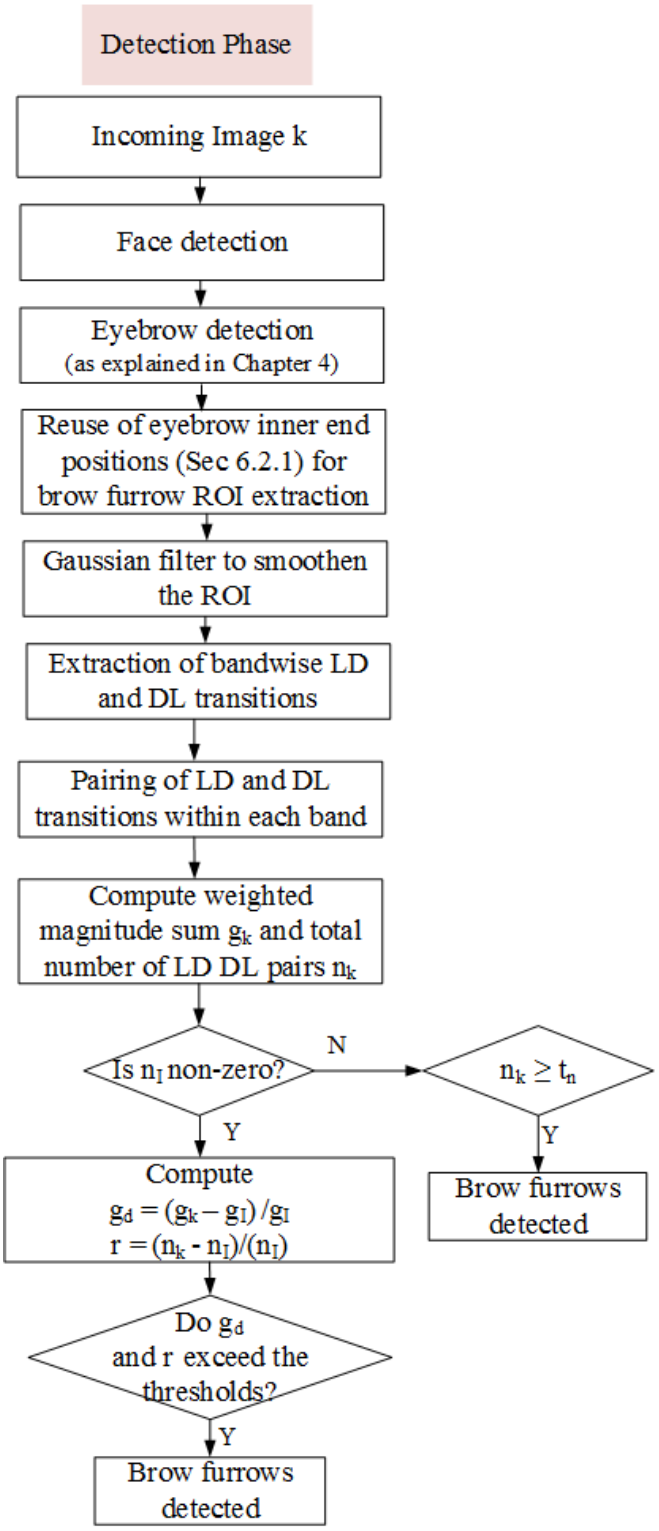


FIGURE 5.25: Steps in the brow furrow extraction technique during the detection phase

Then, if g_d exceeds t_{g_2} and r exceeds t_{n_2} or g_d exceeds $t_{g_2'}$ or r exceeds $t_{n_2'}$, furrows are detected. The thresholds for gradient magnitude and count are set

based on heuristic analysis of the databases. The steps in brow furrow extraction during detection phase are summarized in Fig. 5.25.

(d) Extraction of Wellness Indicators from Brow Furrows

The persistence of brow furrows over time is the wellness indicator extracted based on the temporal analysis of frame-level detection of brow furrows. Based on the frame level detection described above, each frame is classified as 'furrows present' or 'furrows absent'. Then, the percentage of frames d_{F_j} with furrows present is computed within a window. d_{F_j} is then computed across N_f such overlapping windows and $\mathbf{d_F}$ is an array of d_{F_j} where j ranges from 1 to N_f:

$$\mathbf{d_F} = [d_{F_1}, d_{F_2}, \cdots d_{F_j}, \cdots d_{F_{N_f}}] \tag{5.21}$$

Thus, $\mathbf{d_F}$ gives information about the persistence of brow furrows across a longer duration of time.

5.6 Performance Evaluation of Brow Furrow Extraction

FIGURE 5.26: Sample images from the DISFA database showing (a) the reference images of 4 subjects and (b) images of the 4 subjects detected with brow furrows

The brow furrow detection technique was evaluated on two standard databases: the Extended Cohn Kanade database [196] and DISFA database [186]. A subset of 3454 images of 9 subjects from the DISFA database were considered, of which 2841 images

had brow furrows. Sample images from DISFA database are shown in Fig. 5.26. The action unit annotation provided with the database could not be directly used, because in some frames, although the brow lowering action was greater than zero (e.g. 1 or 2 units), the furrows were not visible. Hence, the images were manually annotated for the presence of brow furrows. A total of 2841 images were annotated to have brow furrows.

4 out of 9 subjects had brow furrows developed as a natural intransient feature, the brow furrows were seen in the reference image, relative to which the furrows as a result of brow lowering were detected in the incoming images. An example of each of these cases is shown in Fig. 5.27 and 5.28 respectively. The evaluation results on DISFA database are summarized in Table. 5.7. A precision and recall of 93.8% and 86.7% respectively were achieved. The lower recall is contributed by images in which the furrows were relatively lesser prominent and cases where the furrows occurred closer to the eyebrow inner ends, and thus, not fully captured within the eyebrow ROI.

TABLE 5.7: Evaluation of brow furrow detection technique on DISFA database

	Precision	Recall
DISFA	93.8%	86.7%

The evaluation of brow furrow detection technique on the Extended Cohn-Kanade database was done at sequential level. A subset of 351 sequences of 72 subjects were considered for evaluation. Sample images from the Extended Cohn-Kanade database are shown in Fig. 5.29. The image sequences where brow lowering action (AU 4) has been performed was known from the ground truth FACS labels provided with the database. Out of 351 sequences, 37 sequences were found to have the brow lowering action, and the brow furrows resulting from it. Out of 37 sequences, the proposed method was able to detect 35 sequences with brow furrows, under the same threshold settings used for all the sequences.

The evaluation results of the brow furrow detection on Cohn Kanade database is summarized in Table. 5.8. The resulting precision and recall were 90% and 94.5% respectively. The lower precision is contributed by cases in which the noise gained prominence due to the high contrast, and hence were detected as furrows.

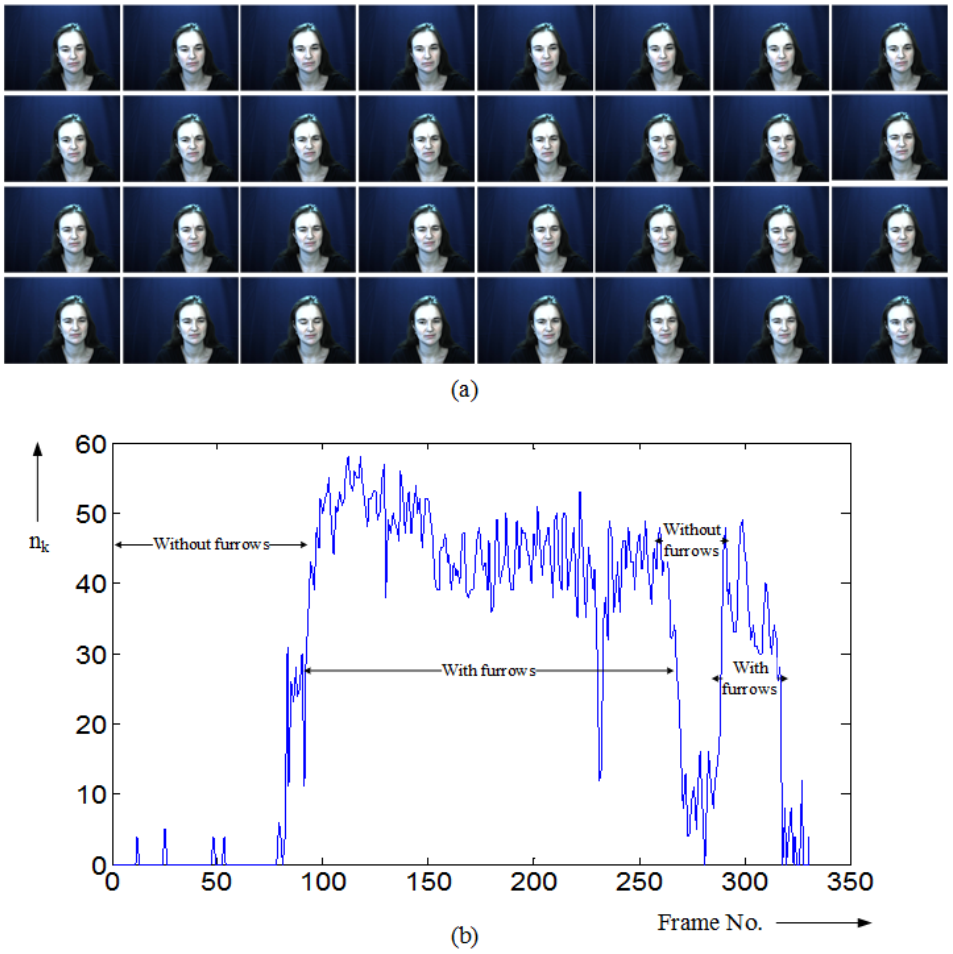


FIGURE 5.27: An example where the subject does not have natural intransient furrows: the furrows detected as temporal features, indicated in the (b); the thumbnails visually illustrate the groundtruth (a)

TABLE 5.8: Evaluation of brow furrow detection technique at sequence level on Cohn-Kanade database

	Precision	Recall
Cohn-Kanade	90%	94.5%

5.7 Summary

In this chapter, techniques to extract wellness indicators from mouth and brows furrows were presented. Firstly, a compute-efficient technique to extract the upper and lower bounds of the mouth (the upper and lower lip positions) was proposed, which was used to detect mouth state. The technique was based on simple operations such as accumulations and mean performed on reduced ROIs. The techniques to extract upper and lip

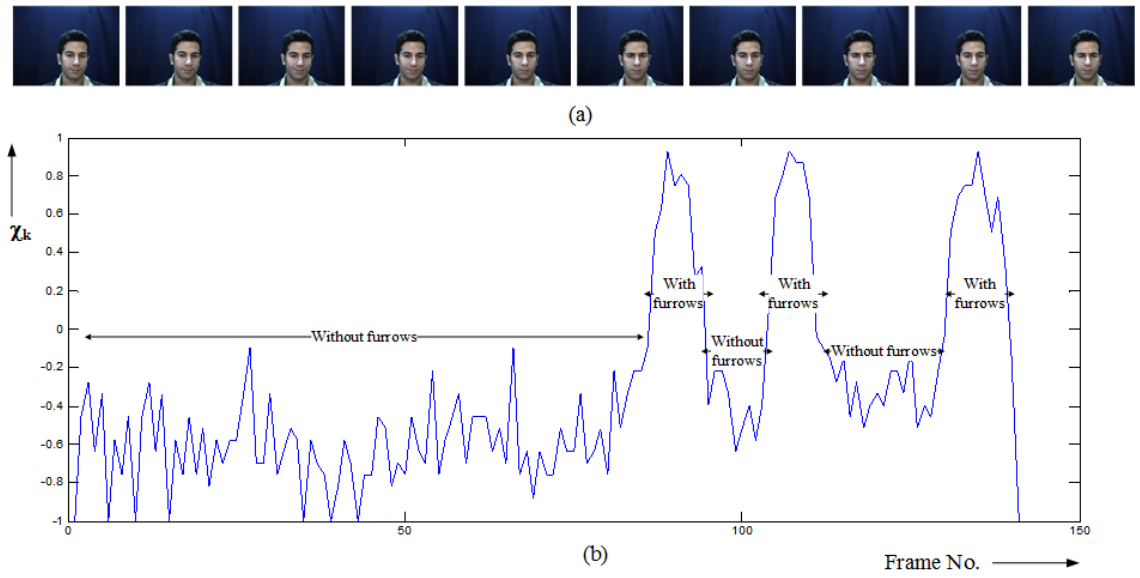


FIGURE 5.28: An example where the subject has natural intransient furrows: the furrows detected as temporal features, indicated in the (b); the thumbnails visually illustrate the groundtruth (a)

FIGURE 5.29: Sample images from the sequences in Cohn Kanade database with brow furrows

positions, and eventually the distance between the lips showed an accuracy of 95.5% and 91.4% respectively upon evaluation on subsets of the extended Cohn-Kanade and KDEF databases. The proposed method for mouth state detection was evaluated on the same databases, and an average accuracy of 95% was achieved. Temporal analysis of the detected mouth state was done to extract the wellness indicators such as - mouth kept open, talking and yawning. The YawDD database was used to evaluate the detection of *yawning* and a 100% detection rate was achieved. The detection of *talking* was tested on Talking Face database, and the *mouth kept open* was tested using the proposed WellCam database. A simple technique based on partial gradient maps was proposed

for brow furrow detection. Band-wise extraction of edge pixels from the partial gradient map and the number of pairs and weighted accumulation of the gradient magnitudes were used to detect brow furrows. The proposed method was evaluated on the Cohn-Kanade and DISFA databases and an average precision and recall of 92% and 91% were achieved. Having seen the extraction of facial features and the associated wellness indicators, which used face detection as a pre-processing step, the next chapter focuses on reducing the computational cost involved in face detection. A technique for patient face localization is proposed that takes advantage of the controlled setting of an indoor patient monitoring scenario is proposed.

CHAPTER 6

Accelerating Patient Face Localization

6.1 Introduction

In chapters 3 to 5, compute-efficient techniques to localize facial features and extract wellness indicators based on their analysis were presented. Face detection was a pre-processing step in these techniques, as explained in Sec.3.3.1.

In addition to the reduction in computational complexity in the facial feature extraction, further reduction is explored in the face detection step, as conventional face detection is compute-intensive. In this chapter, a technique to accelerate face detection is proposed. The technique involves reducing the search area or search space to specific areas with a higher probability of face, and in these areas, the conventional face detection is applied, as against the entire image. The technique takes advantage of the controlled setting of on-bed patient monitoring.

6.2 Computational Challenges in Face Detection

The face detection technique used as a pre-processing step in this work is based on the widely used Viola and Jones approach for face detection [180]. Currently, most methods for face detection rely on grayscale information [197], and among the robust approaches for face detection is the Viola Jones approach [180]. The recent progresses in face detection are made within the cascade framework proposed in the Viola Jones approach [197]. It is a sliding window-based approach, which can result in significant feature computations within every instance of the sliding window [198]. The resulting high computational power can create bottlenecks in the realization of such techniques on embedded platforms, and hence for large scale deployment.

Hardware level optimizations are introduced to enable the embedded realization of the face detection algorithms [199] [200] [201] [202] [203] [204]. [202], [199] and [204] introduce parallelization in architecture and [203] employs multi-core architecture, while [200] uses a multi-GPU implementation. [201] uses the genetic algorithm to optimize the adaboost training and [199], [202] and [204] use pipelined architecture.

Introducing optimizations in the algorithm itself can further reduce the computational costs. One approach is to introduce peripheral algorithms such as search space reduction techniques [198], [205], [206], [207], which have shown a significant reduction in computational complexity. Skin color is one of the commonly used attributes for reducing the possible search space for face detection [206]. However, this requires sensitive skin color models to accurately segment the required regions of interest (ROIs). Upper body detection is explored in [205], to reduce the ROIs for face detection. Oriented Integration of Gradients (OIG) is proposed in [205] as a feature to describe the sub-parts of human head-shoulder curves, which are then detected using a classifier followed by Hough voting scheme to localize their position. This, introducing context-awareness into the algorithm can aid in eliminating redundant computations and hence increase computational efficiency [208].

6.3 Search Space Reduction for Face Localization

In this research, an *on-bed patient monitoring* scenario is considered as shown in Fig. 6.1. The posture of the patient on the bed can vary from lying down to sitting inclined or upright on the bed. The camera is placed such that the patient is facing the camera, and at a certain height and angle with respect to the patient's face.

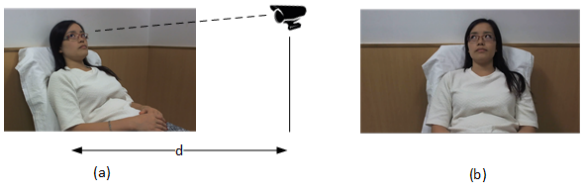


FIGURE 6.1: (a) Setup of camera with respect to the subject in an on-bed patient monitoring scenario (b) frontal view

The head and shoulder profile has been considered as a feature unique to humans and used in face localization [209] and human detection [205]. In the above described scenario of patient monitoring, the head-shoulder profile of the human is a feature that can be efficiently captured under the constrained setting.

The proposed technique processes the edge information of the image and is aimed at detecting the head and shoulder curves, and eventually shortlisting sub-windows that give a higher probability of the presence of face. The computational cost of applying the face detection within shortlisted sub-windows is computed and compared with the cost incurred in running the face detection algorithm on the entire image.

6.3.1 Head Shoulder Curve Detection to Reduce Search Space

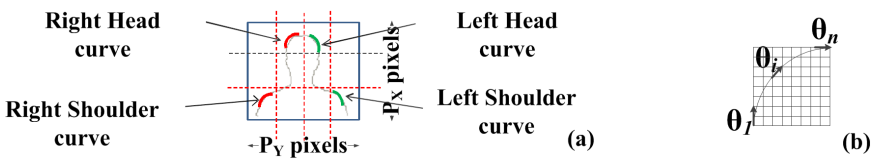


FIGURE 6.2: (a) Head and shoulder curves of a human of scale defined by P_X and P_Y pixels. (b) Illustration showing the linear approximation of the curve [210].

The proposed method is aimed at detecting the head and shoulder curves of the human being of a given scale. As shown in Fig. 6.2(a), the scale is defined by the ratio $P_X : P_Y$, where P_X is the distance between the top of the head to the shoulders, and P_Y is the distance between the two shoulders in pixels. Given this scale, the proposed method detects possible right and left, head and shoulder curves as shown in Fig. 6.2, that satisfy the given scale. Note that in this chapter and the rest of the thesis, *right* and *left* are defined relative to the subject's right and left side.

The detection of curve using Gradient Angle Histograms (GAH) is presented first, followed by a block-based approach to detect human head-shoulder curves. Then, the association of the curves to localize a human face is presented.

6.3.1.1 Gradient Angle Histograms (GAH) for Curve Detection

As shown in [211], a curve can be divided into smaller segments such that each segment can be approximated by the tangent passing through the mid-point of that segment. This is illustrated in Fig. 6.2(b). A curve C has a gradual change in tangential orientation from θ_1 to θ_n.

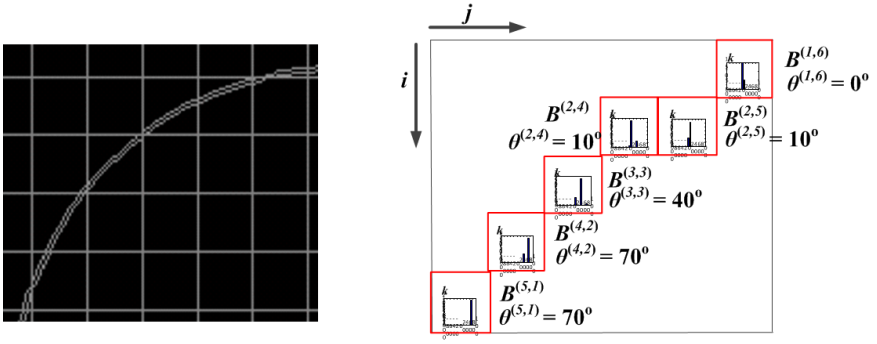


FIGURE 6.3: A convex curve is divided into blocks $B^{(i_1,j_1)}$, $B^{(i_2,j_2)}$, ..., $B^{(i_N,j_N)}$. The curve appears as linear segments and GAHs show peaks corresponding to the gradient angles of these linear edges in each block.

(a) GAH to detect linear edges

In this work, Gradient angle histograms (GAH) have been used to identify curves of specific curvatures that can be associated with head and shoulder curves of a human, as

shown in Fig. 6.2(a). GAH was shown to be an effective way in [212] to detect linear edges in a block-based approach.

Consider an image block I_B, whose edge pixels have been extracted in the edge map E_B. The gradient angle of each edge pixel is computed. Then, GAH (represented by $\mathbf{h}$) is the histogram of gradient angles, where $h_i \in \mathbf{h}$ represents the count of edge pixels having the gradient angles in the i-th bin. Given the edge block E_B and the gradient angles $\theta(x, y)$ of every edge pixel in E_B, GAH can be used iteratively in the following way to extract possible linear edges in E_B: In each iteration j, bin k with the maximum height in $\mathbf{h}$ is obtained, which represents a possible linear edge in E_B [212]. Edge pixels with gradient angles in the range of $\theta_k - \epsilon \leq \theta \leq \theta_k + \epsilon$, centered around the k-th bin in GAH will result in $E_j \subset E_B$ with possible linear edges. This can be summarized as the following:

$$k = \arg\max_i \mathbf{h} \tag{6.1}$$

$$E_j = \{e(x, y) \in E_B | \theta_k - \epsilon \leq \theta(x, y) \leq \theta_k + \epsilon\} \tag{6.2}$$

The above equations are repeated after removing the k-th entry in the GAH $\mathbf{h}$, to get the next peak in $\mathbf{h}$. This is repeated until a termination condition T is reached. This operation is denoted as

$$\{\mathbf{\Theta}^B, \mathbf{E}^B\} = \mathbf{\Phi}(\mathbf{h}^B, E^B, T) \tag{6.3}$$

where $\mathbf{\Theta}^B$ denotes the set of angles θ_ks in the GAH $\mathbf{h}^B$ for block B, and the boldfaced $\mathbf{E}^B$ denotes the set of edge maps that are obtained using θ_ks using equations (6.1 and 6.3). The termination condition depends on the application in which GAH is applied and is defined later in this section, in the context of this work. It is to be noted that GAH differs from HoG (Histogram of Oriented Gradients) [213] in terms of how they are computed and used.

(b) GAH for curve detection

The above GAH formulation can be used to detect a curve as follows. A curved edge, with a known curvature, i.e. given it is a convex or concave curve, is divided using a set

of blocks $B^{(i_1,j_1)}, B^{(i_2,j_2)}, \cdots, B^{(i_N,j_N)}$, such that they are along the curve as shown in Fig. 6.3. It can be seen that the segments of the curve appear as linear edge segments in each block. GAHs are computed in each block as shown in Fig. 6.3. These GAHs are represented by the set $\mathbf{H}$, given by:

$$\mathbf{H} = \{\mathbf{h}^{(i_1,j_1)}, \mathbf{h}^{(i_2,j_2)}, \cdots, \mathbf{h}^{(i_N,j_N)}\} \tag{6.4}$$

Applying $\mathbf{\Phi}(\cdot)$ on each block of $\mathbf{H}$, $\mathbf{\Theta}^{(i,j)}$ is obtained. If it is considered that highest peaks in each GAH, $\Phi(\cdot)$ on $\mathbf{H}$ will give peaks at $\mathbf{\Theta} = \{\theta_k^{(i_1,j_1)}, \cdots, \theta_k^{(i_N,j_N)}\}$. As indicated in Fig. 6.3, these gradient angles correspond to the edge pixels of the line segments that form the curve. If these linear segments are approximated as the tangents of the curved segments in the blocks, then they should satisfy (6.5), i.e.,

$$\theta_k^{(i_1,j_1)} > \theta_k^{(i_2,j_2)} > \cdots > \theta_k^{(i_N,j_N)} \tag{6.5}$$

Having seen how a single curve is extracted using GAH in a block-based approach, we will now see how this is applied in detecting head and shoulder curves in humans.

6.3.1.2 Block-based GAH for Detecting Shoulder & Head Curves

The overview of the proposed method for detecting head and shoulder curves is provided in Fig. 6.4. The steps shown in Fig. 6.4 are discussed in the paragraphs to follow.

Given an $X \times Y$ sized grayscale image I, it is envisaged to detect shoulder-head region that can be captured within a window of size $P_X \times P_Y$ pixels. I is first divided into $b_s \times b_s$ sized blocks, such that a curve is decomposed into smaller linear segments in each block. In every block $B^{(i,j)}$, Sobel filters [181] are applied and the edge map computed is denoted by $E^{(i,j)}$. For every edge pixel $E^{(i,j)}_{(x,y)}$ in the (i,j)-th block, its gradient angle $\theta^{(i,j)}_{(x,y)}$ is computed using the same Sobel kernels. In every block $B^{(i,j)}$, GAHs are computed for the constituent edge pixels. Therefore, for the $X \times Y$ image,

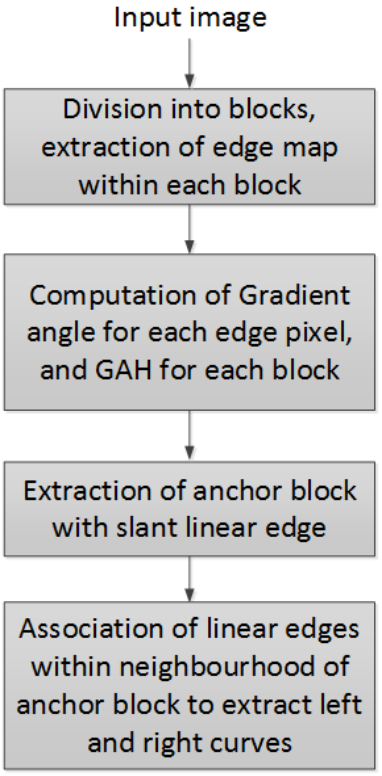


FIGURE 6.4: Overview of the proposed method for extracting head and shoulder curves

$\mathbf{H}$ is the set of $\frac{X}{b_s} \times \frac{Y}{b_s}$ GAHs corresponding to all blocks, i.e.,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}^{(0,0)} & \mathbf{h}^{(0,1)} & \dots & \mathbf{h}^{(0,\frac{Y}{b_s}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}^{(\frac{X}{b_s}-1,0)} & \mathbf{h}^{(\frac{X}{b_s}-1,1)} & \dots & \mathbf{h}^{(\frac{X}{b_s}-1,\frac{Y}{b_s}-1)} \end{bmatrix} \tag{6.6}$$

Applying $\Phi(\cdot)$ on $\mathbf{H}$, the following is obtained:

$$\mathbf{\Theta} = \mathbf{\Phi}(\mathbf{H}, E, T) = \begin{bmatrix} \mathbf{\Theta}^{(0,0)} & \mathbf{\Theta}^{(0,1)} & \dots & \mathbf{\Theta}^{(0,\frac{Y}{b_s}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Theta}^{(\frac{X}{b_s}-1,0)} & \mathbf{\Theta}^{(\frac{X}{b_s}-1,1)} & \dots & \mathbf{\Theta}^{(\frac{X}{b_s}-1,\frac{Y}{b_s}-1)} \end{bmatrix}$$

The termination condition T is defined as a simple threshold $b_s/4$. In other words, $\mathbf{\Theta}^{(i,j)}$ will have θ_ks corresponding to all bins in the GAH $\mathbf{h}^{(i,j)}$ which are higher than $b_s/4$. The selected gradient angles in each $\mathbf{\Theta}^{(i,j)}$ are further constrained in different ways to detect the shoulder and head curves. This will be explained next.

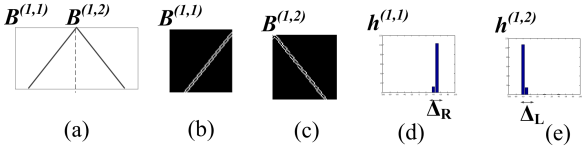


FIGURE 6.5: (a) blocks B_1 and B_2 (b) corresponding edge maps of B_1 and B_2 (c) GAH divided into right and left ranges Δ_L and Δ_R that will be used to find the right and left shoulder and head curves.

Referring to Fig. 6.2(a), four curves that are of interest are the right and left head curves, and right and left shoulder curves. In terms of the direction of convexity, the right head and shoulder curves are called as right curves. The left head and shoulder curves will hence forth be also called as left curves, unless otherwise stated explicitly.

In order to detect the left and right curves, the GAH $\mathbf{h}$ is divided into two ranges called the *right* and *left* angle ranges denoted by Δ_R and Δ_L respectively. Referring to Fig. 6.5, a peak in Δ_R represents a linear edge that slants diagonally upwards, i.e. $\diagup$, whereas a peak in Δ_L of the GAH indicates a linear edge slanting diagonally downwards, i.e. $\diagdown$. It can be seen in Fig. 6.5 that the right and left slant edges, shown in Fig. 6.5(b) & (c), result in distinct peaks in different regions of their respective GAHs shown in Fig. 6.5(d) & (e).

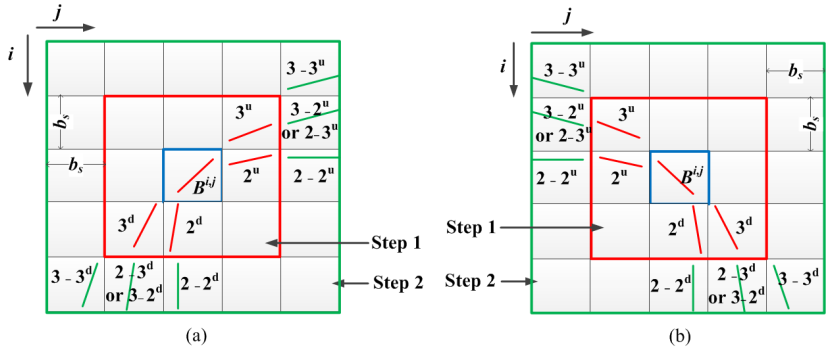


FIGURE 6.6: Ranking of blocks for right and left curve detection.

The right curve detection process is explained next. The same can be applied for detecting the left curve by changing the different parameters such as the angle ranges, and the direction of association between the blocks.

(a) Extraction of anchor block with slant linear edge

Referring to Fig. 6.6 (a), let us consider a block $B^{(i,j)}$, which is part of the right curve. The $\mathbf{\Theta}^{(i,j)}$ for this block is checked to find $\theta_k \in \Delta_R$, i.e. if there are any edge pixels that

are forming a right slant linear edge. This step is the first and critical step because it is considered that the right curve must necessarily have a segment that has a gradient angle in Δ_R. If no such θ exists, the next block is processed. If θ_ks exist in Δ_R, then the block is considered for further processing. The $\theta_k \in \Delta_R$ with maximum $\mathbf{h}_k$ is considered as a the anchor angle $\theta_a^{(i,j)}$ and block $B^{(i,j)}$ is considered as the anchor block. This anchor block will be used to check further if there are left and right shoulder curves.

(b) Association of linear edges within the neighbourhood of anchor block

With $B^{(i,j)}$ as the center, a 3×3 neighborhood of blocks (as shown in Fig. 6.6 (a)) is considered first, which shows the possible linear edges that can form the right curve with the center being $B^{(i,j)}$ block. These blocks are ranked as p^u or p^d where $p = 2, 3$, and u and d indicate up and down (with respect to $B^{(i,j)}$). p indicates the rank of the block, i.e. the order in which it will be processed when going up or down to form the right curve. These ranks were decided based on the manual inspection of human images in datasets like Buffy dataset [214], CASIA dataset [215]. It was seen that the right shoulder tends to be flat (or horizontal) in block $B^{(i,j+1)}$ as compared to slanting further up in block $B^{(i-1,j+1)}$. Therefore, $B^{(i,j+1)}$ is given as higher rank, i.e. 2^u, as compared to $B^{(i-1,j+1)}$ (ranked 3^u). Similar observations can be made about the blocks below $B^{(i,j)}$ to rank them as shown in Fig. 6.6 (a).

Therefore, given $B^{(i,j)}$ and the anchor angle $\theta_a^{(i,j)}$, 2^u ranked block ($B^{(i,j+1)}$) is first checked for θ_ks in $\mathbf{\Theta}^{(i,j+1)}$ such that the following condition is satisfied: $\theta_k^{(i,j+1)} \leq \theta_a^{(i,j)} - \delta_2$ where δ_2 is the expected change in the gradient angle that should occur if the linear edge segment in the anchor block *curves* as we go towards the outer blocks, i.e. $B^{(i,j+1)}$. If there are multiple $\theta_k^{(i,j+1)}$ that satisfy the above condition, the θ_k which has the highest count in the GAH $\mathbf{h}^{(i,j+1)}$ is considered. These conditions will ensure the curvature condition defined in (6.5) is satisfied. If $B^{(i,j+1)}$ does not satisfy any of these conditions, then the next ranked block, i.e. $B^{(i-1,j+1)}$ which is ranked 3^u is considered next. This must also satisfy the same condition for θ_k but has a smaller δ_3 such that $\delta_3 < \delta_2$. This is because the linear edge in $B^{i-1,j+1}$ is slanting upwards more than $B^{(i,j+1)}$ and hence, it is expected to have a lesser gradient angle variation than $B^{(i,j+1)}$ (based on observation from datasets that was described above).

The same is repeated for blocks $B^{(i+1,j)}$ and $B^{(i+1,j-1)}$ that are below the anchor block, which are ranked 2^d and 3^d respectively. The angles must meet similar conditions as above but with a positive δ_2 and δ_3 because the gradient angles in blocks lower than the anchor block are higher than the anchor block (according to (6.5). If any of these conditions are not met in the 3×3 neighborhood of $B^{(i,j)}$, no further processing for $B^{(i,j)}$ is done and the next block is processed for identifying the anchor block.

After identifying the block in the 3×3 neighborhood of the anchor block, another check is performed with the blocks that surround this neighborhood. This is the second stage of processing. This is an optional step depending on user requirement in terms of the curvature constraint one wants to ensure. In our experiments, it was found that going for one more layer of blocks helped to reduce the false positives. Therefore, a 5×5 neighborhood around the anchor block is covered to ensure that a curve has been captured in it, if present.

The ranks of the blocks in the outer ring of blocks in the 5×5 neighborhood of the anchor blocks are marked as $p - q^u$ or $p - q^d$, where $p = 2, 3$ indicates the rank of the origin block in 3×3 neighborhood, $q = 2, 3$ indicates the rank of the current block. This is shown in Fig. 6.6 (a). For example, $B^{(i,j+2)}$ has the rank $2 - 2^u$, which implies that it could have a edge from block $B^{(i,j+1)}$ which was previously ranked as 2. Block $B^{(i-1,j+2)}$ has two ranks: $2 - 3^u$ and $3 - 2^u$. If the right curve is detected in $B^{(i,j+1)}$ in the first stage of processing, then $B^{(i-1,j+2)}$ is processed after processing $B^{(i,j+2)}$. If the right curve is detected in $B^{(i-1,j+1)}$ in the first stage of processing, then $B^{(i-1,j+2)}$ is processed first, followed by $B^{(i-2,j+2)}$. A similar approach is taken to rank the blocks in the lower half of the neighborhood as shown in Fig. 6.6.

Now, given the blocks identified in the 3×3 neighborhood of the anchor block $B^{(i,j)}$, which could potentially be having the right curve, these blocks in the 3×3 neighborhood are considered as the new anchor blocks. Therefore, the two new anchor blocks are B_a^u and B_a^d corresponding to the blocks above and below the main anchor block $B^{(i,j)}$. The above process of checking the GAHs with B_a^u and B_a^d is now repeated, using the ranks for the blocks in 5×5 neighborhood (discussed above). The parameters δ_2 and δ_3

are increased or decreased depending on the block that is being processed in a similar approach as described earlier for the first stage so as to satisfy (6.5).

If a curve is traced within the $k \times k$ window ($k = 5$ in our case), then a right curve is considered to be detected, which is anchored at block $B^{(i,j)}$. If there was any discontinuity while traversing from $B^{(i,j)}$ to its $k \times k$ neighborhood, further analysis of $B^{(i,j)}$ is terminated and the next edge block is considered for the entire analysis described above. This is repeated for all blocks in $\mathbf{\Theta}$ to detect anchor blocks that have either the right or the left curve. Maps $\mathbf{R}$ and $\mathbf{L}$ are generated, such that: $\mathbf{R} \in \mathbb{R}^{\frac{X}{b_s} \times \frac{Y}{b_s}}$ & $\mathbf{L} \in \mathbb{L}^{\frac{X}{b_s} \times \frac{Y}{b_s}}$ where an element in $\mathbf{R}$ say $\mathbf{R}(i,j)$ is set to 1 if an anchor block for a right curve at index (i,j) is found. Similarly $\mathbf{L}$ is defined.

6.3.2 Face Localization by Curve Association

Once all the valid left and right curves are recorded in $\mathbf{L}$ and $\mathbf{R}$, the next level of association is performed to identify regions in the image that could possibly have a front facing human. Recalling that our aim is to detect shoulder-head region which is defined within a $P_X \times P_Y$ pixels sized window, an *association window* is constructed by grouping $u \times v$ blocks, where $u = P_X/b_s$ and $v = P_Y/b_s$, where b_s is the block size. This is illustrated in Fig. 6.7 (a).

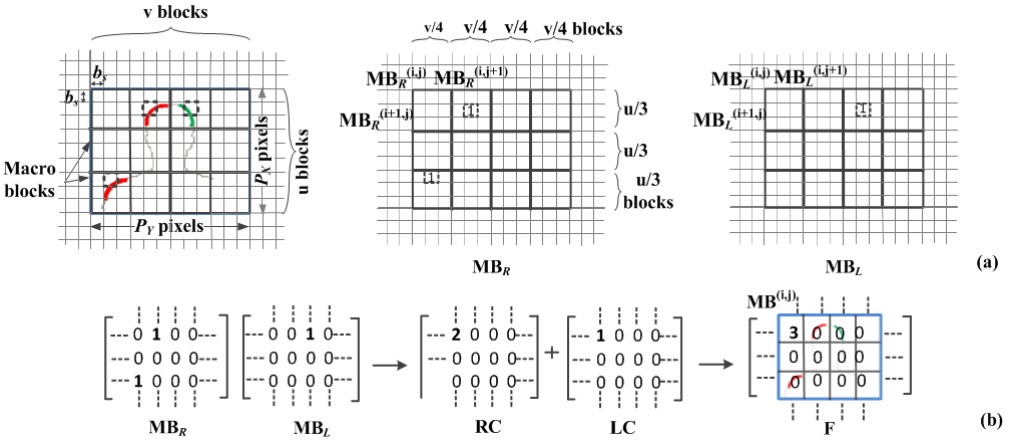


FIGURE 6.7: (a)Association of Right and Left Curves to detect possible shoulder-head regions. (b)Kernels $\mathbb{K}_R$ and $\mathbb{K}_L$ applied on the Macroblocks $\mathbf{MB}_R(i,j)$ and $\mathbf{MB}_L(i,j)$ resulting in Matrices $\mathbf{RC}$ and $\mathbf{LC}$ respectively, and eventually matrix $\mathbf{F}$

Having determined the number of blocks, i.e. $u \times v$, that fit the $P_X \times P_Y$ pixels window, *macroblock left and right arrays* are generated, denoted as $\mathbf{MB}_L$ and $\mathbf{MB}_R$ respectively. Macroblocks are created by grouping a set of blocks of size $b_s \times b_s$ within an association window. In order to compute every element in $\mathbf{MB}_R$ and $\mathbf{MB}_L$, denoted by $\mathbf{MB}_R(i,j)$ and $\mathbf{MB}_L(i,j)$, a group of $u/3 \times v/4$ elements is considered in $\mathbf{R}$ and $\mathbf{L}$, in a rastor scan method, as shown in Fig. 6.7 (a). Therefore, the dimensions of these macroblock arrays are given by: $\mathbf{MB}_R \in \mathbb{R}^{\frac{X}{\frac{u}{3}\times b_s} \times \frac{Y}{\frac{v}{4}\times b_s}}$ and $\mathbf{MB}_L \in \mathbb{R}^{\frac{X}{\frac{u}{3}\times b_s} \times \frac{Y}{\frac{v}{4}\times b_s}}$. Each element $\mathbf{MB}_R(i,j)$ and $\mathbf{MB}_L(i,j)$ is assigned the value 0 or 1 using the following equations:

$$\mathbf{MB}_R^{(i,j)} = \begin{cases} 0 & \text{if } \sum_{r=i\frac{u}{3}}^{(i+1)\frac{u}{3}} \sum_{s=j\frac{v}{4}}^{(j+1)\frac{v}{4}} \mathbf{R}^{(r,s)} = 0 \\ 1 & \text{otherwise} \end{cases} \qquad \mathbf{MB}_L^{(i,j)} = \begin{cases} 0 & \text{if } \sum_{r=i\frac{u}{3}}^{(i+1)\frac{u}{3}} \sum_{s=i\frac{v}{4}}^{(j+1)\frac{v}{4}} \mathbf{L}^{(r,s)} = 0 \\ 1 & \text{otherwise} \end{cases} \tag{6.7}$$

In other words, if any block in the group of $u/3 \times v/4$ blocks (which is one macroblock array element) has an anchor block (either left or right slant edge), 1 is assigned to that group (macroblock array element). Fig. 6.7 (a) & (b) shows an example of macroblock array assignment. It can be seen from Fig. 6.7 (a) & (b) that if a group of $u/3 \times v/4$ blocks has an anchor block, its corresponding position in macroblock array is assigned '1'.

Then, we define 3×4 kernels $\mathbb{K}_L$ and $\mathbb{K}_R$ in the following manner:

$$\mathbb{K}_L = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad ; \quad \mathbb{K}_R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.8}$$

which are convolved with $\mathbf{MB}_R$ and $\mathbf{MB}_L$ respectively as shown in Fig. 6.7 (d). Considering that the $\mathbb{K}$s are not square-odd matrix - which can be centered at the center of the kernel, the top left element of $\mathbb{K}$s, i.e. $\mathbb{K}(0,0)$s are aligned with every element of $\mathbf{MB}_L$ and $\mathbf{MB}_R$. The convolution output is placed on the top left corner element over which the convolution kernels are moved as shown in Fig. 6.7(b). The convolution

operation results in $\mathbf{RC}$ and $\mathbf{LC}$ matrices given by:

$$\mathbf{RC}(i,j) = \sum_{k=1}^{k=3}\sum_{l=1}^{l=4} \mathbf{MB}_R(i+k-1, j+l-1)\mathbb{K}_R(k,l) \tag{6.9}$$

$$\mathbf{LC}(i,j) = \sum_{k=1}^{k=3}\sum_{l=1}^{l=4} \mathbf{MB}_L(i+k-1, j+l-1)\mathbb{K}_L(k,l) \tag{6.10}$$

where $1 \leq i \leq \frac{X}{\frac{u}{3}b_s}$ and $1 \leq j \leq \frac{Y}{\frac{v}{4}b_s}$. The matrices $\mathbf{RC}$ and $\mathbf{LC}$ capture the number of right and left curves in the left and right halves of the $u \times v$ window respectively, as shown in Fig. 6.7 (b). The position of face localization window is then obtained by generating $\mathbf{F}$ matrix given by: $\mathbf{F} = \mathbf{RC} + \mathbf{LC}$. If $\mathbf{F}(i,j) \geq 3$, then it is considered that a window with its top left corner positioned at (i,j)-th position to have a face. This condition allows us to check for the presence of at least 3 of the 4 curves that form the head and shoulder curves. This is shown in Fig. 6.7 (b).

6.4 Performance Evaluation

In this section, the evaluation of the proposed search space reduction algorithm is presented. The algorithm is first evaluated for accuracy. Then, the amount of search space reduced. Then, the reduction in computational cost is presented.

6.4.1 Accuracy Evaluation

The detection rate of the proposed algorithm on two datasets is evaluated. The first dataset is the Biometric Database (CASIA Face Image Database - CASIA-FaceV5 300-399) with 500 images of 100 subjects [185]. This dataset contains five different images of size 640×480 of each subject; the subject being a front facing human in a constrained background set up. The variations among the images include angular movements of the person with respect to the camera, changes in illumination, and imaging distance. Considering that the algorithm is being demonstrated for one particular scale, 480 images from this dataset were chosen, which are of similar scale by manual inspection. The

second dataset is a subset of the Buffy Dataset [214]. This contains images with unconstrained backgrounds. 108 images of size 720×405 of front facing humans from this dataset were considered for the evaluation.

As discussed in Section 6.3.1, the proposed algorithm being a block based approach, parameters that can influence the detection accuracy for a given scale are block size (b_s), number of bins in the GAH, block-wise GAH threshold setting (T), maximum gradient angle change allowed between blocks while detecting the curves (δ_2 and δ_3 in Section 2). It was found that the block size (b_s) is a critical parameter that impacts the detection rate, and other parameters were tuned based on observations with respect to the scale. For example, T is set to a value proportional to b_s, $|\delta_2|$ and $|\delta_3|$ are set to 50° and 20° with respect to the scale of the human that has been considered for detection. However, these parameters can also be varied and their impact on detection rate can be studied. In the scope of this research, the effect of change in block size b_s on the detection rate will be evaluated and discussed.

First, the ground truth was generated, which includes marking a bounding box around the face in each image of the two datasets. During evaluation phase, if the association window resulting from the proposed algorithm has detected a true shoulder-head region and the window encloses the ground truth window, then it is considered as a True Positive (TP) window. In addition to TP windows, there can be false positive (FP) windows. If the proposed algorithm gives at least one TP window for an input image, then it is considered that the head-shoulder curve is correctly detected. Therefore, if there are n_{TP} images with at least one TP window, then the detection rate is given by n_{TP}/n_{total}, where the dataset contains n_{total} number of images.

Table 6.1 gives the detection rates for each dataset and different block setting. It can be seen that the detection rate is 99.5% for CASIA dataset at a block size of 6 and it is 94% for Buffy dataset at a block size of 8. The detection rates reduce for both datasets as the block sizes are increased. This shows that for a given scale of the humans in a dataset, a particular block size gives the highest detection rates. Also, other reasons for that impact the detection accuracy are - extremely low contrast between the intensity

TABLE 6.1: Accuracy analysis results

	CASIA				*Buffy*		
	$P_X \times P_Y = 150 \times 200$ $X \times Y = 240 \times 320$				$P_X \times P_Y = 150 \times 150$ $X \times Y = 405 \times 720$		
Block Size	6	8	12	16	8	12	16
Detection Rate	99.5	85.1	79.9	70.7	94.4	74.0	79.6
False Positives per frame	1.78	0.44	0.27	0.08	6.37	2.71	2.36
Number of windows per frame	3.32	1.44	1.30	0.71	8.15	3.74	3.66

of the clothing worn by the subject and the surrounding background, and occlusion of shoulder by hair.

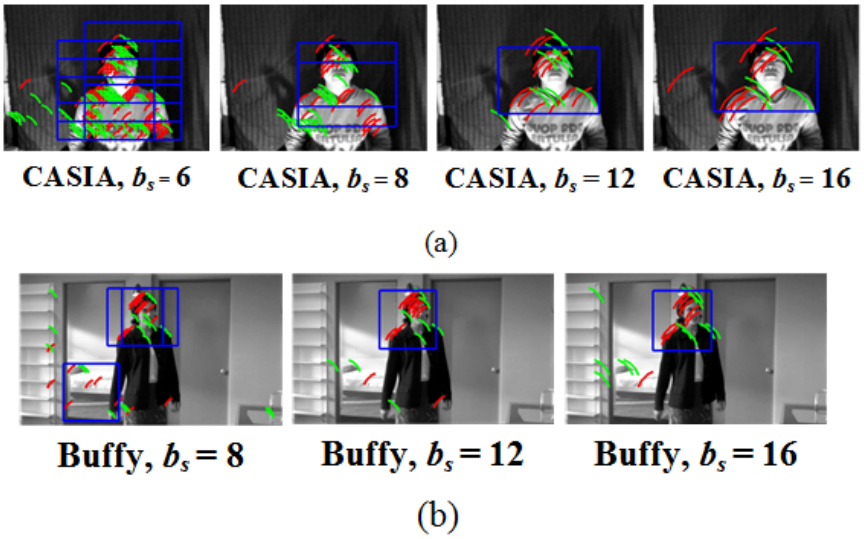


FIGURE 6.8: Detection windows resulting from the proposed algorithm as the block size increases for a specific image in CASIA dataset (a) and Buffy dataset (b).

In Table 6.1, the average total number of windows per frame for each dataset has been included. For each input image, the total number of windows, i.e. all TP windows and FP windows are counted. Table 6.1 gives the average number of windows over the entire dataset. This metric is particularly important to determine the amount of image area that needs to be searched by the face detection algorithm. Fig. 6.8 (a) & (b) show that there are more number of windows detected when the block sizes are smaller as compared to higher block sizes and that the FP windows reduce as the block sizes increase. Fig. 6.9 shows more examples of correct detection windows by the proposed algorithm under

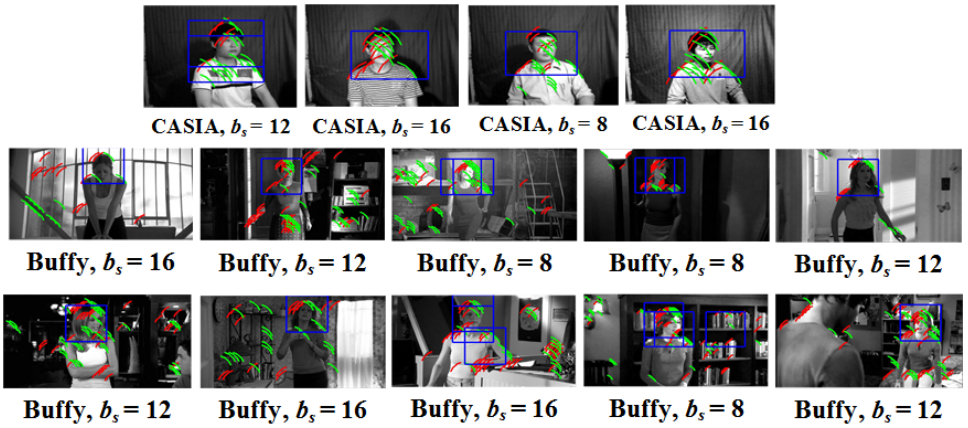


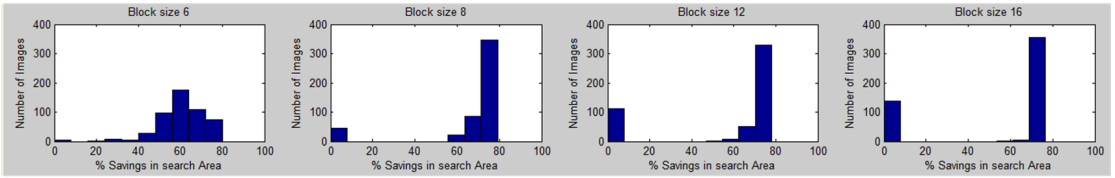
FIGURE 6.9: detection results under varying background conditions and complexities.

varying backgrounds and complexities of the input images. Some images in the last row of Fig. 6.9 show that FP windows may increase in the presence of background clutter. In Table 6.1, the average false positives per frame (FPPF) for each dataset under varying block sizes has also been listed. It can be seen from Table 6.1 that the FPPF is less than 2 and 6.5 for CASIA and Buffy datasets respectively. In other words, an average of 2 false positive windows are detected per frame in the case of CASIA dataset. This number is about 6 in the case of Buffy dataset. The number of false positive windows increases in Buffy because of the unconstrained backgrounds which have more variations as compared to CASIA dataset. In both cases, the FPPF is highest for the smallest block size.

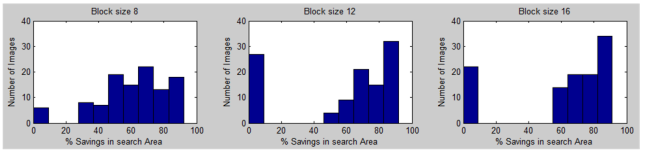
6.4.2 Computational Cost Analysis

6.4.2.1 Search Space Reduction

The percentage savings in search area for face detection is determined. If an image has at least one TP window, the total search area for face detection is the union of all the detection windows, which include both TP and FP windows. The ratio between the total image size ($X \times Y$) and the total number of pixels in the search area enclosed by this union of detection windows is used to determine the percentage savings for each image. This percentage savings in search area directly corresponds to a proportional decrease in cost of conventional sliding window based face detection.



(a)



(b)

FIGURE 6.10: Distribution of search space savings: (a) CASIA (b_s = 6, 8, 12 and 16 (from L to R)) (b) Buffy (b_s = 8, 12, 16 (From L to R)). x-axis: $\%$ savings in search area, y-axis: Number of images

We show the distribution of percentage savings for each dataset under different block size settings in Fig. 6.10 (a) & (b). There is a cluster of distribution around 0%, corresponding to the missed detections, i.e. no TP windows. The second cluster is seen at a higher percentage indicating images where at least one TP window was detected.

Referring to Fig. 6.10 (a) which shows the distributions for CASIA dataset, we observe that the percentage savings in search area is spread across bins ranging from 40% to 80% for a block size setting $b_s = 6$. This shows that with this block setting, there are false positives along with the true positive windows, but overall detection rate is high since there are very few missed-detections (the cluster around zeroth bin the histogram is less than 10). Block sizes of 8, 12 and 16 show a high concentration of the count in the histogram at 80% savings, which means that they result in very low false positive rates and high precision. But, the bin at 0% is also populated for the three block sizes, which accounts for the cases that are missed-detections under these block size settings. Similar observations can be drawn from the histograms for the Buffy dataset as shown in Fig. 6.10 (b) for different block size settings.

6.4.2.2 Reduction of Computational Cost

In this section, the computational complexity of the proposed method is evaluated and the cost reduction is evaluated. Existing face detection algorithms are implemented on

different platforms. Therefore, in order to perform an equivalent comparison, the complexity in terms of equivalent operations is compared, so that the complexity evaluation is across implementation platforms. The equivalence in operations that will be used in this section is derived from [218].

Consider an image I of size $X \times Y$, and a block size of $m \times n$ (i.e, $b_s \times b_s = m \times n$), within which the block-wise GAH is computed. Then, the number of blocks are $X/m \times Y/n$. The following is the calculation of the computational cost involved in the proposed search space reduction algorithm.

(a) Extraction of edge map and computation of GAH: Application of sobel kernel to a pixel results in 10 additions. So, considering the entire image I, it results in $10XY$ additions. Gradient computation involves 2 multiplications and one square root operation per pixel ($\sqrt{(G_x)^2 + (G_y)^2}$), where G_x and G_y are the gradients along x and y directions). A square root operation is considered as a CORDIC [216], [217] operation. Extraction of edge map involves XY comparisons, for selecting the gradient values that cross the threshold. Then, the block-wise GAH computation for the edge pixels results in e_dXY division operations (or equivalent multiplication operations) to compute G_x/G_y for every edge pixel, e_dXY trigonometric operations (for the gradient angle computations) and e_dXY accumulation operations (for the histogram accumulation operations), where e_d is the percentage of edge content in the image.

(b) Extraction of candidate curves from the edge map: In the process of extracting the line segments within each block, $X/m \times Y/n \times n_b$ comparisons are performed to check if the accumulation at each bin of the GAH has crossed a threshold (n_b is the number of bins in the histogram). The threshold is computed relative to block size, hence it involves one multiplication operation. The computational cost of locating the blocks with the right and left slant edges and checking within the neighbouring blocks for segments that can form the right or left curve is computed next. If it is considered that each of the $X/m \times Y/n$ blocks have either a right or a left slant edge, then it results in $10 \times X/m \times Y/n \times$ comparisons (because a maximum of 10 blocks are considered in the neighbourhood of the anchor block while checking for the presence of line segments forming the curve around the anchor block).

(c) Association of right and left curves: The computational cost of associating the potential head and shoulder candidates within the association window is computed next. Within the association window of size $P_X \times P_Y$ pixels, a maximum of 3 additions are incurred during the convolution with the kernels K_R and K_L, and one comparison to check if the convolution sum is greater than or equal to 3. If the association window is scanned with a step size of $P_X/4$ and $P_Y/4$ along the x and y directions respectively, the number of additions and comparisons are $3(X/(P_X/4) * Y/(P_Y/4))$ and $X/(P_X/4) * Y/(P_Y/4)$ respectively.

The above computations can be summarized as follows:

$$\begin{aligned} C_{ADD} &= 10XY + 3(X/(P_X/4) * Y/(P_Y/4)) + e_d XY \\ C_{TRIG} &= e_d XY + XY \\ C_{MUL} &= 2XY + e_d XY + 1 \\ C_{COMP} &= XY + n_b XY/mn + 10X/m * Y/n + X/(P_X/4) * Y/(P_Y/4) \end{aligned} \tag{6.11}$$

Next, the computational cost incurred in running the Viola Jones face detection [180] on the entire input image is computed using the same metrics that was used to compute the computational cost of the proposed method. At the end of the first stage of the Viola Jones technique, nearly 50% of the non-face sub-windows are rejected, and only the sub-windows that qualify at this stage are passed on to the second stage [180]. The computational complexity analysis will be done considering only the first stage of the cascade for classifier computation. For an image of size face detection on a $X \times Y$, the integral image computation results in $2XY$ additions. In the first stage of the classifier, in every sub-window of size $m \times n$, the two-feature classifier are computed, and the corresponding computational cost is as follows - $X/p \times Y/p(m \times n) \times [2 \times (3\ adds, 1\ muls) + 3 \times (3\ adds, 1\ muls) + 1\ add + 1\ comp]$ considering that the sub-window is moved with a step-size of p pixels (note: add stands for addition, mul for multiplication and $comp$ for comparison operations). The equation (6.12) summarizes

the additions involved in obtaining the integral image and equation (6.13) summarizes the computations involved in obtaining the classifiers.

$$C^1_{ADD} = 2XY \tag{6.12}$$

$$\begin{aligned} C^2_{ADD} &= 16mn(X/p * Y/p) \\ C_{MUL} &= 5mn(X/p * Y/p) \\ C_{COMP} &= mn(X/p * Y/p) \end{aligned} \tag{6.13}$$

The above equations in (6.13) are formulated considering that the sub-windows are scanned across the image with a step-size of p pixels. All of the above computations are summarized in Table. 6.2.

TABLE 6.2: Summary of computations in proposed method and Viola-Jones face detection technique (up to the first stage classifier applied)

Operations	Viola-Jones [180]	Proposed method
Additions	$2XY + 16mn(X/p * Y/p)$	$10XY + e_dXY + 3(16XY/P_xP_y)$
Trigonometric Operations	0	$e_dXY + XY$
Multiplications	$5mn(X/p * Y/p)$	$2XY + e_dXY + 1$
Comparisons	$mn(X/p * Y/p)$	$16XY/P_xP_y + (n_b + 10)XY/mn + XY$

Now, the computational cost of applying the conventional face detection to the entire image is compared with applying the same only within the windows shortlisted by the proposed method. Consider an image of size 350×200.

(a) Conventional method: The cost of applying the Viola-Jones face detection technique, considering a moving sub-window of size 16×16 pixels scanned with a gap of 8 pixels, and up to the first stage classifiers is computed. The computational cost is tabulated in the second column of the Table. 6.3. An n-bit multiplication is considered equivalent to n n-bit additions [218].

TABLE 6.3: Comparison of computational cost of the conventional Viola-Jones technique and the proposed method (considering an image of size 360×200.)

Operations	Conventional method [180]	Proposed method	Proposed method + conventional method in shortlisted windows
Additions	4620000	707896	3087896
Trigonometric Operations	0	77000	77000
Multiplications	1400000	147001	847001
Comparisons	280000	77955	217955
Total computations (equivalent additions)	27300000	4369867	18089867
Percentage savings			34%

(b) Proposed method: If the same image is divided into blocks of size 16×16 and the percentage of edge content in each block is assumed to be 10% of the total number of pixels in the block [219], the resulting computational cost of the proposed algorithm is calculated. 18 bins for GAH is considered. The size of the association window considered is 50×75 pixels. The computational cost is tabulated in the third column of Table. 6.3.

(c) Proposed method followed by conventional method: The computational cost of applying conventional face detection within the shortlisted windows is computed and presented in the last column of the Table. 6.3. Based on the results presented in Sec. 6.4.2.1, a search space reduction of up to 80% (an average of 70%) is achieved by the proposed technique. Let us consider a worst case condition of the shortlisted windows covering 50% of image area. The cost of the proposed method, followed by face detection within the shortlisted windows is computed. The integral image computation that was computed considering the entire image and the cost involved in classifier computations within the shortlisted windows is computed. Computational cost savings of 34% is achieved if a worst case of 50% reduction in search space is considered, as shown in Table. 6.3.

6.5 Summary

In this chapter, an effective strategy to reduce search space for face detection in a controlled setting, taking advantage of an on-bed patient monitoring scenario is proposed. The block-based nature of the approach allows for performance gains through parallelism. Evaluation of the algorithm on two standard datasets, CASIA frontal face and Buffy Stickmen was shown to yield a reduction of search space by upto 80% of the image area (an average reduction of 70%). It was established that optimal block settings can be derived for a given scale of humans in the image, depending on the required accuracy and % reduction in search area. It was shown that the method can cater to varying scales of human faces by deploying it iteratively or by combining GAH information in a hierarchical manner. The method was shown to perform well for profile view of persons on the datasets considered. The computational cost savings that result by applying the proposed search space reduction technique as a pre-processing step for face detection were computed, and savings of nearly 34% is achieved.

In the chapters thus far, techniques to localize the face and facial features - eyes, mouth and brow furrows, and extract wellness indicators based on their analysis have been proposed. The next chapter will bring these wellness indicators together in the proposed framework for wellness assessment.

CHAPTER 7

Integrated Framework for Assessing Patient Wellness

7.1 Introduction

In this chapter, a framework for assessing the wellness of a patient is presented. The following are the objectives of the proposed framework for wellness assessment:

- To serve as an assistive system that takes the inputs of the doctor or nurse and visually monitors the wellness of the patient based on the temporal analysis of a set of facial features.
- To continuously assess the wellness of a patient relative to an *initial state, desired* and *undesired* states (as defined by the doctor or nurse), and monitor if the patient is improving, worsening or shows no change in his condition relative to the initial, desired and undesired states over a period of time.

The techniques for extraction of facial features and the associated wellness indicators, and patient face localization discussed in Chapters 3, 4, 5 and 6 are brought together to realize the objectives of the framework.

The term ‘wellness’ is first defined. Then, the proposed framework is presented along with a brief description of the various blocks in the framework. Following this, the process of estimating wellness indicators is described, followed by the determination of wellness state and the wellness assesment. Then, an example scenario is presented to illustrate the wellness assessment process.

7.2 System Overview: Wellness Assessment

The term ‘wellness’ is defined first, following which the proposed wellness assessment framework is presented.

7.2.1 Defining Wellness

As seen from Sec. 2.5.5, the term ‘wellness’ can be defined in many ways, and different aspects of wellness have been considered by different groups while defining ‘wellness’. We define the patient’s wellness state at an instance of time to be described primarily by one of the following states: asleep, awake, drowsy, inactive and discomfort. Then, wellness is assessed using the wellness state determined over time. The wellness assessment will provide the doctor with:

- a continous assessment of whether the patient’s condition has improved, worsened or has been the same relative to certain reference states, accompanied by an *trigger* when necessary,
- a wellness profile based on the history of wellness assessment showing how rapidly or gradually the patient’s condition has altered and how its severity has changed over time.

Thus, we define wellness as a combination of the patient’s wellness state at a point in time and the wellness profile assessed over time.

The reference states used are: (1) the patient's wellness state at the start of assessment, which is also called the *initial wellness state* $\mathbf{s}^{(\mathbf{t_0})}$ or *initial state*, and (2) *desired* state D and *undesired* state U as defined by the doctor based on the patient's initial wellness state. The initial wellness state is first assessed by the system. Then, the assessed initial state may be modified based on the doctor's inputs. Desired state is the state the doctor would *expect* or *desire* to see the patient in, and undesired state is the state the doctor expects the patient to *not be in*. The doctor is alerted when the patient's wellness state crosses certain thresholds while moving towards the desired or undesired states. At this point, the *wellness profile* is generated for the doctor's reference. The wellness profile can also be generated whenever the doctor wishes to see it.

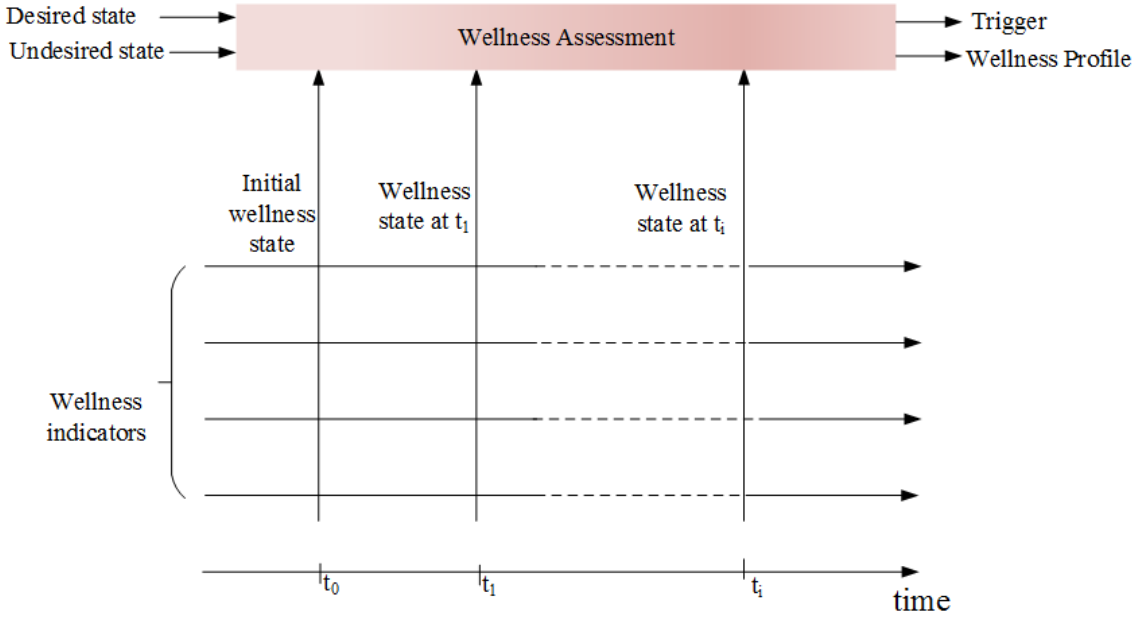


FIGURE 7.1: Illustration of wellness states determined at the different instances of time and the wellness assessment with respect to initial, desired and undesired states, resulting in a trigger and a wellness profile.

In this work, five states namely, awake, asleep, drowsy, discomfort and inactive that can define the wellness state of the patient have been considered. The confidence measures of these states at an instance of time t are computed as follows. The wellness indicators extracted over a window of time just prior to t_i, based on Chapters 4 and 5, such as - eye state over time, blink, eyeball movement, mouth kept open etc. are normalized to interpret their intensity. Some examples of interpretation of wellness indicators are - blink being low or high, mouth kept open for a long duration, or eyes kept open

without much eyeball movement, etc. The normalized wellness indicators are combined to compute the confidence measure of these five states. Then, based on the confidence measures, the most predominant of these states determines the patient's wellness state at that instance of time.

Fig. 7.1 shows wellness indicators extracted continually along the time axis, wellness state determined at different instances of time t. The wellness state $\mathbf{s}^{(\mathbf{t_i})}$ at the current instance of time which is determined as described above, is input to the wellness assessment block. The other inputs to this block are - the desired state D and undesired state U. Wellness assessment is done based on the patient's current wellness state $\mathbf{s}^{(\mathbf{t_i})}$ relative to the initial state $\mathbf{s}^{(\mathbf{t_0})}$ (at time t_0), desired and undesired states. This is for every instance of time t. The outputs of the wellness assessment block are - a trigger based on the wellness assessment, and the wellness profile.

In the above description of wellness assessment, three reference states have been used - the initial state (at t_0) and the desired and undesired states. The reason behind considering three reference states is explained as follows.

Consider a case where (1) wellness was to be assessed entirely based on the changes in current state relative to the initial state, and (2) the presence of the states, namely, drowsiness, inactivity and discomfort were considered signs of unwellness. However, if a patient is detected to be drowsy, it may not necessarily mean he is *unwell*. If the patient has been injected with a drug which would make him drowsy, then, drowsy is a state that would be *expected* by the doctor. This shows that the initial state alone is not sufficient to make the complete wellness assessment.

Hence, a second and third reference state became necessary to make the wellness assessment complete. The 'desired' and 'undesired' states were considered as the second and third reference states. So, taking the above example, if the doctor inputs the desired state indicating that the patient would be drowsy, then the wellness assessment will show that the patient is in a state in line with what he was expected to be in.

The doctor may also modify the initial wellness state determined by the system based on the patient's age or health condition. Let us take an example - a patient's initial wellness

state has been determined to be *inactive* using default parameter bounds. If the doctor feels that the patient is not in an inactive state, then, he may modify the initial state to *not inactive* or may modify the level of *inactivity* that was determined by the system. In this case, the parameter bounds with respect to the *inactive* state will be modified accordingly for that patient.

Having defined the term wellness and provided an overview of the process of wellness assessment, the wellness assessment framework is presented next.

7.2.2 Wellness Assessment Framework

The schematic of the proposed framework for assessing wellness is presented in Fig. 7.2. The video frames of the patient under on-bed monitoring are input to the system. The input frames are processed to first localize and detect the face (as explained in Chapter 6), detect the eyebrows (as explained in Chapter 3) to localize the facial features, then extract them at frame level, followed by extraction of wellness indicators over a local window of time (as explained in Chapters 4 and 5). The wellness indicators are normalized between minimum and maximum bounds, in order to estimate wellness indicators. The wellness indicators are stored in a memory buffer. They are combined to compute the condifence measure of the states, namely asleep, awake, drowsy, inactive and discomfort which define the wellness state of the patient. The wellness state determined at each instance of time are stored in a memory buffer. Then, wellness is assessed to check if the patient's condition has improved, worsened or has been the same relative to the initial, desired and undesired states, accompanied by a *trigger* if the thresholds are crossed, and a wellness profile based on the history of wellness assessment is provided.

The framework for wellness assessment is explained in detail in the following paragraphs:

- **Face Localization and Detection:** A front-facing camera placed within a distance of 4 feet from the patient under on-bed monitoring captures the frames of the patient. The degree of head rotation considered is $+/-15^{o}$. As the video

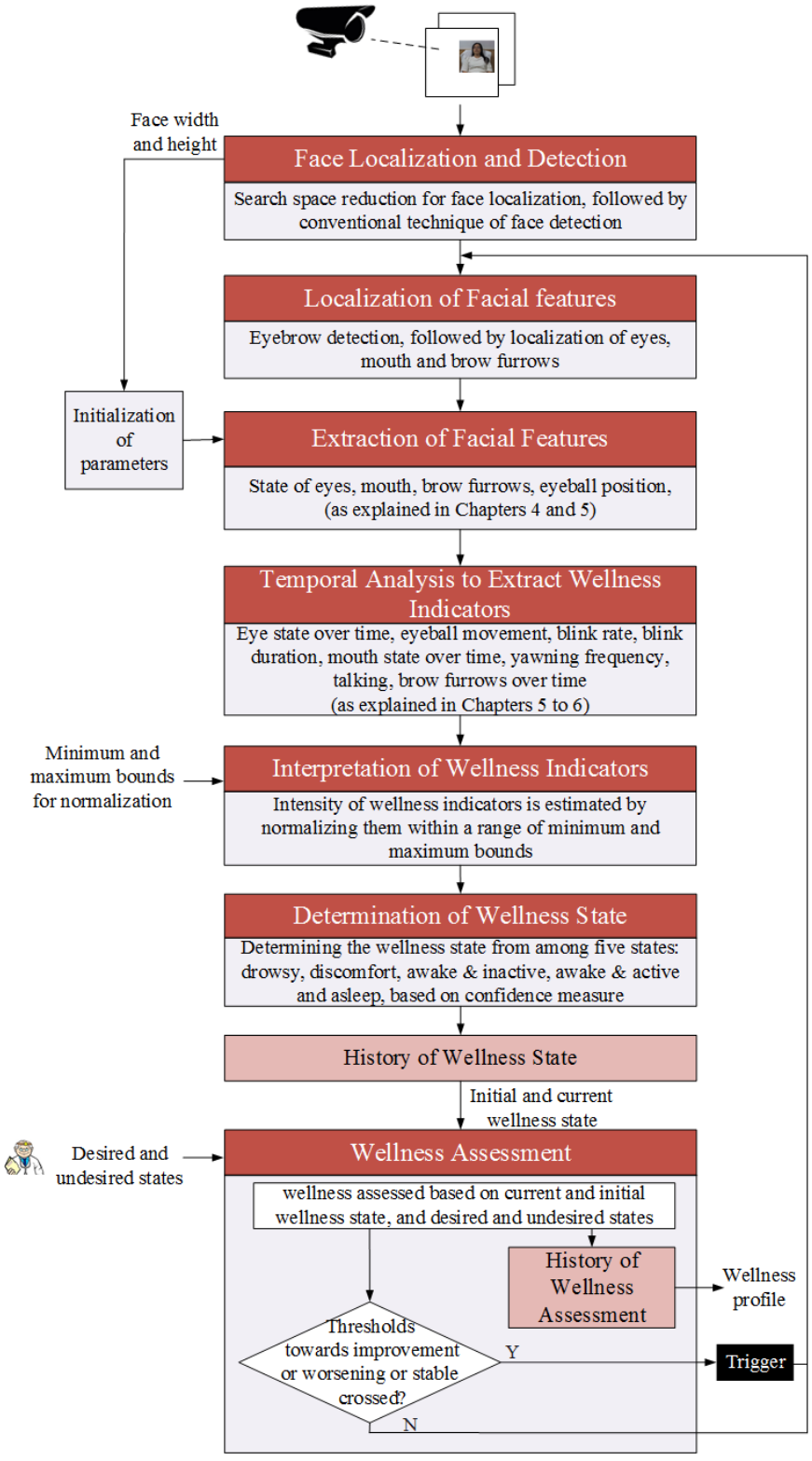


FIGURE 7.2: Schematic of the proposed framework

frames of the patient input to the system, search space reduction for face localization (as explained in Ch. 6) is followed by a conventional face detection technique

to detect the patient's face.

- **Localization of facial Features:** Once the patient's face is detected in the frame, the eyebrow detection (as explained in Ch. 3) is applied on the detected face. Using the eyebrows as anchor points, the regions of interest for eyes, mouth and brow furrows is estimated.

- **Extraction of Facial Features:** This step involves detection of the states of the facial features - eyes, mouth and brow furrows and the eyeball position within the estimated regions of interest (as explained in Ch. 4 and 5) in every frame. The facial features extracted in this work - eyeball position, eye state, mouth state and brow furrows are summarized in Fig. 7.3. Each incoming image are size and intensity-normalized with respect to the reference image.

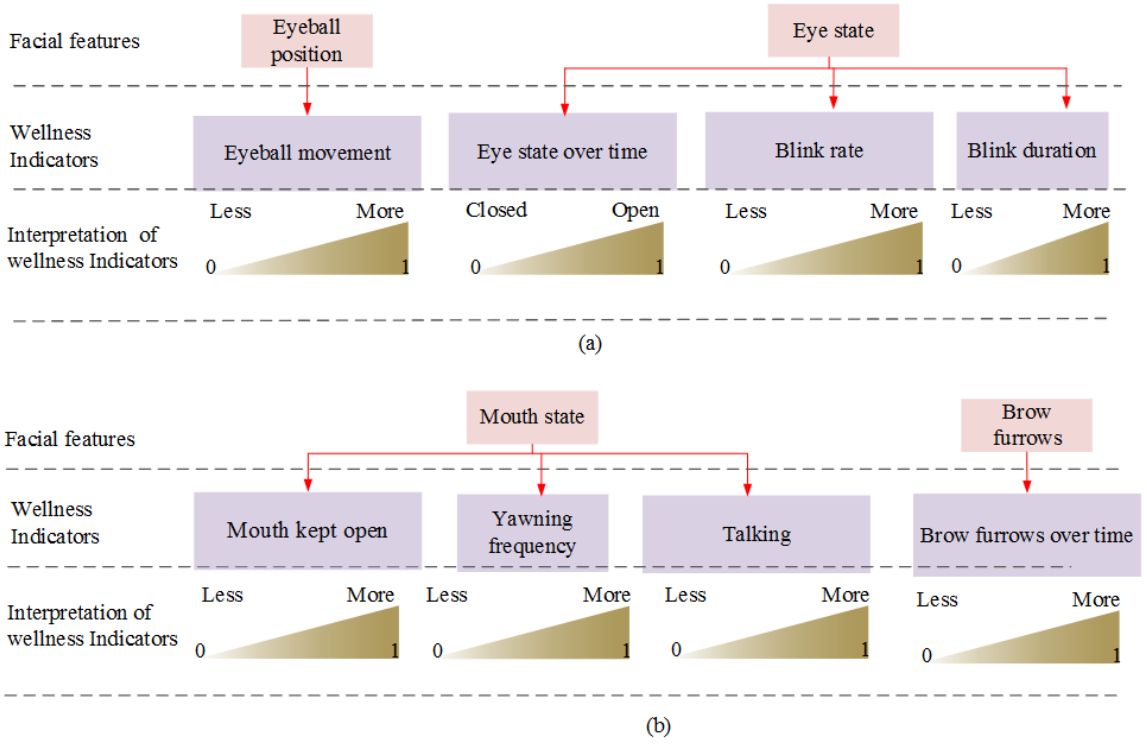


FIGURE 7.3: Facial features and wellness indicators related to the (a) eye, (b) mouth and brow furrows, and their associated interpretation

- **Initialization of parameters:** During the localization and extraction of the facial features as discussed above, initial thresholds are set at the start of detection. The input to the 'initialization of parameters' block is the width and height of the face detected.

- Thresholds for the length and width of the eyebrow, the eyeball eye, size of the lip and approximate distances between the eyebrows and eyes, nostrils and eyebrows, are set to an initial value based on heuristic analysis of anthropometric measurements using the width and height of the face.
- Other thresholds such as those for eye state detection are set after learning the parameters for the open and closed states in the initial p frames. A similar process is followed for the brow furrow detection in order to detect if the patient has brow furrows as intransient features.

During the initialization, if certain facial features are not detected either due to extremely low contrast between features and skin intensities or due to the width or length of the features being comparatively very low or very high, then such missed detections across p such frames will prompt the system to send a alert, and manual inputs are used to modify the associated thresholds.

Another example where manual intervention is needed is when the eyes of the patient at the initialization phase are partially open. The system would by default learn the parameters of partially open eye as fully open state, which would be incorrect. If manual input is given to the system at this point, then the system computes the thresholds for fully open eye accordingly.

- **Temporal analysis to Extract Wellness Indicators:** The wellness indicators are extracted through the temporal analysis of the facial features (as discussed in chapters 4 and 5). The wellness indicators extracted are: eyeball movement, eye state over time, blink rate, blink duration, mouth kept open, yawning frequency, talking and brow furrows over time. The wellness indicators extracted from the facial features related to eye, mouth and brow furrows are summarized in Fig. 7.3. The wellness indicators at time t are extracted within a local window of time $[t - w,\ t]$, with the windows spaced at an interval w.

- **Interpretation of Wellness Indicators:** The wellness indicators are intrepreted to infer their intensity or level. Some examples of intrepretation of wellness indicators are:

  - eye is kept partially open

- eyeball movement is very less
- blink rate is very high
- mouth is kept open for very long
- blink duration has increased
- yawning frequency has increased

Referring to Fig. 7.3, we have ranges for e.g. less to more, closed to open that are used to measure the intensity of the wellness indicator. In order to measure the intensity of the wellness indicators, they are normalized using unity-based normalization. Maximum and minimum bounds are used in normalization. The normalized values are indicative of where the absolute value of the wellness indicator lies in the range of minimum to maximum. This step of interpretation of wellness indicators is explained further in Sec. 7.3.

- **Determination of Wellness State:** The normalized wellness indicators are combined to determine the *wellness state* $\mathbf{s}^{(\mathbf{t_i})}$ at the current time instance t_i. First, this involves the computation of *confidence measure* of the states: asleep, awake, drowsy, inactive and discomfort. Based on the confidence measure, the most predominant state determines the patient's *wellness state*, and is also called the primary state.

  A voting system is used while combining the wellness indicators in order to compute the confidence measure of the states. Fig. 7.4, the *voting scale* shows what intensity or level of a wellness indicator votes higher for a state. For e.g., in Fig. 7.4, the voting scale shows that more the eye is open, more is the vote for awake state. Weighing constants α are used to prioritize the influence of the wellness indicators when voting for these states, and these are provided as inputs to this block. In Fig. 7.4, α_1 and α_2 are the weighing constants used.

  An example of wellness indicators, which when combined would vote *for* an *awake* state is shown in Fig. 7.4. Eye state over time and talking are used as the wellness indicators for computing the confidence measure of awake state. Longer the duration the eye is kept open within the window it is measured, higher is the

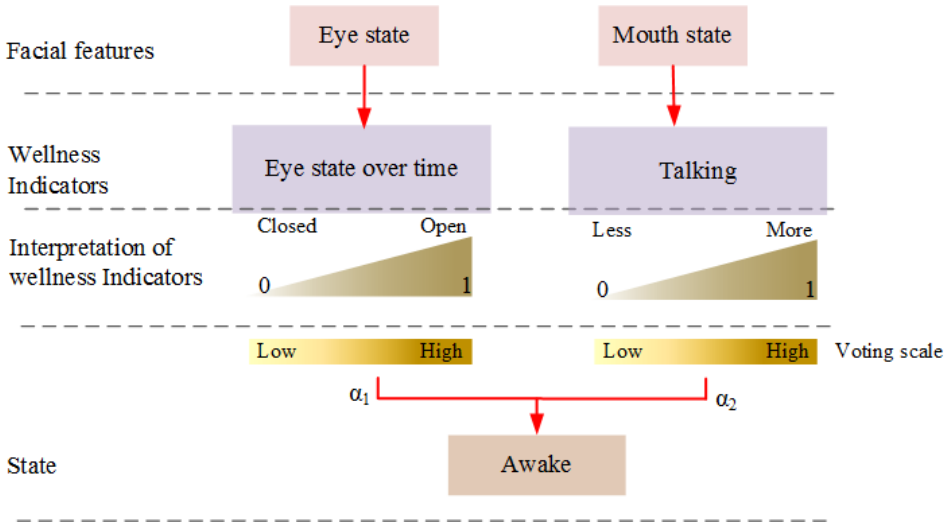


FIGURE 7.4: Illustration of how states are extracted starting from facial features, taking the example of *awake* state; α_1 and α_2 are the weighing constants

vote on the voting scale for awake state. More the patient is detected to be talking, more is the vote on the voting scale for awake state. Weights α_1 and α_2 are the weights assigned to eye state over time and talking when they vote for awake state. This step of determination of wellness state is explained further in Sec. 7.4.

- **History of Wellness State:** The wellness state determined at each current time instance t_i are stored in a memory buffer which are used in wellness assessment.

- **Wellness Assessment:** Wellness is assessed based on the shift in patient's current wellness state $\mathbf{s}^{(\mathbf{t_i})}$ with respect to the initial, desired and undesired states. The initial and current wellness state are retrieved from the memory buffer and input to the wellness assessment block. The overall improvement or worsening or stability of the patient's current wellness state $\mathbf{s}^{(\mathbf{t_i})}$ with respect to initial, desired and undesired states is assessed. The history of the wellness assessment is stored in a memory buffer which is called *history of wellness assessment* in Fig. 7.2. The doctor is alerted through a *trigger* when the patient's current wellness state approaches the desired or undesired states and crosses the thresholds that are set and a wellness profile based on the history of wellness assessment is provided to the doctor, accompanying the assessment as explained in Sec. 7.2.1. This step of wellness assessment is explained further in Sec. 7.5.

An overview of the various functional blocks of the framework was provided thus far. The functional blocks, namely - face localization and detection, extraction of facial features and the temporal analysis to extract wellness indicators have been discussed in detail in Chapters 3 to 6. The functional blocks, namely, interpretation of wellness indicators, determination of wellness state and wellness assessment are discussed in detail in this chapter in the sections to follow.

7.3 Interpretation of Wellness Indicators

In the chapters 4 and 5, techniques to extract wellness indicators were presented, which were the actual values of blink rate, blink duration, eyeball movement, etc. The wellness indicators that are extracted from the eyes, mouth and brow furrows have been summarized in Fig. 7.3.

In order to estimate the intensity of the wellness indicators, a reference is necessary. This is achieved by providing maximum and minimum bounds for each of them. We then normalize the wellness indicators using unity-based normalization. The normalized values of wellness indicators are indicative of where the actual value of the wellness indicators lies is the range of minimum to maximum. Next, by setting thresholds to this range, the normalized values are discretized for e.g. as high/medium/low and these discretized values are used to create a profile of the wellness indicators. The discretization enables the doctor to easily infer the profile of the intensity of the wellness indicators over time.

An example to illustrate the interpretation of wellness indicators is presented in Fig. 7.5 taking blink rate as an example, where (a) occurrence of blinks plotted along time-axis which is divided into 3 windows. The number of blinks within each window is computed and the derived blink rate is indicated, and in (b) the corresponding normalized values plotted taking the range as [0 60].

The normalization of wellness indicators is described as follows:

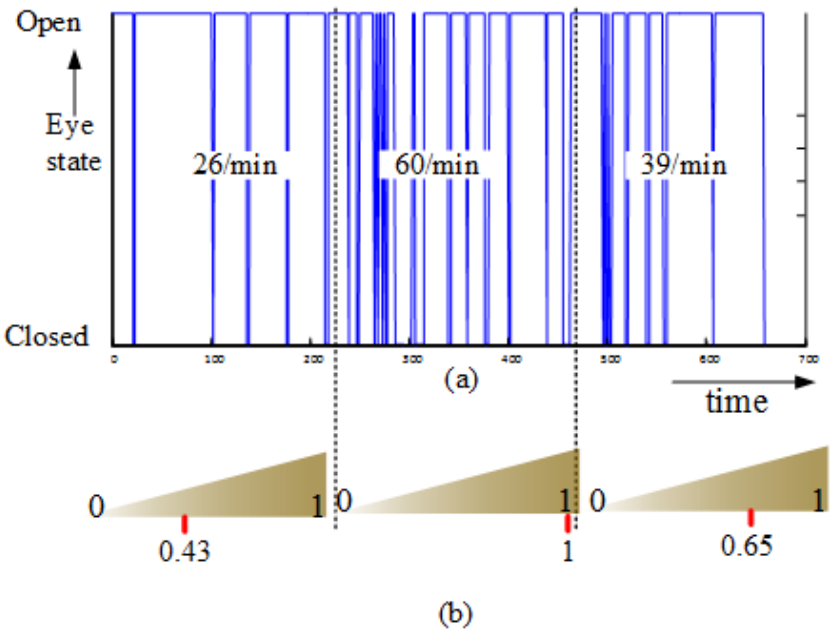


FIGURE 7.5: Illustration of the interpretation of wellness Indicators, taking the example of blink rate (a) Occurrence of blinks plotted along time-axis, (b) The corresponding normalized values plotted taking the range as [0 60]

Let $q^{(t)}$ be the measured value of a wellness indicator over a window $[t-w,\ t]$ at t, and q_{min} and q_{max} be the preset minimum and maximum bounds for that wellness indicator. Then, the normalized wellness indicator $\gamma_q^{(t)}$ for $q^{(t)}$ is computed as:

$$\gamma_q^{(t)} = (q^{(t)} - q_{min})/(q_{max} - q_{min}) \tag{7.1}$$

The above is done for all the wellness indicators. The minimum and maximum bounds given as inputs to the system are pre-defined based on literature or extracted from data. However, the doctor or expert can modify them for a given subject. The setting of the minimum and maximum bounds for the normalization of each of the wellness indicators is discussed in the following paragraphs:

Normalization of Wellness Indicators from Eyes

1. *Eyeball movement*: Eyeball movement is extracted as explained in Sec. 4.3.3. The minimum and maximum bounds for eyeball movement $\beta^{(t)}$ are set based on

empirical analysis. The subjects of the WellCam database were asked to simulate low and high eyeball movement, and the average $\beta^{(t)}$ across subjects was used to set the bounds for eyeball movement.

2. *Eye state over time*: Eye state over time is extracted as explained in Sec. 4.3.1. For eye state over time ${S_W}^{(t)}$, the minimum and maximum bounds are set based on the appearance of the eye in closed and open states respectively. The minimum bound is set based on closed state, because eye openness is zero in closed state. For setting the maximum bound, the following two cases are possible:

   Case 1: Patient is awake during initialization. During the initialization step, the height of the open eye detected across p frames is taken as fully open eye size, unless the doctor chooses to change the observation to partially open.

   Case 2: Patient is asleep during initialization. Since the eye will be closed during the initialization step, the height of the open eye is initialized either based on patient history or anthropometric estimations of height of open eye. While extracting the wellness indicator for eye state, the most predominant eye state within the local window is taken as $S_W^{(t)}$.

3. *Blink rate*: Blink rate is derived from blink detection as explained in Sec. 4.3.2. The minimum and maximum bounds for blink rate $R^{(t)}$ are set based on literature. The blink rate when a person is resting lies in the range of 8 to 21 blinks per minute and can go up to 19 to 26 blinks per minute when the person is engaged in conversation [220]. So, the minimum and maximum bounds are set to 0 and 40 respectively.

4. *Blink duration*: Blink duration is derived from blink detection as explained in Sec. 4.3.2. The minimum bound for blink duration $b_D^{(t)}$ is set based on literature to about 150 to 300ms (5 to 10 frames while 30fps). The maximum blink duration bound is set to 1 second.

Normalization of Wellness Indicators from Mouth and Brow Furrows

1. *Mouth kept open*: Mouth state over time is extracted as explained in Sec. 5.4.1.2. The bounds for mouth kept open $d_{mo}^{(t)}$ are computed based on the duration for

which the mouth is kept *open*. The minimum bound for mouth kept open is 0 and the maximum bound is equal to the duration for which the indicator was measured w. So, the longer the duration for which mouth is kept open, higher will be the estimated intensity for mouth kept open.

2. *Talking*: The event of talking is extracted as explained in Sec. 5.4.1.2. The duration for which talking is detected $d_T^{(t)}$ is used to estimate the intensity of talking. So, 0 is the minimum bound and the duration of the window w over which the indicator is estimated is the maximum bound.

3. *Yawning frequency*: The event of yawning is extracted as explained in Sec. 5.4.1.2. The intensity of yawning is estimated based on the frequency or number of yawns within a certain time duration, defined as the yawning frequency $f_Y^{(t)}$. The minimum bound for yawning frequency is 0 and the maximum bound is set to the duration within which $f_Y^{(t)}$ is measured divided by the average duration of a yawn.

4. *Brow furrows over time*: The brow furrows over time were extracted based on the temporal analysis of brow furrows as explained in Sec. 5.5. In order to estimate the intensity of this spatio-temporal feature, the time duration is used as the measure, i.e., the longer the persistence of brow furrows, higher will be the intensity estimated. If the brow furrows were for a very short period, as a momentary occurrence, then, the intensity is considered as *low*. A persistent occurrence of brow furrows will be discretized as *high*. The persistence of brow furrows in between low and high is discretized as *medium*. So, the limits for normalization are set accordingly. If $d_F^{(t)}$ is the duration for which brow furrows over time were detected, the minimum bound is 0 and the maximum bound is the total duration over which the indicator was measured, which is $d_F^{(t)}$.

The system assumes the above default minimum and maximum bounds during normalization. However, the doctor may choose to change these settings specific to the patient if necessary.

7.4 Determination of Wellness State

The normalized wellness indicators are combined in order to determine the wellness state. As explained in Sec. 7.2.1, at an instance of time t_i, the confidence measure of a set of five states namely, awake, asleep, inactive, drowsy and discomfort, are first computed, and then the wellness state $\mathbf{s}^{(\mathbf{t_i})}$ is determined. The five states will be addressed with the numbering from j = 1 to 5 as follows: inactive - $s_1^{(t)}$, discomfort - $s_2^{(t)}$, drowsy - $s_3^{(t)}$, awake - $s_4^{(t)}$, asleep - $s_5^{(t)}$, for the rest of the chapter. $s_j^{(t_i)}$ refers to the *confidence measure* of the respective states. Following the computation of confidence measure for the five states, the *wellness state* at that time instance is determined.

In order to compute the confidence measure of each state at an instance of time t, the wellness indicators at that time instance t are multiplied by weighing constants in order to prioritize their influence while computing the confidence measure. A voting system is used and the wellness indicators vote for the states they are indicative of.

The process of determination of wellness state at a time instance is explained as follows:

Prioritization of Wellness Indicators using Weighing Constants

The wellness indicators are combined through a *voting system* in order to compute the confidence measure of the states (asleep/awake/drowsy/inactive/discomfort). In this process, certain indicators are given priority over others. For example, brow furrows are given priority over eye state while voting for discomfort state. The wellness indicators are multiplied by weighing constants assigned to them that quantify their significance in voting for a particular state. The most significant wellness indicators with respect to a state are considered as primary indicators, and will be assigned a higher weightage while voting for that particular state. Secondary indicators are those that strengthen the voting of a certain state, and by themselves may not have significance independent of primary indicators. Tertiary indicators are similar to secondary indicators, but have even lesser weightage than them. The set of weights will be addressed as α and are tabulated in Table. 7.1. As seen from Table. 7.1, the non-zero values in a certain row imply that they are the indicators that are considered in the computation of the confidence measure of a particular state. Each individual weight assigned to the q^{th}

wellness indicator voting for the state $s_j^{(t)}$ will be addressed as $\alpha(q, j)$. In order to demonstrate the proof-of-concept of the framework, the weights α were derived based on simple experiments carried out on the WellCam dataset in this work.

TABLE 7.1: Weighing constants α assigned to wellness indicators γ while voting for the different states

	blink rate	eye state over time	blink duration	eyeball movement	mouth kept open	yawning frequency	talking	brow furrows over time
asleep	0	1	0	0	0	0	-1.25	0
awake	0	0.7	0	0	0	0	0.3	0
inactive	0.475	0	0	0.475	0.05	0	-0.5	0
drowsiness	0	0.15	0.85	0	0	0.85	0	0
discomfort	0	0.225	0	0	0.15	0	0	0.625

Apart from the primary, secondary and tertiary indicators that vote for a certain state, wellness indicators that vote *against* a certain state are also defined. These are called counter indicators and are defined in order to increase the robustness of the computation of the confidence measure of the states. The same wellness indicator gets different weights when being considered for voting for different states. The sum of weights for a particular state is equal to 1 (excluding the counter indicator).

Inverse voting by wellness indicators: Certain wellness indicators vote directly for a wellness state, which means the higher their value, higher is their vote for a particular state. However, there are several cases where the wellness indicators vote inversely for a wellness state, which means lower the value of $\gamma_q^{(t)}$ increases the vote for a wellness state $s_j^{(t)}$ and vice versa. For example, we have - low eyeball movement votes for *inactive* state. This has been illustrated in Fig. 7.6. This means, lower the eyeball movement, higher is the vote for inactive state. So, during the computation of the confidence measure for the inactive state, the normalized value of eyeball movement is first transformed through inverse voting and then used.

For a wellness indicator $\gamma_q^{(t)}$, if value of $\gamma_q^{(t)}$ tends to 0, then an inverse voting will tend $\gamma_q^{(t)}$ to infinity. Hence, the limits are normalized to lie between the limits [0,1] by

transforming the value of $\gamma_q^{(t)}$ in such cases to the transformed value of the wellness indicator $\gamma_q^{\prime(t)}$:

$$\gamma_q^{\prime(t)} = (e^{-\gamma_q^{(t)}} - e^{(-1)}) \times (0-1)/(e^{(0)} - e^{(-1)}) + 0 \tag{7.2}$$

Computation of Confidence Measure of the Five States

The description of each of the five states, the wellness indicators that vote for them as primary, secondary, tertiary or counter indicators for the computation of confidence measure are as follows:

1. *Inactive*: We define inactive state by lower eyeball movement, inability of eye being fully open, blink rate being low, increased blink duration and supported by mouth being kept open. So, the primary indicators for the inactive state are eye state over time and eyeball movement. Secondary indicators are blink rate and blink duration, and tertiary indicator is mouth kept open, while talking is a counter indicator. The wellness indicators are combined for computing the confidence measure of inactive state are summarized in Fig. 7.6. Eyeball movement, blink rate and eye state over time inversely vote for inactive state. Hence, the transformed values of these three wellness indicators as per (7.2) are computed and the respective weighing constants are multiplied. Talking is a counter indicator for inactive state, and is hence represented with a (-) sign in Fig. 7.6. The wellness indicators that are combined for computing the confidence measure of the rest of the three states (drowsy, asleep and discomfort) are shown in Appendix A.

2. *Drowsy*: The drowsy state is defined by eyes being closed for longer duration during the blinks, yawning and hence, the primary indicators are blink duration, yawning frequency and eye state over time. Higher the value of either blink duration or frequency of yawning, higher is the vote in favor of the drowsy state. Eye state over time votes inversely, i.e, lesser the value of eye state over time (in other words, lesser is the eye openness over time), more is the vote in favor of drowsy

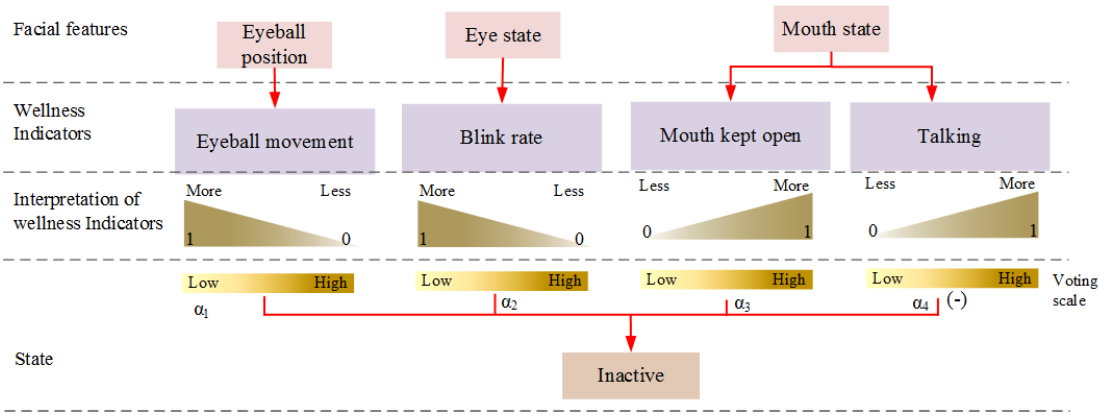


FIGURE 7.6: Wellness indicators that are combined for computing the confidence measure of inactive state, (-) indicates that *talking* is a counter indicator for inactive state

state. Referring to Table. 7.1, the weight assigned to blink duration and yawning frequency

3. *Awake*: The wellness indicators that are combined for computing the confidence measure of awake state are summarized in Fig. 7.4. The awake state is primarily defined based on the duration and amount of the eye being kept open. Hence, the primary indicator for awake attribute is eye state over time and the secondary indicator that supports that the awake state is *talking*.

4. *Asleep*: The asleep state is characterized by the amount and duration of the eye being kept closed. The primary indicator for asleep attribute is eye state over time. The counter indicator is *talking*, which means that even if the person appears to have the eyes closed, if he is talking, it is counter indicative of the subject being asleep. The eye state over time inversely votes for asleep state, which means the lesser the eye is open, more is the vote for asleep state. So, inverse of zero would tend to infinity. Hence, the transformation of limits is done to ensure the range of γ_{S_W} is between 0 and 1. So, the transformed value $\gamma'^{(t)}_{S_W}$ is derived as per (7.2). Talking is a counter indicator for asleep state.

The *confidence measure* $s_j^{(t)}$ of each of the five states is computed by a weighted summation of primary, secondary, tertiary and counter indicators. The confidence measure is computed based on the number of wellness indicators voting in favor of the state.

More the wellness indicators with higher weightage and higher their normalized value and longer their persistence over time, higher is their score for a particular state.

So, in order to compute $s_j^{(t)}$, we formulate state equations using the wellness indicators. A state equation is given by a dot product of the wellness indicators γ_q and the respective weighing constants $\alpha_{q,j}$ which gives the confidence measure $s_j^{(t)}$ for a state:

$$s_j^{(t)} = \sum_{q=1}^{8} \alpha_{(q,j)} \gamma_q^{(t)} \tag{7.3}$$

where j = 1 to 5 denoting the five states (inactive, discomfort, drowsy, awake and asleep). In cases where the transformed value of the wellness indicator needs to be used, $\gamma_q^{(t)}$ will need to be replaced by $\gamma_q'^{(t)}$. q ranges from 1 to 8, since we have eight wellness indicators considered in this work. The confidence measure $s_j^{(t)}$ for each state ranges from 0 to 1. The computation of the confidence measures for the five states is explained in Appendix. A.

Determining the Wellness State from among the States

If $s_j^{(t)}$ exceeds a certain preset threshold T_{2_j}, then, the state $s_j^{(t)}$ is determined to be the *patient's wellness state* $\mathbf{s^{(t)}}$ and the corresponding value of j is the *wellness state label* and will be denoted as $\mathbf{j^{(t)}}$ (this is done for ease of referring to a particular state among the five states). However, if $s_j^{(t)}$ exceeds the preset threshold for multiple states at time 't', i.e., if the patient is in multiple states simultaneously, the state with the highest confidence measure is called the primary state and the other states which cross the threshold will be called the secondary states. However, the decision while determining the primary state based on the value of $s_j^{(t)}$ is subject to the following conditions:

- The discomfort, inactive and drowsy states will be considered mutually exclusive states, so only one of them can have a significantly higher confidence measure than the other two at any given time. Asleep and awake states may co-occur with the discomfort state and awake may co-occur with inactive state. In the event of co-occurrence, the discomfort, inactive and drowsy states are given priority over awake and asleep states. Amongst discomfort, inactive and drowsy, the priority can be modified by the doctor if necessary.

- In the event that the patient is not in drowsy, inactive or discomfort states in his initial wellness state, the system sets default thresholds to detect the onset of any of these three states during the course of assessment.

Thus, we have $\mathbf{s}^{(\mathbf{t})}$ which is the confidence measure of the wellness state determined at time t along with the wellness state label $\mathbf{j}^{(\mathbf{t})}$. The wellness state assessed at the current instance of time is $\mathbf{s}^{(\mathbf{t_i})}$ and the wellness state label is $\mathbf{j}^{(\mathbf{t_i})}$.

7.5 Wellness Assessment

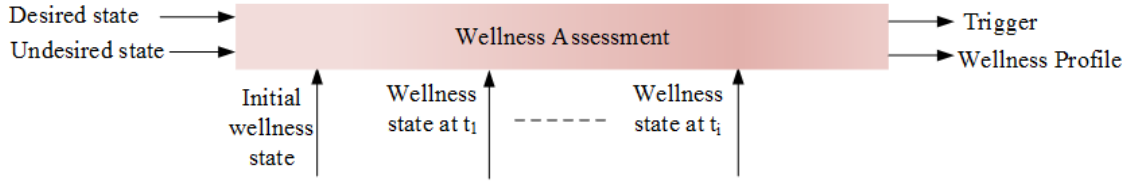


FIGURE 7.7: Illustration of wellness assessment with respect to initial, desired and undesired states, resulting in a trigger and a wellness profile.

Referring to Fig. 7.7, the inputs to the wellness assessment block at the current instance of time t_i are:

1. the initial wellness state $\mathbf{s}^{(\mathbf{t_0})}$ and the corresponding wellness state label $\mathbf{j}^{(\mathbf{t_0})}$ retrieved from the memory buffer,
2. current wellness state $\mathbf{s}^{(\mathbf{t_i})}$ and the corresponding wellness state label $\mathbf{j}^{(\mathbf{t_i})}$,
3. desired state D and undesired state U along with the associated thresholds T_D and T_U provided as input by the doctor at t_0. A threshold T_S for alerting if the patient's state is the same, without changing is also provided. T_S is a threshold on the duration beyond which if a patient is in the same wellness state continuously, must alert the doctor.

The outputs of the wellness assessment block are:

1. A ***wellness profile*** that shows the wellness state at each time instance, the rate at which the patient's wellness state has improved or worsened and change in severity of the wellness state over time. It is generated based on the history of the wellness states and the assessment made at each time instance.

2. A ***trigger*** to alert the doctor when the the wellness state $\mathbf{s}^{(\mathbf{t_i})}$ at the current instance of time t_i crosses the *thresholds* towards desired or undesired states, or the threshold T_S

The wellness assessment done for every instance of time is stored in a memory buffer, which is referred to as *History of wellness assessment* in Fig. 7.2. **Wellness Profile** The *wellness profile* is generated based on the history of wellness assessment over time. It contains vital information on how the patient's wellness state improved, worsened or remained the same over time. The wellness profile is generated when the wellness assessment at the current instance of time leads to a trigger. However, it can also be generated whenever the doctor wants as well. Fig. 7.8 shows a template of wellness

	t_0	t_1	t_2	t_3	t_4
Wellness state					
Change in wellness state					
% Change in wellness state					
Time since t_0 (min)					
Threshold reached (T_D/T_U/T_S): Time taken to reach threshold:					

FIGURE 7.8: A template of wellness profile

profile with the columns referring to the time instances starting with the initial time instance t_0. The wellness profile resembles a *patient's chart* that is filled by a nurse during the hourly rounds, where information on the the various parameters that are checked are noted down every hour. In the proposed work, the system takes note of the wellness state and the associated assessment, which are used to create the wellness profile. The following information can be extracted using the wellness profile:

- the label of the wellness state at each instance of time and the associated confidence measure $\mathbf{s}^{(t)}$,

- information on whether the wellness state approach improvement, worsening or stability at each instance of time,
- percentage change in wellness state with respect to the initial state at each instance of time,
- thresholds (T_D, T_U, T_S) that were crossed and the time taken to cross the threshold.

The importance of the wellness profile is illustrated through the following examples:

- *Monitoring the after-effect of drug*: A patient in a state of discomfort has been injected with a drug by a doctor and is being monitored for the response to the drug. Let us say the threshold towards improvement has been crossed and a trigger was sent, and a wellness profile was generated. Then, based on the wellness profile, the doctor can infer how long the patient took to respond to the drug, given the initial wellness state of the patient when he was injected with the drug. In case the trigger towards worsening was crossed, then, through the wellness profile, the doctor can infer at what rate the patient's condition worsened and since what time.
- *Monitoring recovery in post-operative care*: A patient under post-operative care, who is in an inactive state is being monitored for recovery. Let us say the patient begins to show an improvement from his initial state, threshold towards improvement is crossed and a wellness profile is generated. From the wellness profile, the doctor can make an inference about the time the patient began to show a an improvement from the inactive state and the rate at which he showed an increase in his level of activity.

Trigger generated when thresholds crossed

Thresholds are set by the doctor, so that he is alerted when the patient's wellness state approaches a desired state D, undesired state U or when the wellness state is the same for a long period of time. U. Let D = $[d_j]$, U = $[u_j]$ and S = $[s_j]$ where j = 1 to 5 which corresponds to the inactive, discomfort, drowsy, awake and asleep states in order. The doctor may specify the thresholds for one or more states, which when crossed

during the wellness assesment, must alert him, i.e., he may set the d_j, u_j and s_j of such states to the thresholds T_{D_j}, T_{U_j}, T_{S_j} respectively, where j = 1 to 3. Also, if the states - discomfort, drowsy and inactive are not in the initial states, then, the system will monitor these three states as undesired states by default, and sets default thresholds for the confidence measure that qualify them as the primary state. So, u_1, u_2 and u_3 will be set to the default thresholds T_{U_j} where j = 1 to 3, which may be modified by the doctor if necessary.

At every current instance of time t_i, patient's current state $\mathbf{s^{(t_i)}}$ is compared with the desired and undesired states. In order to detect the percentage change (improvement, worsening or stability), we define wellness parameters $\theta_{s_j}^{(t')}$ and $\theta_{I_j}^{(t')}$. $\theta_{s_j}^{(t')}$ and $\theta_{I_j}^{(t')}$ will be referred to as the wellness parameters. $\theta_{s_j}^{(t')}$ is given by $+$ or $-$ and indicates if there is an improvement or worsening in the assessment of the current wellness state in comparison with the initial state, desired and undesired states. If $\theta_{s_j}^{(t_i)} =$ '$+$', then, there is an improvement, else there is a worsening which is denoted by '-'. $\theta_{I_j}^{(t')}$ gives the percentage improvement or worsening. Depending on whether the initial wellness state was one of - inactive, discomfort and drowsy, or not, we have two possible cases:

- If the initial wellness state is one of - inactive, discomfort and drowsy, then, the wellness state label of the current state $\mathbf{j^{(t_i)}}$ matches the wellness state label in initial wellness state $\mathbf{j^{(t_0)}}$. The current wellness state $\mathbf{s^{(t_i)}}$ is compared with the initial wellness state $\mathbf{s^{(t_0)}}$ and the corresponding thresholds in the desired d_j and undesired u_j states to compute the wellness parameters $\theta_{s_j}^{(t')}$ and $\theta_{I_j}^{(t')}$.

$$\theta_{s_j}^{(t_i)} = \begin{cases} improvement\ (+), & \text{if} |(d_j - \mathbf{s^{(t_i)}})| < |(d_j - \mathbf{s^{(t_0)}})| \\ worsening\ (-), & \text{if} |(u_j - \mathbf{s^{(t_i)}})| < |(u_j - \mathbf{s^{(t_0)}})| \end{cases}$$

$$\theta_{I_j}^{(t_i)} = \mathbf{s^{(t_i)}} - \mathbf{s^{(t_0)}} \tag{7.4}$$

The system sends a *trigger* to the doctor if $\theta_{I_j}^{(t_i)}$ exceeds certain preset thresholds T_D and T_U: threshold T_D is used to check the relative change or shift in current

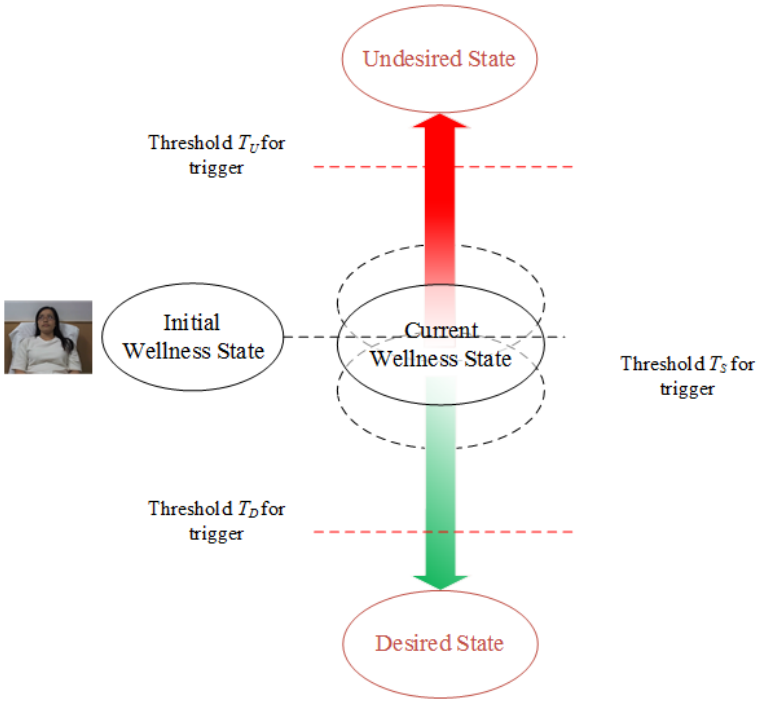


FIGURE 7.9: An illustration of generation of a trigger when the thresholds T_D towards improvement,T_U towards worsening or T_S are crossed

state with respect to the initial and desired states. Threshold T_U checks the relative change or shift in current state with respect to initial and undesired states. Although the system assumes default values, the doctor may specify different thresholds in T_D and T_U for each of the states. An illustration of generation of trigger is shown in Fig. 7.9.

- If the initial wellness state is *not* one of - inactive, discomfort and drowsy; and if the current wellness state is one of - inactive, discomfort and drowsy, then the current wellness state $\mathbf{s}^{(\mathbf{t_i})}$ will then be considered as the initial state for further instances of time.

7.6 Case study

Having discussed the steps in wellness assessment in the previous sections, the wellness state determination and wellness assessment are illustrated through a case study provided in this section.

(a) Example of Wellness State Extraction

We will first look at an example to illustrate the wellness state extraction when we have the wellness indicators over a period of time as input. Consider the wellness indicators measured at three time instances - t_1, t_2 and t_3 within windows of width w as shown in Fig. 7.10. They are extracted based on the techniques described in chapters 4 and 5. The corresponding normalized values of wellness indicators are plotted underneath each of these wellness indicators as discussed in Sec. 7.3. Thresholds t_H and t_L are set to discretize the wellness indicators into 'low', 'medium' and 'high'. The measured values of the wellness indicators and wellness indicators are summarized in Table. 7.2.

TABLE 7.2: Measured and normalized values of wellness indicators at t_1, t_2 and t_3

Wellness Indicator	[min. max]	Meas. at t_1	Norm. at t_1	Meas. at t_2	Norm. at t_2	Meas. at t_3	Norm. at t_3
Eyeball movt.	[0 20]	4.15	0.2	3.75	0.18	3	0.15
Blink rate	[0 60]	52.5	0.87	15	0.25	7	0.11
Eye state	[0 1]	1	1	1	1	0	0
Blink duration	[0.1 2]	0.22	0.11	0.2	0.1	0.25	0.125
Furrows	[0 15]	2	0.13	13	0.86	15	1
Mouth open	[0 80]	0	0	0	0	40	0.5
Talking	[0 80]	0	0	0	0	0	0
Yawning	[0 6]	0	0	0	0	0	0

Meas. = measured value of wellness indicator, Norm. = normalized value of wellness indicator, movt. = movement.

Then, the weights α from Table. 7.1 are applied and the weighted combination of the wellness indicators are used to extract the states at t_1, t_2 and t_3 as explained in Sec. 7.4. The resulting confidence measures for the states $s^{(t_i)}$ extracted at t_1, t_2 and t_3 on applying (7.3) are summarized in Table. 7.3.

It can be seen from Fig. 7.10 that the patient is awake, with eyes open, blinking, but also has brow furrows for most of the period from t_2 to t_3. During the period from t_2 to t_3, the eyes are closed, and brow furrows persist for the whole duration from t_2 to t_3. Mouth kept open for a short duration in t_3 also contributes to the increase in confidence

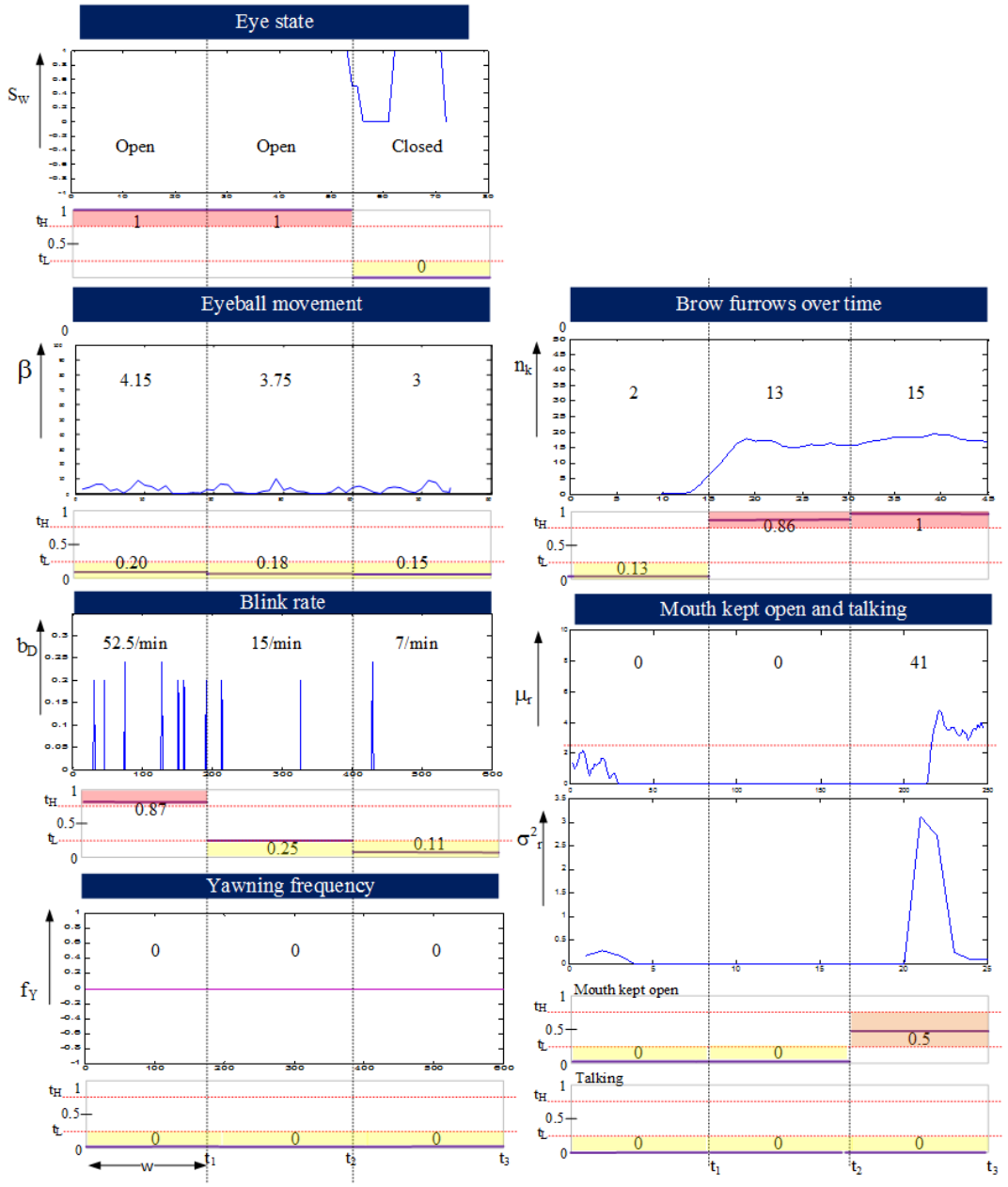


FIGURE 7.10: wellness indicators $I_q^{(t)}$ extracted at t_1, t_2 and t_3, and the corresponding wellness indicators $\gamma_q^{(t)}$ plotted underneath $I_q^{(t)}$. S_W = eye state, β = variance of eyeball position (eyeball movement), b_D = blink duration, f_Y = yawning frequency, μ_r and σ_r^2 are the mean and variance of the dark pixels between lips (to extract mouth kept open, and talking), n_k = number of pairs forming brow furrows (brow furrows over time); the time intervals marked on x-axis for the wellness indicators may vary. This is because, different window and overlap sizes have been used in the extraction of the wellness indicators, but the total duration is the same.

measure. As seen in Table. 7.3, this reflects as an increase in confidence measure of discomfort state from t_1 to t_2 and t_2 to t_3.

From Table. 7.3, we also infer that the wellness state was assessed to be awake with a

TABLE 7.3: Wellness State $s^{(t_i)}$ extracted at t_1, t_2 and t_3 indicated by the values in *bold*

	$s^{(t_1)}$	$s^{(t_2)}$	$s^{(t_3)}$
Inactive	0.37	0	0
Discomfort	0.08	**0.54**	**0.93**
Drowsy	0.08	0.08	0.18
Awake	**0.7**	0.7	0
Asleep	0	0	1

confidence measure of 0.7 at t_1. At t_2, discomfort state is given priority over inactive state in this example. The wellness state at time t_2 and t_3 are - discomfort with a confidence measure of 0.54 and 0.93 respectively. Mouth kept open and the persistence of furrows increasing have both contributed to the increase in confidence measure of the discomfort state at t_2 and t_3.

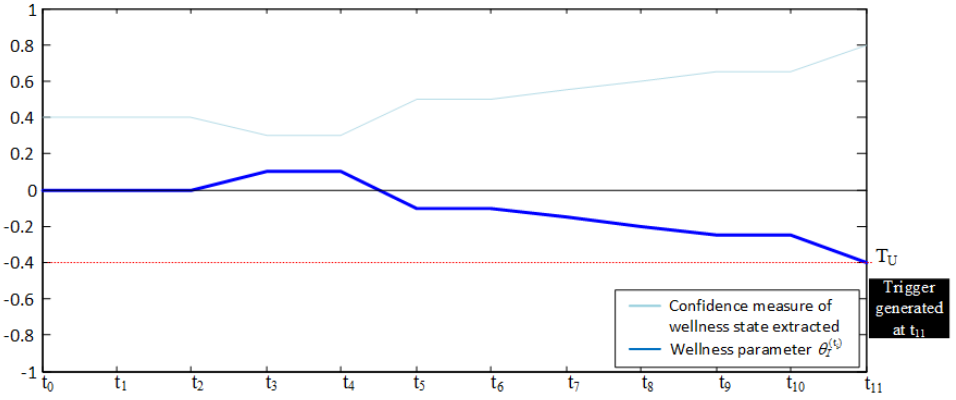
(b) Example of Wellness Assessment

Let us now see an example of wellness assessment, given the initial wellness state at t_0, desired and undesired states and the associated thresholds, and the wellness state determined at different time instants. The confidence measures for each of the five states $s_j^{(t_i)}$ for the instances of time t_0 to t_{11} are computed. So, we have $s_j^{(t_0)}$ to $s_j^{(t_{11})}$ tabulated in Table. 7.4. $s_j^{(t_0)}$ are the of confidence measures of states extracted in initial state. The primary state which determines the wellness state at each time instance is marked in *bold*.

TABLE 7.4: Confidence measures $s^{(t)}$of the states assessed at t_0 and the current time instances t_1 to t_{11}

	$s^{(t_0)}$	$s^{(t_1)}$	$s^{(t_2)}$	$s^{(t_3)}$	$s^{(t_4)}$	$s^{(t_5)}$	$s^{(t_6)}$	$s^{(t_7)}$	$s^{(t_8)}$	$s^{(t_9)}$	$s^{(t_{10})}$	$s^{(t_{11})}$
Inactive	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1
Discomfort	**0.4**	**0.4**	**0.4**	**0.3**	**0.3**	**0.5**	**0.5**	**0.55**	**0.6**	**0.65**	**0.65**	**0.8**
Drowsy	0.08	0.08	0.07	0.07	0.07	0.07	0.08	0.08	0.07	0.07	0.07	0.07
Awake	0.8	0.7	0.68	0.5	0.3	0.2	0.2	0.2	0.1	0.1	0.1	0.1
Asleep	0.1	0.1	0.12	0.15	0.12	0.22	0.26	0.39	0.4	0.41	0.42	0.45

From Table. 7.4, we find that the patient's initial wellness state was discomfort with a confidence measure of 0.4. The doctor then provides the desired state $D = [0\ 0.2\ 0\ 0\ 0]$, and undesired state $U = [0\ 0.4\ 0\ 0\ 0]$. This means that the doctor desires the patient's state of discomfort be reduced and would like to be alerted when it reduces by 0.2, and warned if it increases by 0.5 compared to the value in initial wellness state.



(a)

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
Wellness state $(j^{(t)})$	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.
$(s^{(t)})$	40%	40%	40%	30%	30%	50%	50%	55%	60%	65%	65%	80%
Change in wellness state $\theta_s^{(t)}$	-	stable	stable	improve	improve	worsen	worsen	worsen	worsen	worsen	worsen	worsen
% Change in wellness state $\theta_I^{(t)}$	-	0	0	10%	10%	10%	10%	15%	20%	25%	25%	40%
Time since t_0 (min)	-	5	10	15	20	25	30	35	40	45	50	55
Threshold reached : T_U **Time taken to reach threshold: 55 min**												

dis. = discomfort state

(b)

FIGURE 7.11: (a) Plot of confidence measure of wellness state at each instance of time t_0 to t_{11} and the computed wellness parameter $\theta_I^{(t_i)}$, along with threshold T_U, (b) Wellness profile of the assessment

The plot of confidence measure of the wellness state (which has been determined by discomfort state from t_0 to t_{11}) is shown in Fig. 7.11(a). The wellness parameters $\theta_s^{(t_i)}$ and $\theta_I^{(t_i)}$ are computed based on (7.4) for time instances t_1 to t_{11}. The wellness parameter $\theta_I^{(t_i)}$ has also been plotted for each time instance in Fig. 7.11(a), from which we can observe that the threshold of 0.4 towards undesired state is crossed at t_{11} for the discomfort state and hence, a *trigger* is generated.

The *wellness profile* is generated, as shown in Fig. 7.11(b). The wellness profile shows the wellness states assessed at each time instance and the associated confidence measure, which shows that *discomfort* is the wellness state from t_0 to t_{11}. Then, stability

is seen until t_2, then the condition improves until t_4, then it worsens continuously until t_{11}. The time taken for reaching the threshold T_U of 0.4 is 55 minutes, taking each time instance to be spaced at a duration of 5 min.

7.7 Summary

In this chapter, an integrated framework for wellness assessment was proposed. The term 'wellness' is first defined in the context of this thesis and a wellness assessment framework is proposed. The framework brings together the techniques proposed in the previous chapters for the extraction of wellness indicators from facial features. The process of wellness state determination and wellness assessment is described. The wellness indicators were first normalized to interpret the wellness indicators and a weighted combination of the normalized wellness indicators was used to extract the wellness state at a given time (t). Wellness is then assessed relative to the initial state, and the desired and undesired state input by the doctor. The wellness assessment involves the generation of a trigger if there was an improvement, worsening or stability in the patient's condition over a period of time, along with a wellness profile. Examples to show the wellness state determination and wellness assessment are also presented.

CHAPTER 8

Conclusions and Future Work

The contributions described in this thesis have led to the development of a framework for assessing wellness of a patient based on the analysis of the following facial features: (a) eyes, (b) mouth and (c) brow furrows. Techniques for extracting these facial features and their temporal analysis led to a framework to infer patient's state and his wellness.

The distinct property of eyebrows, of being relatively more stable in appearance was exploited to use them as anchor points for facial feature localization. The technique uses a conventional method for face detection as a pre-processing step. Deploying partial gradient maps enabled selective extraction of the upper and lower edges of the eyebrow. The iterative nature of the proposed method with decreasing threshold for eyebrow edge extraction was shown to be effective in dealing with low contrast between the eyebrows and skin, partial occlusions and variations in lighting conditions. Exploiting the properties of eyebrows to verify the extracted edges led to an average detection rate of 96% when the technique was evaluated on three standard databases and computational savings of 35% compared to a state-of-art method was achieved. Yet, false negatives occurred mainly when the skin and eyebrows both were of high intensities or when strong shadows were present in the eye socket region. False positives were contributed by presence of hair fringes just above the eyebrow, closely resembling an eyebrow, and

dark colored, framed glasses worn by the subject. Also, detection of eyebrows when head rotation exceeds $\pm 15^{\circ}$ needs further investigation.

The systematic approach based on computation of band-wise accumulation of intensities and gradient magnitudes rendered to be effective in detecting the eyeball, based on its distinct properties. The iterative increase in band size enabled the detection of eyeball in partially open eye or in eyes that appeared smaller in size than the standard anthropometric estimations. The choice of initial band size was critical in determining the eyeball detection accuracy. Based on an empirical study, an initial band-size of 0.6 $\times$ eyeball diameter led to a detection rate of 95% on the BioID database. However, the algorithm encountered misdetections when strong shadows were present in the eye region or the ROI was incorrectly estimated.

The eyeball detection technique when combined with blob analysis and inter-feature distance measures differentiated the following eye states: open, closed and partially open, achieving an average recall and precision of 91.3% and 96% respectively on a subset of the proposed WellCam database. The computations performed on a reduced set of pixels resulted in computational cost savings of as high as 67% compared to a state-of-art method. The size of the eye ROI window and the initial band-size were found to impact the detection accuracy, and required modifications during initialization for subjects with eyes that appeared smaller with respect to the face height.

The temporal analysis of the extracted eye features - eyeball position and eye state led to the extraction of wellness indicators - eye blink, eye state over time and eyeball movement. The evaluation of eye blink detection technique on standard databases resulted in similar mean accuracy as the state-of-the-art. However, high reflections due to glasses, and images with relatively lower resolution led to false negatives. Considering that there were no equivalent databases to test the eye state over time and eyeball movement, these wellness indicators were simulated in the WellCam database and validated. The evaluation of eye state over time on the WellCam database resulted in a recall and precision of 98.2% and 97% respectively. The inner ends of the eyebrows and the position of nostrils that were detected using simple techniques, aided in effective estimation

of the ROI(s) for mouth state detection. Operations performed on narrow, vertical cross-sections of the mouth, as opposed to the entire mouth region, substantially reduced the computations involved. A combination of the upper and lower lip positions and the analysis of the region between them was effective in detecting the mouth state as open or closed. Deploying computations such as accumulations and mean intensity profile on a reduced set of pixels led to significant computational savings compared to the state-of-art. Evaluation of the technique confirms that the accuracy of the proposed mouth state detection technique can be as much as 95%. However, limitations were observed in cases of frowning expressions, in which nostril positions were wrongly detected and in the presence of strong shadows in region beneath the lower lip.

The temporal analysis of mouth state led to the extraction of wellness indicators such as mouth kept open, yawning and talking. Evaluation of the yawning detection method achieved a detection rate of 100%. The proposed method is limited to distinguishing talking from mouth kept open or closed. Further efforts will be required to extend it to distinguish between a person actually talking or making movements with his mouth.

The reuse of inner ends of eyebrows for estimating ROI for brow furrow detection contributed to the computational efficiency. Partial gradient maps along with their magnitude-based weight assignment contributed to the selective extraction of brow furrow edges against surrounding noise. Computing the relative change with respect to an initialization phase allowed detection of brow furrows even when they were present as natural intransient features. The technique was shown to detect the absence or presence of brow furrows with an average precision and recall of 92% and 91% respectively.

It is concluded that setting the window size in the extraction of wellness indicators is an important parameter that can impact the detection accuracies. The window sizes to detect certain wellness indicators such as blinking and yawning are more assertive to estimate, because of the definite duration that they occur for. But, for other wellness indicators such as talking, eyeball movement, mouth state over time, eye state over time, the window size has an impact on the detection accuracy. Window sizes to achieve the best accuracy for each of the wellness indicators were set based on the experiments carried out.

Additionally, in order to further reduce the computational cost, a technique for accelerating patient face localization is proposed. The technique relies on detecting human head shoulder profile in a controlled setting of patient lying on the bed facing the camera. The use of block-based gradient angle histograms (GAH) with analysis of selected sets of angles aided in making the technique robust to noise. Setting the block size is critical in determining the detection rate and percentage search space savings. Evaluation of the method on standard databases showed that an average reduction in search space of 70% is achievable. Presence of background clutter led to lesser reduction of search space, due to more number of windows shortlisted with human head-shoulder curves. Extremely low contrast between the intensity of the clothing worn by the subject and the surrounding background and occlusion of shoulder by hair are the main reasons for false negatives. The technique led to a computational cost savings of 34% compared to applying face detection on the entire image.

Finally, an integrated framework for assessing patient wellness that brings together the localization of patient's face, extraction of facial features and wellness indicators as explained in the chapters 3 to 6, was proposed. Unlike existing solutions that are aimed at a specific medical condition or emotion, the methods proposed in this research work have led to a unified framework which is comprehensive, versatile, configurable and affordable. The framework acts as an assistive tool to the doctor. A novel aspect of the framework is its ability to capture relative changes in the wellness state, thus enabling it to monitor a variety of medical conditions. The wellness state is determined with a confidence measure, which is used for assessing wellness. A profile of the assessed wellness that contains important information for e.g. the change in wellness and its severity over time is provided for the doctor's reference. The system also generates a trigger in the event the patient's state is approaching improvement, worsening, or has remained stable for a very long duration. At present, the system is limited to a head rotation of $\pm 15^{\circ}$. Also, a linear model with manually assigned weights to the wellness indicators has been adopted in the wellness state determination process and this could be further investigated for improving the confidence measures of the wellness state determined.

8.1 Future Work

We recommend the following directions for further research:

- **Improving accuracy of face localization in the presence of background clutter**: The technique to reduce search space for face localization was shown to work well for the case of front-facing human. However, the number of shortlisted windows is still high mainly in the presence of background clutter, thereby pointing to opportunities for further reduction in the computation costs. Hence, it is worth relying skin color for reducing complexity and consider extracting additional features such as head to neck profile to make the technique more robust to background clutter.

- **Enhancing robustness of facial feature detection**: Although the facial feature detection techniques were shown to be robust across facial expressions, ethnicities and lighting conditions, the robustness of these techniques can be further enhanced by mainly addressing cases of low contrast between skin and the facial feature, occlusions, reflections due to glasses, and non-uniform and drastic changes in lighting conditions. Inclusion of tracking, advanced filtering techniques and color image processing could be considered for enhancing the robustness of the techniques.

- **Identifying other features for wellness assessment**: Wellness indicators based on temporal analysis of features such as eyes, mouth and brow furrows were extracted in this research work. It is worth extending the methods to extract wellness indicators caused due to other facial actions such as nose wrinkling, orbit tightening, lip stretch, head pose and head movement so as to further enhance the confidence measure in the detection of the wellness states. Other features that are worth exploring are coloration in the sclera and face, and color of the nose ad iris. Such additional features can be effectively incorporated into the current wellness assessment framework to realize a more robust and deterministic PMS.

- **Evaluation on datasets from hospitals**: In this thesis, we have proposed an overall framework for assessing wellness based on the extraction and analysis of facial features. Considering the administrative and privacy-related challenges in procuring a sizable dataset for this application from hospitals, the individual techniques were evaluated on the proposed WellCam dataset and standard datasets. However, it would be of immense value to evaluate the techniques on real datasets of patients from hospitals and wellness centers that are obtained under realistic operating conditions.

- **Architecture/implementation considerations**: The techniques proposed in this work for the extraction of the facial features and wellness indicators have been evaluated for their computational efficiency, and the resulting cost savings have been presented. As a next step, methods for integrating the proposed techniques into an embedded platform can be undertaken with a view to evaluate the appropriateness of deploying these techniques to realize a real-time capable embedded PMS.

- **Addressing large variations in head pose**: The techniques in this research are proposed by considering a single front-facing camera with head rotations (yaw, pitch and roll) up to $\pm 15^{\circ}$. Extending these techniques for changes in roll more than $\pm 15^{\circ}$ is worth exploring, especially for accommodating realistic real-life patient monitoring scenarios. One option is to consider incorporating a multi-camera system to handle larger head rotations. This would necessitate the calibration of the system and seamless integration among cameras. It is envisaged that a three camera arrangement can be considered whereby one camera situated in the center while the two on either side of the center camera.

- **Multi-modal health monitoring**: The proposed wellness assessment framework is currently based only on information extracted from a CMOS camera based vision sensor. Incorporating other sensors such as those used for monitoring the body's vital parameters, infrared sensors and audio sensors to realize a robust sensing through multi-modal data fusion should pave way for a reliable PMS

that can be safely deployed to facilitate in the monitoring of patients requiring intensive care.

APPENDIX A

Computation of Confidence Measures of Wellness States

The computation of confidence measure $s_j^{(t)}$ of the wellness states using (7.3), as a linear combination of the weighing constants $\alpha(q, j)$ and wellness indicators $\gamma_q^{(t)}$:

1. *Awake*: The awake state is primarily defined based on the duration and amount of the eye being kept open. Hence, the primary indicator for awake state is eye openness and the secondary indicator that supports that the awake state is 'talking'. The state equation is defined based on these two indicators as follows.

$$s_{awake}^{(t)} = \alpha_{(d_{S_W}, A)} \gamma_{d_{S_W}}^{(t)} + \alpha(d_T, A) \gamma_{d_T}^{(t)} \tag{A.1}$$

   The wellness indicators that are combined for computing the confidence measure of awake state are summarized in Fig. 7.4.

2. *Asleep*: The asleep state is characterized by the amount and duration of the eye being kept closed. The primary indicator for asleep state is eye openness. The counter indicator is 'talking', which means that even if the person appears to have the eyes closed, if he is talking, it is counter indicative of the subject being asleep. The eye openness inversely votes for asleep state attribute, which means the lesser

the eye is open, more is the vote for asleep state. So, inverse of zero would tend to infinity. Hence, the transformation of limits is done to ensure the range of $\gamma_{d_{S_W}}$ is between 0 and 1. So, the transformed value of γ, γ' derived as per (7.2)

Next, $\alpha_{d_T} \times \gamma_{d_T}$ is the term that represents talking in the state equation for sleep states, which will appear as a negative term, talking being a counter indicator of asleep state.

$$s^{(t)}_{asleep} = \alpha_{(d_{S_W},S)}\gamma'^{(t)}_{d_{S_W}} - \alpha_{(d_T,S)}\gamma^{(t)}_{d_T} \tag{A.2}$$

The wellness indicators are combined for computing the confidence measure of asleep state are summarized in Fig. A.1.

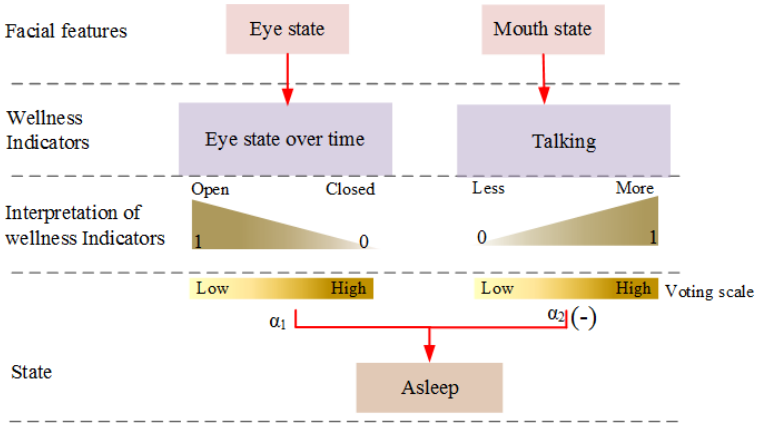


FIGURE A.1: Wellness indicators that are combined for computing the confidence measure of asleep state, (-) indicates *talking* is a counter indicator for asleep state

3. *Inactive*: We define inactive by lower eyeball movement, inability of eye being fully open, blink rate being low, increased blink duration and supported by mouth being kept open in awake state. So, the primary indicators for the inactive state are eye openness, eyeball movement. Secondary indicators are blink rate and blink duration and tertiary indicator is mouth kept open. The counter indicator is talking. The state equation for Inactive attribute is evaluated only if the eye openness is $\geq 20\%$ of fully open eye. Eyeball movement $\gamma^{(}_{\beta}t)$, blink rate γ_R and eye state over time $\gamma_{d_{eo}}$ inversely vote for inactive state attribute. Hence, the transformed values of these three indicators as per (7.2) is computed and the respective weighing constants are multiplied. Mouth kept open will be represented by $\gamma_{d_{mo}}$.

$\alpha_{(d_{S_W},I)} \times 1/\gamma_{d_{S_W}}$ will be the term representing eye openness in the state equation for drowsy and inactive state attributes. For inactive state attribute, the blink rate inversely votes for the state, i.e., lesser the blink rate, more is the vote for inactive state attribute. So, $\alpha_{(R,I)} \times 1/\gamma_R$ will be the term corresponding to blink rate in the state equation for inactive state attribute. $\alpha_{(d_T,I)} \times \gamma_{d_T}$ is the term that represents talking in the state equation for inactive attribute. The following is the state equation for inactive attribute:

$$\begin{aligned} s^{(t)}_{Inactive} &= \alpha_{(d_{S_W},I)}\gamma'^{(t)}_{d_{S_W}} + \alpha_{(\beta,I)}\gamma'^{(t)}_{\beta} + \alpha_{(R,I)} \times 1/\gamma'^{(t)}_{R} \\ &\quad + \alpha_{(b_D,I)}\gamma^{(t)}_{b_D} + \alpha_{(d_{mo},I)}\gamma^{(t)}_{d_{mo}} + \alpha_{(d_T,I)}\gamma^{(t)}_{d_T} \end{aligned} \tag{A.3}$$

4. *Discomfort*: In the proposed framework, the discomfort attribute is primarily characterized by the eyebrow inner ends being drawn together that result in brow furrows and the eyelids being drawn together. This implies that the primary indicators are brow furrows and eye state. Mouth being open is a secondary indicator that supports the detection of discomfort state attribute.

$$s^{(t)}_{Discomfort} = \alpha_{(d_F,Di)}\gamma_{d_F} + \alpha_{(d_{S_W},Di)}\gamma'^{(t)}_{d_{S_W}} + \alpha_{(d_{mo},Di)}\gamma^{(t)}_{d_{mo}} \tag{A.4}$$

The wellness indicators that are combined for computing the confidence measure of discomfort state are shown in Fig. A.2 .

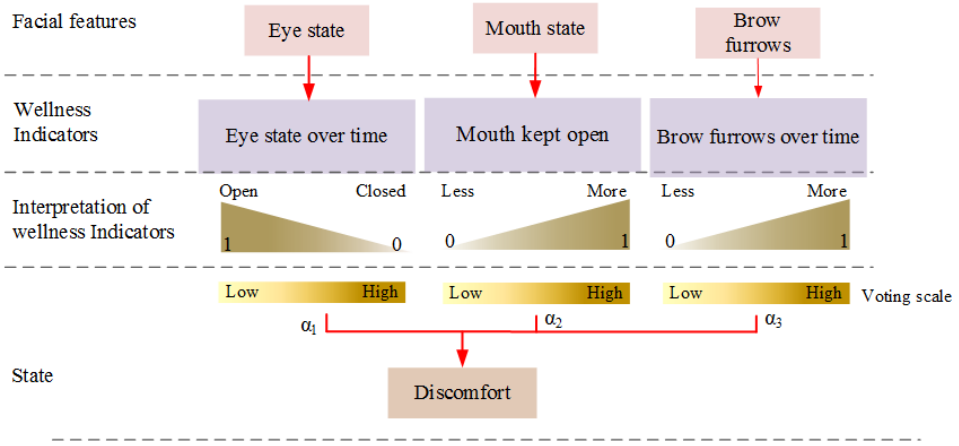


FIGURE A.2: Wellness indicators that are combined for computing the confidence measure of discomfort state

5. *Drowsy*: The wellness indicators are combined for computing the confidence measure of drowsy state are summarized in Fig. A.3. The drowsy attribute is

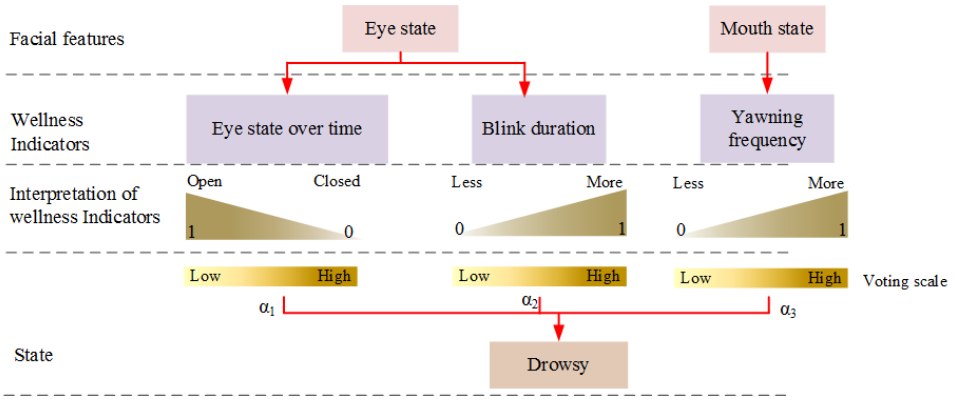


FIGURE A.3: Wellness indicators that are combined for computing the confidence measure of drowsy state

defined by eyes being closed for longer duration during the blinks, increased yawning and Hence, the primary indicators are blink duration, yawning and eye openness. For drowsy state, higher the value of blink duration and frequency of yawning, higher is the vote in favor of the drowsy state. Eye openness votes inversely, i.e, lesser the value of eye openness, more is the vote in favor of drowsy state. Mouth kept open is an additional indicator that supports the assessment of drowsy attribute.

$$s^{(t)}_{drowsy} = \alpha_{(f_Y,D)}\gamma^{(t)}_{f_Y} + \alpha_{(d_{S_W},D)}\gamma'^{(t)}_{d_{S_W}} + \alpha_{(b_D,D)}\gamma^{(t)}_{b_D} + \alpha_{(d_{mo},D)}\gamma^{(t)}_{d_{mo}} \quad \text{(A.5)}$$

APPENDIX B

Description of databases used in this thesis

TABLE B.1: Details of the WellCam dataset listing the various wellness indicators and their average duration in the dataset

Name of database	Type	pose	variation in lighting	Natural setting
Cohn-Kanade Facial Expression database	486 sequences from 97 subjects, in grayscale	30^o yaw	yes	No
The Japanese Female Facial Expression (JAFFE) database	213 grayscale images of 10 subjects	Front-facing	No	No
AR Face database	4000 color images from 126 subjects	frontal	yes	No, feature occlusions
BioID database	1521 grayscale images of 23 subjects	Slight	yes	Yes
ZJU eye blink database	80 video clips of 20 individuals, in color	frontal and upward view	yes	Yes
Talking Face video database	5000 color images of 1 person	Natural variation	No	Yes
Denver Intensity of Spontaneous Facial Action (DISFA) database	stereo videos of 27 adults, in color	Natural variation	No	Yes
Yawning Detection Dataset (YawDD)	322 videos from front mirror and 29 videos from camera placed on dash, in color	Natural variation	No	Yes
Extended Cohn-Kanade Facial Expression database	Type	pose	Yes	No
Karolinska Directed Emotional Faces (KDEF) database	4900 images of 70 subjects, in color	5 different angles	No	No
CASIA face image database	2500 images of 500 subjects	3 different angles	yes	No
Buffy Stickmen dataset	748 video frames, in color	natural setting	yes	yes

The description of the databases used in this thesis have been summarized in the Table B.1.

APPENDIX C

Examples of Implementation on MATLAB

All the techniques proposed in this work were implemented and evaluated on MATLAB 2013A.

- **Eyeball detection** The flowchart summarizing the major steps in detection of the eyeball is provided in Fig. 4.10 in Section 4.2.2.2. The top level structure of the MATLAB code and an excerpt from the code is provided below.

```
% Top level structure
face = face_detect(input image in grayscale)
eyebrow positions = eyebrow_detect(face, parameters for eyebrow detection)
eye ROI = ROI_estimate(face, eyebrow positions)
eyeball and eye corner positions = loop(eye_ROI,bs,thresholds)
    bandwise summation = bandwise_sum(eye_ROI,bs)
    peaks and valleys = peak_valley_extract(bandwise summation)
    top 2 bands = bands_prioritize(peaks and valleys, height, width & number of
                                   peaks)
    PV combinations = peak_valley_analysis(peaks & valleys in top 2 bands)
    eyeball and eye corner positions = property_check(PV combinations in the
                                                      two bands, thresholds)
```

```
% Excerpt of MATLAB code
face_detect %Face detection; computer-vision system toolbox is used
eyebrow_detect % Eyebrow detection
```

```
ROI_estimate % Estimation of eye ROI for left and right eyes
loop % loop to detect eyeball position, with iterative reduction in band size
        while i<=size(arr) %iterations in band sizes
            bs = arr(i)
            bandwise_sum %compute band-wise intensity-weighted summation, after
            %division into bands
                for b = 1:round(bs/2):size(R,1)-bs-1 %50% overlap between bands,
                %bs refers to band size & R refers to eye ROI
                    for j = 1:size(R,2)
                        sum_j = sum(R(b:b+bs-1,j)); %bs is band size
                        sum_v_temp = [sum_v_temp, sum_j];
                    end
                    sum_temp = sum_v_temp/max(sum_v_temp);
                    sum_vector = [sum_vector;sum_temp]; %sum_vector represents
                    %bandwise summation
                end
            peak_valley_extract %Extract the significant valleys and peaks
            bands_prioritize %Prioritize the bands based on presence of
            %significant valley, height and number of peaks
            for k = 1:2 %consider the top two prioritized bands
                peak_valley_analysis %Peak valley analysis in each band - get
                %the PVP, VPV,VP and PV combinations, reorder them based on
                peak/valley height property_check %Perform checks to verify
                %the peak-valley combination to be eyeball-sclera
            end
            if eyeball_found ==1
                break % come out of while loop if eyeball is found & output
                    %eyeball and eye corner positions
            else % else continue with next band size
                continue
            end
        end
```

- **Mouth feature point detection** The flowchart summarizing the major steps in detection of the mouth feature points (upper and lower lip positions) is provided in Fig. 5.8 in Section 5.2.2.2. The top level structure of the MATLAB code and an excerpt from the code is provided below.

```
% Top level structure
face = face_detect(input image in grayscale)
eyebrow positions = eyebrow_detect(face, parameters for eyebrow detection)
eyebrow inner ends = eyebrow_inner_ends(eyebrow positions, face)
nostril positions = nostril_detect(face, eyebrow positions, eyebrow inner ends)
mouth ROI(s) R1,R2,R2_1 = mouth_ROI_estimate(face, eyebrow & nostril positions)
M1,M2,p1,v,p2 = intensity_accu(R1,R2,w) %w is the band size for generating
%profile, approximate vallue set based on anthropometric measures
```

```
P_ref = mean_int_profile_ref(p2 & R2_1 of reference image, w) %for reference image
OR
P, jW_m = mean_int_profile(P_ref, p2 & R2_1 of incoming image, w) %for incoming
%image

--------------------------------------------------------------------------------
% Excerpt of MATLAB code
face_detect %Face detection; computer-vision system toolbox is used
eyebrow_detect %Eyebrow detection
eyebrow_inner_ends %detection of eyebrow inner ends
nostril_detect % Detection of nostril positions
mouth_ROI_estimate % Estimation of mouth ROI(s) R1,R2,R2'
intensity_accu %Generation of intensity accumulation map M1 and M2 of R1 and R2
%respectively, and detection of upper lip position p2
if image == reference image
    mean_int_profile_ref % Generation of bandwise mean intensity profile P_ref of
    %R2' of reference image
else
    mean_int_profile % Generation of P of R2_1 of incoming image, detection of
    %lower lip position jW_m
        R2_1_norm = double(R2_1)./max(double(R2_1(:)));
        P = []; %mean intensity profile of R2_1 of incoming image
        for i = 1:round(0.625*w2):size(R2_1,1)-w
            P = [P;i mean(mean(R2_1(i:i+w,:)))];
        end
        P_ref = P_ref./max(P_ref);
        D = []; %Array of Euclidean distances
        for j = 1:size(P,1)-size(P_ref,1) % Scan P, compute Euclidean distance
        %between every P_ref and Pw_j, then find best matching window
            Pw_j = P(j:j+size(P_ref,1)-1,2);
            Pw_j = Pw_j./max(Pw_j);
            D = [D; j norm(Pw_j - P_ref)];
        end
        pA = find(P(:,1)>p2);
        jW_m = Mouth_int_sum_prof4(D(D(pA(1):end,2)==min(D(pA(1):end,2)),1)+
                                   pA(1)-1,1);
end
```

- **Eyebrow detection** The flowchart summarizing the major steps in eyebrow detection is provided in Fig. 3.13 in Section 3.3.4. The top level structure of the MATLAB code is provided below.

```
% Top level structure
face = face_detect(input image in grayscale)
eyebrow ROI = eyebrow_ROI(face)
G_y = gradient_map(eyebrow ROI)
        P_R and P_L = loop(eyebrow ROI,thresholds,band size)loop to extract upper
```

```
        %and lower eyebrow edges iteratively for right and left halves of the face,
        %followed by verification
            E_y+,E_y- = signed_map_gen(G_y, thresholds)
            Segment pairs = signed_pairs(E_y+,E_y-,band size)
            P_R and P_L = candidate_verification(Segment pairs)
        end loop
```

- **Brow furrow detection** The flowchart summarizing the major steps in brow furrow extraction is provided in Fig. 5.24 in Section 5.5. The top level structure of the MATLAB code is provided below.

```
% Top level structure
face = face_detect(input image in grayscale)
eyebrow positions = eyebrow_detect(face, parameters for eyebrow detection)
eyebrow inner ends = eyebrow_inner_ends(eyebrow positions, face)
G_x = gradient_map(brow furrow ROI)
F_x+,F_x- = signed_map_gen(G_x, thresholds)
F_xP, G_xP = pairs(F_x+,F_x-, band size) %extraction of pairs
if Initialization phase:
    G_xPW,g_i,n_i = weighted_grad_mag_init(G_xP) %g_i is weighted gradient
                    %magnitude sum & n_i is the total number of pairs forming
                    %the furrows
if Detection phase:
    G_xPW,g_d,r = weighted_grad_mag_detect(G_xP) %g_d & r are differences in
                  %weighted gradient magnitude sum & total number of pairs
                  %forming the furrows with respect to initialization phase
```

Bibliography

[1] P. Rashidi and A. Mihailidis, "A Survey on Ambient-Assisted Living Tools for Older Adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, 2013.

[2] C. Crispim, V. Bathrinarayanan, B. Fosty, A. Konig, R. Romdhane, M. Thonnat, and F. Bremond, "Evaluation of a monitoring system for event recognition of older people," in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug. 2013, pp. 165–170.

[3] S. Colantonio, G. Coppini, D. Germanese, D. Giorgi, M. Magrini, P. Marraccini, M. Martinelli, M. A. Morales, M. A. Pascali, G. Raccichini, M. Righi, and O. Salvetti, "A smart mirror to promote a healthy lifestyle," *Biosystems Engineering*, vol. 138, pp. 33–43, Oct. 2015.

[4] C. W. Wang, A. Hunter, N. Gravill, and S. Matusiewicz, "Unconstrained Video Monitoring of Breathing Behavior and Application to Diagnosis of Sleep Apnea," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 396–404, Feb. 2014.

[5] K. Sikka, A. Dhall, and M. S. Bartlett, "Classification and weakly supervised pain localization using multiple segment representation," *Image and Vision Computing*, vol. 32, no. 10, pp. 659–670, Oct. 2014.

[6] S. Alghowinem, R. Goecke, M. Wagner, G. Parker, and M. Breakspear, "Eye movement analysis for depression detection," in *2013 20th IEEE International Conference on Image Processing (ICIP)*, Sep. 2013, pp. 4220–4224.

[7] T. H. Lee, H. J. Kwon, D. J. Kim, and K. S. Hong, "Design and Implementation of Mobile Self-care System Using Voice and Facial Images," in *Ambient Assistive Health and Wellness Management in the Heart of the City*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Jul. 2009, no. 5597, pp. 249–252.

[8] A. O'Brien and R. Mac Ruairi, "Survey of Assistive Technology Devices and Applications for Aging in Place," in *Second International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2009. CENTRIC '09*, Sep. 2009, pp. 7–12.

[9] M. Shoaib, T. Elbrandt, R. Dragon, and J. Ostermann, "Altcare: Safe living for elderly people," in *Pervasive Computing Technologies for Healthcare (Pervasive-Health), 2010 4th International Conference on-NO PERMISSIONS*, Mar. 2010, pp. 1–4.

[10] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, "A Survey on Ambient Intelligence in Health Care," *Proc IEEE Inst Electr Electron Eng*, vol. 101, no. 12, pp. 2470–2494, Dec. 2013.

[11] J. Biswas, A. Tolstikov, M. Jayachandran, V. Foo, A. A. P. Wai, C. Phua, W. Huang, L. Shue, K. Gopalakrishnan, J.-E. Lee, and P. Yap, "Health and wellness monitoring through wearable and ambient sensors: exemplars from home-based care of elderly with mild dementia," *Ann. Telecommun.*, vol. 65, no. 9-10, pp. 505–521, May 2010.

[12] G. Souto, "Guardian: A Pervasive Environment to Monitor Elderly People in Medical Treatment at Home," in *Inclusive Society: Health and Wellbeing in the Community, and Care at Home*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Jun. 2013, no. 7910, pp. 286–291.

[13] UN (2012) World population ageing 1950-2050, *Department of Economic and Social Affairs, Population Division, United Nations*, 2012.

[14] B. OMullane, B. Bortz, A. OHannlon, J. Loane, and R. B. Knapp, "Comparison of Health Measures to Movement Data in Aware Homes," in *Ambient Intelligence*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Nov. 2011, no. 7040, pp. 290–294.

[15] M. Alwan, S. Dalal, D. Mack, S. Kell, B. Turner, J. Leachtenauer, and R. Felder, "Impact of monitoring technology in assisted living: outcome pilot," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 192–198, Jan. 2006.

[16] R. Paradiso, "Wearable health care system for vital signs monitoring," in *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003*, Apr. 2003, pp. 283–286.

[17] A. Pantelopoulos and N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis," *IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev.*, vol. 40, no. 1, pp. 1–12, Jan. 2010.

[18] A. M. Tabar, A. Keshavarz, and H. Aghajan, "Smart Home Care Network Using Sensor Fusion and Distributed Vision-based Reasoning," in *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, ser. VSSN '06. New York, NY, USA: ACM, 2006, pp. 145–154.

[19] R. Orpwood, C. Gibbs, T. Adlam, R. Faulkner, and D. Meegahawatte, "The design of smart homes for people with dementiauser-interface aspects," *Univ Access Inf Soc*, vol. 4, no. 2, pp. 156–164, Jul. 2005.

[20] A. Helal, D. J. Cook, and M. Schmalz, "Smart Home-Based Health Platform for Behavioral Monitoring and Alteration of Diabetes Patients," *J Diabetes Sci Technol*, vol. 3, no. 1, pp. 141–148, Jan. 2009.

[21] N. Suryadevara and S. Mukhopadhyay, "Determining Wellness through an Ambient Assisted Living Environment," *IEEE Intelligent Systems*, vol. 29, no. 3, pp. 30–37, May 2014.

[22] T. S. Barger, D. E. Brown, and M. Alwan, "Health-status monitoring through analysis of behavioral patterns," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, no. 1, pp. 22–27, Jan. 2005.

[23] B. Ni, N. C. Dat, and P. Moulin, "RGBD-camera based get-up event detection for hospital fall prevention," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp. 1405–1408.

[24] Y.-T. Peng, C.-Y. Lin, M.-T. Sun, and M.-W. Feng, "Sleep condition inferencing using simple multimodality sensors," in *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, May 2006, pp. 4.

[25] R. Murthy, I. Pavlidis, and P. Tsiamyrtzis, "Touchless monitoring of breathing function," *Conf Proc IEEE Eng Med Biol Soc*, vol. 2, pp. 1196–1199, 2004.

[26] N. Sun, M. Garbey, A. Merla, and I. Pavlidis, "Imaging the cardiovascular pulse," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 2, Jun. 2005, pp. 416–421 vol. 2.

[27] R. Barea, L. Bergasa, E. Lopez, M. Ocana, D. Schleicher, and A. Leon, "Robotic assistants for health care," in *IEEE International Conference on Robotics and Biomimetics, 2008. ROBIO 2008*, 2009, pp. 1099–1104.

[28] T. Gao, D. Greenspan, M. Welsh, R. R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *2005 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-7*. New York: Ieee, 2005, pp. 102–105.

[29] T. Gao, C. Pesto, L. Selavo, Y. Chen, J. Ko, J. H. Lim, A. Terzis, A. Watt, J. Jeng, B.-r. Chen, K. Lorincz, and M. Welsh, "Wireless Medical Sensor Networks in Emergency Response: Implementation and Pilot Results," in *2008 IEEE Conference on Technologies for Homeland Security*, May 2008, pp. 187–192.

[30] E. Monton, J. F. Hernandez, J. M. Blasco, T. Herve, J. Micallef, I. Grech, A. Brincat, and V. Traver, "Body area network for wireless patient monitoring," *IET Commun.*, vol. 2, no. 2, pp. 215–222, Feb. 2008.

[31] U. Varshney, "Pervasive Healthcare and Wireless Health Monitoring," *Mob. Netw. Appl.*, vol. 12, no. 2-3, pp. 113–127, Mar. 2007.

[32] M. Fischer, Y. Y. Lim, E. Lawrence, and L. K. Ganguli, "ReMoteCare: Health Monitoring with Streaming Video," in *7th International Conference on Mobile Business, 2008. ICMB '08*, Jul. 2008, pp. 280–286.

[33] E. P. Scilingo, A. Lanat, and A. Tognetti, "Sensors for Wearable Systems," in *Wearable Monitoring Systems*, A. Bonfiglio and D. D. Rossi, Eds. Springer US, Jan. 2011, pp. 3–25.

[34] C. Doukas and I. Maglogiannis, "Advanced patient or elder fall detection based on movement and sound data," in *Second International Conference on Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008*, Jan. 2008, pp. 103–107.

[35] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer Methods and Programs in Biomedicine*, vol. 117, no. 3, pp. 489–501, Dec. 2014.

[36] G. Chen, P. Govindaswamy, N. Li, and J. Wang, "Continuous Camera-based Monitoring for Assistive Environments," in *Proceedings of the 1st International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '08. New York, NY, USA: ACM, 2008, pp. 31:1–31:8.

[37] K. Malakuti and A. Albu, "Towards an Intelligent Bed Sensor: Non-intrusive Monitoring of Sleep Irregularities with Computer Vision Techniques," in *2010 20th International Conference on Pattern Recognition (ICPR)*, Aug. 2010, pp. 4004–4007.

[38] H. Cheng, Z. Liu, Y. Zhao, and G. Ye, "Real world activity summary for senior home monitoring," in *2011 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2011, pp. 1–4.

[39] G. Balakrishnan, F. Durand, and J. Guttag, "Detecting Pulse from Head Motions in Video," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013, pp. 3430–3437.

[40] S. Alghowinem, R. Goecke, M. Wagner, G. Parkerx, and M. Breakspear, "Head Pose and Movement Analysis as an Indicator of Depression," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, Sep. 2013, pp. 283–288.

[41] Z. Li, A. da Silva, and J. Cunha, "Movement quantification in epileptic seizures: a new approach to video-EEG analysis," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 6, pp. 565–573, Jun. 2002.

[42] Y. Takemura, J.-y. Sato, and M. Nakajima, "A Respiratory Movement Monitoring System Using Fiber-Grating Vision Sensor for Diagnosing Sleep Apnea Syndrome," *OPT REV*, vol. 12, no. 1, pp. 46–53, Jan. 2005.

[43] S. Zambanini, J. Machajdik, and M. Kampel, "Detecting falls at homes using a network of low-resolution cameras," in *2010 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)*, 2010, pp. 1–4.

[44] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall Detection With Multiple Cameras: An Occlusion-Resistant Method Based on 3-D Silhouette Vertical Distribution," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 290–300, 2011.

[45] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall Detection from Depth Map Video Sequences," in *Toward Useful Services for Elderly and People with Disabilities*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Jan. 2011, no. 6719, pp. 121–128.

[46] G. Mastorakis and D. Makris, "Fall detection system using Kinects infrared sensor," *J Real-Time Image Proc*, pp. 1–12, Mar. 2012.

[47] Z. Zhang, U. Kapoor, M. Narayanan, N. Lovell, and S. Redmond, "Design of an unobtrusive wireless sensor network for nighttime falls detection," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society,EMBC*, Aug. 2011, pp. 5275–5278.

[48] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiol Meas*, vol. 25, no. 2, pp. R1–20, Apr. 2004.

[49] U. Anliker, J. A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. B. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic, "AMON: A wearable multiparameter medical monitoring and alert system," *IEEE Transactions on Information Technology In Biomedicine*, vol. 8, no. 4, pp. 415–427, Dec. 2004.

[50] H. Ghasemzadeh, E. Guenterberg, and R. Jafari, "Lightweight Signal Processing for Wearable Body Sensor Networks," in *Wearable Monitoring Systems*, A. Bonfiglio and D. D. Rossi, Eds. Springer US, Jan. 2011, pp. 99–122.

[51] F. G. Miskelly, "Assistive technology in elderly care," *Age Ageing*, vol. 30, no. 6, pp. 455–458, Nov. 2001.

[52] A. Mihailidis, B. Carmichael, and J. Boger, "The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 3, pp. 238–247, Sep. 2004.

[53] M. Martinez and R. Stiefelhagen, "Breath rate monitoring during sleep using near-ir imagery and PCA," in *2012 21st International Conference on Pattern Recognition (ICPR)*, Nov. 2012, pp. 3472–3475.

[54] M. C. Yu, H. Wu, J. L. Liou, M. S. Lee, and Y. P. Hung, "Multiparameter Sleep Monitoring Using a Depth Camera," in *Biomedical Engineering Systems and Technologies*, ser. Communications in Computer and Information Science, Springer Berlin Heidelberg, Jan. 2013, no. 357, pp. 311–325.

[55] F. C. Yang, C. H. Kuo, M. Y. Tsai, and S. C. Huang, "Image-based sleep motion recognition using artificial neural networks," in *2003 International Conference on Machine Learning and Cybernetics*, vol. 5, Nov. 2003, pp. 2775–2780.

[56] K. Nakajim, Y. Matsumoto, and T. Tamura, "Development of real-time image sequence analysis for evaluating posture change and respiratory rate of a subject in bed," *Physiol Meas*, vol. 22, no. 3, pp. N21–28, Aug. 2001.

[57] W. H. Liao and C. M. Yang, "Video-based activity and movement pattern analysis in overnight sleep studies," in *19th International Conference on Pattern Recognition, 2008. ICPR 2008*, Dec. 2008, pp. 1–4.

[58] Chekmenev SY, Rara H, Farag AA (2005) Non-contact, wavelet-based measurement of vital signs using thermal imaging. In: The first Intl. Conf on graphics, vision, and image processing (GVIP), pp. 107–112.

[59] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, Jan. 2013.

[60] F. Hijaz, N. Afzal, T. Ahmad, and O. Hasan, "Survey of fall detection and daily activity monitoring techniques," in *2010 International Conference on Information and Emerging Technologies (ICIET)*, 2010, pp. 1–6.

[61] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *10th International Conference on e-health Networking, Applications and Services, 2008. HealthCom 2008*, 2008, pp. 42–47.

[62] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. E. Lundy, "Fall detection - Principles and Methods," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007*, 2007, pp. 1663–1666.

[63] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, no. 1, p. 66, Jul. 2013.

[64] D. N. Olivieri, I. Gmez Conde, and X. A. Vila Sobrino, "Eigenspace-based fall detection and activity recognition from motion templates and machine learning," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5935–5945, Apr. 2012.

[65] M. Y. Chen, "Long Term Activity Analysis in Surveillance Video Archives," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 2010.

[66] K. Zhan, F. Ramos, and S. Faux, "Activity recognition from a wearable camera," in *2012 12th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec. 2012, pp. 365–370.

[67] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, vol. 2, Jun. 2004, pp. 819–826.

[68] Z. Khan and W. Sohn, "Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1843–1850, Nov. 2011.

[69] M. Pediaditis, M. Tsiknakis, L. Koumakis, M. Karachaliou, S. Voutoufianakis, and P. Vorgia, "Vision-based absence seizure detection," *Conf Proc IEEE Eng Med Biol Soc*, vol. 2012, pp. 65–68, 2012.

[70] M. Pediaditis, M. Tsiknakis, P. Vorgia, D. Kafetzopoulos, V. Danilatou, and D. Fotiadis, "Vision-based human motion analysis in epilepsy - Methods and challenges," in *2010 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)*, Nov. 2010, pp. 1–5.

[71] Y. M. Kuo, J. S. Lee, and P. C. Chung, "A Visual Context-Awareness-Based Sleeping-Respiration Measurement System," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 255–265, 2010.

[72] J. Kroutil, A. Laposa, and M. Husak, "Respiration Monitoring During Sleeping," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, ser. ISABEL '11. New York, NY, USA: ACM, 2011, pp. 33:1–33:5.

[73] I. Sato and M. Nakajima, "Non-contact Breath Motion Monitor ing System in Full Automation," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, 2005, pp. 3448–3451.

[74] J. Fei and I. Pavlidis, "Analysis of breathing air flow patterns in thermal imaging," *Conf Proc IEEE Eng Med Biol Soc*, vol. 1, pp. 946–952, 2006.

[75] L. L. Chen, K. W. Chen, and Y. P. Hung, "A sleep monitoring system based on audio, video and depth information for detecting sleep events," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2014, pp. 1–6.

[76] V. K. Somers, D. P. White, R. Amin, W. T. Abraham, F. Costa, A. Culebras, S. Daniels, J. S. Floras, C. E. Hunt, L. J. Olson, T. G. Pickering, R. Russell, M. Woo, and T. Young, "Sleep Apnea and Cardiovascular Disease in an American Heart)," *Journal of the American College of Cardiology*, vol. 52, no. 8, pp. 686–717, Aug. 2008.

[77] D. Falie, L. David, and M. Ichim, "Statistical algorithm for detection and screening sleep apnea," in *International Symposium on Signals, Circuits and Systems, 2009. ISSCS 2009*, Jul. 2009, pp. 1–4.

[78] J. Fei, I. Pavlidis, and J. Murthy, "Thermal vision for sleep apnea monitoring," *Med Image Comput Comput Assist Interv*, vol. 12, no. Pt 2, pp. 1084–1091, 2009.

[79] T. Gault and A. Farag, "A Fully Automatic Method to Extract the Heart Rate from Thermal Video," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2013, pp. 336–341.

[80] M. Z. Poh, D. McDuff, and R. Picard, "A Medical Mirror for Non-contact Health Monitoring," in *ACM SIGGRAPH 2011 Emerging Technologies*, ser. SIGGRAPH '11, New York, NY, USA: ACM, 2011, pp. 2:12:1.

[81] X. Li, J. Chen, G. Zhao, and M. Pietikainen, "Remote Heart Rate Measurement from Face Videos under Realistic Situations," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014, pp. 4264–4271.

[82] J. Kranjec, S. Begu, G. Gerak, and J. Drnovek, "Non-contact heart rate and heart rate variability measurements: A review," *Biomedical Signal Processing and Control*, vol. 13, pp. 102–112, Sep. 2014.

[83] C. A. Gilbert, C. M. Lilley, K. D. Craig, P. J. McGrath, C. A. Court, S. M. Bennett, and C. J. Montgomery, "Postoperative pain expression in preschool children: validation of the child facial coding system," *Clin J Pain*, vol. 15, no. 3, pp. 192–200, Sep. 1999.

[84] K. M. Prkachin, "The consistency of facial expressions of pain: a comparison across modalities," *Pain*, vol. 51, no. 3, pp. 297–306, Dec. 1992.

[85] K. M. Prkachin, "Assessing pain by facial expression: Facial expression as nexus," *Pain Res Manag*, vol. 14, no. 1, pp. 53–58, 2009.

[86] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, 1978.

[87] Y.-L. Tian, T. Kanade, and J. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, Feb. 2001.

[88] G. C. Littlewort, M. S. Bartlett, and K. Lee, "Faces of Pain: Automated Measurement of Spontaneousallfacial Expressions of Genuine and Posed Pain," in *Proceedings of the 9th International Conference on Multimodal Interfaces*, ser. ICMI '07. New York, NY, USA: ACM, 2007, pp. 15–21.

[89] B. Gholami, W. M. Haddad, and A. R. Tannenbaum, "Agitation and pain assessment using digital imaging," *Conf Proc IEEE Eng Med Biol Soc*, pp. 2176–2179, 2009.

[90] L. Nanni, S. Brahnam, and A. Lumini, "A local approach based on a Local Binary Patterns variant texture descriptor for classifying pain states," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7888–7894, Dec. 2010.

[91] A. B. Ashraf, K. Prkachin, T. Chen, S. Lucey, P. Solomon, Z. Ambadar, J. F. Cohn, and B.-j. Theobald, "The painful face - pain expression recognition using active appearance models," in *In ICMI*, 2007.

[92] P. Werner, A. Al-Hamadi, and R. Niese, "Pain recognition and intensity rating based on Comparative Learning," in *2012 19th IEEE International Conference on Image Processing (ICIP)*, Sep. 2012, pp. 2313–2316.

[93] A. B. Ashraf, S. Lucey, J. F. Cohn, T. Chen, Z. Ambadar, K. M. Prkachin, and P. E. Solomon, "The painful face Pain expression recognition using active appearance models," *Image and Vision Computing*, vol. 27, no. 12, pp. 1788–1796, Nov. 2009.

[94] P. Lucey, J. Cohn, I. Matthews, S. Lucey, S. Sridharan, J. Howlett, and K. Prkachin, "Automatically Detecting Pain in Video Through Facial Action Units," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 3, pp. 664–674, 2011.

[95] Z. Hammal and J. F. Cohn, "Automatic Detection of Pain Intensity," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ser. ICMI '12, New York, NY, USA: ACM, 2012, pp. 47–52.

[96] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews, "Painful Monitoring: Automatic Pain Monitoring Using the UNBC-McMaster Shoulder Pain Expression Archive Database," *Image and Vision Computing*, vol. 30, no. 3, pp. 197–205, Mar. 2012.

[97] A. A. H. Philipp Werner, "Towards Pain Monitoring: Facial Expression, Head Pose, a new Database, an Automatic System and Remaining Challenges," 2013.

[98] R. Khan, A. Meyer, H. Konik, and S. Bouakaz, "Pain detection through shape and appearance features," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2013, pp. 1–6.

[99] P. L. Manfredi, B. Breuer, D. E. Meier, and L. Libow, "Pain assessment in elderly patients with severe dementia," *Journal of Pain Symptom Management*, vol. 25, no. 1, pp. 48–52, Jan. 2003.

[100] A. Horgas and L. Miller, "Pain assessment in people with dementia," *American Journal of Nursing*, vol. 108, no. 7, pp. 62–70; quiz 71, Jul. 2008.

[101] M. Kunz, S. Scharmann, U. Hemmeter, K. Schepelmann, and S. Lautenbacher, "The facial expression of pain in patients with dementia," *PAIN*, vol. 133, no. 13, pp. 221–228, Dec. 2007.

[102] G. McIntyre, R. Gocke, M. Hyett, M. Green, and M. Breakspear, "An approach for automatically measuring facial activity in depressed subjects," in *3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009*, Sep. 2009, pp. 1–8.

[103] J. Cohn, T. Kruez, I. Matthews, Y. Yang, M. H. Nguyen, M. Padilla, F. Zhou, and F. De la Torre, "Detecting depression from facial actions and vocal prosody," in *3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009*, Sep. 2009, pp. 1–7.

[104] J. Joshi, A. Dhall, R. Goecke, M. Breakspear, and G. Parker, "Neural-net classification for spatio-temporal descriptor based depression analysis," in *2012 21st International Conference on Pattern Recognition (ICPR)*, Nov. 2012, pp. 2634–2638.

[105] J. Joshi, R. Goecke, G. Parker, and M. Breakspear, "Can body expressions contribute to automatic depression analysis?" in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Apr. 2013, pp. 1–7.

[106] J. Joshi, R. Goecke, S. Alghowinem, A. Dhall, M. Wagner, J. Epps, G. Parker, and M. Breakspear, "Multimodal assistive technologies for depression diagnosis and monitoring," *J Multimodal User Interfaces*, vol. 7, no. 3, pp. 217–228, Sep. 2013.

[107] Y. Dai, Y. Shibata, T. Ishii, K. Hashimoto, K. Katamachi, K. Noguchi, N. Kakizaki, and D. Cai, "An associate memory model of facial expressions and its application in facial expression recognition of patients on bed," in *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001*, Aug. 2001, pp. 591–594.

[108] W. Liao, W. Zhang, Z. Zhu, and Q. Ji, "A Real-Time Human Stress Monitoring System Using Dynamic Bayesian Network," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops*, Jun. 2005, pp. 70–70.

[109] M. K. Mandal, R. Pandey, and A. B. Prasad, "Facial expressions of emotions and schizophrenia: a review," *Schizophr Bull*, vol. 24, no. 3, pp. 399–412, 1998.

[110] V. Bevilacqua, D. D'Ambruoso, G. Mandolino, and M. Suma, "A new tool to support diagnosis of neurological disorders by means of facial expressions," in *2011 IEEE International Workshop on Medical Measurements and Applications Proceedings (MeMeA)*, May 2011, pp. 544–549.

[111] J. Hamm, C. G. Kohler, R. C. Gur, and R. Verma, "Automated Facial Action Coding System for Dynamic Analysis of Facial Expressions in Neuropsychiatric Disorders," *J Neurosci Methods*, vol. 200, no. 2, pp. 237–256, Sep. 2011.

[112] P. Wang, F. Barrett, E. Martin, M. Milanova, R. E. Gur, R. C. Gur, C. Kohler, and R. Verma, "Automated Video Based Facial Expression Analysis of Neuropsychiatric Disorders," *J Neurosci Methods*, vol. 168, no. 1, pp. 224–238, Feb. 2008.

[113] C. Alvino, C. Kohler, F. Barrett, R. E. Gur, R. C. Gur, and R. Verma, "Computerized measurement of facial expression of emotions in schizophrenia," *J. Neurosci. Methods*, vol. 163, no. 2, pp. 350–361, Jul. 2007.

[114] M. N. Bin Mansor, S. Yaacob, R. Nagarajan, and M. Hariharan, "Detection of facial changes for hospital ICU patients," in *2010 6th International Colloquium on Signal Processing and Its Applications (CSPA)*, May 2010, pp. 1–5.

[115] M. Naufal Mansor, S. Yaacob, R. Nagarajan, L. S. Che, M. Hariharan, and M. Ezanuddin, "Detection of facial changes for ICU patients using KNN classifier," in *2010 International Conference on Intelligent and Advanced Systems (ICIAS)*, Jun. 2010, pp. 1–5.

[116] N. Brusco and A. Paviotti, "3d quantification of facial edemas," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005. ISPA 2005*, Sep. 2005, pp. 191–196.

[117] C. R. Rogers, K. L. Schmidt, J. M. VanSwearingen, J. F. Cohn, G. S. Wachtman, E. K. Manders, and F. W.-B. Deleyiannis, "Automated facial image analysis: detecting improvement in abnormal facial movement after treatment with botulinum toxin A," *Ann Plast Surg*, vol. 58, no. 1, pp. 39–47, Jan. 2007.

[118] M. Smith and L. Puczko, "Health and Wellness Tourism - 1st Edition," 2009. [Online]. Available: https://www.elsevier.com/books/health-and-wellness-tourism/smith/978-0-7506-8343-2

[119] N. Suryadevara, M. Quazi, and S. Mukhopadhyay, "Intelligent Sensing Systems for Measuring Wellness Indices of the Daily Activities for the Elderly," in *2012 8th International Conference on Intelligent Environments (IE)*, Jun. 2012, pp. 347–350.

[120] N. Suryadevara and S. Mukhopadhyay, "Wireless Sensor Network Based Home Monitoring System for Wellness Determination of Elderly," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1965–1972, Jun. 2012.

[121] N. D. Lane, M. Lin, M. Mohammod, X. Yang, H. Lu, G. Cardone, S. Ali, A. Doryab, E. Berke, A. T. Campbell, and T. Choudhury, "BeWell: Sensing Sleep, Physical Activities and Social Interactions to Promote Wellbeing," *Mobile Netw Appl*, vol. 19, no. 3, pp. 345–359, Jan. 2014.

[122] G. Morgavi, R. Nerino, L. Marconi, P. Cutugno, C. Ferraris, A. Cinini, and M. Morando, "An Integrated Approach to the Well-Being of the Elderly People at Home," in *Ambient Assisted Living*, ser. Biosystems & Biorobotics, B. And, P. Siciliano, V. Marletta, and A. Monteri, Springer International Publishing, 2015, no. 11, pp. 265–274.

[123] Y. Andreu-Cabedo, P. Castellano, S. Colantonio, G. Coppini, R. Favilla, D. Germanese, G. Giannakakis, D. Giorgi, M. Larsson, P. Marraccini, M. Martinelli, B. Matuszewski, M. Milanic, M. Pascali, M. Pediaditis, G. Raccichini, L. Randeberg, O. Salvetti, and T. Stromberg, "Mirror mirror on the wall #x2026; An intelligent multisensory mirror for well-being self-assessment," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, Jun. 2015, pp. 1–6.

[124] N. Yang, X. Zhao, and H. Zhang, "A non-contact health monitoring model based on the Internet of things," in *2012 Eighth International Conference on Natural Computation (ICNC)*, May 2012, pp. 506–510.

[125] H. Eisenbarth, G. W. Alpers, "Happy mouth and sad eyes: scanning emotional facial expressions, *Emotion*, vol. 11, no. 4, pp. 860-865, 2011.

[126] "Eye Health: Indicator of Overall Wellness," 2014. [Online]. Available: http://www.ottawalife.com/2014/03/eye-health-indicator-of-overall-wellness/

[127] I. Garcia, S. Bronte, L. Bergasa, J. Almazan, and J. Yebes, "Vision-based drowsiness detector for real driving conditions," in *2012 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2012, pp. 618–623.

[128] H. Tan and Y.-J. Zhang, "Detecting eye blink states by tracking iris and eyelids," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 667–675, Apr. 2006.

[129] R. Hammoud, A. Wilhelm, P. Malawey, and G. Witt, "Efficient real-time algorithms for eye state and head pose tracking in advanced driver support systems," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 2, Jun. 2005, pp. 1181.

[130] R. Oyini Mbouna, S. Kong, and M.-G. Chun, "Visual Analysis of Eye State and Head Pose for Driver Alertness Monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1462–1469, Sep. 2013.

[131] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition*, vol. 47, no. 9, pp. 2825–2838, Sep. 2014.

[132] T. Gevers, R. Valenti, "Accurate Eye Center Location through Invariant Isocentric Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1785–1798, Sep. 2012.

[133] A. George and A. Routray, "Fast and accurate algorithm for eye localisation for gaze tracking in low-resolution images," *IET Computer Vision*, vol. 10, no. 7, pp. 660-669, Oct. 2016.

[134] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[135] D. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[136] T. Drutarovsky and A. Fogelton, "Eye Blink Detection Using Variance of Motion Vectors," in *Computer Vision - ECCV 2014 Workshops*, ser. Lecture Notes in Computer Science, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Springer International Publishing, Sep. 2014, no. 8927, pp. 436–448.

[137] M. Divjak and H. Horst, "Robust Eye blink based fatigue detection for prevention of computer vision syndrome," 2009, pp. 350–353.

[138] K. Grauman, M. Betke, J. Gips, and G. R. Bradski, "Communication via Eye Blinks - Detection and Duration Analysis in Real Time," in *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR*, 2000.

[139] M. Szwoch and P. Pieniek, "Eye Blink Based Detection of Liveness in Biometric Authentication Systems Using Conditional Random Fields," in *Computer Vision and Graphics*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Sep. 2012, no. 7594, pp. 669–676.

[140] M. Chau and M. Betke, "Real time eye tracking and blink detection with USB cameras," In Boston University Computer Science Technical Report No. 2005–12, Tech. Rep., 2005.

[141] W. O. Lee, E. C. Lee, and K. R. Park, "Blink detection robust to various facial poses," *Journal of Neuroscience Methods*, vol. 193, no. 2, pp. 356–372, Nov. 2010.

[142] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," in *2010 International Conference on Machine and Web Intelligence (ICMWI)*, Oct. 2010, pp. 230–233.

[143] Y. Kurylyak, F. Lamonaca, and G. Mirabelli, "Detection of the eye blinks for human's fatigue monitoring," in *2012 IEEE International Symposium on Medical Measurements and Applications Proceedings (MeMeA)*, 2012, pp. 1–4.

[144] K. Takahashi and Y. Mitsukura, "Eye blink detection using monocular system and its applications," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, Sep. 2012, pp. 743–747.

[145] Y. Ying, S. Jing, and Z. Wei, "The Monitoring Method of Driver's Fatigue Based on Neural Network," in *International Conference on Mechatronics and Automation, 2007. ICMA 2007*, Aug. 2007, pp. 3555–3559.

[146] S. Abtahi, B. Hariri, and S. Shirmohammadi, "Driver drowsiness monitoring based on yawning detection," in *2011 IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, May 2011, pp. 1–4.

[147] W. Zhang, Y. L. Murphey, T. Wang, and Q. Xu, "Driver yawning detection based on deep convolutional neural learning and robust nose tracking," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2015, pp. 1–8.

[148] R. Zheng, C. Tian, H. Li, M. Li, and W. Wei, "Fatigue Detection Based on Fast Facial Feature Analysis," in *Advances in Multimedia Information Processing – PCM 2015*, ser. Lecture Notes in Computer Science, Springer International Publishing, Sep. 2015, no. 9315, pp. 477–487.

[149] B. Hariri, S. Abtahi, S. Shirmohammadi, and L. Martel, "Demo: Vision based smart in-car camera system for driver yawning detection," in *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, Aug. 2011, pp. 1–2.

[150] M. Omidyeganeh, S. Shirmohammadi, S. Abtahi, A. Khurshid, M. Farhan, J. Scharcanski, B. Hariri, D. Laroche, and L. Martel, "Yawning Detection Using Embedded Smart Cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 570–582, Mar. 2016.

[151] P. Smith, M. Shah, and N. d. V. Lobo, "Determining driver visual attention with one camera," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 4, pp. 205–218, Dec. 2003.

[152] T. Wang and P. Shi, "Yawning detection for determining driver drowsiness," in *Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology, 2005*, May 2005, pp. 373–376.

[153] E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan, "Drowsy Driver Detection Through Facial Movement Analysis," in *HumanComputer Interaction*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Oct. 2007, no. 4796, pp. 6–18.

[154] T. Azim, M. A. Jaffar, M. Ramzan, and A. M. Mirza, "Automatic Fatigue Detection of Drivers through Yawning Analysis," in *Signal Processing, Image Processing and Pattern Recognition*, ser. Communications in Computer and Information Science, Springer Berlin Heidelberg, 2009, no. 61, pp. 125–132.

[155] L. Li, Y. Chen, and Z. Li, "Yawning detection for monitoring driver fatigue based on two cameras," in *12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC '09*, Oct. 2009, pp. 1–6.

[156] W. Rongben, G. Lie, T. Bingliang, and J. Lisheng, "Monitoring mouth movement for driver fatigue or distraction with one camera," in *The 7th International IEEE Conference on Intelligent Transportation Systems, 2004. Proceedings*, Oct. 2004, pp. 314–319.

[157] E. Vural, M. etin, A. Eril, G. Littlewort, M. Bartlett, and J. Movellan, "Machine Learning Systems for Detecting Driver Drowsiness," in *In-Vehicle Corpus and Signal Processing for Driver Behavior*, K. Takeda, H. Erdogan, J. H. L. Hansen, and H. Abut, Eds. Springer US, 2009, pp. 97–110.

[158] J. Jimnez-Pinto and M. Torres-Torriti, "Driver alert state and fatigue detection by salient points analysis," in *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009*, Oct. 2009, pp. 455–461.

[159] C. Bouvier, A. Benoit, A. Caplier, and P.-Y. Coulon, "Open or Closed Mouth State Detection: Static Supervised Classification Based on Log-Polar Signature," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Oct. 2008, no. 5259, pp. 1093–1102.

[160] P. C. Yuen, J. H. Lai, and Q. Y. Huang, "Mouth state estimation in mobile computing environment," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, May 2004, pp. 705–710.

[161] Y.-L. Tian, T. Kanada, and J. Cohn, "Recognizing upper face action units for facial expression analysis," in *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings*, vol. 1, 2000, pp. 294–301 vol.1.

[162] "The Nursing Shortage and the Quality of Care," *New England Journal of Medicine*, vol. 347, no. 14, pp. 1118–1119, 2002. [Online]. Available: http://www.nejm.org/doi/full/10.1056/NEJM200210033471419

[163] D. Stefanov, Z. Bien, and W.-C. Bang, "The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 2, pp. 228–250, 2004.

[164] Z. Htike, S. Egerton, and K. Y. Chow, "A Monocular View-Invariant Fall Detection System for the Elderly in Assisted Home Environments," in *2011 7th International Conference on Intelligent Environments (IE)*, 2011, pp. 40–46.

[165] P. V. K. Borges and N. Nourani-Vatani, "Vision-based detection of unusual patient activity," *Stud Health Technol Inform*, vol. 168, pp. 16–23, 2011.

[166] P. Varady, L. Nagy, and L. Szilagyi, "On-line detection of sleep apnea during critical care monitoring," in *Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2000*, vol. 2, 2000, pp. 1299–1301.

[167] Y. W. Bai, W. T. Li, and Y. W. Chen, "Design and implementation of an embedded monitor system for detection of a patient's breath by double Webcams," in *2010 IEEE International Workshop on Medical Measurements and Applications Proceedings (MeMeA)*, 2010, pp. 171–176.

[168] S. Fleck and W. Strasser, "Smart Camera Based Monitoring System and Its Application to Assisted Living," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1698–1714, 2008.

[169] S. Iwasawa, J. Ohya, K. Takahashi, T. Sakaguchi, K. Ebihara, and S. Morishima, "Human body postures from trinocular camera images," in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000. Proceedings*, 2000, pp. 326–331.

[170] Z. Hammal and M. Kunz, "Pain monitoring: A dynamic and context-sensitive system," *Pattern Recognition*, vol. 45, no. 4, pp. 1265–1280, Apr. 2012.

[171] K. Mishima and T. Sugahara, "Analysis methods for facial motion," *Japanese Dental Science Review*, vol. 45, no. 1, pp. 4–13, May 2009.

[172] J. Song, L. Wang, and W. Wang, "Eyebrow Segmentation Based on Binary Edge Image," in *Intelligent Computing Technology*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Jan. 2012, no. 7389, pp. 350–356.

[173] Y. Li, H. Li, and Z. Cai, "Human eyebrow recognition in the matching-recognizing framework, *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 170-181, Feb. 2013.

[174] D. Kelly, J. R. Delannoy, J. McDonald, and C. Markham, "Incorporating facial features into a multi-channel gesture recognition system for the interpretation of irish sign language sequences, *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1977–1984, Sept. 2009.

[175] Q. Chen, W. k. Cham, and K. k. Lee, "Extracting eyebrow contour and chin contour for face recognition," *Pattern Recognition*, vol. 40, no. 8, pp. 2292-2300, Aug. 2007.

[176] L. Ding and A. Martinez, "Precise detailed detection of faces and facial features," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, 2008, pp. 1–7.

[177] A. Nikolaidis and I. Pitas, "Facial feature extraction and pose determination, *Pattern Recognition*, vol. 33, no. 11, pp. 1783–1791, Nov. 2000.

[178] Y. Li, Y. Dong, DL. Woodard, "Automatic segmentation of eyebrows for biometric recognition using modified level set, in *2012 IEEE International Conference on Image Processing*, 2012.

[179] S. Suchitra, R. K. Satzoda, and T. Srikanthan, "Detection amp; classification of arrow markings on roads using signed edge signatures," in *2012 IEEE Intelligent Vehicles Symposium (IV)*, 2012, pp. 796–801.

[180] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[181] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2008.

[182] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000. Proceedings*, 2000, pp. 46–53.

[183] M. J. Lyons, M. Kamachi, J. Gyoba, "Japanese Female Facial Expressions (JAFFE), Database of digital images, 1997. www.kasrl.org/jaffe.html

[184] A. Martinez and R. Benavente, "The AR face database, *CVC technical report*, no.24, June 1998. http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html

[185] D. Lundqvist, A. Flykt, and A Öhman, "The karolinska directed emotional faces (KDEF), *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, pp. 91–630, 1998.

[186] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J. Cohn, "DISFA: A Spontaneous Facial Action Intensity Database," *IEEE Transactions on Affective Computing*, vol. 4, no. 2, pp. 151–160, Apr. 2013.

[187] BioID Technology Research, "The bioid face database," 2001. http://www.bioid.com.

[188] P. M. Prendergast, "Facial Proportions," in *Advanced Surgical Facial Rejuvenation*, Springer Berlin Heidelberg, 2012, pp. 15–22.

[189] A. S. M. Sohail and P. Bhattacharya, "Detection of Facial Feature Points Using Anthropometric Face Model," in *Signal Processing for Image Enhancement and Multimedia Processing*, ser. Multimedia Systems and Applications Series, Springer US, 2008, no. 31, pp. 189–200.

[190] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, Jun. 2008, pp. 1–8.

[191] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera," in *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, Oct. 2007, pp. 1–8.

[192] "Talking Face Video Database, *Face and Gesture Recognition Working group*, http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html

[193] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based Fatigue Detection for Prevention of Computer Vision Syndrome, in *IAPR Conference on Machine Vision Applications*, 2009.

[194] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A Yawning Detection Dataset," in *Proceedings of the 5th ACM Multimedia Systems Conference*, ser. MMSys '14. New York, NY, USA: ACM, 2014, pp. 24–28.

[195] C. Anitha, M. Venkatesha and B. Suryanarayana Adiga, "A Two Fold Expert System for Yawning Detection," *Procedia Computer Science*, vol. 92, pp. 63-71, 2016.

[196] T. Kanade, J. Cohn, and Y. Tian, "The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression," in *Third IEEE International Workshop on CVPR for Human Communication Behavior Analysis, 2010. Proceedings*, 2010, pp. 94101.

[197] S. Z. Li and A. Jain, *Handbook of Face Recognition*. Springer Science & Business Media, Aug. 2011.

[198] R. Sznitman and B. Jedynak, "Active testing for face detection and localization," *IEEE Trans Pattern Anal Mach Intell*, vol. 32, no. 10, pp. 1914–1920, Oct. 2010.

[199] K. Khattab, J. Miteran, J. Dubois, and J. Matas, "Embedded system study for real time boosting based face detection," in *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, Nov 2006, pp. 3461–3465.

[200] D. Hefenbrock, J. Oberg, N. Thanh, R. Kastner, and S. Baden, "Accelerating viola-jones face detection to fpga-level using gpus," in *Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on*, May 2010, pp. 11–18.

[201] M. Yang, J. Crenshaw, B. Augustine, R. Mareachen, and Y. Wu, "Adaboost-based face detection for embedded systems," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1116 – 1125, 2010, special issue on Embedded Vision.

[202] O. Bilaniuk, E. Fazl-Ersi, R. Laganiere, C. Xu, D. Laroche, and C. Moulder, "Fast lbp face detection on low-power simd architectures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 616–622.

[203] H. Ren and M. Che, "A multi-core architecture for face detection," in *Multimedia Technology (ICMT), 2011 International Conference on*, July 2011, pp. 3354–3357.

[204] J. Cho, S. Mirzaei, J. Oberg, and R. Kastner, "Fpga-based face detection system using haar classifiers," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '09, pp. 103–112.

[205] F. He, Y. Li, S. Wang, and X. Ding, "A novel hierarchical framework for human head-shoulder detection," in *2011 4th International Congress on Image and Signal Processing (CISP)*, vol. 3, 2011, pp. 1485–1489.

[206] D. Xu, Y. L. Chen, X. Wu, Y. Ou, and Y. Xu, "Integrated approach of skin-color detection and depth information for hand and face localization," in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2011, pp. 952–956.

[207] J. J. D. Dios and N. Garca, "Feature Extraction Used for Face Localization Based on Skin Color," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Jan. 2005, no. 3656, pp. 1032–1039.

[208] C. Zhang and Z. Zhang, *A Survey of Recent Advances in Face detection*, 2010.

[209] Y. Sun, Y. Wang, Y. He, and Y. Hua, "Head-and-Shoulder Detection in Varying Pose," in *Advances in Natural Computation*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Jan. 2005, no. 3611, pp. 12–20.

[210] A. Kolesnikov, "Constrained piecewise linear approximation of digital curves." In *19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[211] S. W. Zucker, C. David, A. Dobbins and L. Iverson, "The organization of curve detection: Coarse tangent fields and fine spline coverings," in *2nd International Conference on Computer Vision*, 1988, pp. 568-577.

[212] R. K. Satzoda, S. Suchitra, and T. Srikanthan, "Gradient angle histograms for efficient linear hough transform," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 3273–3276.

[213] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 1, 2005, pp. 886–893.

[214] V. Ferrari, M. Eichner, M. Marin-Jimenez, and A. Zisserman, "Buffy stickmen v 3.01 dataset www.robots.ox.ac.uk/~vgg/data/stickmen/

[215] "CASIA-FaceV5 dataset, Chinese Academy of Sciences Institute of Automation, C.A. http://biometrics.idealtest.org/dbDetailForUser.do?id=9

[216] R. Andraka, "A survey of cordic algorithms for fpga based computers," in *Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays*, ser. FPGA '98. New York, NY, USA: ACM, 1998, pp. 191–200.

[217] S. Sathyanarayana, R. Satzoda, and S. Thambipillai, "Unified cordic based processor for image processing," in *Digital Signal Processing, 2007 15th International Conference on*, July 2007, pp. 343–346.

[218] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford University Press, 2010.

[219] D. D. S. Deng, and H. ElGindy, "High-speed parameterizable Hough transform using reconfigurable hardware," in *Proc. Pan-Sydney Area Workshop Visual Information Processing*, May. 2001, pp. 51–57.

[220] A. R. Bentivoglio, S. B. Bressman, E. Cassetta, D. Carretta, P. Tonali, and A. Albanese, "Analysis of blink rate patterns in normal subjects," *Mov. Disord.*, vol. 12, no. 6, pp. 1028–1034, Nov. 1997.