ROBUST DISCRIMINATIVE ANALYSIS FRAMEWORK FOR GAZE AND HEAD-POSE ESTIMATION

by

Salahaldeen Rabba

Master of Applied Mathematics, Ryerson University, 2014

Bachelor of Electrical Engineering, Lakehead University, 2002

A dissertation presented to Ryerson University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2019

Declaration

I hereby declare that I am the sole author of this dissertation. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

ROBUST DISCRIMINATIVE ANALYSIS FRAMEWORK FOR GAZE AND HEAD-POSE ESTIMATION

Doctor of Philosophy, 2019, Salahaldeen Rabba

Electrical and Computer Engineering, Ryerson University

Abstract

Head movements, combined with gaze, play a fundamental role in predicting a person's action and intention. In non-constrained head movement settings, the process is complex, and performance can degrade significantly in the presence of variation in head-pose, gaze position, occlusion and ambient illumination. In this thesis, a framework is therefore proposed to fuse and combine head-pose and gaze information to obtain more robust and accurate gaze estimation.

Specific contributions include: the development of a newly developed graph-based model for pupil localization and accurate estimation of the pupil center; the proposal of a novel iris region descriptor feature using quadtree decomposition, that works together with pupil localization for gaze estimation; the proposal of kernel-based extensions and enhancements to a fusion mechanism known as Discriminative Multiple Canonical Correlation Analysis (DMCCA) for fusing features (proposed and traditional) together, to generate a refined, high quality feature set for classification; and the newly developed methodology of head-pose features based on quadtree decompositions and geometrical moments, to better integrate roll, yaw, pitch and jawline into the overall estimation framework.

The experimental results of the proposed framework demonstrate robustness against

variations in illumination, occlusion, head-pose and is calibration free. The proposed framework was validated on several datasets and scored: 4.5° using MPII, 4.4° using Cave, 4.8° using EYEDIAP, 5.0° using ACS, 4.1° using OSLO and 4.5° using UULM datasets respectively.

Acknowledgements

I am grateful to my parents, Ribhi Rabba and Rabiah Rabba, for their unconditional support throughout my life and especially during the ups and downs of this dissertation.

I am grateful to the many individuals who have contributed towards shaping this dissertation. I would like to express my appreciation to Professor Ling Guan for his advice during my doctoral research endeavor for the past four years. As my supervisor, he has constantly advised me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. I thank him for providing me with the opportunity to work with a talented team of lab mates. I would like to thank Dr. Yifeng He, who helped me kick start my doctoral research. Surprisingly, after a few months of his supervision, he announced his plan of joining an outside company, which triggered my independent exploration and self-dependence. This enabled me to excel above and beyond my expectations. I would also like to thank Dr. Matthew Kyan and Dr. Lei Gao for their contributions. I would like to thank Ryerson University, the administrative and technical staff members of the university (especially Dawn Wright) who have been kind enough to advise and help in their respective roles.

I would like to express my sincere thanks to Yasmeen Almukamis, for generously sharing her time and skills in polishing the presentation of my dissertation. I thank Dr. Ali Zandi and Dr. Azhar Quddus (Alcohol Countermeasure System Corp. Toronto, Canada) for sharing information on the gaze estimation application and providing data that enabled me to develop my research.

Table of Contents

List of Tables

List of Figures

List of Acronyms

HCI	**Human Computer Interaction**
HCC	Human Computer Communication
DAF	Discriminative Analysis Framework
DMCCA	Discriminative Multiple Canonical Correlation Analysis
K-DMCCA	Kernel Discriminative Multiple Canonical Correlation Analysis
CCA	Canonical Correlation Analysis
PCA	Principal Component Analysis
MSE	Mean Square Error
LBP	Local Binary Patterns
QD	Quadtree Decomposition
IR	Infrared
EMDR	Eye Movement Desensitization and Reprocessing
WHO	World Health Organization
PTSD	Post-Traumatic Stress Disorder
GT	Gaze Tracking
VF	Visual Field
ALR	Adaptive Linear Regression
RBF	Radial Basis Function
CNN	Convolutional Neural Network
AR/VR	Augmented Reality/ Virtual Reality
ROI	Region of Interest
SNR	Signal-to-noise Ratio
GEV	Generalized Eigen Value
FOV	Field of View
SVM	Support Vector Machine
COM	Center Of Mass
FITECOC	Fit Multiclass Error-Correcting Output Code
RESubPredict	Response Ensemble reSubstitution Predict
t-SNE	t-distributed Stochastic Neighbor Embedding
$\widetilde{\boldsymbol{X}}$	Mapping features
$\widetilde{\boldsymbol{C_{\tilde{x}_k\tilde{x}_k}}}$	Cross-correlation matrix

$\widetilde{C}_w$	Correlation matric within one class
$\widetilde{C}_b$	Correlation matric between classes
$\widetilde{w}^T$	Projected features matrix
V	Cartesian product
π_z	The reflection operator in the z axis
λ_z	The rotation operator around the z axis
S_i	Spatial distribution of polynomial
m_{pq}	Geometrical moment of order (p,q)
M_{01}, M_{10}	Zeroth order geometrical moment
μ_{20}	2nd order geometrical moment
φ	Angle of the principal axis with respect to the x axis
Sk_x	skewness of an object
M'_{pq}	Transformational, reflectional or rotational moment
$Pt_{E\ Right_x}$	Center of right eye along x axis
Pt_{Nose_x}	Center of nose along x axis
ζ_i	Slack variable
ϕ	Kernel mapping function

Chapter 1

1.1 Introduction

Eye movements is one of many ways to get information about a person's thoughts and intentions [1]. Thus, the study of eye movement may help determine what people are thinking based on where they are looking. Eye tracking is the measurement of eye movement/activity and gaze (point of regard and understanding a person's interest or intent), while gaze tracking is the analysis of eye tracking data with respect to the head/visual scene. Researchers of this field often use the terms eye-tracking, gaze-tracking or eye-gaze tracking.

The integration of eye and head-pose position is used to compute the location of the gaze in the visual scene. Simple eye trackers report only the direction of the gaze relative to the head (with head-mounted systems, electrodes, scleral coils) or for a fixed position of the eyeball (systems which require head fixation). Such eye tracking systems are referred to as intrusive or invasive systems because some special contacting devices are attached to the skin or eye to catch the user's gaze. The systems which do not have any physical contact with the user and the eye tracker apparatus are referred to as non-intrusive systems or remote systems. Eye movement tracking techniques are constructed by either measuring the position of the eyes in relation to the head-pose or by measuring the orientation of the eye in space -the point of regard (POR) -used to identify fixated elements in a visual scene. In some cases, gaze estimation is used in the field of regard (FOR), which is the total area that can be captured by both eyes. It should not be confused with the field of view (FOV), which is the angular cone perceivable by a person's eyes at a particular time instant. When a user looks at a 2D screen, such as a desktop, the depth is fixed. So, the gaze direction from one eye is sufficient to determine a 2D gaze point. However, if we want to get a 3D gaze point, such as looking at a point in the real world, we need the depth, see Fig. 1.1. A 2D gaze is defined when obtaining (x,y) coordinate of the point of gaze on a screen. However, the depth element, (x,y,z) coordinate, gives more data about the other perspective (3D) such as thickness, illumination and orientation.

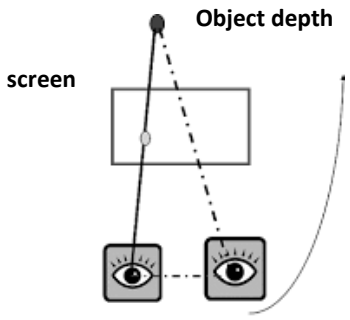


Figure 1.1: Gaze direction and gaze depth [1].

The gaze location of a user depends both on the gaze direction and on the head orientation [1]. The illustration in Fig 1.2, presents the relationship between user reference gaze directions $d_{k_{ref}}$, head-pose direction d_k and actual gaze direction $d_{k_{gaze}}$ which is a result of both head and eye rotation. The effect of head movement has to be compensated for prior to applying the gaze mapping algorithm.

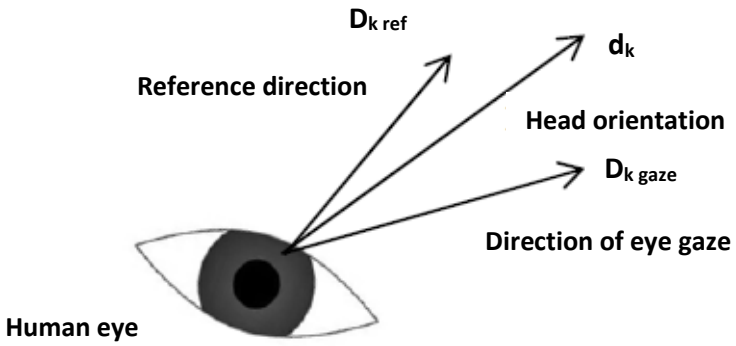


Figure 1.2: Illustration of the relation between gaze direction and head orientation (head-pose) [1].

The design of robust and accurate gaze-tracking systems is one of the most important objectives in the eye-tracking field. The sensitivity to calibration in traditional gaze estimation methods face some challenges: errors in model parameters, noise in pupil center estimation, and head fixation errors during calibration. There is a need to attempt to determine if subject calibration can be eliminated and there is a need to reduce the amount of hardware needed to solve for head-pose and gaze estimation.

1.2 Background

The advancement of eye gaze tracking technology has led to the development of head-pose and gaze estimation techniques and applications. Invasive eye tracking techniques were introduced, which included electro-oculography using pairs of electrodes placed around the eyes or the scleral search methods which involved coils embedded into a contact lens adhering to the eyes. The first video-based eye gaze tracking study was made on pilots operating airplane controls [37]. With increasing computing power in devices, real time operation of eye gaze trackers became possible. Rapid advancements in computing speed, digital video processing and low cost hardware brought eye gaze tracking equipment closer to users, with applications in gaming, virtual reality and web-advertisements [38,39]. As discussed earlier, gaze estimation is concerned with tracking the eyes (iris/pupil) along with head orientation (head-pose).

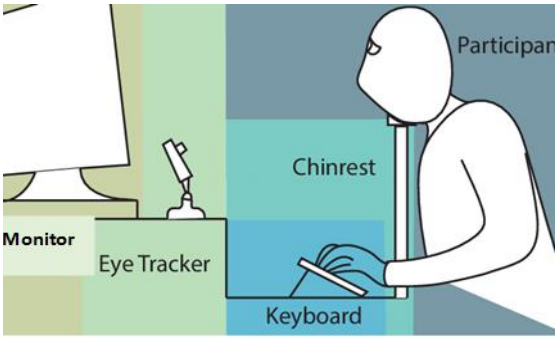
The pupil is one of the darkest areas in the eye region, characterized by its high pixel intensity compared to the surrounding area (eyebrows, eyelashes etc.). The pixel intensity-based approach can effectively locate the pupil center and requires low computational complexity. The intensity values of the pixels in the eye image range from 0 to 255, where 255 represents the highest intensity (black pixels) and 0 represents the lowest intensity (white pixels). The target pixel (the pupil center pixel) has a high intensity value. The pupil center pixel is identified by contrasting intensity between the different areas of the eye region, with the center of the pupil hypothesized to have the highest pixel intensity. Identifying the pupil center is a step towards eye gaze estimation, which plays an essential role in human-computer interaction (HCI). Gaze tracking has a variety of applications. For example, gaze tracking can be utilized to detect distracted drivers. A comprehensive survey of the earlier works can be found in [40] and [41]. However, it is difficult to localize the pupil center for the cases in which the pupil is partially covered by the upper and lower eyelids. Soelistio et al. analyzed each part of the face separately, including the pupil center [42]. Lu et al. located the center of gaze by mapping high-dimensional eye image features to low-dimensional gaze position, and then used an adaptive linear regression (ALR) for testing [43]. Leo et al. proposed an unsupervised eye pupil localization technique using differential geometry and local self-similarity matching [45]. Pupil/iris localization is an essential step towards eye movements and gaze estimation. Eye movements are analyzed as cognitive indicators of visual attention and thought process of a

person [45]. The diversion of attention or lack of focus is accounted for in the eye movements or gaze.

1.3 Methods of Eye Gaze Tracking

Video-based eye gaze tracking systems comprised fundamentally of one or more digital cameras, near infra-red camera and a computer with a screen displaying a user interface where the user gaze is tracked. The steps commonly involved in passive video-based eye tracking include user calibration, capturing video frames of the face and eye regions of the users, eye detection and mapping with gaze coordinates on screen. The user interface for gaze tracking can be active or passive, single or multimodal [46,47]. In an active user interface, the user's gaze can be tracked to activate a function and gaze information can be used as an input modality. A passive interface is a non-command interface where eye gaze data is collected to understand user's interest or attention. Single modal gaze tracking interfaces use gaze as the only input variable whereas a multimodal interface combines gaze input along with mouse, keyboard, touch, or blink inputs for command.

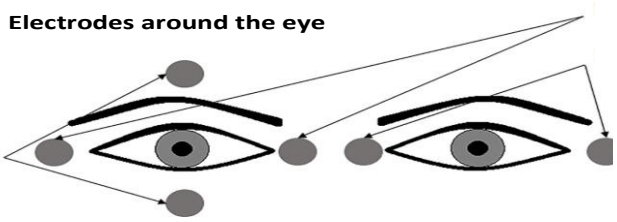
An example of devices used in the market for eye gaze tracking is shown in Fig. 1.3. These devices may include a chin-rest to fix head movement (Fig. 2.1.a), head mounted video-based eye gaze tracker (Fig. 2.1.b), a scleral ring used as a contact lens coil (Fig. 2.1.c) or using electrodes placed around the eye to measure the skin's potential differences (Fig. 2.1.d).



(a) Head-fixed set-up (using chin-rest) to provide accuracy in object's eye gaze tracking estimation.

(b) Head mounted video-based eye gaze tracker, suitable to graphical interactive systems.

(c) Scleral ring used as a contact lens coil, which tracks gaze movements.



(d) Electro-oculography relies on measurement of skin's differences, using electros placed around the eye.

Figure 1.3: Examples of devices used in the market for eye gaze tracking [46,47].

The first step is to detect the eye location in the image. Based on the information obtained from the eye region and possibly head-pose, the direction of gaze can be estimated. The most important parts of the human eye are: the pupil – the aperture that lets light into the eye, the iris – the colored muscle group that controls the diameter of the pupil and the sclera – the white protective tissue that covers the remainder of the eye. Eye gaze detection and tracking remains a very challenging task due to several unique issues, including illumination, viewing angle, occlusion of the eye, head-pose etc. Two types of imaging processes are commonly used in eye gaze tracking: visible and infrared spectrum imaging. Infrared eye gaze tracking typically utilizes either bright pupil or dark pupil technique [47]. In this thesis, we focus on gaze estimation methods based on analysis of the image data. These methods are broadly grouped into feature-based and appearance-based gaze estimation.

1.3.1 Feature-based Gaze Estimation

Feature-based methods explore the characteristics of the human eye to identify a set of distinctive features of the eyes like contours (limbus and pupil contour), eye comers and cornea reflections. The aim of feature-based methods is to identify informative local features of the eye that are generally less sensitive to variations in illumination and viewpoint [48]. These systems have performance issues in the outdoors or under strong ambient light. In addition, the accuracy of gaze estimation decreases when accurate iris and pupil features are not available. There are two types of feature-based approaches that exist [49]: model-based (geometric) and interpolation-based (regression-based).

1.3.1.1 Model-based approaches

Model-based approaches use an explicit geometric model of the eye to estimate 3D gaze direction vector. Most 3D model-based (or geometric) approaches rely on metric information and thus require camera calibration and a global geometric model (external to the eye) of light sources, camera and monitor position and orientation. Most of the model-based methods follow a common strategy; first the optical axis of the eye is reconstructed in 3D, then the visual axis is reconstructed and finally, the point of gaze is estimated by intersecting the visual axis with the scene geometry. Reconstruction of the optical axis is done by estimation of the cornea and pupil center. By defining the gaze direction vector and integrating it with information about the objects in the scene, the point of gaze is estimated [49]. For 3D model-based approaches, gaze directions are estimated as a vector from the eyeball center to the iris center [46,50,51,52].

1.3.1.2 Interpolation-based approaches

Interpolation-based methods assume the mapping from image features to gaze co-ordinates (2D or 3D) have a particular parametric form such as a polynomial or a nonparametric form as in neural networks. Since the use of a simple linear mapping function in the first video-based eye tracker [53], polynomial expressions have become one of the most popular mapping techniques [47,54,55]. Interpolation-based methods avoid explicitly modeling the geometry and physiology of the human eye but instead describe the gazed point as a generic function of image features. Calibration data are used to calculate the unknown coefficients

of the mapping function by means of a numerical fitting process, such as multiple linear regressions. As an alternative to parametric expressions, neural network-based eye gaze trackers [56,57,58] assume a nonparametric form to implement the mapping from image features to gaze coordinates. In these approaches, the gaze tracking is done by extracting the coordinates of certain facial points and sending them through a trained neural network, its output is the coordinates of the point where the user is looking.

1.3.2 Appearance-based Gaze Estimation

Appearance-based methods detect and track eyes directly based on the photometric appearance. Appearance-based techniques use image content to estimate gaze direction by mapping image data to screen coordinates [59,60]. The major appearance-based methods [61] are based on morphable model [62], gray scale unit images [52], appearance manifold [63], Gaussian interpolation [64] and cross-ratio [65]. Appearance-based methods typically do not require calibration of cameras and geometry data since the mapping is made directly on the image contents.

1.4 Applications

Eye gaze estimation has a wide range of applications. Examples of such applications are listed below.

1.4.1 Eye tracking used in the applications of drowsiness detection

Drowsiness is the transition between an alert, awake state and sleep during which one's abilities to observe and analyze are substantially reduced. The authors in [23] present an algorithm for drivers' drowsiness detection based on visual signs that can be extracted from the analysis of a high frame rate video. A study of different visual features is proposed to evaluate their relevance to detect drowsiness by data-mining. Then, an algorithm that merges the most relevant blink features (duration, percentage of eye closure, frequency of the blinks and amplitude-velocity ratio) using fuzzy logic is employed.

1.4.2 Iris recognition and enhancement for biometric applications

In many images, the iris is often occluded by the eyelid and partially by the eyelashes. If these noises cannot be removed, this negatively impacts the performance of the iris recognition system. Similarly, low contrast and non-uniform brightness will also increase the difficulty of feature extraction and matching [24].

1.4.3 Eye typing for physically disabled individuals

Eye typing provides a means of communication for persons with severe disabilities, and those who are only capable of moving their eyes. The authors in [25] consider the features, functionality and methods used in the eye typing systems developed in the literature.

1.4.4 Cognitive and behavioral therapy

The authors in [26] present a study that aims to evaluate the efficiency and flexibility of virtual reality as a therapeutic tool in the confines of a social phobia behavioral therapeutic program. The study's goal is to use the confines of virtual exposure to objectively evaluate a specific parameter present in social phobia, namely eye contact avoidance, by using an eye-tracking system.

1.4.5 Visual search/ marketing/advertising

Familiarity with the distractors around an unfamiliar target facilitates visual search. Four experiments are examined in [27]: (a) shorter and fewer, (b) shorter, but more abundant, (c) equally long, but fewer, or (d) longer, but fewer when distractors are familiar. In a fifth experiment, a gaze-contingent moving window paradigm is used to control peripheral processing. Results reveal a wider span of effective processing for familiar distractors.

1.4.6 Psychology and Neuroscience

During normal vision, when subjects attempt to fix their gaze on a small stimulus feature, small fixational eye movements persist. The authors in [28] recorded the impulse activity of single neurons in primary visual cortex while the individual's fixational eye movements moved over and around a stationary stimulus.

Recent studies of eye movements in reading and other information processing tasks, such as music reading, typing, visual search, and scene perception, are reviewed in [29]. The major emphasis of their review is on reading as a specific example of cognitive processing. Basic topics discussed with respect to reading are (a) the characteristics of eye movements, (b) the perceptual span, (c) eye movement control, and (e) individual differences (including dyslexia). Similar topics are discussed with respect to the other tasks examined. The basic theme of their review is that eye movement data reflect moment-to-moment cognitive processes in the various tasks examined.

The mind can track not only the changing locations of moving object, but also can predict the next move of the object. By tracking the movement of a ball for instance, gaze locations can be continuously recorded with a video-based eye tracker and we can make predictive saccades to the locations where we expect the ball to land.

1.4.7 Human Computer Interaction (HCI)

An eye tracking study was conducted in [28] to evaluate specific design features for a prototype web portal application. Each participant navigated across multiple web pages while conducting six specific tasks, such as removing a link from a portlet. Specific experimental questions included (1) whether eye tracking-derived parameters were related to page sequence or user actions preceding page visits, (2) whether users were biased to traveling vertically or horizontally while viewing a web page, and (3) whether specific sub-features of portlets were visited in any particular order.

1.4.8 Gaze-Mind Connection

Figure 1.4: The Eyes-Mind Relationship. the relationship between eyes and the brain starts in the first days of life [30].

Sight is so crucial that a main part of the brain is dedicated to vision and seeing, see Fig. 1.4. Conventional medicine shows that specific eye movement patterns may relate to mental health conditions. Moreover, certain existing literature suggests that mental health conditions involving attention (such as ADHD, dyslexia and anxiety) are accompanied by increases in erratic eye movements [30].

Consciousness experimentation suggests that eyes and breathing patterns directly influence one's mental and emotional state. It is easier to analyze and measure eye movements and breathing levels, rather than brain activity [30] .

Scholars in the field of psychology are developing theories and methodologies based on the same principle; focusing one's eyes allows for focusing one's mind. One of the theories is EMDR (Eye Movement Desensitization and Reprocessing), which is a modality for treating trauma. Mental and emotional states affect eye movements. They can also affect the mind, and even manage trauma, by doing certain practices with the eyes [30].

The eyes and eyelids are constantly making small, subtle movements, to make sure that the image falling onto the retina is constantly changing (this is called Troxler's Phenomenon). The eyes do this so that the object in one's field of vision continues being registered by the brain; otherwise, by constantly staring at an object for long enough, it tends to disappear from one's perception. In fact, the eyes can focus on multiple things every second. In today's modern world, the eyes are restless more than ever with the intensive use of computer and smartphones, which may be contributing to shorter attention spans. By stilling the micro movement of the eyes, it is argued that stilling of the mind could occur [30].

Neuropsychology research shows that there is a definite relationship between eye position and the dominant hemisphere of the brain; so much so that changing eye position can directly affect one's mood and experience of the world. The left hemisphere is activated when the eyes gaze to the right, and the opposite is true. It is reasonable to assume that holding a perfectly centered and forward gaze produces a balanced brain activity in both hemispheres.

The relevance of this observation can explain the importance of activating the whole brain by holding a central gaze; a practice commonly used in meditation [30]. Gaze meditation technique involves focusing the eyes (and, in turn, the mind) through intent but relaxed gazing. Initially, this practice is done with open eyes on an external object. It then progresses to internal practice (with eyes closed), and to gazing in to space.

Gaze meditation may contribute to the following benefits [30]: improving concentration, memory, visualization skills, cognitive function, symptoms of some eye diseases, strengthening the eyes, eye clarity, insomnia, clearing mental/emotional complexes, calming the anxious mind and enhancing self-confidence and patience.

1.5 Motivation and Challenges

Figure 1.5: Illustrates variations of head-pose angles, illumination and gaze positions [6].

An individual's visual gaze is important in many applications in human computer interaction (HCI) and human behavioral analysis [1]. An example of variations of head-pose angles, illumination and gaze positions is presented in Fig. 1.5. In applications where human activity is under observation from a fixed camera, the estimation of visual gaze provides important information about the interest and intent of the subject, which is commonly used as control devices for persons with disabilities [2,3], to analyze the individual's attention while driving [4,5] or for videogame applications. Studies have shown that visual gaze is a field of two contributing factors [6]: the head-pose and the eye locations. The estimation of these two factors is often achieved using expensive, bulky or limiting hardware [7]. Therefore, the problem is simplified by either considering the head-pose or the eye center

locations as the only feature to understand the gaze intention of a user [8,9]. There is an abundance of literature concerning these two distinct topics: recent surveys on head-pose and eye center location estimation can be found in [10] and [11]. There are two commonly researched fields for head-pose and gaze estimation: appearance-based and geometrical-based methods. The appearance-based methods for eye location proposed in literature, [12,13,14,15], present evidence that accurate appearance-based eye center localization can be used for various gaze related applications. However, we are concerned with studies performed on the feasibility of an accurate appearance-based visual gaze estimation which considers both head-pose and eye location. Therefore, the goal is to build a framework capable of analyzing the visual gaze of an individual starting from video frames or images. This allows for the analysis of the movement of the individual's head and eyes in a more natural manner than traditional methods, as there are no additional hardware requirements needed to implement the framework.

The eye location and eye center algorithms found in commercially available eye trackers share the problem of sensitivity to head-pose variations and require the individual to be either equipped with a head-mounted device or to use a high-resolution camera combined with a chin rest to limit head movement, as shown in Fig. 1.6.

Figure 1.6: Eye gaze applications in various platforms of eye tracking setups (from left to right): head-mounted eye tracker, participant using a desktop with camera, PC with an eye tracker using chinrest, tablet [10,11].

Appearance-based methods that make use of standard low-resolution cameras are considered to be less invasive and, thus, more desirable in a large range of applications. In [16], an online tracking algorithm employing adaptive view-based appearance models is proposed. The method provides drift-free tracking by maintaining a dynamic set of keyframes with views of the head under various poses and registering the current frame to the previous frames and keyframes.

Within geometrical-based methods, studies in the literature propose to integrate a skin-

tone edge-based detector into a Kalman-filter-based robust head-pose tracker and hidden-Markov-model-based pose estimator [17]. Hu et al. described a coarse-to-fine head-pose estimation method by combining facial appearance asymmetry and 3D head model [18]. A generic 3D face model and an ellipsoidal head model are utilized in [19].

A number of studies that focus on combining head-pose and eye information for gaze estimation are available in literature. Newman and Matsumoto [20] and Matsumoto et al. [21] consider a tracking scenario equipped with stereo cameras and employ 2D feature tracking and 3D model fitting. The work proposed in [22] describes a real-time eye gaze, and head-pose tracker for monitoring driver's attention. The authors use IR (infrared) illumination to detect the pupils and derive the head-pose by building a feature space from them. Most of these gaze estimation methods usually regress gaze directions directly from a single face or eye image. However, due to important variabilities in eye shapes and inner eye structures amongst individuals, these methods obtain limited accuracies and their output usually exhibit high variance as well as biases which are subject dependent. Therefore, increasing accuracy is usually done through calibration.

A novel multi-information head-pose and gaze estimation framework are the aim of this thesis and the motivations are the following:

1) Rather than combining different features/techniques (as proposed in literature) through a sequential combination, we propose a framework that fuses and combines appearance-based and geometrical-based features for head-pose and gaze estimation.

2) The fused multi features are constructed in parallel and not in sequence.

3) In this thesis, the normal working range of the gaze location is extended. The short-comings of the reported eye locators due to extreme head-pose are considered and analyzed.

4) The head-pose and eye location information are used together in a multi information gaze estimation framework, which uses the eyes to adjust the gaze location whilst considering the head-pose.

4) Developing an affordable, reliable and non-intrusive eye estimation framework, which requires minimum hardware (a single camera).

The formulation of human head-pose and gaze is a challenging problem in computer vision

and image processing. It is desirable because the head-pose and gaze position provide necessary information about communicative intent, noticeable regions in a scene based on focus of attention, group detection, crowd behavioral dynamics and tracking [33], and variance detection, leading to the joint analysis (fusion) of multimodal/multi-view information.

Although many attempts have been made to improve information fusion techniques, it is still a challenging field for several reasons. The majority of these reasons arise from the data to be fused, imperfection and diversity of the camera technologies, and the nature of the application environment. Data relevance, conflicting data, data correlation, data dimensionality and data imperfection (data provided by cameras) are all issues that may influence and pose challenges while processing data [34,35].

While many of the above-mentioned problems have been identified and actively investigated, no existing information fusion algorithm is capable of addressing all these problems at once. Furthermore, constructing new features to replace existing features in literature is a complex process.

Although intuition indicates that fusion of multi-feature data should help in many information processing tasks, this is not always true. The major difficulties lay in the design of a fusion system that can effectively correlate the information presented in different features.

It is important for a fusion method to be able to identify the discriminatory representation amongst different features [36]. In addition, multi-feature data may carry redundant or even contradictory information.

Occlusions such as facial hair, bangs, eyelids occluding the iris, accessories like glasses and/or low resolution make the estimation computationally difficult and complex.

In summary, we address these problems by presenting the framework and implementation of a new technique that:

- exploits high and low resolution images as well as multiple imaging modalities, i.e. RGB and depth where possible;
- presents alternative state of the art features that results in higher classification accuracy.

Although many investigations have been carried out to address these challenges, the performance of information fusion systems are still not satisfactory. To obtain good multi-feature fusion performance, this thesis focuses on developing a novel framework to tackle the above-mentioned challenges.

1.6 Objective

Enhancing gaze estimation, in terms of accuracy and head movement tolerance, is one of the most cited objectives in gaze estimation technology. Appearance-based and geometry-based methods investigate gaze estimation models that allow for free head movement and that are based on mathematical and geometrical principles.

An example of typical gaze and head-pose estimation is shown in Fig. 1.7 presenting a system that locates the face and eye positions, then applying a modified appearance-based technique for face and eye detection [31]. Another example of a head-pose and gaze estimation is presented in [32], which shows the whole training and estimation pipeline of their system using multimodal convolutional neural networks for appearance-based gaze estimation.

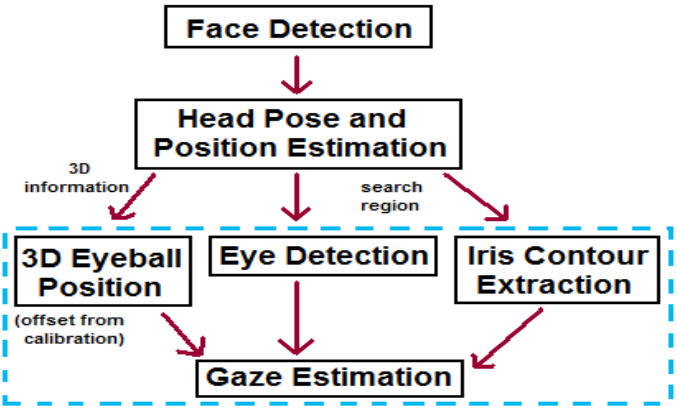


Figure 1.7: Gaze estimation system overview, presenting a system that locates the face and rough eye positions, then applying a modified appearance-based technique for face and eye detection [31].

This thesis is concerned with exploring and finding new features that are more robust and accurate than the existing methods in literature. Figure 1.8 illustrates the general block

diagram of the proposed framework, which shows multiple proposed features are fused together for improved classification.

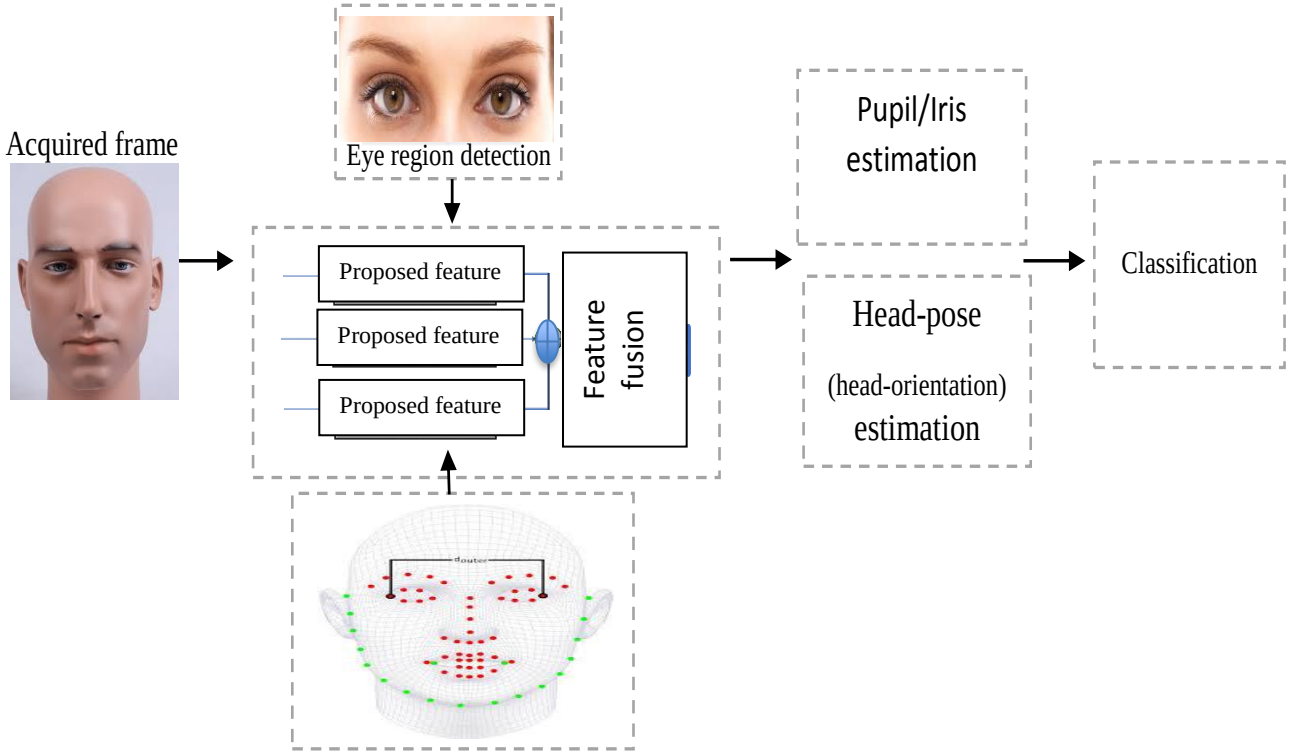


Figure.1.8: The general block diagram of the proposed framework for robust gaze and head-pose estimation. It shows multiple proposed features are fused together for improved classification.

The design of robust and high-performance gaze-tracking systems is one of the most important objectives in the eye-tracking field. In general, a calibration procedure is needed to learn system parameters and to be able to estimate the gaze direction accurately. In this thesis, we attempt to determine if subject calibration can be eliminated. A geometric analysis of a gaze-tracking system is conducted to identify user calibration requirements. This thesis determines the minimal number hardware and user calibration points needed to solve for head-pose and gaze estimation.

The proposed framework is designed to enhance the accuracy of head-pose and gaze estimation, for its application in real life settings. Furthermore, the proposed framework considerably extends its operating range by overcoming the problems introduced by variations of head-pose, occlusion and illumination.

1.7 Contributions

We have identified a gap in the field of unconstrained head-pose and gaze estimation in medium and low resolution images, which we address in this thesis. Each chapter not only presents its own contribution but builds on the previous. However, a high-level overview of the main contributions is as follows:

- We first develop a state of the art graph-based model approach for pupil localization and accurate estimation of the pupil center; a crucial step towards gaze estimation.
- We then employ the pupil localization feature along with another newly developed state of the art iris region descriptor feature using quadtree decomposition, both of which are used for gaze estimation.
- We then fuse and combine these two features with existing head-pose estimation features in literature. In addition, we enhance the performance of DMCCA with the kernel (K-DMCCA) for the purpose of feature fusion, then employed a classifier to provide accurate classification.
- We further develop the framework by replacing the existing head-pose features used in literature with newly developed features (to estimate roll, yaw & pitch and jawline) through the development of quadtree decomposition and the employment of geometrical moments.

1.8 Thesis Organization

This thesis flows from one chapter to the next to narrate the stages taken to complete the structure of the proposed framework. Each chapter will present its own proposed approach, the reason behind the choice and the theory. While identifying its own contribution at its respective stage, each chapter also outlines its contribution towards the thesis as a whole. Each chapter is built is on the previous by extending it and preparing it for the next.

Chapter 2 introduces related work, feature extraction methods/approaches, estimation approaches and fusion.

Chapter 3 presents pupil localization for gaze estimation using an unsupervised graph-based model. We have developed this state of the art feature to estimate the pupil center; an

important step towards gaze estimation.

Chapter 4 presents discriminative robust gaze estimation using Kernel-DMCCA fusion. We employ the feature produced by chapter 3, and by developing a new state of the art feature (iris region descriptor using quadtree) and employing other existing features in literature, we are able to accurately estimate the head-pose and gaze.

Chapter 5 presents robust classifications for head-pose and gaze estimation using quadtree decomposition and geometrical moments. This chapter investigates an alternative state of the art methodology to replace the existing features in literature that were used in chapter 4. This is achieved by extending the region descriptor feature using quadtree which was presented in the previous chapter and developing it further to structure state of the art features, for the purpose of achieving more accurate head-pose and gaze estimation.

Finally,, Chapter 6 draws conclusions.

Chapter 2

2.1 Related Work

A comprehensive survey of gaze and head-pose estimation in the field of computer vision and its applications is summarized in [45]. Recent appearance-based methods have considered gaze estimation in everyday scenarios, which account for different light illumination and head-pose [66]. They do not use the whole face but rather only use image information from one or both eyes. Gaze estimation methods in [67] only study 2D gaze estimation.

Gaze estimation has a wide range of potential applications on tablets, smart phones and hands-free human devices [68]. Cost-effective hardware and gaze tracking are common in gaming, AR/VR and online-advertisements [69]. An eye-tracker called GazeCapture is proposed in [70], which uses a large dataset to train a deep convolutional neural network (CNN) for gaze prediction.

Gaze estimation for identifying points or regions in the real-world consists of many parameters, such as head-pose measured by the angles of the complete facial features [71]. Canonical correlational analysis (CCA) and discriminative multiple CCA (DMCCA) are data driven frameworks for studying the correlation of two or more features [72,73], for efficiently combining a diverse range of free parameters.

Both appearance-based and geometric-based models have been investigated in related studies. Several geometrical-based methods are considered, such as deriving equations to solve for the 3D eyeball center, the 3D pupil center and the 3D visual axis using the tracked facial feature points [74].

Another geometrical-based approach was investigated in [75], where a transformation matrix is obtained from the head-pose and by using the location of both eyes and head information to estimate the gaze. Also, researchers in [76] developed a model-based gaze estimation by eye localization using cascade classifiers and a shape-based approach, limbus

ellipse fitting to identify the 2D limbus positions and point-of-gaze estimation using limbus back projection and gaze smoothing.

Examples of appearance-based methods and training data are as follows: using neural network based gaze tracking to determine the position of a user's gaze from the appearance of the user's eye [77], employing an appearance manifold model using the original set of sparse appearance samples and using linear interpolation among a small subset of samples to approximate the nearest manifold point [78], mapping images to continuous output spaces using Bayesian learning techniques and sparse Gaussian process regression model [79], using single-directional flow model to handle eye image variations due to head motion [80], capturing the user's head-pose and eye images with a monocular camera where samples are adaptively clustered according to the estimated head-pose [81] and by using a 3D rectification process that renders head-pose dependent eye images into a canonical viewpoint to compute the line-of-sight in the 3D space [82].

Gaze estimation is related to head-pose estimation and recent methods have included different head-pose estimations [77,79,81,83]. CNN is used for classification and a regression for gaze estimation with spatial encoding, accommodating various regions [77]. The method used in [84] aims to solve the appearance-based gaze estimation problem under free head-pose, while the approach in [85] presents a novel learning-based method for eye region landmark localization using an appearance-based method.

Adaptive Linear Regression (ALR) is another methodology that uses sparse training samples and L_1 optimization; it is aimed at reducing training samples and improving the accuracy of estimation [86]. In the field of gaze estimation, the eye is the region of interest (ROI). A method to capture ROI is based on the maxima of the derivative and a model-fitting for the ellipse, which includes shape analysis for detecting the iris edges [66].

Gaze estimation depends on the angular precision of the head-pose, thus, roll, yaw and pitch provide 3-degrees-of-freedom to improve the accuracy. Roll is the actual rotation of the head in the frontal plane, yaw is the side to side movement and pitch is along the longitudinal axis of motion.

Generally, head-pose or orientation is based on the symmetrical metrics with three reference planes namely frontal, transverse and sagittal planes. Quadtree decomposition analysis is discussed in literature, however, has not been introduced to the field of head-pose and gaze estimation or to localize the feature in terms of face symmetry. A full face quadtree decomposition may provide a boundary sketch of the image for calculating the angles such as roll, yaw and pitch, based on focus points. We can further analyze the symmetry and rotation aspects of the head-pose using these angles. Considering bilateral symmetry of humans, symmetry can be estimated with full facial features [87]. Potential application of gaze and symmetry analysis is used to detect the behavioral significance in HCI [87].

CNN-based estimation significantly addresses everyday gaze interactions such as head-pose alignments and person independent issues [80]. Portable devices with user experience application testing for enhancing features and content on the screen is becoming very proficient. In immersive environments such as virtual reality (VR) and augmented reality (AR) with digital projections gaze and eye tracking is extremely realistic. Gaze tracking has become common in gaming, AR/VR and online-advertisements [80] due to the advancements in digital signal processing speed and the affordability of cost-effective hardware.

2.1.1 Engineered Features

As will be illustrated throughout this thesis, the proposed framework consists of engineered/designed features. The methods in 2.1.1.1 and 2.1.1.2 are employed and explained below.

2.1.1.1 Local Binary Patterns (LBP)

Local Binary Patterns (LBP) is commonly used in texture analysis and face recognition applications for identification or classification, and it is employed in this thesis to structure features. The LBP technique uses labeling of pixel intensity in the surrounding neighborhood reducing the pixels to a 1 or a 0 [88]. By using a threshold value, the LBP is created using uniform or non-uniform patterns [88]. The LBP creation is done based on directional wavelet decomposition and creating a quadtree for identifying the directional coefficients [88], hence, coupling two sets of operations for operating on multi-region based sub-bands of coefficients. Another algorithm, where quadtree is representing an image or is used to represent an object

and eigen-decomposition is investigated in [89]. This algorithm is tested with objects without occlusions [89], their work relies on training sample accuracy and is unable to detect occlusion or background clutter. Thus, using recursive normalization of quadtree may help overcoming the object detection with reasonable occlusions.

2.1.1.2 Graph model-based information fusion

The graph model combines calculation with graph theory to provide a better tool for structuring features. The graph-based model is employed in this thesis for the purpose of pupil localization. With the help of multi-scale analysis, undirected graph is widely used in scene segmentation, video content analysis, text semantic understanding and so on. Segmentation, detection and tracking of human motion in video are based on the Markov field model. Graph estimation based on grouping/fusion is used to retrieve missing features in multi-channel information, classification of text, video and audio.

2.1.2 Learned Features: Neural network model for multimodal information fusion

The convolutional neural network model has good performance in nonlinear function fitting. The neural networks with deep structure are widely used in speech recognition, man-machine dialogue, machine translation, semantic understanding, object recognition, gesture detection and tracking, human body detection and gaze estimation and tracking. In this thesis, we are comparing the performance of our framework with methods that used Neural network models.

2.1.3 Fusion Methods

This thesis is concerned with fusing multiple features of various information and sizes. Starting from Canonical Correlation Analysis method to Multiple Canonical Correlation Analysis then employing Discriminative Multiple Canonical Correlation Analysis. Therefore, it was worth studying some of the available fusion methods which exist in literature. Below is a list of some fusion methods.

2.1.3.1 Multi-modal information fusion methods

Multi-information analysis is becoming an increasingly important research topic in the field of multimedia. Various types of information channels exist in multi-information interaction. Although, the information acquisition and storage methods of these channels are different, they share some common characteristics in information processing. Starting from single channel information, then progressing towards the processing of multimodal information to developing a framework of multi-modal information fusion. Some of the multi-information fusion methods are presented below [90]:

2.1.3.2 Multiple Canonical Correlation Analysis for Information Fusion

Multi-modal information fusion refers to a process which attempts to achieve more reliable and robust analysis performance by integrating a set of multiple data sources, extracted features, and intermediate decisions. Multi-feature fusion is a special case of multimodal fusion. In multi-feature fusion, different sets of features are extracted from the same modality data but using different extraction methods and it is likely to carry richer information. Therefore, the fusion of the multi-featured sets could lead to better estimation results.

Canonical correlation analysis (CCA) is a statistical method dealing with the mutual relationship between two random information vectors, and a valuable multi information processing method [91,92,93]. However, the effective implementation of multi-information is a challenging problem, when extracting complementary and discriminatory features from different sources. The authors in [73] addressed these challenges, by introducing Discriminative Multiple Canonical Correlation Analysis (DMCCA) as an information fusion framework. DMCCA is capable of simultaneously maximizing the within-class correlation and minimizing the between-class correlation, revealing the intrinsic structure and complementary representations from different modalities to improve the performance, leading to better correlation of the multimodal information.

The DMCCA is formulated as the following optimization problem:

Using canonical correlation analysis (CCA) with two vectors represented as $X = {w_1}^T.x$ and $Y = {w_2}^T.y$, where $w = [{w_1}^T, {w_2}^T]^T$ are the projections or solutions of the problem formulated as:

$$arg \max_{w_1 w_2} \mu = {w_1}^T R_{XY} w_2 \tag{2.1}$$

R_{XY} is the cross-correlation matrices given as XY^T.

Generalizing for a DMCCA, with N set of mapping features X=$[x_1, x_2, x_3, \ldots . x_N]$, find solutions in the form $w^T = [{w_1}^T, {w_2}^T, {w_3}^T, \ldots, {w_N}^T\ ]^T$ that satisfies:

$$arg \max_{w_1\ w_2 \ldots . w_N} \beta = \frac{1}{N(N-1)} \sum_{\substack{k,l \\ k \neq l}}^{N} {w_k}^T C_{x_k x_l}\, w_l \tag{2.2}$$

$$\text{Subject to: } \sum_{k=1}^{N} {w_k}^T C_{x_k x_l}\, w_l = N \tag{2.3}$$

where $C_{\mathrm{x}_k \mathrm{x}_k} = {x_k}^T.x_k$ is cross-correlation matrix, $C_{\mathrm{x}_k \mathrm{x}_l} = C_w - \delta C_b$, $\delta > 0$ with C_w *and* C_b representing the correlation within and between different features, which are written as follows:

$$C_b = -x_i A {x_u}^T\ , C_w = x_i A {x_u}^T \tag{2.4}$$

$$A = \begin{bmatrix} x_{n_{i_1}}.x_{n_{i_1}} & \cdot & \cdot \\ \cdot & x_{n_{i_p}}.x_{n_{i_p}} & \cdot \\ \cdot & \cdot & x_{n_{i_Z}}.x_{n_{i_Z}} \end{bmatrix} \epsilon\, \mathbb{R}^{nxn} \tag{2.5}$$

Eqs. (2.2) and (2.3) are further expressed as follows:

$$\frac{1+\delta}{N-1}(C - D)w = \rho D w \tag{2.6}$$

where C and D are the transformational correlation matrices of multiple sets in the mapping space, as presented in [73], and ρ is the generalized canonical correlation.

The information fusion algorithm based on DMCCA is given as follows: extract information from multi-modal sources to form the training samples, convert the extracted information into the normalized form and compute the matrices C and D then compute the eigenvalues and eigenvectors of Eq. 2.6.

In addition to the above multi-channel information fusion models, there are many other models also used for multi-channel information fusion, such as multi-level support vector machine, decision regression tree and random forest.

2.2 Datasets Description

Several datasets are available. These datasets offer variability with respect to magnitude, head-pose angles, occlusion, illumination and facial appearance. A brief description of these datasets is presented in Table 2.1.

Table 2.1: Brief description of available datasets that are used for validation.

Dataset	Description
CAVE [94]	21 gaze classes operating under lighting changes and occlusions
MPII [79]	Variable appearance, illumination, head -pose
UULM [95]	Wide range of gaze positions, as well as four degrees of freedom of head-pose displacement. The dataset is labeled with vertical and horizontal gaze offsets.
EYEDIAP [96]	Diversity of head-poses, gaze targets. Data collection with Kinect for RGB and depth video streams
OSLO [97]	Variable appearance, illumination, head-pose
ACS [98]	Simulation of real-life driving settings, not available to the public

2.3 Classification

In image processing and machine learning, classification is a learning approach in which the computer program learns from the data input and then uses this learning to classify new observations. This data set may simply be bi-class or multi-class. Some examples of classification problems are: speech recognition, handwriting recognition, biometric identification, gaze classification etc. An example of some of the types of classification algorithms [99]: Naive Bayes Classifier (Generative Learning Model), Logistic Regression (Predictive Learning Model), Decision Trees, Random Forest, Neural Network, K-Nearest Neighbor and Support Vector Machine (SVM). In this thesis, we decided to employ the SVM as a classifier. The SVM is explained briefly in section 2.5.1.

2.3.1 Support Vector Machine (SVM)

In machine learning, support-vector machines are learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick; implicitly mapping their inputs into high-dimensional feature spaces.

Thus far, we have explored existing methods in literature; however, we have identified a gap in the research field and believe we can improve accuracy by introducing state of the art features, for the purpose of head-pose and gaze estimation. The proposed framework aims to use information from the full face (using both eyes) by fusing both appearance-based and geometric-based approaches.

Chapter 3

Pupil Localization for Gaze Estimation Using Unsupervised Graph-Based Model

Overview

In this chapter, we propose a novel graph-based method for pupil localization, which is a step towards gaze estimation. The proposed method can differentiate the key points located at the eyelashes, eyebrows and eye white regions. We first crop the eye region with an ellipse and then estimate the pupil center within the ellipse, thus reducing the computational complexity. We also consider the light reflections in the pupil region, which could lead to inaccuracy in pupil localization. We construct an undirected graph in the eye region based on the following key points: corner points in the eye region, the centers of light reflection regions and the multiple pixels with the highest intensity in the pupil region. The pupil center is initially estimated as the weighted center of the revised graph after vertex/edge removal. In addition, we shift the initial pupil center to a revised position based on the line segments in the pupil region. We evaluate the proposed method on eye images from a public database. The experimental results demonstrate that the proposed method can achieve a more accurate result compared to the existing work in literature.

Different from the existing methods, the proposed method is constructed as follows: we first crop the eye region using an ellipse and then search the pupil center in the ellipse, thus reducing the computation. Second, we construct an undirected graph and then estimate the pupil center based on the trimmed graph. Third, we revise the initial estimation of pupil localization, leading to a more accurate result.

The remainder of this chapter is organized as the following: Section 3.1 provides a detailed description of the proposed method. Section 3.2 provides the performance evaluation and finally, section 3.3 outlines the summary.

3.1. The Proposed Method

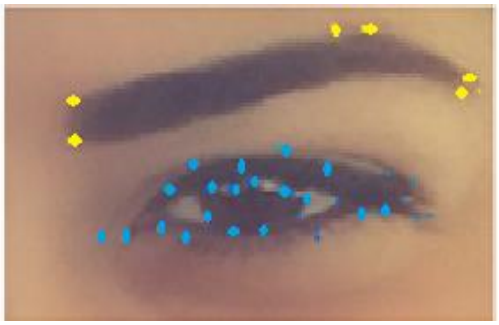


Figure 3.1: The targeted eye region contains a large number of corners.

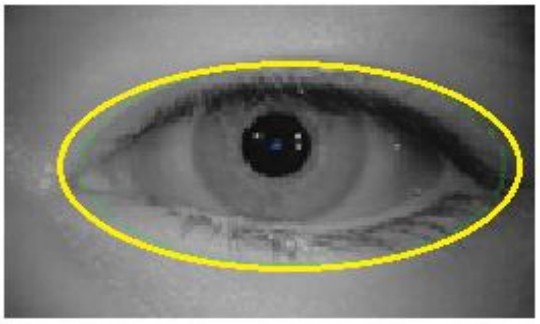


Figure 3. 2: Cropping the eye region with an ellipse.

The proposed algorithm for pupil center detection is structured in five stages. In the first stage, we detect the eye region based on the eye image corners, as will be shown in Section 3.1.1. In the second stage, we fit all corners in the eye region with an ellipse, as shown in Sections 3.1.1 and 3.1.2. In the third stage, we detect the light reflection regions in the pupil area, as shown in Section 3.1.3. In the fourth stage, we identify an enclosed dark region (i.e. pupil, eyelashes), as shown in Section 3.1.4. Stages in Sections 3.1.1-3.1.4 produce point coordinates (referred to as vertices) which are used in the fifth stage, as shown in Section 3.1.5, to establish a graph. We construct a (2n+1) × (2n+1) local patch for each vertex. Each vertex is linked to all other vertices via edges, establishing a complete graph. Each edge has a weight based on the average intensity of pixels along the edge. The graph will direct us to obtaining the location of the pupil center.

3.1.1 Eye Region Detection

A greyscale eye image with $N \times M$ pixels can be represented by: $I = \{I(x_1, y_1), \dots, I(x_i, y_j), \dots, I(x_N, y_M)\}$, where I represents the set of pixels in the greyscale eye image. The intensity value for the pixel located at (x_i, y_j) is given by $I(x_i, y_j)$. The corner detection algorithm [4] finds the corners in a grayscale image. The set of detected corner points is denoted by C. We observe that corner points are gathered in different areas, namely, eyebrows and the eye region (eyelids, pupil contour, eyelashes), as shown in Fig. 3.1. In most cases, it is observed that the eye region has a high number of corner points, as shown in Fig. 3.1. As a result, we direct our attention to the corner points in that region. We adopt the method in [100], which is specific to fitting an ellipse, for cropping the eye region, as shown in Fig.

3.2. The long/short diameters for the ellipse are fixed and the ellipse is designed to fit all different eye sizes, as specified in [100]. The objective of the eye region cropping is to fit the maximal number of the corner points into the fixed ellipse. We start by taking all corner points detected by the corner detection algorithm [101], and then find the mean for all corner points to be the initial center of the ellipse. Corner points within the ellipse are referred to as inliers, and all other corner points outside the ellipse are referred to as outliers. At the initial stage, the corners located at the eyebrows are typically left outside the ellipse because the eye region contains much more corner points. However, some corners of interest near the eye region were left outside of the first ellipse. We repeat the process by taking the mean of the corners inside the first ellipse to get the revised ellipse center. At this stage, we have managed to fit most corners (near the eye region) inside the ellipse, as shown in Fig. 3.2. To optimize the previous process and to ensure that all eye region corners are contained within the ellipse, we revise the ellipse center in an iterative way. The process is illustrated in Fig. 3.3. Taking only the x-coordinate of each outlier corner point, our algorithm will shift the ellipse location horizontally to include more corners within the ellipse. The proposed algorithm will ensure that the number of inlier-corners inside the ellipse is larger than that of the previous iteration. The proposed algorithm will terminate the iteration process if no further inlier points can be added into the ellipse. After termination, the mean of all inliers is chosen as the final ellipse center. The pupil localization will be conducted within the ellipse, rather than the whole image, thus reducing computational complexity.

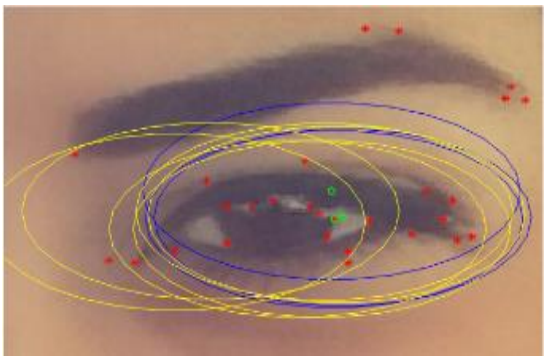


Figure 3.3: Blue ellipse is the initial ellipse. Yellow ellipses are obtained in the revised iterations.

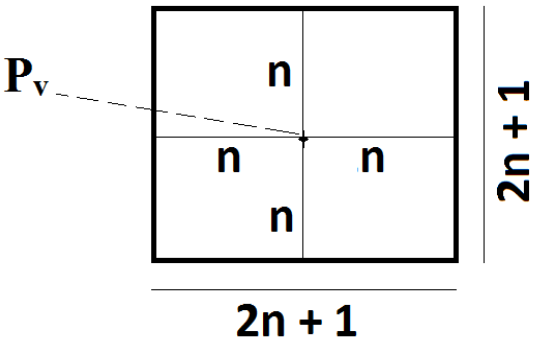


Figure 3.4: Local patch of (2n+1) × (2n+1) centered at each vertex.

3.1.2 Corners Detection

At this stage, we have *k* inlier corners within the final ellipse. The set of *k* inlier corners is represented by:

$$S_c \in I \mid S_c = \{P_1, P_2, P_3, \dots, P_k\}, P_i \in I, i = 1, 2, \dots, k. \quad (3.1)$$

The set of inlier corners S_c will be used to construct the graph-based model, as explained in section 3.1.5.

3.1.3 Light Reflection Detection

In some cases, white regions might be found in the pupil area due to light reflections, which may cause an error in our pupil detection process. To combat this issue, we identify small reflection areas (e.g., white areas with low intensity) within the pupil. Suppose there are z reflection regions in the pupil area, and each center of the reflection region is denoted by P_i. The set of light reflection region's centers is given by:

$$S_o \in I \mid S_o = \{P_1, P_2, P_3, \dots, P_z\}, P_i \in I, i = 1, 2, \dots, z \quad (3.2)$$

3.1.4 Dark Closed Region Detection

In some cases, the contour of the pupil could be smooth, where no corners can be detected. To combat this issue, we identify an area of large connected components that have pixels of high intensity, such as the darkest closed region within the ellipse (i.e. eyelashes, eye corner, and pupil). We randomly select *w* pixels in the dark region with intensity higher than the pre-defined threshold. The set of chosen pixels is denoted by S_d, represented as follows:

$$S_d \in I \mid S_d = \{P_1, P_2, P_3, \dots, P_w\}, P_i \in I, i = 1, 2, \dots, w. \quad (3.3)$$

3.1.5 Graph-Based Model

Consider a graph $G(S_{vertex}, E)$, in which S_{vertex} is the set of vertices represented by:

$$S_{vertex} \in I \mid S_{vertex} = S_c \cup S_o \cup S_d. \quad (3.4)$$

E is the set of edges that link two vertices, represented by:

$$E = \{e_1, e_2, \dots, e_f\}, \; i = 1, 2, \dots, f. \quad (3.5)$$

G is an undirected graph. Every vertex is connected to all other vertices by an edge, thus constructing a complete graph. The weight of an edge, denoted by $W(e_i)$, is represented by the sum of the intensity values for the pixels along the edge.

Each point in S_{vertex} is referred to as a vertex P_v (The vertices are the points in $S_c \cup S_o \cup S_d$). We construct a (2n+1)×(2n+1) local patch centered at each vertex, and use the *average neighborhood intensity* for the local patch, denoted by $B(P_v)$, to describe the vertex P_v, as shown in Fig. 3.4 and Eq. 3.6.

$$B(P_v) = \frac{1}{(2n+1)^2} \sum_{j=-n}^{n} \sum_{k=-n}^{n} I(P_{v\ (x_{+j}, y_{+k})}),\ P_v \in S_{vertex.} \qquad (3.6)$$

We observe that in most cases, the points in S_{vertex} are mostly located in the pupil region, eyelashes and eyelids. Edges are connecting each vertex with all other vertices. The weights of those edges (average pixel intensity along the edge), as well as the value of $B(P_v)$ (average neighborhood intensity of local patch) at each vertex, are of importance and are our focus at this step. It is worth noting that an edge connecting two vertices and passing through the eye white region would have a lower weight in comparison to the edge connecting two vertices located in the pupil region. The undirected weighted complete graph, as in Fig. 3.5, will go through a sequence of modifications as follows:

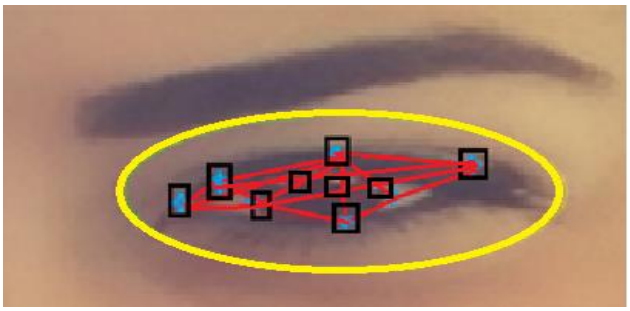


Figure 3.5: Graph consists of edges connecting vertices. A local patch constructed at each vertex.

- Starting at an arbitrary vertex, search the entire set of vertices and remove the vertices with an *average neighborhood intensity* below the threshold value (ignoring seventy percent of the vertex with low a*verage neighborhood intensity* and also having at least 30 vertices in the final selection). The remaining set of vertices is denoted by S'_{vertex} , and each point in the set represented by $P'_v \in S'_{vertex}$

- Perform a search for the edges and remove the edges with the weight below the threshold. The remaining set of edges is denoted by E'.

- Remove the vertices that are isolated from the main graph.

After the graph trimming, we have a revised graph that consists of less vertices and edges, denoted by $G(S'_{vertex}, E')$, where the vertices have a high *average neighborhood intensity*, and the edges have a high weight. It is observed that S'_{vertex} and E' are mostly located in the pupil region. Each P'_v has the coordinates of (X,Y) and the intensity of $I(P'_v)$. We take the weighted average of the coordinates for all vertices in $G(S'_{vertex}, E')$ to be the estimated center ($\overline{X}_c$, $\overline{Y}_c$)of the pupil center, as shown in Eq. 3.7:

$$\overline{X}_c = \frac{\sum_{i=1}^{n}\{I(P'_v).X_i\}}{\sum_{i=1}^{n} I(S'_{vertex})} \quad , \quad \overline{Y}_c = \frac{\sum_{i=1}^{n}\{I(P'_v).Y_i\}}{\sum_{i=1}^{n} I(S'_{vertex})} \tag{3.7}$$

3.1.6 Pupil Center Localization Revision

It is observed in some cases, that ($\overline{X}_c$, $\overline{Y}_c$) may not be close to the pupil center since part of the iris or pupil is covered by eyelashes. To combat this issue, the following revision procedure is proposed, as shown in Fig. 3.6. The revision procedure will generate a revised center, ($\overline{X}'_c$, $\overline{Y}'_c$), that is located closer to the real pupil center. In particular, we have search for the high pixel gradient along eight different directions with 45-degree difference, which results in P_1 to P_8.

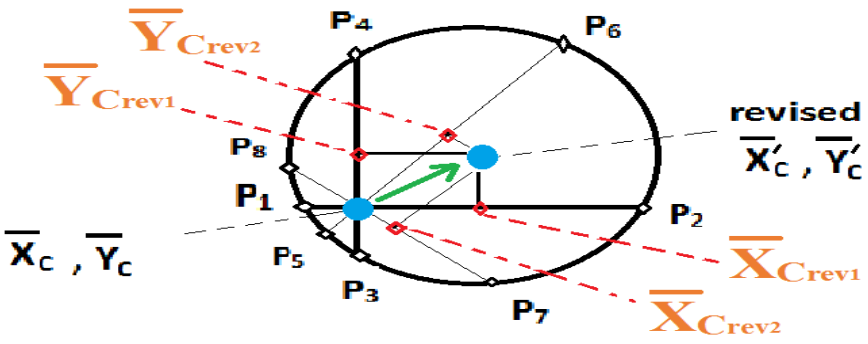


Figure 3.6: Shifting ($\overline{X}_c$, $\overline{Y}_c$) coordinate towards the pupil center.

As shown in Fig. 3.6, the circular area represents the pupil area and it is identified to be the

dark area (high pixel intensity), and the surrounding area is identified to be the white area (low pixel intensity). We construct four line-segments passing through the estimated center ($\overline{X}_c$, $\overline{Y}_c$) obtained by Eq. 3.7, and then find the center of each line segment as follows:

$$\text{Center of line segment from } P_1 \text{ to } P_2\text{: } \overline{X}_{c_{rev1}} = \frac{X_{P1} + X_{P2}}{2} \tag{3.8}$$

$$\text{Center of line segment from } P_3 \text{ to } P_4\text{: } \overline{Y}_{c_{rev1}} = \frac{Y_{P3} + Y_{P4}}{2} \tag{3.9}$$

Eq. 3.8 will shift $\overline{X}_c$ to the midpoint of $P_1 and\ P_2$, and Eq. 3.9 will shift $\overline{Y}_c$ to the midpoint between P_3 and P_4. Furthermore, we find the midpoints for line segment from P_5 to P_6 with the slope of 1 and line segment from P_8 to P_7 with the slope of -1, and then shift $\overline{X}_c$ and $\overline{Y}_c$ to $\overline{X}_{c\_rev2}$ and $\overline{Y}_{c\,rev2}$, respectively. The revised pupil center, denoted by ($\overline{X'}_c$, $\overline{Y'}_c$), is then calculated by:

$$\overline{X'}_c = \frac{\overline{X}_{C_{rev1}} + \overline{X}_{C_{rev2}}}{2}, \quad \overline{Y'}_c = \frac{\overline{Y}_{C_{rev1}} + \overline{Y}_{C_{rev2}}}{2} \tag{3.10}$$

A line segment may pass through the light reflection region, which leads to the error in the estimation of pupil center. Therefore, we take four-line segments instead of two- line segments to improve the robustness and accuracy for the pupil center estimation.

3.2 Experiment

We have used the National Laboratory of Pattern Recognition database [94], to evaluate the accuracy of the proposed method. We applied the proposed pupil localization method on grayscale images with various pupil positions. Among the images, some have dark eyelashes, dark eyebrows, glasses and some have partially occluded pupils. Our experiment was conducted on a desktop computer with the following configurations: 64-bit OS, Intel Core i7 3.07 GHz CPU, and 4GB RAM. The error is expressed in Eq. 3.11.

$$\text{Error} = \frac{|TP|}{|AB|} = \frac{\sqrt{(X_T - X'_P)^2 + (Y_T - Y'_P)^2)}}{\sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2)}} \times 100\%, \tag{3.11}$$

where |AB| is the length of the eye, |TP| is the distance between the ground truth and the estimated pupil center, $\frac{|TP|}{|AB|}$ is a relative error represented in percentage, X and Y are coordinates, T is the pupil center ground truth, and P is the pupil center estimated by the proposed method. The error calculation is illustrated in Fig. 3.7.

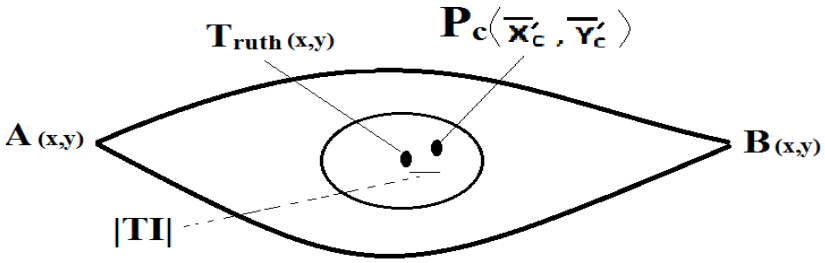


Figure 3.7: Calculating the relative error based on the ground truth.

Table 3.1: Performance comparison between the proposed method and existing method in [102] and [103].

	Proposed method	[102] Circle-based eye center localization	[103] Pupil localization using differential geometry
Average Error	0.4%	8.8%	12.69%
Average Processing Time	0.8 sec	0.2 sec Their system had different configurations than our system	0.33 sec

The comparison of the proposed method against the method in [102] (Circle-based eye center localization) is presented in Fig. 3.8. The error, in Eq. 3.11, was calculated based on the difference in pixels between the estimated pupil center and the ground truth. the As can be seen, the error in the proposed method (green) is much smaller than the existing method (red) [102]. It is worth noting that the method in [102] localizes the pupil center from face images, while the proposed method finds the pupil center from eye images.

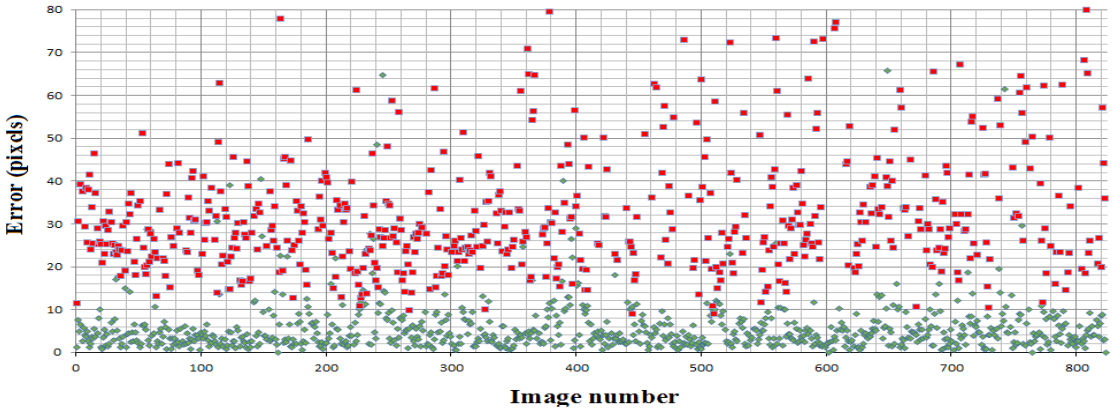


Figure 3.8: Error for our method, in green, against the method in [102], in red.

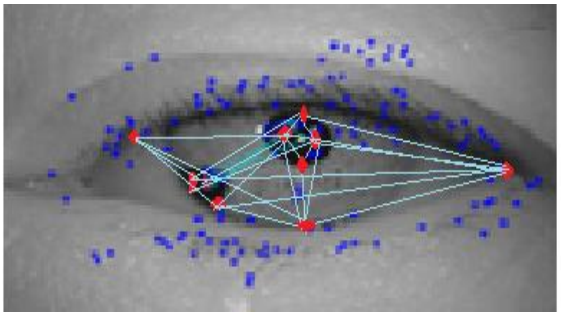


Figure 3.9: A graph that consists of vertices and edges. Red vertices have higher average neighborhood intensity than blue vertices. Edges connect red vertices only.

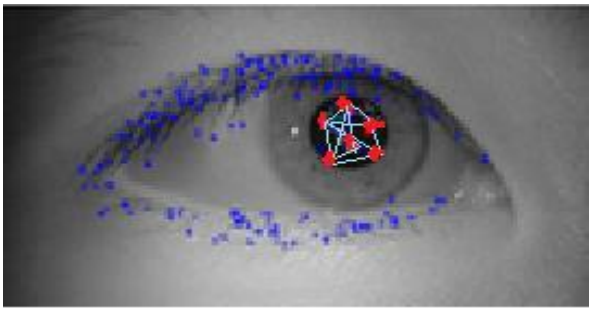


Figure 3.10: Removing the edges with a low weight and keeping the edges with a high weight. Edges on the pupil area have high pixel intensity.

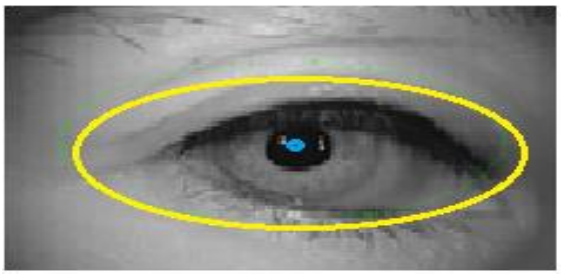


Figure 3.11: Small blue circle indicates the estimated pupil center.

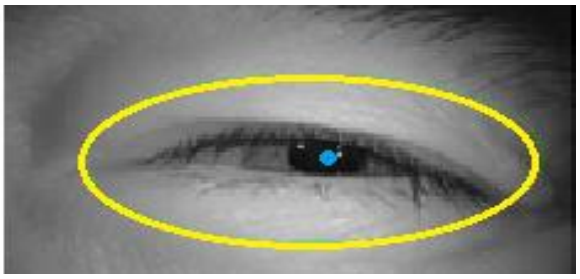


Figure 3.12: Occluded pupil, blue circle indicates the estimated pupil center.

The comparison of average error, average accuracy, and average processing time ,between the proposed method and the method in [102] and [103], is shown in Table 3.1. The proposed method achieves an average accuracy of 99.6%, which is much higher than the method in [102] and [103]. We employ more computation to increase the localization accuracy, thus leading to a larger average processing time than the method in [102] and [103].

We encountered some challenges in the experiment for the individuals with thick eyebrows, which are very close to the eye region, in which the corners at the eyebrows were included as inliers within the ellipse. We also encountered images with no eyelashes, in which we were not able to locate the corners at the eyelashes or dark regions. In some cases, as shown in Fig. 3.9, there are two dark regions within the ellipse, which produced a pupil center location that was not close to the real pupil center. The proposed method can handle such cases, by removing the edges connecting the vertices between the two regions, because edges have a low weight as a result of passing through the eye-white area. Furthermore, the pupil is a bigger region and its vertices have a higher weight. Taking the weighted average of vertices from both dark regions will result in an estimated pupil center that is close to the pupil region. Our revision

procedure further shifts the estimated center closer to the actual pupil center. The case with only one enclosed dark region is shown in Fig. 3.10. The proposed method takes the weighted average of all vertices to get an accurate pupil center localization. A similar example is shown in Fig. 3.11. A difficult case where the pupil was partially covered by the eyelids is seen in Fig. 3.12. The proposed algorithm can deal with such a case successfully with an accurately estimated pupil center.

3.3 Summary

In this chapter, we proposed a graph-model based pupil localization method, which is a crucial step towards gaze estimation. The proposed method can distinguish the key points in the pupil region from those in the other locations. The proposed revision process further improves the accuracy of the pupil center estimation. The experimental results demonstrated that we were able to locate the pupil center with a 99.6% accuracy.

Chapter 4

Discriminative Robust Gaze Estimation Using Kernel-DMCCA Fusion

Overview

In this chapter, we present an algorithm employing discriminative analysis for gaze estimation using kernel discriminative multiple canonical correlation analysis (K-DMCCA), which utilizes multiple feature vectors that account for variations of head-pose, illumination and occlusion. The features used by this algorithm include spatial indexing, statistical and geometrical elements and the feature produced in Chapter 3 (pupil localization). Gaze estimation is constructed by feature aggregation and transforming features into a higher dimensional space then fed into the RBF classifier with kernel γ and a spread factor. The output of fused features through K-DMCCA is robust to illumination, occlusion and is calibration free. The proposed algorithm is validated on MPII, CAVE and EYEDIAP datasets. We also used ACS dataset, a dataset collected at Alcoholic Countermeasure Systems Corp and are not available for the public, in validation. The two main contributions of the algorithm are the following: Enhancing the performance of DMCCA with the kernel and introducing quadtree as an iris region descriptor. Spatial indexing using quadtree is a robust method for detecting which quadrant the iris is situated, detecting the iris boundary and it is inclusive of statistical and geometrical indexing that are calibration free.

The remainder of this chapter is structured as follows: Section 4.2 outlines the proposed algorithm, Section 4.3 details the extracted features, Section 4.4 develops the implementation of K-DMCCA, Section 4.5 presents experimental evaluation, Section 4.6 highlights the results and Section 4.7 summarizes the chapter.

4.1 Proposed Framework

The proposed framework is illustrated in Fig. 4.1. The features for gaze estimation include: Graph-based approach to estimate pupil center [104], gaze vector [105], facial landmark angles [71], 15D eigen values that represent the collective eye region [43] and a quadtree-based iris region descriptor with local and global coordinates. A gaze position/direction is assumed to be of a given class (identifying where the person is looking at). We propose a feature fusion method for gaze estimation that fuses features to give a correlation within and between classes using K-DMCCA. The output of fused features through K-DMCCA is robust to illumination, occlusion and is calibration free. We correlate the feature sets and direction by employing K-DMCCA to study the modulation profile of the extracted feature sets, which gives discriminative correlations between two or more classes.

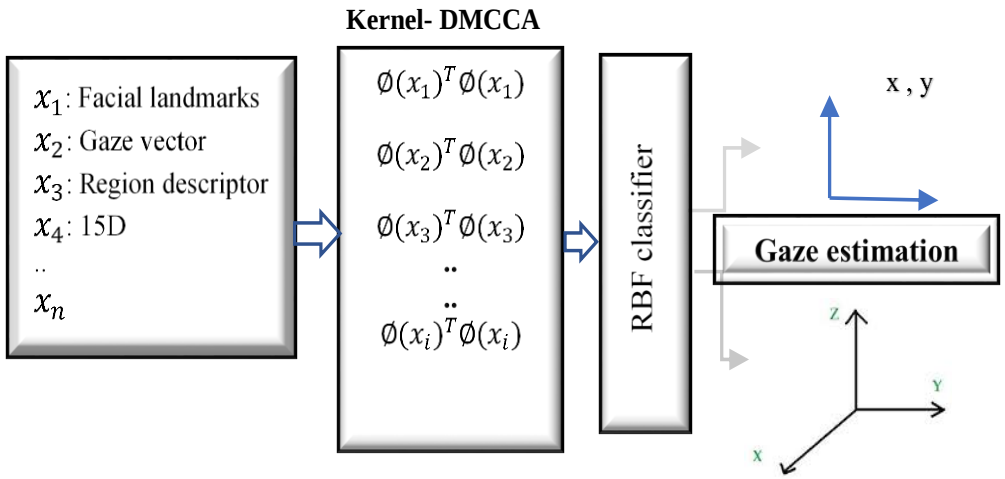


Figure 4.1: The proposed framework consists of extracting difference features, applying K-DMCCA on extracted features to project the extracted features to a high dimensional space and to produce discriminative correlations, for the purpose of 2D and 3D gaze estimation.

4.2 Feature Extraction

4.2.1 Distances and Angles Feature Extraction

The feature in [71] is concerned with the estimation of head-pose and extraction of relative facial feature distances, such as distance from ear-to-nose, distance between the eyes and the distance between the nose base and nose tip. By collecting facial landmark points, this feature is able to calculate the distance/angle between facial landmarks and monitor changes over time [71]. See Fig. 4.8 for illustration. This is referred to as X_1 {feature vector that contains the distances between facial landmarks and the angles between them}.

4.2.2 Eye-center to Pupil-center Vector Feature

The eye is assumed to be an ellipse with three parameters [105]: eye radius, iris radius and relative position of the eye center with respect to the head-pose. The gaze direction is a vector that starts at the eye center and passes through the pupil center. This feature is then merged with the feature in section 4.2.1. This refers to X_2 {feature vector that contains directional information of (x,y) coordinates of gaze vector}.

4.2.3 Iris Region Descriptor Extracted Using Quadtree

Shape descriptors are vectors that define a region of interest (ROI) that is not sensitive to rotation or translation. We propose an iris region descriptor using quadtree decomposition that generates a set of points defining the iris geometrical boundary. This is referred to as X_3 {feature vector that contains (x,y) coordinates on the iris boundary}.

4.2.3.1 Quadtree Structure

A quadtree uses a tree-based spatial indexing data-structure in which each node has exactly four children [89]. This representation is constructed from a root node (representing the entire image) which is progressively sub-divided into 4 quadrants (represented as children of the root), with further sub-divisions as children of each of these quadrants, and so on. Each node is a quadrant consisting of a BLACK pixel that is '1' and a WHITE pixel that is '0'. Each node is labeled as north-west (NW), north-east (NE), south-west (SW) and south-east (SE). Each sub-quadrant has a collection of BLACK and WHITE binary pixel values. Fig. 4.2 shows a 3-level decomposition of an image. The root nodes are {Q_1, Q_2, Q_3, Q_4}. The next level of sub-quadrants is {SQ_1, SQ_2, SQ_3, SQ_4}, while the third level is {sq_{11}, sq_{12}, sq_{13}, sq_{14}}. The quadrants and sub-quadrants are also structured as {NW, NE, SW, SE} which indicates the directional mapping of the pixels in the image data. Homogenous quadrants (uniform BLACK or uniform WHITE) are not stored. Furthermore, each node stores a label/compass point. Only labeled nodes are stored (in the order in which the tree is traversed) and considered for encoding.

We use this spatial indexing to build geometrical features such as a segment or an arc forming a circle. To represent a line segment or an arc of a circle, we count the number of 1s and 0s values. The merging of sub-quadrants, where the iris lies, is based on similar boundary information [106].

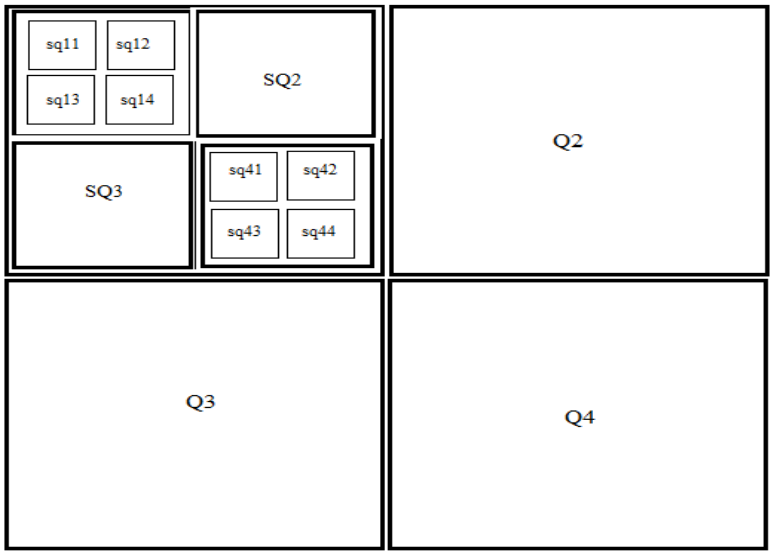


Figure 4.2: Three levels of quadrant and sub-quadrants division for creating a quadtree indexing [89].

Unlike a line segment, which can be identified by continuity of edges, we need to look at a circle as a polygon with a finite set of points situated over the neighboring nodes at a particular level. Consider a quadtree, T, of an image with depth h, each block of size $2^q \; x \; 2^q$, q is an integer. Each node has information about the parent node, the node itself and the direction; along with the data (1s and 0s) of the decomposed pixels. As the level increases so does the depth of the tree, hence we use the depth first algorithm for storing nodes at each level. The iris could be situated in a single or in multiple sub-quadrants at any given level. The narrowing down of possibilities (where the iris is situated) is based on the statistical aspect (by counting the number of 1s and 0s values) of the sub-quadrant with respect to the horizontal, vertical and diagonal sub-quadrants at each level. The merging of the sub-quadrants depends on the data/label driven determination of the iris's geometrical shape.

4.2.3.2 Geometrical Approximation

To select the iris region and its occupancy in the image coordinates, we assume a polygon inscribed in a circle [107]. For a regular polygon of M sides, with a set of vertices C that lie on the curve, we need a dominant set of points:

$$D = \{x_i, y_i\}_{i=1}^{M} \; , \; C = \{x_i, y_i\}_{i=1}^{N} \tag{4.1}$$

where D is a subset of C and $M \leq N$.

To get a measure of such approximation error, we use the integral square error (ISE). Polygon approximations with a number of lines are compared with the angles/curves with an increased

number of points [107]. We use curves with an increased density of points to approximate the polygons with small lengths that result in minimizing ISE. The centering and shifting in the positions of the iris gives an assessment of correctness in terms of estimated pose. On the circumference of a circle, we choose points of the polygon with an angle of separation at least $\frac{2\pi}{M}$, as shown in Fig. 4.3.

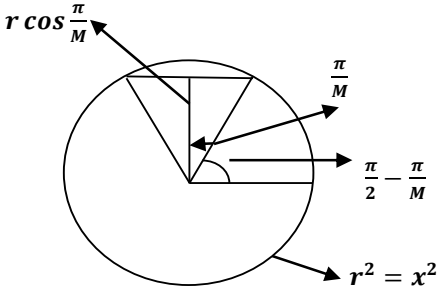


Figure 4.3: Polygon inscribed within a circle of radius r [107].

4.2.3.3 Quadtree Complexity

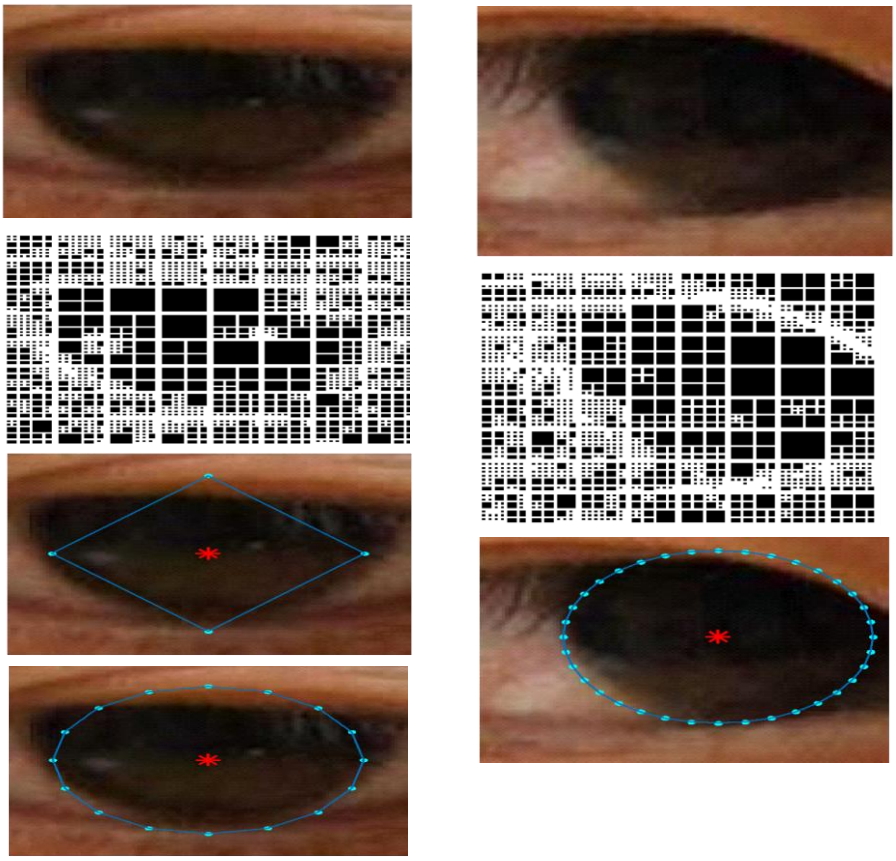
An image P is used to create a tree T with a depth h. We employ the Warnock algorithm, which uses a divide-and-conquer technique to reconstruct hidden objects in an image [106,108]. The Warnock algorithm is used for recognizing a polygonal boundary with perimeter p, resolution parameter q and ϑ number of vertices, with a complexity of Θ(p+ q+ ϑ).

4.2.3.4 Index Extraction

The image representation of the quadtree decomposition, with a pre-defined threshold, presents the background in BLACK pixel region while foreground is WHITE, indicating the iris and cornea part of the eye. An image cropped from CAVE dataset in Fig. 4.4.a, showing quadtree decomposition and how the polygonal approximation is established. Fig. 4.4.b shows spatial shift of iris in 4.4.a, allowing us to shift the center with an offset relative to the earlier frame.

The points for generating the circle are derived from the selection of percentages (statistical count) of 1 and 0 transition values (one-to-zero, zero-to-one) that are obtained after quadtree decomposition. The curvature identification of an iris is based on finding the transition blocks by counting the number of 0-1 and 1-0 (row and column-wise) from the decomposed set. The count determines the selection of neighboring blocks. We take the mean of these transition points count to be the center for generating the circle radius, see Fig. 4.6 for illustration. For generation of the circle, we must verify its boundary and create a ground truth to compare it to.

To do that, we employ an edge operator [108] to create a set of edges and compare points on the edges with the points generated previously by the iris region descriptor. We take all the overlapping points of both procedures and we refer to it as the dominant set. The dominant set is used as a measure of performance and ground truth for the iris.



(a) (b)

Figure 4.4: (a) An image cropped from CAVE dataset, showing quadtree decomposition and how the polygonal approximation is established. (b) Spatial shift of iris in (a), allowing us to shift the center with an offset relative to the earlier frame

4.2.3.5 Iris Ground Truth

Using a standard edge detector [108], we extract the boundary of the iris. However, we have no way of confirming this is the iris. Hence, the dominant set is required. In Eq. 4.2, 'A' represents the number of points of the iris region descriptor while 'B' is the set of points obtained from the edge operator. Edge detection measure is given as:

$$P(I) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \qquad (4.2)$$

$P(I)$ normalizes in the range of [0, 1] as a measure for the dominant set. P(I) =1 implies that all chosen points coincide. If *P(I)*=0 then there are no coinciding points [108].

In an image, points extracted by the edge operator are referred to as μ_i while points of the iris region descriptor are I_i. Maximizing the points, *P*, can be expressed as:

$$P_{combined}(I, \mu) = \{P(I_i, \mu_i)\} \qquad (4.3)$$

Eq. 4.3 is the factor of improvement when obtaining a dominant set and after removing the unwanted edges.

4.2.3.6 Edge Detection Complexity

Grouping of boundaries is extremely challenging in terms of generalization where the detection of the false edges and the inflection points are specific to every image [108]. Thus, we consider an edge operator that provides an improved signal-to-noise ratio (SNR [108]) and localization of edges. We need image-dependent parameters for selecting and improving the dominant set performance by either increasing the edges of the polygon or by shifting the center of an iris that has spatially shifted. Fig. 4.6 shows the combination by overlapping the boundary detection points over the detected edges after the removal of outlier edges.

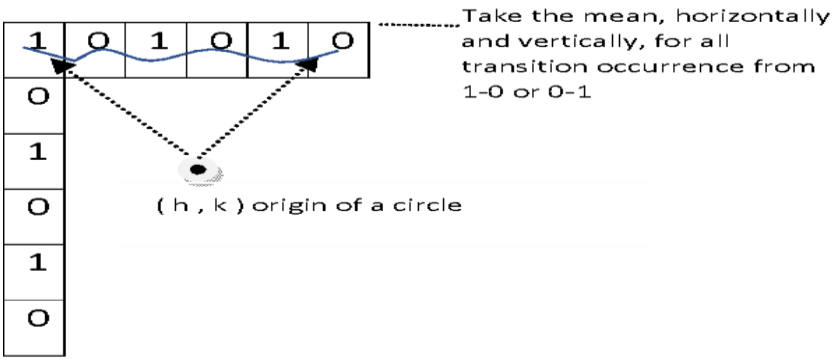
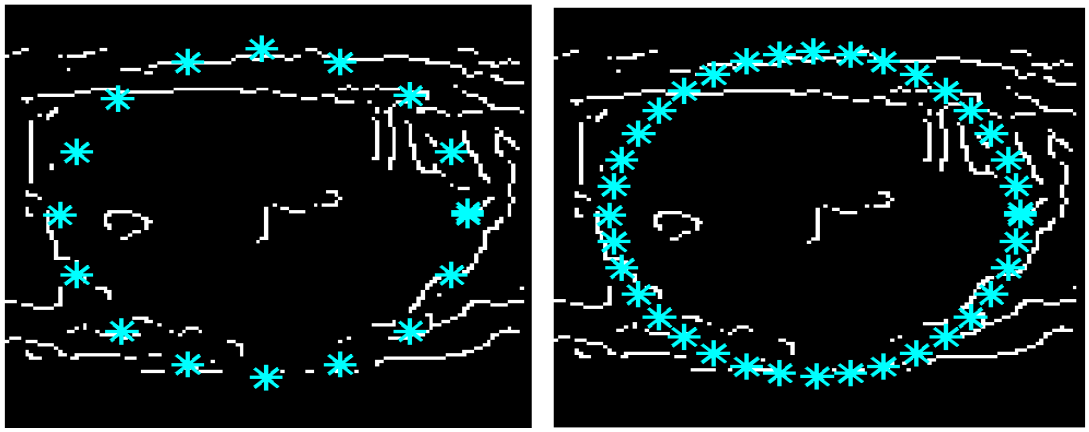


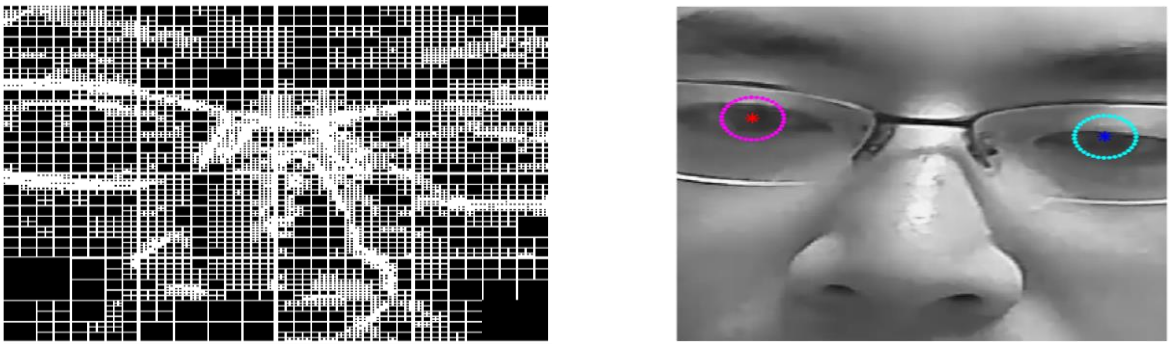
Figure 4.5: Shows how the mean of the transition point count (horizontally and vertically) is taken to be the origin. To calculate the radius, we take the furthest point from the mean to a point on the transition count.



(a) (b)

Figure 4.6: (a) Cyan '*' represents the points generated by the iris region descriptor overlaid over the edges. (b) Increased set of points to cover a greater number of edge points.

To shed some light on the outcome of the iris region descriptor, we demonstrate some results in Fig. 4.4, which shows how the iris has shifted spatially. This allows us to shift the iris center with an offset relative to the earlier frame. Furthermore, Fig. 4.7 illustrates quadtree level-2 decomposition. It clearly shows the background (BLACK) while foreground (WHITE) pixel regions of the image content such as skin and glasses. Using the zero-to-one transition to define the iris circle center and the boundary of each sub-quadrant at the region where there is no transition from 0-1 or 1-0, then radius and circle are established as illustrated in Fig. 4.5 and Fig. 4.7. To establish a circle, Fig. 4.5 shows how the mean of the transition point count (horizontally and vertically) is taken to be the origin. To calculate the radius, we take the furthest point from the mean to a point on the transition count.



(a) (b)

Figure 4.7: (a) Quadtree level 2 decomposition, (b) Blue and pink circles are the Iris boundary for the eyes along with centers.

4.2.4. 15D Feature Extraction

We adopt appearance-based features introduced in [43] because it presented mapping high-dimensional eye image features to low-dimensional without losing classification accuracy. We define the region of interest P x Q, where P and Q are multiplies of 3 and 5, respectively. The region of interest is divided into sub-regions forming a 2D buffer of size 3 x 5. Each block is normalized, as a result, the 2D 3x5 buffer is reduced to 1D (1x15) eigen values that provide a gaze position in the eigen space. Unlike the approach [43] , we took the whole 15D feature vector without applying PCA on it. A 15D intensity feature from the eye region pertaining to 3 x 5 sub-regions is computed as a feature set. See Fig. 4.9 for illustration. This is referred to as X_4 {feature vector that contains 1x15 eigen values}.

4.3 Implementation

4.3.1 Extending DMCCA to K-DMCCA

Multi-feature processing of the dataset involves extracting features and fusing them to perform correlation analysis. DMCCA in [73] is capable of simultaneously maximizing the within-class correlation and minimizing the between-class correlation, revealing the intrinsic structure and complementary representations from different modalities to improve the performance. The integration of statistical and geometrical features aims at unifying and evaluating the associations between variables for improving the accuracy of gaze prediction [72]. Due to the success of DMCCA [73] and multiple features used in our framework, we extend the method in [73] and employ a kernel to project the features to a high dimensional space for better separation and identification, especially on nonlinear distributions.

Using canonical correlation analysis (CCA) with two vectors represented as

$X = {w_1}^T.x$ and $Y = {w_2}^T.y$, where $w = [{w_1}^T, {w_2}^T]^T$ are the projections or solutions of the problem formulated as:

$$arg \max_{w_1 w_2} \mu = {w_1}^T R_{XY} w_2 \tag{4.4}$$

R_{XY} is the cross-correlation matrices given as XY^T.

Generalizing for a kernel DMCCA, with N set of mapping features $\tilde{X}$=[$\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \ldots . \tilde{x}_N$], where $\tilde{x}_N = \emptyset(x_N)^T \emptyset(x_N)$ is the kernelization of x_N, finding solutions in the form $\widetilde{w}^T = [\widetilde{w}_1{}^T, \widetilde{w}_2{}^T, \widetilde{w}_3{}^T, \ldots, \widetilde{w}_N{}^T\ ]^T$ that satisfies:

$$arg \max_{\widetilde{w}_1 \widetilde{w}_2 \ldots . \widetilde{w}_N} \beta = \frac{1}{N(N-1)} \sum_{\substack{k,l \\ k \neq l}}^{N} \widetilde{w}_k{}^T \widetilde{C_{\tilde{x}_k \tilde{x}_l}}\, \widetilde{w}_l \tag{4.5}$$

$$\text{Subject to: } \sum_{k=1}^{N} \widetilde{w}_k{}^T \widetilde{C_{\tilde{x}_k \tilde{x}_l}}\, \widetilde{w}_l = N \tag{4.6}$$

where $\widetilde{C_{\tilde{x}_k \tilde{x}_k}} = \tilde{x}_k{}^T . \tilde{x}_k$ is cross-correlation matrix,$\widetilde{C_{\tilde{x}_k \tilde{x}_l}} = \tilde{C}_w - \delta \tilde{C}_b$, $\delta > 0$ with $\tilde{C}_w$ and $\tilde{C}_b$ representing the correlation within and between different features, which are written as follows:

$$\tilde{C}_b = -\tilde{x}_i \tilde{A} \tilde{x}_u{}^T , \tilde{C}_w = \tilde{x}_i \tilde{A} \tilde{x}_u{}^T \tag{4.7}$$

$$\tilde{A} = \begin{bmatrix} \tilde{x}_{n_{i_1}} . \tilde{x}_{n_{i_1}} & . & . \\ . & \tilde{x}_{n_{i_p}} . \tilde{x}_{n_{i_p}} & . \\ . & . & \tilde{x}_{n_{i_Z}} . \tilde{x}_{n_{i_Z}} \end{bmatrix} \epsilon\, \mathbb{R}^{nxn} \tag{4.8}$$

Eqs. (4.5) and (4.6) are further expressed as follows:

$$\tfrac{1+\delta}{N-1} \left(\tilde{C} - \widetilde{D}\right) \widetilde{w} = \rho \widetilde{D}\ \widetilde{w} \tag{4.9}$$

where $\tilde{C}$ and $\widetilde{D}$ are the transformational correlation matrices of multiple sets in the mapping space, as presented in [73], and ρ is the generalized canonical correlation. Then Eq. 4.9 can be solved as the generalized eigenvalue (GEV) problem.

With the projected matrix of $\widetilde{w}^T = [\widetilde{w}_1{}^T, \widetilde{w}_2{}^T, \widetilde{w}_3{}^T, \ldots, \widetilde{w}_N{}^T\ ]^T$ from Eq. 4.9, the projection of training dataset in K-DMCCA space is calculated as follows:

$$X_{\text{Train}} = \begin{bmatrix} \widetilde{X}_{1\ \text{Train}_{\text{Projection}}} = \widetilde{w}_1{}^T . \ \widetilde{x}_{1\ \text{Train}} \\ \widetilde{X}_{2\ \text{Train}_{\text{Projection}}} = \widetilde{w}_2{}^T . \ \widetilde{x}_{2\ \text{Train}} \\ \ldots \\ \widetilde{X}_{N\ \text{Train}_{\text{Projection}}} = \widetilde{w}_N{}^T . \ \widetilde{x}_{N\ \text{Train}} \end{bmatrix} \tag{4.10}$$

4.3.2 Classification

The output of K-DMCCA is fed into an RBF classifier with the 3-σ rule (three sigma [70]) as a correlation analysis index. The classification is performed by considering the distance between X_{Train} to Y_{Test} with X_{Train} being the projection of training dataset in K-DMCCA space, while Y_{Test} is the projection of the testing dataset in K-DMCCA space. The pseudo code of K-DMCCA and classification is outlined as follows:

1. Extract all features and produce (x,y) tuple coordinates, or (x,y,z) for 3D, corresponding to each feature.
2. Employ K-DMCCA as explained in section A to calculate X_{Train} .
3. For testing, we calculate:

$$Y_{Test} = \begin{bmatrix} Y_{1\,\mathrm{Test_{Projection}}} = \widetilde{w}_1{}^T . y_{1\,\mathrm{Test}} \\ Y_{2\,\mathrm{Test_{Projection}}} = \widetilde{w}_2{}^T . y_{2\,\mathrm{Test}} \\ Y_{N\,\mathrm{Test_{Projection}}} = \widetilde{w}_N{}^T . y_{N\,\mathrm{Test}} \end{bmatrix} \quad (4.11)$$

4. Compare the distance between training data and testing data in K-DMCCA space with Eq. 4.12.

$$K(X_{Train}, Y_{Test}) = e^{\frac{-\gamma \| Y_{Test} - X_{N_{Train_{Projection}}} \|^2}{2\delta^2}} \quad (4.12)$$

   where γ is the spread factor of the RBF.

5. The outcome of K-DMCCA transformation and classification produces a space in the range of [0,1].
6. Compute the 3-σ, which is the probability of 0.68, 0.95 and 0.997.
7. Calculate accuracy of estimation based on the discriminative statistics produced by step 6.

The RBF provides a measure of similarity between the training and testing feature vectors. Eq. 4.12 is the norm of the vectors that brings the points closer or further using the spread factor, it is a bell-curve that changes based on γ value (spread factor [109]).

4.4 Evaluation

To evaluate the proposed algorithm, we conducted experiments on three public datasets: Cave (5880 images) [94], MPIIGaze (3000 images of left and right eyes) [79] and EYEDIAP 2D and 3D (15 frames for each VGA video per participant) [96]. We also used ACS dataset (10 subjects; not available to the public) [98]. The datasets consisted of variations of illumination, occlusion and head-pose. We compared the proposed full-face method with recent existing methods [77], [79], [84] and [85].

We adopted the method in [71] and were able to monitor the head-pose change over time. Fig. 4.8 shows the detection of facial landmarks, the distances and the angles between them. We also adopted the method used in [105], which gave us a clear indication of the gaze direction. We integrated the method [105] into [71], which allowed us to track the head-pose coordinates along with the gaze vector, see Fig. 4.8.

We show the implementation of the method used in [43] to extract the 15D feature vector, see Fig. 4.9. The implementation of quadtree decomposition on different datasets is illustrated in Fig. 4.10 and Fig. 4.11. Moreover, we have investigated samples with different illuminations as shown in Fig. 4.10, head-pose as shown in Fig. 4.11a, occlusion as shown in Fig. 4.11.b, blurring as shown in Fig. 4.11.c and full face as in Fig. 4.11.d.

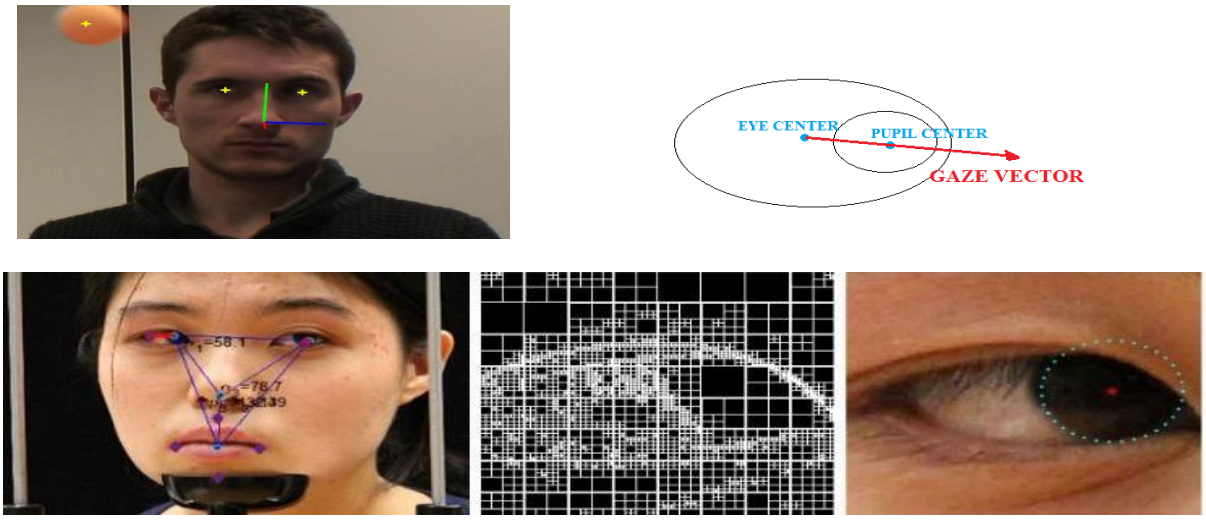
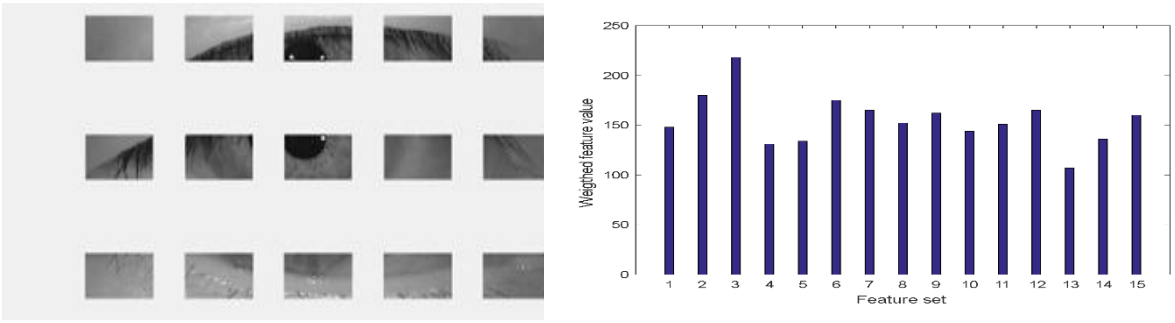


Figure 4.8: Samples are taken from EYEDIAP and CAVE datasets. We detected facial landmarks and captured the distances and angles between them. We further detected the eye center, the pupil center and the relative position of the pupil center with respect to the head-pose. Lastly, we employed quadtree region descriptor to detect the iris boundary.



(a) (b)

Figure 4.9: Sample from CAVE dataset. (a) The image is divided into a 3x5 sub-regions. (b) Each region is normalized resulting in a 15D feature vector.

The above sets of features provide a richer feature set, enabling higher precision in gaze estimation. We fused all feature sets from section 4.2 using K-DMCCA with discriminative correlation criteria based on the 3-σ rule. The training and testing datasets were compared using the RBF function, which gave us correlations between any two sets of samples.

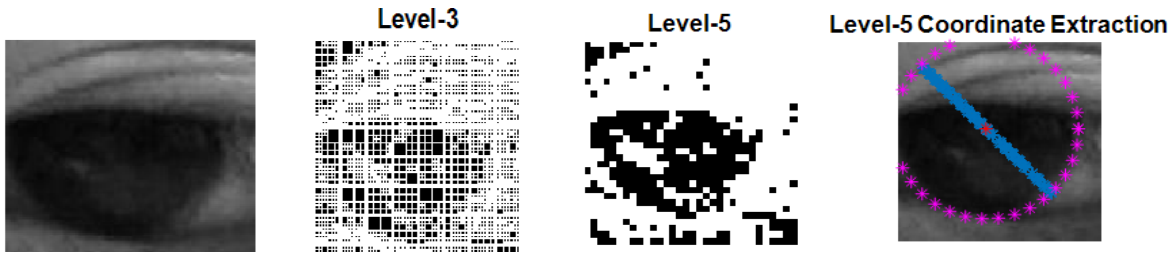
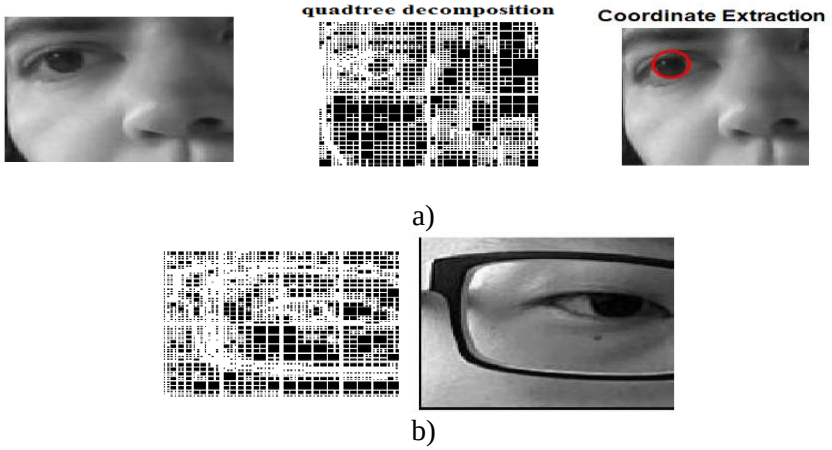
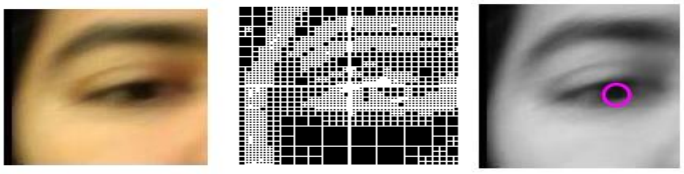


Figure 4.10: Samples were taken from MPII dataset, showing quadtree region descriptor stages to arrive at iris boundary estimation.

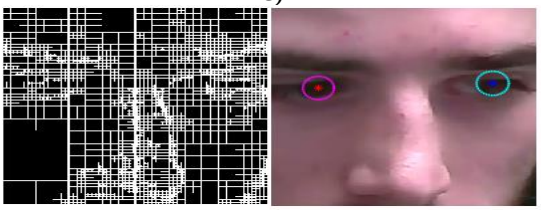


a)

b)



c)



d)

Figure 4.11: Samples are taken from MPII dataset showing quadtree region descriptor and presenting cases of: (a) Head-pose, (b) Occlusion, (c) Blurry image and (d) Full face frame.

4.4.1 EYEDIAP 3D Dataset

The EYEDIAP 3D dataset is based on the RGBD format. The extraction of 3D coordinates, namely (x,y,z),was based on the procedure presented in [110]:

1) Kinect RGB VGA video with a resolution of 640 x 480 was encoded using MPEG-4 of a front view.
2) Kinect depth video with a resolution of 640 x 480 was encoded using Zlib and was further processed using Python script.
3) Parameters were calibrated for the RGB camera and extrinsic camera (camera 3D pose with respect to the world coordinate system).
4) Depth camera parameters were calibrated similar to the RGB camera. They also included mapping parameters from depth map values to actual depth measurements.

Depth was encoded as a disparity values map, which was encoded in the RGB image. With capture device Kinect, 11 bits ranging [0, $2^{11} - 1$] for a given disparity value was encoded, such that 8 least significant bits (LSB) were assigned to the B channel of the RGB image, while the remaining 3 most significant bits (MSB) were taken as a byte, shifted left by 5 bits and then assigned to the G channel. The disparity map value was recovered by converting each color channel, in the corresponding RGB image, from an 8-bit to 16-bit unsigned integer [111].

$$disp\_undistorted = disparity + \beta(u, v) * e^{-(\alpha_0 - \alpha_1) * disparity}$$

β *(u,v)* are image coordinates that indicated the spatial distortion pattern for each pixel value at position *(u,v)*, and $(\alpha_0 - \alpha_1)$ signifying the decay of distortion effect. Finally,, the relation between the obtained disparity value and depth 'z_d' contained two parts: a scaled inverse and a distortion correction as modeled by Eq.4.13:

$$z_d = \frac{1}{disparity_{\,undistorted} \cdot k_1 + k_0} \tag{4.13}$$

where k_0 and k_1 are part of the depth camera intrinsic parameters known as k coefficients. The depth camera coordinates (x_c,y_c, z_c) were obtained based on the following formulas:

$$\begin{aligned} x_c &= \frac{x_d - c_{x_d}}{f_{x_d}} . z_d \\ y_c &= \frac{y_d - c_{y_d}}{f_{y_d}} . z_d \\ z_c &= y_d \end{aligned} \tag{4.14}$$

where f_{x_d} and f_{y_d} are focal lengths of the camera in (*x*,*y*) and (*y*,*d*) positions.

From depth camera to RGB camera, a point $P_d = [x\ y\ z]^T$ in depth camera coordinate was transformed to RGB coordinate as $P_{rgb} = R_d . P_d + T_d$. The data preparation involved separating 4400 frames from each video. We decided to implement the region descriptor feature only in the 3D gaze estimation, since it was proven to be invariant to illumination, occlusion and rotation. The 3D frames were cropped to fit the region descriptor algorithm. The workflow is shown below:

- Spatial indexing of the front view frames using region descriptor.
- Estimate boundary for the point cloud based on circle points using region descriptor.
- Extract (x,y,z) from the point cloud.
- Apply discrimination K-DMCCA with depth info.

We applied quadtree decomposition on the front view of the camera, then detected iris boundary using region descriptor statistics and tracked position change. These frames are transformed from (x,y) to (x,y,z) coordinates. See Fig. 4.12.

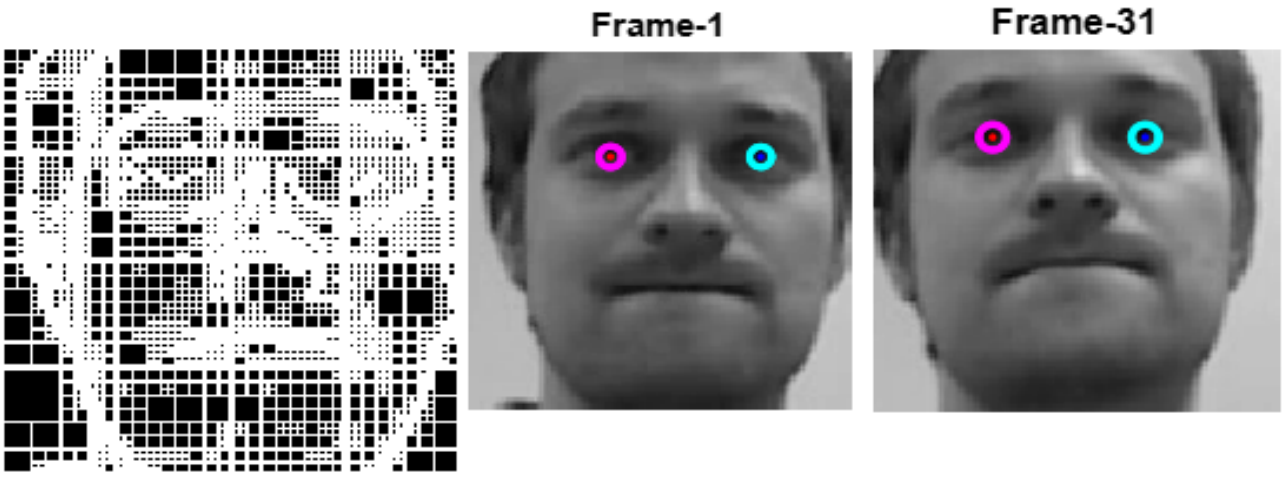


Figure 4.12: Quadtree decomposition of the front view of the camera. Detecting iris boundary of both eyes using quadtree decomposed statistics and tracking position change.

4.5 Results and Discussion

It was worth analyzing the effect of using one feature or two features versus all features combined. A comprehensive analysis was performed and outlined in section 4.5.1. To measure the accuracy of the proposed framework, we used the mean angular error which is defined as the angular distance between the proposed framework's estimate of the gaze and the true gaze.

4.5.1 One Feature vs. all Features

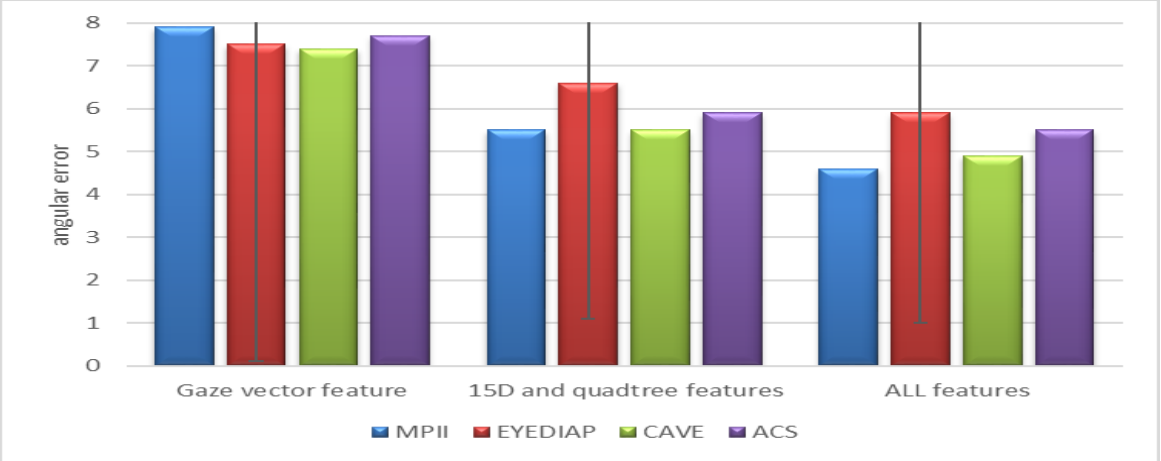


Figure 4.13: Using the average error of each dataset, we show the comparison of employing one feature vs. all features, on all datasets.

Applying the gaze vector feature only and estimating the gaze on all datasets, we were able to get an average angular error of 7.6º across all datasets as shown on the left of Fig. 4.13.

Using two features (15D and quadtree region descriptor) to estimate the gaze, improved the results noticeably as shown in the middle of Fig. 4.13. Finally, combining all features together achieved a much better result with an average angular error of 5.1˚ on all datasets as shown on the right of the same figure. The angular error was illustrated in Fig. 4.13. Particularly, we use MPII dataset for illustration as shown in Fig. 4.14. Using K-DMCCA, the results on the testing samples are almost the same as those on the training data, indicating that the feature extracted, and the fusion and recognition methods are representative across the datasets.

Multiple features fusion significantly improved the performance of the proposed algorithm. In feature fusion, sufficient information exists by combining all features, as a result, it can be expected that features fusion can achieve greater performance. However, the processing time and the computational demands of such a system are higher than one-feature systems.

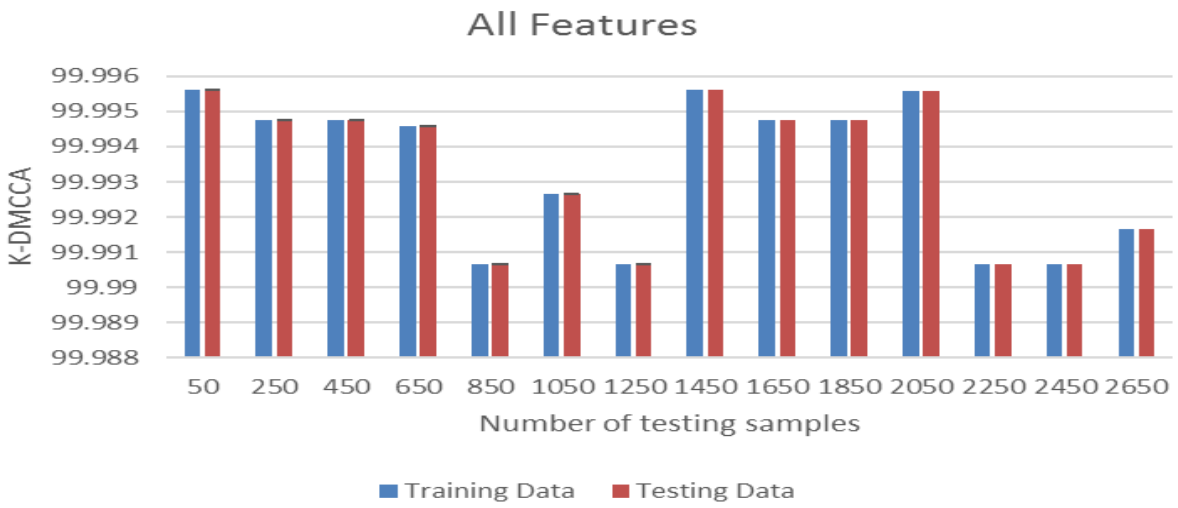


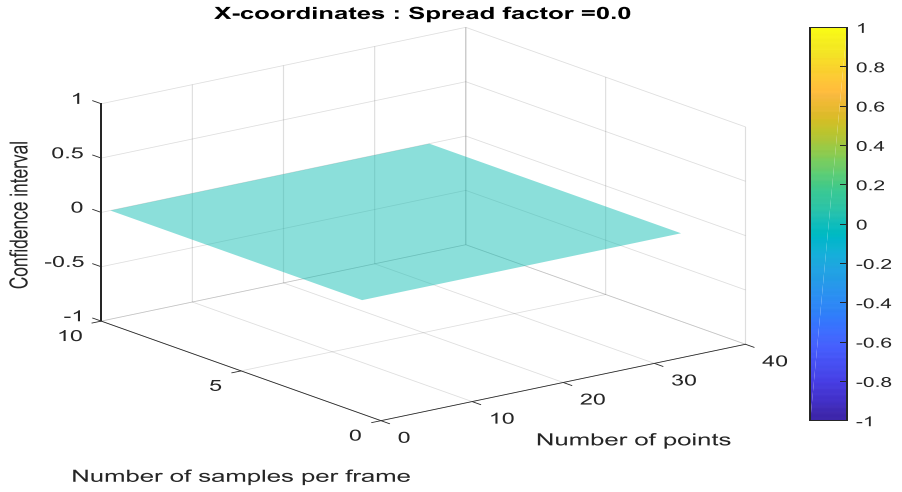
Figure 4.14: Using MPII dataset for illustration, we compare testing vs. training datasets using K-DMCCA.

4.5.2 Validating the results on EYEDIAP 3D

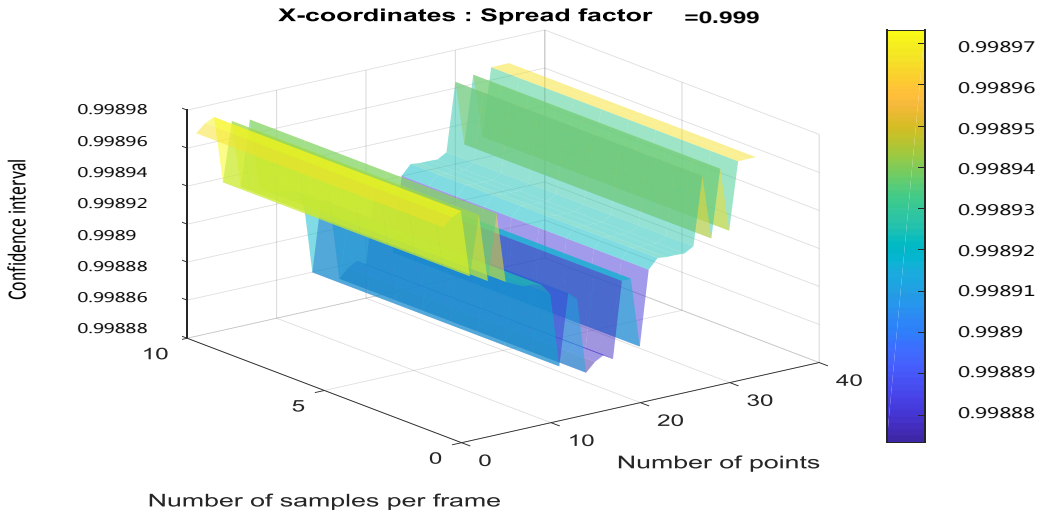
The EYEDIAP training dataset consisted of videos of 4000+ frames for each training sample (clip). We have extracted (x,y,z) coordinates for each video. Only the iris region descriptor feature was implemented in the 3D gaze estimation to demonstrate that it is invariant to illumination, occlusion and rotation.

The values for predicted gaze estimation were very close for many samples. We employed

an extra parameter to show the numerical precision by transforming the computed values using a spread factor. The spread factor changes the distance of the vectors based on a predefined threshold value [109]. The distance values vary the degree of similarity based on the spread factor (γ) used. If the $\gamma \approx 1$, then they belong to a particular class, while $\gamma \approx 0$, then there is no considerable difference and we cannot specify which class it belongs to. The plot in Fig. 4.15.a shows spready factor of $\gamma = 0$, which indicated no significant distance between the testing samples, therefore, it was very difficult for the proposed algorithm to select which gaze label/class the testing sample belongs to. Furthermore, in Fig. 4.15.b, we introduced a spread factor $\gamma = 0.999$, which created a distance between testing samples, as a result, the proposed algorithm was able to distinguish which gaze label/class the testing sample belongs to.



(a)



(b)

Figure 4.15: (a) Surface plot with spread factor $\gamma = 0$, (b) Surface plot with spread factor $\gamma = 0.99$, showing several levels representing the variations.

We have compared our results with the methods in [77], [79] [84] and [85]. Table 4.1 illustrates the performance and accuracy of our method in comparison to other methods. Our method achieved high accuracy, which was measured using Eq. 4.15. The mean angular error is the difference between the estimated feature vector and the ground truth feature vector.

$$\text{Mean Angular Error} = \text{Cos}^{-1}[\text{estimated gaze - ground truth}] \qquad (4.15)$$

Table 4.1: Shows the experimental results validated over MPII, EYEDIAP, Cave and ACS (not available to the public) datasets. The accuracy of the predicted gaze direction was indicated in mean angular error.

	MPII	EYEDIAP	CAVE	ACS
Proposed	4.6°	5.9°	4.8°	5.1°
Method in [77]	4.8°	6°	N/A	N/A
Method in [79]	5.9°	10.5°	N/A	N/A
Method in [85]	4.6°	7.5°	6.2°	N/A
Method in [84]	5.99° using their own dataset			

It is worth noting that none linear features call for a none-linear classifier. In our case, features such as QD coordinates, distances and angles , gaze vector, etc … , are all features that do not bear a linear relation with one another. Hence using a kernel was essential (Since the DMCCA takes linear features). To validate this, we ran the same experiment as in section 4.4, but without using the kernel. Only DMCCA and a classifier were employed. See figure 4.16 illustration. By using the DMCCA only, the graph on the left clearly shows that the testing samples are not close to those of the training samples, because our classification threshold is none-linear due to the fact that we have none linearly features.

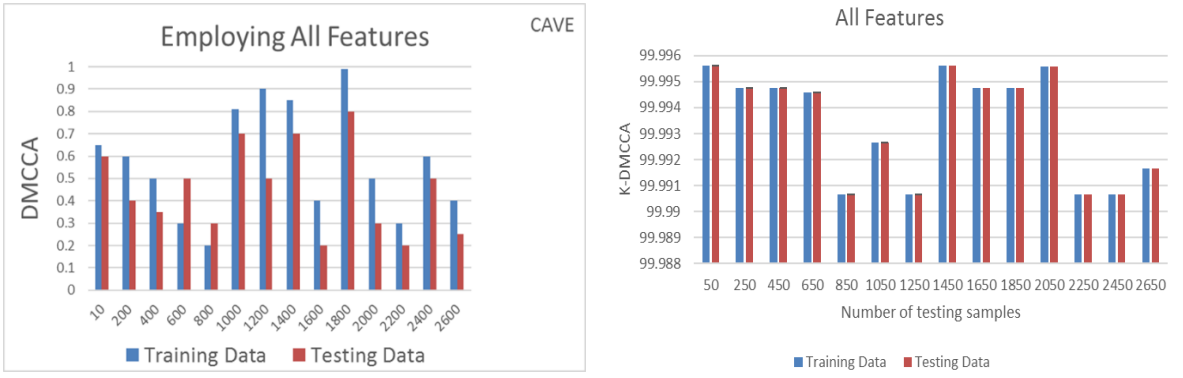


Figure 4.16: Illustration of training vs. testing using DMCCA (on the left) and using KDMCCA (on the right) on CAVE dataset. The graph on the left (using DMCCA) clearly shows that the testing samples are not close to those of the training samples.

4.6 Summary

In this chapter, in addition to employing the feature produced in chapter 3, we have introduced a new iris region descriptor using quadtree decomposition. The proposed algorithm is composed of appearance-based and geometric-based features. The feature extraction process starts with facial landmarks and zooms-in to the eye region along with the region descriptor spatial indexing and the statistical 15D of the eye region. Each feature acted as a mask/label for each sample. Based on the recent success of DMCCA, we proposed a kernel-DMCCA feature fusion approach and employed a classifier for optimal gaze estimation. The discriminative algorithm transforms the features using kernel to a high dimensional space, which established a better correlation between the training and testing datasets. Furthermore, the number of parameters to be estimated in the new feature space becomes independent of the dimension of the feature space. The output of fused features through K-DMCCA is robust to illumination and occlusion, and is calibration free. The proposed framework achieved an accurate gaze estimation of 4.8º using Cave, 4.6° using MPII, 5.1º using ACS and 5.9° using EYEDIAP datasets respectively.

Chapter 5

Robust Classification for Head-pose and Gaze Estimation Using Quadtree Decomposition and Geometrical Moments

Overview

This chapter investigates newly developed features to replace the existing features in literature that were used in chapter 4. This is achieved by extending the region descriptor feature using quadtree and developing it further to structure new features, for the purpose of achieving more accurate head-pose and gaze estimation.

The proposed framework employs multiclass analysis for head-pose and gaze estimation using an integrated spatial indexing, statistical and geometrical moments. The proposed framework is calibration free (user independent and requires no adjustment) and accounts for variations of head-pose, illumination and occlusion. We extract the following feature sets: shape of the eyes, tip of the nose, jawline, head angles (three degrees of freedom; roll, yaw and pitch) and 2^{nd} and 3^{rd} order moments. The feature sets are structured using geometrical moments then fused together. The main contribution is the development of a new framework by introducing a newly developed jawline feature along with spatial indexing technique using quadtree decomposition and geometrical moments. The outcome is exhibited using binary and one-vs-one multi-class SVM, which produces high quality discrimination between true and predicted results with loss function.

The remainder of this chapter is structured as follows: Section 5.1 outlines the proposed framework, section 5.2 details the extracted features, section 5.3 develops the implementation and experiment, section 5.4 presents the evaluation and results, section 5.5 summarizes the chapter.

5.1 Proposed Framework

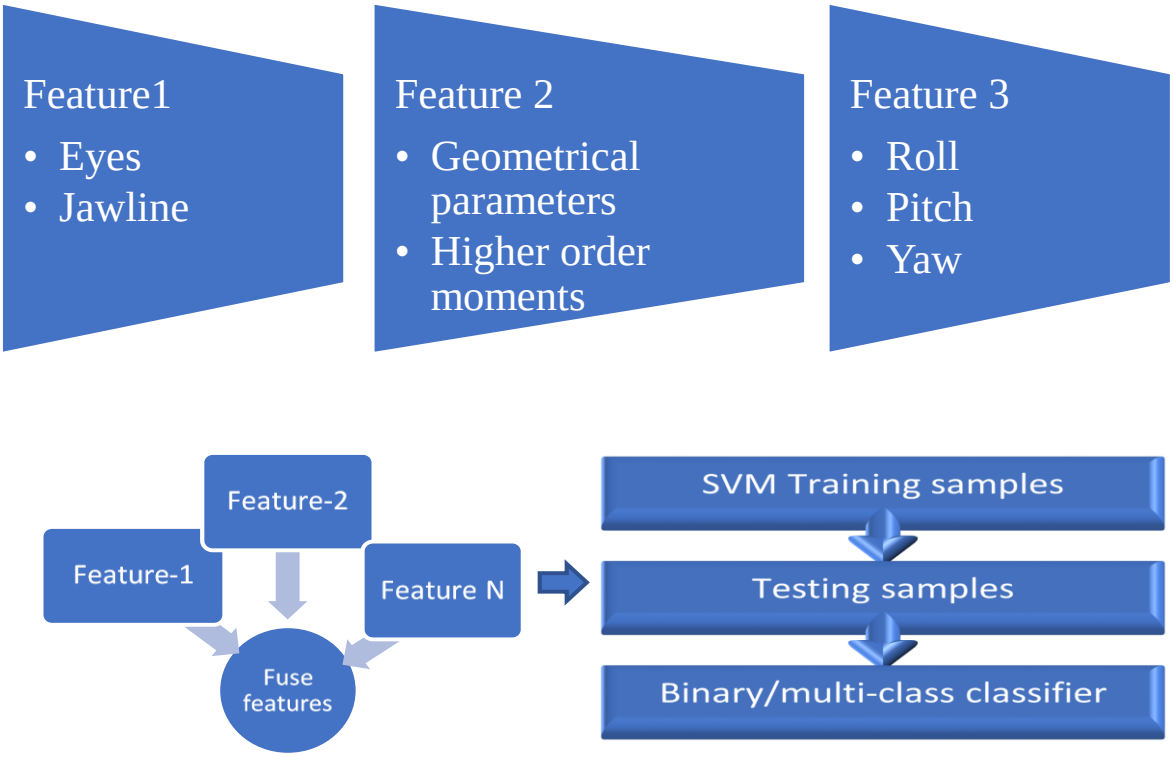


Figure 5.1: Shows the workflow of extracting features and classification of the testing set based on SVM.

The proposed framework is illustrated in Fig. 5.1, which highlights the approach towards head-pose and gaze estimation using independent training and testing sets. We extract features from the binary pattern obtained after quadtree decomposition, which defines the geometrical features of the shape of the eye, face boundary, jawline and nose tip. Using these primary features, we compute the symmetry metrics in terms of roll, yaw and pitch, as discussed in section 5.2 The geometrical moments masked onto the quadtree decomposition (QD) binary patterns identify the object's centroid and the orientation along major and minor axes of the eyes. The training sets of samples are used for creating the SVM model. With this model, prediction of the testing set is carried out by either a binary or multi-class classifier.

5.2 Feature Extraction

5.2.1 Quadtree Decomposition (QD) and Geometrical Moments

5.2.1.1 Principle

In chapter 4.2.3, the QD algorithm was outlined as an iris region descriptor. Below, we will develop this principle further to describe an ROI. The narrowing down of possibilities to localize the ROI is based on the statistical aspect (by counting the number of 1s and 0s values) of the sub-quadrant, and by traversing the horizontal, vertical and diagonal sub-quadrants at each level. The sewing (merging) of the sub-quadrants depends on the data/label driven determination of the ROI's geometrical shape. Fig. 5.2 illustrates level 3 of quadrant and sub-quadrant divisions for creating QD indexing.

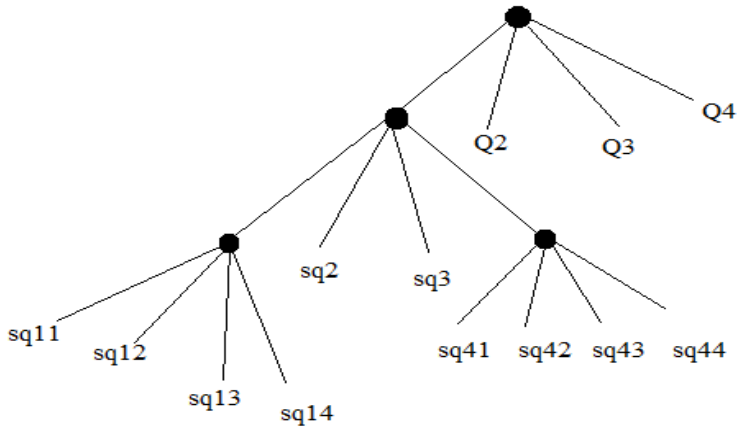


Figure 5.2: Level 3 of quadrant and sub-quadrant divisions for creating QD indexing. QD representation, where the dark circles are the root and the sub-quadrants, are the leaf nodes.

5.2.1.2 Tessellation

We use QD as a ROI descriptor in an affine plane with tiling of image features. We can identify the symmetry of the face by comparing the Euclidean space to the 2D image space. QD creates non-overlapped tiles with defining lines of boundaries forming a "tessellation".

Consider a Euclidean plane $\mathbb{R}^2$, with a point represented as (*x, y*), subtending an angle $\boldsymbol{\theta}$ that ranges between 0 and π. Let the origin be '*O*'. *X* and *Y* are sets of points on the x and y axes. Cartesian product $V = X \times Y \{(i, j)|, X \in x, \ Y \in y\}$, is a quadratic lattice formed by a set of points (d_x, d_y) on the x and y axes, which forms a graph *G* [112,113]. A collection of cells describing the ROI is identified iteratively. We then capture the area by a set of points (x, y) and the neighboring pairs of points.

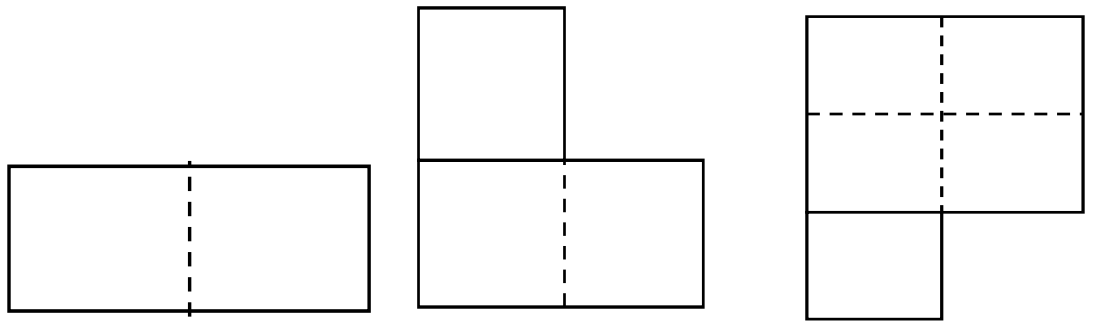
$Cell(i,j) = x^i \, y^j$, forms a base cell, adjacent to one another, such that (*i-1,j*) ,(*i+1,j*), (*i,j-1*) and (*i,j+1*) are the four adjacent nodes that relate to the tree [112,113].

The quadtree technique recursively decomposes the original image into tessellated segments such that if they are recombined into groups, then the original image can be reconstructed. The hierarchical decomposition enables addressing for rapid access to any geographical part of the image. It retains explicitly in the data structure a hierarchical description of image patterns, elements, and their relationships. Also its data structure distinguishes the object from background and thereby can focus on the interesting tessellated subsets of the data.

The detention of several tessellation schemes for different planar topologies is presented below using the concept of shape polynomials and tessellation matrices. The shape polynomial represents the geometry of the planar region while the tessellation matrix reveals the spatial adjacency of the regions.

Definition 1: A tile τ_k , is a group of base cells that are of similar shape as the larger form.

Definition 2: Notations of tiles is based on the shape polynomial.



(a) (b) (c)

Figure 5.3: (a) Showing two adjacent tiles (b) Tile with polynomial shape $S(L_k)$, (c) Tile with polynomial $S(P_k)$.

(i) $S(I_k) = \sum_{i=0}^{k-1} x^i$. e.g: I_2 , as shown in Fig. 5.3.a.

(ii) $S(L_k) = \sum_{i=0}^{m-1} x^i + \sum_{j=0}^{s} y^j$, m+s=k , for k >2, as shown in Fig. 5.3.b.

(iii) $S(P_k) = \sum_{i=1}^{s} x^i . \sum_{j=m}^{n-1} y^j + \sum_{j=0}^{m-1} y^j$

$S(n-m)+m$=k , for k>2, as shown in Fig. 5.3.c.

Definition 3: Shape polynomial of a tile is given as:

$$S(\tau_k) = \sum_{i,j\in\mathbb{R}^2} \gamma_{ij} x^i y^j \quad (5.1)$$

The area of such a tile is equal to $S(\tau_k)$

$$\gamma_{ij} = \begin{cases} 1 & Iff\ \forall(x,y) forming\ a\ cell\ (i,j)\ \in\ \tau_k \\ 0 & elsewhere \end{cases} \quad (5.2)$$

Definition 4: A tile τ_k is rotated by an angle Ψ around a σ −axis. Using an operator Ψ_σ , the new formation or shape of the tile is $\Psi_\sigma . S(\tau_k)$.

The tile orientation in the Cartesian coordinates allows certain sets of permissible operators.
Definition 5: Common transformations are reflections and rotations of the tiles.

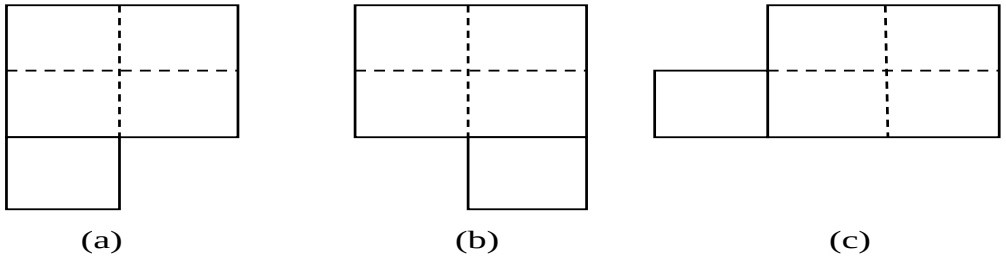


Figure 5.4: (a) Tile $S(P_5)$, (b) $S(P_5)$ reflected around y-axis, (c) $S(P_5)$ is rotated around z-axis.

On a given tile, say P_5 **,** we can have a set of permissible operators, such as π_x , π_y , π_z, λ_x , λ_y , λ_z. The reflection operator π_z is in the z axis, while λ_z is the rotation around the z axis. The change in orientation of the tile is seen in Fig. 5.4.
Definition 6: Given a region R=$\bigcup_{i=1}^{n} R_i$, then we have a spatial distribution of polynomial S_i, which allows alternative orientations.

Definition 7: Two regions are "homothetic" if they are collinear and have an affinity or are transformed in the affine plane. Consider two regions, R_i *and* R_j, with shape distributions of S_i *and* S_j; both are said to be homothetic if a sequence of permissible orientations makes them similar within the plane [112,113].

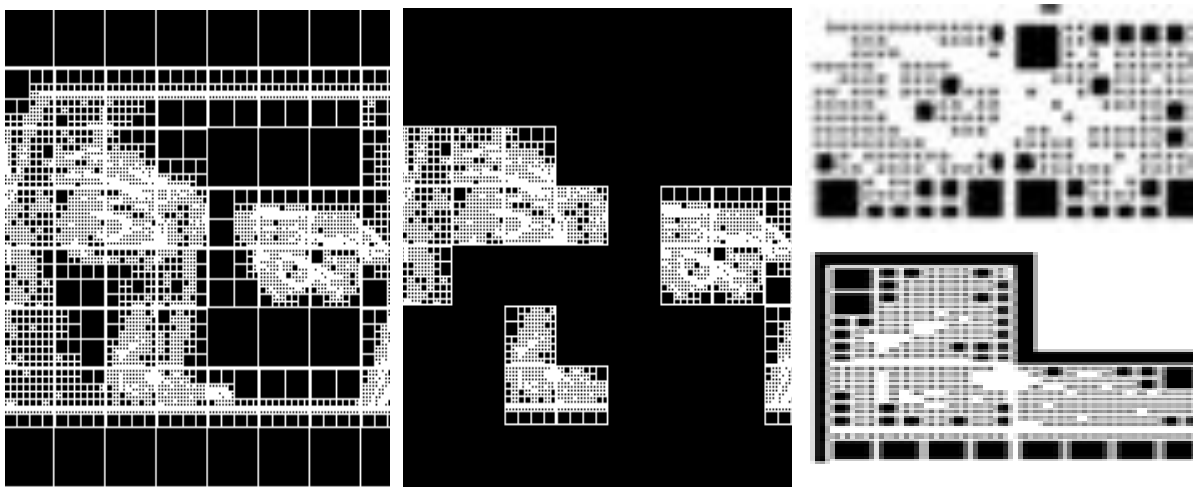
With reference to QD, a cell represents a point, edge and segment, as an attribute in each sub-quadrant. Collectively, such features in a cell or sub-quadrant form boundaries to describe

the object in the global space [68,112,113,114,115,116]. The next section will illustrate how features are constructed using the principle of tessellation.

5.2.1.3 Tessellation of features

An image of size 256 x 256, with level 3 decomposition results in Fig. 5.5.a. For identifying the features in the form of tessellation tiles, we use the one count (defined in section 4.2.3.1). Choosing the denser tiles results in Fig. 5.5.b. The proposed algorithm spatially identifies the eye region and nose based on the one count and its neighborhood, as seen in Fig. 5.5.c.

Tessellation of a polygon is used in the grouping of facial features. The regular square tile and its neighbor form the basis of the eyes and nose. Each tile at every level results in a square tile of size $2^{N/L} \times 2^{N/L}$ **,** where L is the particular level of decomposition. Furthermore, translational and rotational symmetry can be easily identified using the zeros and ones count, as discussed in section 5.2.1.4.1.



(a) (b) (c)

Figure 5.5: (a) QD image with a predefined threshold, (b) tiles selected by the algorithm based on one-count of QD, (c) tiles of nose and eyes facial features.

The tessellation of features through QD is presented in Fig. 5.5. The image is decomposed, as shown in Fig, 5.5.a, then showing the selected tiles by the algorithm based on ones-count of QD, as shown in Fig. 5.5.b and Fig. 5.5.c.

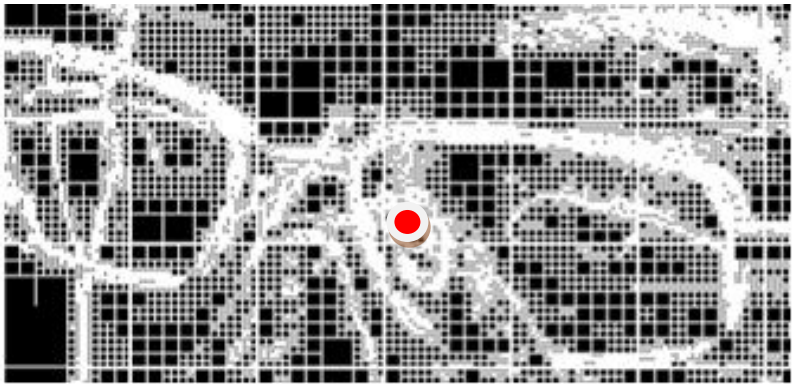
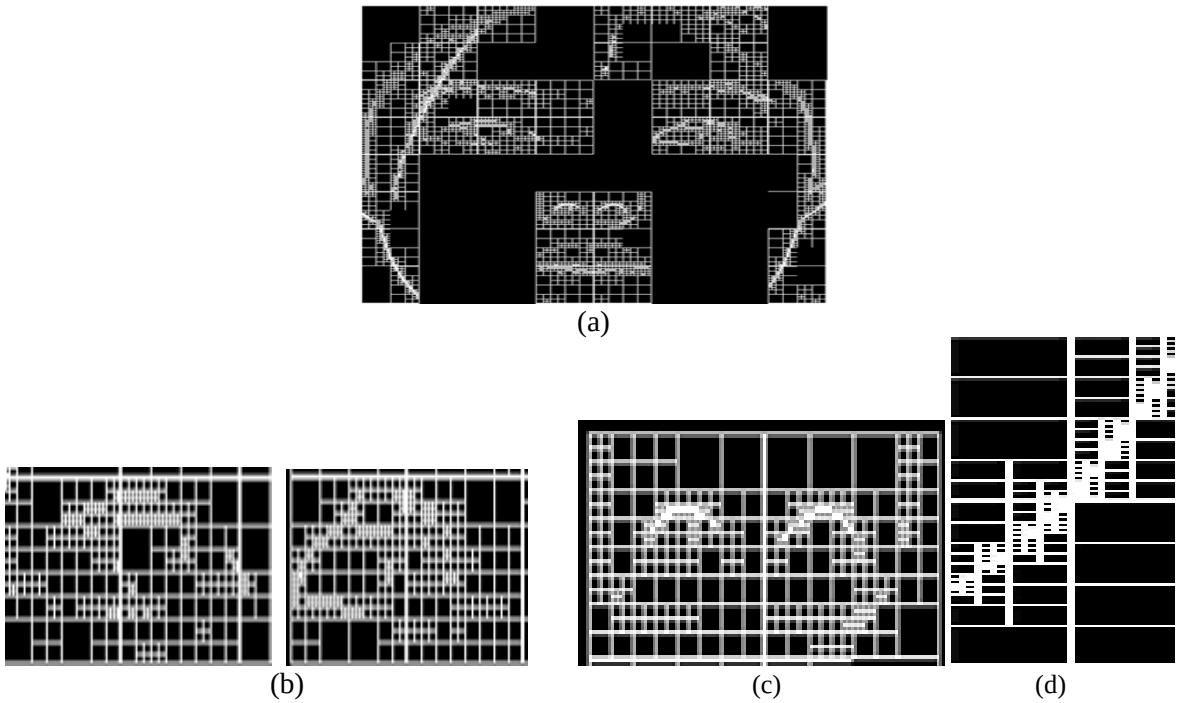


Figure 5.6: Lateral view QD from ACS dataset. Red dot is the glass frame.

The occluded image in Fig. 5.6, exhibits the one count as denser around the edges of the glass frame. However, in this complex case with glass frames, the proposed algorithm does not consider the choice of tiles with the maximum one count at the frame. Instead, it scans and compares the one count at the glass frame region with the one count at the eye region. Only the eye region, which exhibits the less dense one count will be chosen as the ROI.



(a)

(b) (c) (d)

Figure 5.7: Image from CAVE dataset. (a) Presenting selected tiles from the one count after QD; (b) Two sub-quadrants identifying the right and left eyes; (c) Presenting the sub-quadrants of the nose and (d) Presenting the sub-quadrants of the face boundary.

Samples from the CAVE dataset with removal of non-dense one count of the QD are shown in Fig. 5.7. The sub-quadrants of the facial feature's tiles, as shown in Fig. 5.7.a and 5.7.b, are considered for the facial feature extraction of the eyes, nose and jawline (face boundary). The

jawline is passing through multiple sub-quadrants and the choice of selecting multiple sub-quadrants is mandatory for the face boundary, since it extends to more than two sub-quadrants.

5.2.1.4 Geometrical Moments

Moments characterize distances, such as length, area and width based on the reference points in a given ROI [113].

Consider pixels in an affine plane represented as Cartesian coordinates, moment of order (p,q) in the continuous domain of image function f(x,y) is given in the equation below:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy \quad (5.3)$$

In the discrete image of size M x N, g(x,y) the moments are given as shown below:

$$M_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} x^p y^q \, g(x,y) \quad (5.4)$$

p,q are the order of the moments.

The set of moments $\{M_{0q}\}$ and $\{M_{0p}\}$ are the projections on x and y axis, respectively. The one-dimensional projection is given as:

$$m_p = \int_{-\infty}^{\infty} x^p v(x)dx \quad (5.5)$$

$v(x)$ is the vertical image projection, with a statistical distribution of the image data.

For example, we can consider second order moments μ_{20}:

$$\mu_{20} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^2 f(x,y)dx$$

μ_{20} represents the variance of the image data [10].

Properties of the moments:

1. The zeroth order moment M_{00} gives the total mass of the distribution of the image data.
2. 1st order moments $\{M_{01}, M_{10}\}$ are called center of mass (COM), which are parallel to the x and y axis, respectively. COM is used to locate central moments that are given by (x, y) coordinates: $\tilde{x} = \frac{M_{10}}{M_{00}}$ and $\tilde{y} = \frac{M_{01}}{M_{00}}$.

If the central moment coincides with the origin, then the central moments $\tilde{x} = 0$ and $\tilde{y} = 0$ are represented as μ_{10} and μ_{01}.

3. 2nd order moments are called moment of inertia, represented as $\{ M_{20}, M_{11}, M_{02}\}$. They provide the principle axes and the orientation of the object φ:

$$\varphi = \frac{1}{2} \tan^{-1} \frac{2\,\mu_{11}}{\mu_{20} - \mu_{02}} \qquad (5.6)$$

where φ is the angle of the principal axis with respect to the x axis. The orientation axis mainly depends on $\mu_{11}, \mu_{20}, \mu_{02}$

The ROI of the eye is presumed to be an ellipse, with the major axis (Alpha) and minor axis (Beta) values estimated based on the second order moments:

$$\text{(Alpha, Beta)} = \sqrt{\frac{\mu_{20}+\mu_{02}}{2} \pm \sqrt{4 * \mu_{11}^2 + [\mu_{20} - \mu_{02}]^2}} \qquad (5.7)$$

Radii of gyration (RoG) indicates the concentration of mass without change about the axis.

$$RoG_x = \sqrt{\frac{M_{20}}{M_{00}}} \quad RoG_y = \sqrt{\frac{M_{02}}{M_{00}}} \qquad (5.8)$$

Around the center $(\tilde{x}, \tilde{y})$, $RoG = \sqrt{\frac{\mu_{20}+\mu_{02}}{\mu_{00}}}$.

4. The 3rd order moment identifies the skewness is given below:

$$Sk_x = \frac{\mu_{30}}{\mu_{20}^{3/2}} \quad Sk_y = \frac{\mu_{03}}{\mu_{02}^{\frac{3}{2}}} \qquad (5.9)$$

The skewness provides the degree of deviation around the x and y axis, respectively. The skewness coefficient sign provides the degree of skew on each side of the axis. For instance, the case Sk_x and Sk_y are zero indicates symmetry. If both are positive, the distribution is skewed left of the y axis and below the x axis [32].

5.2.1.5 Moments transformation

Geometric transformations, scaling, translation, rotation and reflection can be detected in binary images by using moments [113].

Scale:
Transformed moments scaled by (δ,ε), of image $f(x, y)$, is given as:

$$M\acute{}_{pq} = \delta^{p+1}\varepsilon^{q+1}M_{pq} \qquad \delta = \varepsilon$$

$$M\acute{}_{pq} = \delta^{p+q+2}M_{pq} \qquad \delta \neq \varepsilon \tag{5.10}$$

Translation:

$$M\acute{}_{pq} = \sum_{r=0}^{p}\sum_{s=0}^{q}\binom{p}{r}\binom{q}{s}\delta^{p-r}\varepsilon^{q-s}M_{rs} \tag{5.11}$$

Rotation:

$$M\acute{}_{pq} = \sum_{r=0}^{p}\sum_{s=0}^{q}\binom{p}{r}\binom{q}{s}(-1)^{q-s}\ (cos\theta)^{p-r+s}(sin\theta)^{q+r-s}.M_{p+q-r-s,r+s} \tag{5.12}$$

Reflection:

$$M\acute{}_{pq} = (-1)^{p}x\ M_{pq} \tag{5.13}$$

Based on the materials presented in sections 5.2.1.1-5.2.1.4, we are now able to extract features as follows:

5.2.2 Eye Region and Nose tip Feature Extraction

Eyes and nose are selected by the one-count of the tiles. For example, if the image size is 512 x 512, we have a tile of size $\frac{2^9}{2^3}$ that is 64 x 64 bits. The quadtree data structure is stored in the form of an array, whose size is 2^6 which is represented in the form of 16 x 4. As a general rule, for higher dimension images, the eye region is selected based on four neighboring tiles, while the nose is selected based on two neighboring tiles. Lower resolution images require more neighboring tiles.

Once the tiles are picked, the moments are calculated. We extract the shape of the eye based on the 1st and 2nd order moments. For the detection of the center of the nose, we calculate the 1st

through 3rd order moments, with the addition that we are interested in the center of the tiles along with the statistical aspects.

5.2.3 Head-pose: Distances and Angles Feature Extraction

Facial landmarks analysis detects eyes, nose and jawline. All facial landmarks have a frame of reference, which accounts for any changes of orientation. Peng *et al*. [117] proposed a technique for eye detection with two directional gradients relying on light reflectance as a distinctive feature. Similar considerations are adopted for the jaw and boundary/edges of the face. Using a medial axis, the face pixels are projected along the nose [118].

We adopt Lucas-Kanade algorithm [119], which is used for measuring displacements within a small neighborhood of successive frames of images. This method minimizes the sum of square error between the image and the given template. With a tracking window size 10 x10, they track three points namely nose, right and left eyes [118].

As a measure of symmetry, we take the equations for these three reference points, namely, the centers of right and left eyes and the center of the nose, to calculate the movement in images along the longitudinal axis as follows:

$$roll = \arctan\left(\frac{Pt_{E\ Left_y} - Pt_{E\ Right_y}}{Pt_{E\ Left_x} - Pt_{E\ Right_x}}\right) \quad (5.14)$$

$$yaw = Pt_{Nose_{\acute{x}}} - Pt_{Nose_x} \quad (5.15)$$

$$pitch = Pt_{Nose_y} - Pt_{Nose_{\acute{y}}} \quad (5.16)$$

where $(Pt_{E\ Right_x}, Pt_{E\ Right_y})$ is the center of right eye and $(Pt_{E\ Left_x}, Pt_{E\ Left_y})$ is center of the left eye. (Pt_{Nose_x}, Pt_{Nose_y}) is the COM (center of mass) of the nose while $(Pt_{Nose_{\dot{x}}}, Pt_{Nose_{\dot{y}}})$ is the projection point on the longitudinal axis.

Roll angle is based on the current location of the eye. Estimation of this angle is accomplished by the line passing through the eyes, parallel to the frontal plane. Yaw and pitch angles are based on the points of reference on the longitudinal axis and COM of the nose. These two parameters (yaw and pitch) are analyzed as displacements, and not angles, which are easier to calculate, even when encountering changes in frontal pose [118].

5.2.4 Frame of reference

The frame of reference is illustrated in Fig. 5.8. We employ the QD analysis to localize the feature set based on facial symmetry. With full facial features constructed from the image in sagittal and frontal planes, we provide a measure of displacement and rotation around the longitudinal axis. Head-pose (head orientation) is based on the symmetrical metrics with these reference frames.

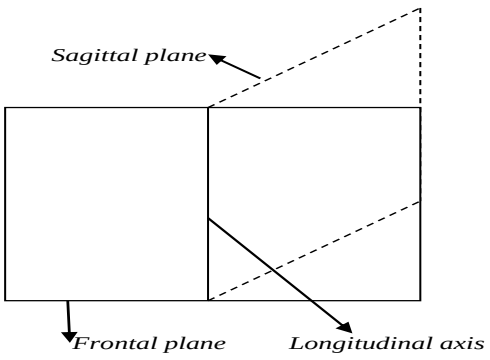


Figure 5.8: Presenting two anatomical planes commonly used for symmetry measures.

Anatomical planes are commonly defined using MRI and other imaging techniques. For calculations of yaw and pitch, the longitudinal axis is the frame of reference. For an image of

size 256 x 256, after QD, this axis lies on the following coordinates (128,0) , (128,128) and (128,256).

5.2.5 Classification

5.2.5.1 Binary SVM

SVM is a classifier that provides an optimal boundary solution based on the feature set or raw data. This will be employed in MPII and CAVE. We have used binary SVM with Gaussian kernel (Gk-SVM) function indicating a non-linearly separable class of data. With training and testing feature sets, the transformation from non-linearly-separable to linearly-separable form is accomplished by "kernelization" and is represented as shown in Eq. 5.17 [120]:

$$\phi(\mathrm{x}).\phi(\acute{\mathrm{x}}) = e^{-\gamma\|x-\acute{x}\|^2}, \forall \boldsymbol{x}, \acute{\boldsymbol{x}} \in \mathbb{R}^{\boldsymbol{d}} \tag{5.17}$$

To find the margin separating the feature space, we use a quadratic problem formulation [120] as follows:

$$\begin{aligned} &\min_{w,b,\zeta_i} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i \\ &St: (y_i(w.\phi(x_i) + b) \geq 1 - \zeta_i \\ &\zeta_i \geq 0 \quad \forall i \end{aligned} \tag{5.18}$$

where constant C > 0 is a tuning parameter, ζ_i is slack variable, and w is the projection.

In a binary SVM, with a feature set X=[x_1, x_2...x_n] $\in \mathbb{R}^{dxn}$, corresponding to two sets of labels $y_i = \pm 1$, the mapping of the input feature is constructed by the function $\boldsymbol{\phi}$, as seen in Eq. 5.17. The mathematical notation in Eq. 5.18 signifies the inverse margin among two classes and contains parameter b, to be adjusted based on the value of slack variable ζ_i [120]. Once we solve Eqs. 17 and 18, the decision making for the testing dataset is as follows:

$$y = sgn(\sum y_i \phi(\mathrm{x}).\phi(\acute{\mathrm{x}})\lambda_i + \mathrm{b}) \tag{5.19}$$

Where 'b' can be estimated by a set of support vectors ranging from 0 to C of a given λ_i (λ_i refers to a class).

5.2.5.2 Multi-class SVM

For any data structure of over 2 classes, one-vs-one multi-class SVM is commonly used. The multi-class SVM will be employed on EYEDIAP.

i. One-vs-One [120]:

The number of classifiers is N(N-1)/2, each class is trained as a positive and a negative one (± 1).

$$\min_{w_{ij},b_{ij},\zeta_{ij}} \frac{1}{2} {w_{ij}}^T w_{ij} + C \sum_{j=1}^{l} \zeta_{ij}^j {w_{ij}}^T$$

$$St: \ ({w_{ij}}^T . \phi(x_p) + b_{ij}) \geq 1 - \zeta_{ij}^p \ \ if\, y_p = i$$

$$ {w_{ij}}^T . \phi(x_p) + b_{ij} \ \leq \ -1 + \zeta_{ij}^p \ \ if\, y_p \neq i \qquad and \qquad \zeta_j^i \geq 0,1,2, \dots p \tag{5.20}$$

where w is the projection, b can be estimated by a set of support vectors, slack variable ζ_i and ϕ is the mapping function. The value belonging to a particular feature set (which belongs to the ith or jth class) is illustrated in Eq. 5.20.

The training set selects only the positive classes from the data set. For the testing phase, each classifier is calculated using the decision function. The classifier selects the highest value, resulting in the most accurate head-pose and gaze estimation.

5.3 Implementing the Experiment

We conducted experiments on three public datasets: Cave (5880 images) [94], MPIIGaze (3000 images of left and right eyes) [79] and EYEDIAP (15 frames for each VGA video per participant) [96]. We also conduced the experiment on ACS dataset (10 subjects; not available to the public) [98]. The datasets consisted of different samples with variations of illumination, occlusion and head-pose. We structured 50% of the dataset for training and 50% for testing. We compared the proposed framework with recent existing methods [77], [79], [84], [39] and

[85]. Using a full-face image, QD starts at 256 x 256 size with a minimum block size of 32 x 32, indicating a level 3 decomposition.

The localization of the eye shape (the ROI in this case) is typically an ellipse characterized by major and minor axes. We employed M_{20} , M_{02} and M_{11}, as given by Eq. 5.6, where the orientation/tilt of the ROI is given by angle φ. The second order moments are also called the "principal axis" method. The tilt of the ROI can be determined by angle φ, For example, if $M_{20} - M_{02} = 0$ and $M_{11} = 0$, $then$ φ =0; if both are positive then 0<φ <45.

Moments are used to get the dimensions of principal axis that are also independent of direction [121]. When the direction is used for identification, there is a higher precision in pattern recognition. Higher moments are used for discriminatory purposes [121]. Hence, we have considered RoG and skewness for differentiating rotational invariance and statistical measure of deviation around the axis of symmetry [113], as presented in Eqs. 5.8 and 5.9.

5.3.1 Experiments on Datasets

5.3.1.1 MPII

The dataset is labeled as RIGHT ('r') or LEFT ('l'), for gaze direction and head-pose estimation. Fig. 5.9 below presents samples selected from MPII, illustrating the detection of eyes and nose using second order moments.

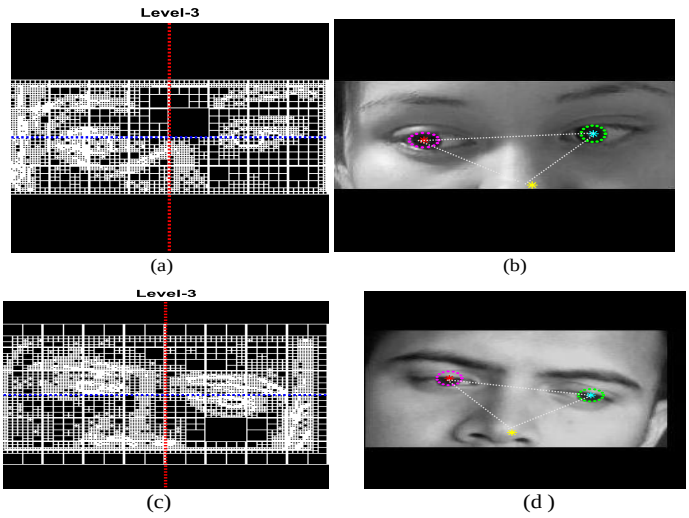


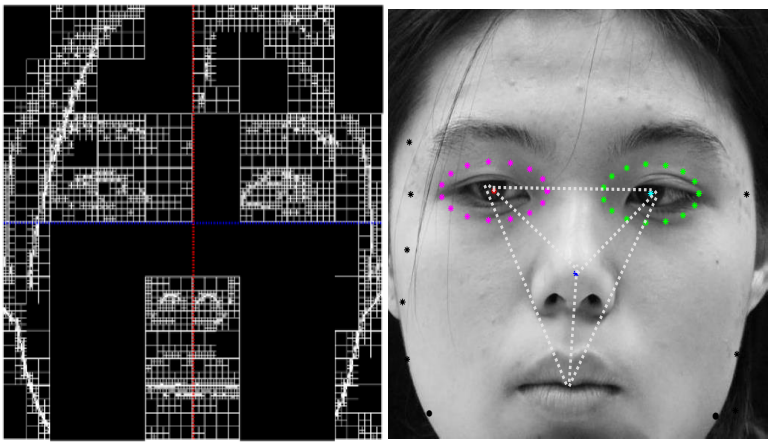
Figure 5.9: (a) QD of a sample selected from MPII, (b) Localization of eyes and nose (COM), (c) QD of another sample with a different head orientation, (d) Eyes and nose identified using second order moments.

5.3.1.2 CAVE

The image acquired from the dataset consisted of 3 parameters: head-pose, vertical and horizontal displacement, see Table 5.1. Fig. 5.10 presents a sample taken from CAVE illustrating the detection of eyes, nose and jawline.

Table 5.1: List of image acquisition, labeled in 3 variables.

Head-pose	Horizontal offsets	Vertical offsets
0 P	0°, ±5°, ±10°, ±15°	0°, ±5°, ±10°, ±15°
± 15P	0°, ±5°, ±10°, 15°	0°, ±5°, ±10°, 15°
30 P	0°	0°



(a) (b)

Figure 5.10: (a) QD of selected tiles of an image, (b) Identifying eyes, nose and jawline.

5.3.1.3 EYEDIAP

This dataset captured images in RGB and depth camera [82,96]. We selected frames that spread over 100s to capture variations of head-pose and gaze direction. Figure 5.11 presents two consecutive frames from EYEDIAP, illustrating the detection of eyes, nose and face boundary.

Using geometrical moments, we were able to deduce the roundness and eccentricity of ROI as shown below:

Roundness $\kappa = \frac{P^2}{2\pi A}$, where P denotes the perimeter and A denotes the area.

$$\text{Eccentricity } \varepsilon = \frac{\sqrt{Alpha^2 - Beta^2}}{Alpha} = \frac{[\mu_{20} - \mu_{02}]^2 - 4[\mu_{11}]^2}{[\mu_{20} + \mu_{02}]^2} \quad (5.21)$$

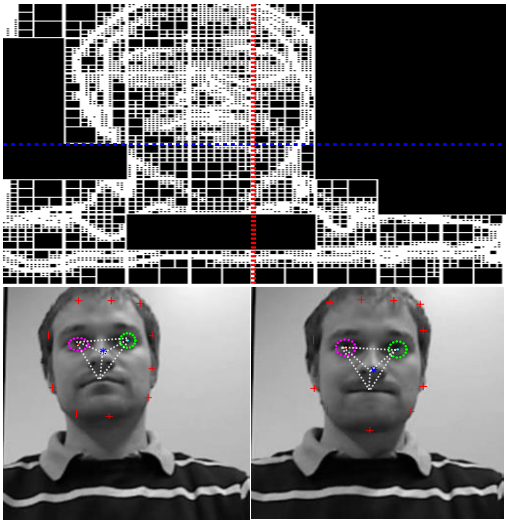


Figure 5.11: QD of a frame showing (a) Localization of ROI (eyes, nose and face boundary).
(b) Localization of ROI in two consecutive frames.

.

5.4 EVALUATION AND RESULTS

To evaluate the robustness of the proposed framework, we analyzed the effect of employing one feature, two features versus all features combined. A comprehensive analysis was performed and outlined in this section.

We have considered the following feature sets: eye region, nose COM, jawline, distance and angles between facial landmarks (distance between the eyes center, distance from the center of each eye to the nose COM, distance from the points on the jawline to the points on the symmetry axis), roll, pitch, yaw, 2^{nd} and 3^{rd} order moments. Each dataset was structured by 50% for training and 50% for testing.

5.4.1 Evaluation on MPII

Using a binary classifier (two classes), with 50% hold out, we create a model to train and cross-validate SVM by mapping the prediction data using a kernel function. Based on the trained classification SVM model, the model is developed and the prediction process (estimation) is completed using predicted class labels. The SVM model returns a matrix of scores indicating the likelihood that a label belongs to a particular class.

The predictor returns a score, for the input feature set, as a row vector that transforms the input linearly with a coefficient factor based on the model [109,120]. The classification has positive and negative set of classes based on the input feature values.

Classification has an associated loss function that determines the inaccuracy of the prediction [109,120]. For a binary feature set y_j, the classes are +1 and -1, $m_j = y_j f(X_j)$, where X is the predictor and m_j is the loss value.

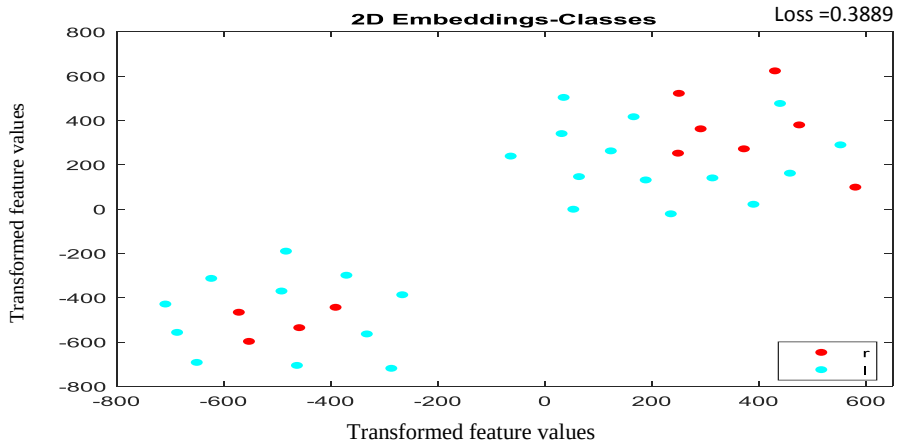
5.4.1.1 Employing one feature: eye shape

We adopt the k-fold cross-validation [122] which is a procedure used to estimate the skill (in terms of a loss value) of the model on new data (testing set), where k refers to the number of groups that a given dataset is split into (true vs. predicted). As seen in Table 5.2 (selection of a handful of samples for illustration purpose), the loss value is presented by a score for true vs. predicted labels of the eye region (negative score corresponds to l 'left' whereas positive corresponds to r 'right'). The algorithm, using one feature only, failed in two cases to predict the correct gaze label. The algorithm, using one feature only, failed in two cases to predict the correct gaze label.

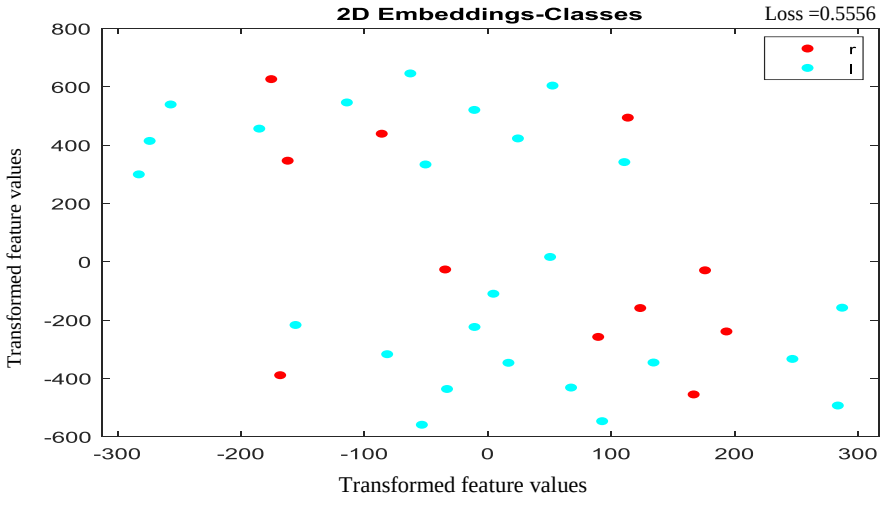
Table 5.2: True vs. predicted label with a score value for eyes.

True	Predicted
'r'	'r'
'l'	'l'
'l'	'l'
'r'	'r'
'l'	'l'
'r'	'r'
'l'	'l'
'l'	'r'
'r'	'r'
'l'	'l'
'l'	'l'
'l'	'l'
'l'	'l'
'l'	'l'
'r'	'r'
'l'	'l'
'r'	'l'
'l'	'l'

To visualize the gaze-class association, we used the t-distributed Stochastic Neighbor Embedding (t-SNE) for two gaze direction classes, namely, ‘right’ and ‘left’. The t-SNE is a machine learning algorithm for visualization, It uses Euclidean distance to embed high-dimensional data points into low-dimensional data points [123], in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability. Fig. 5.12 shows the classification in t-SNE 2D embedding with a loss value with a range of 0.3889 and 0.5556. A loss value of 0.3889 presents a better separation of gaze-class. The t-SNE is a machine learning algorithm for visualization.



(a)



(b)

Figure 5.12: (a) t-SNE 2D embedding of right and left gaze direction points with a loss value of 0.3889 (b) t-SNE 2D embedding of right and left gaze direction points with a loss value of 0.5556.

5.4.1.2 Employing Two Features: axis values and moments

By applying two features, as seen in Table 5.3, the loss value is presented by a score for true vs. predicted labels of the axis values and moments. Employing two features improved the results,

however, we are still getting errors. Refer to Section 5.4.1.4 for results of employing all features combined. The algorithm, using these feature, failed in one case to predict the correct gaze label.

Table 5.3: True vs. predicted for axis values and moments.

True label	Predicted
'l'	'l'
'l'	'l'
'r'	'r'
'l'	'l'
'l'	'l'
'r'	'r'
'l'	'l'
'r'	'r'
'l'	'l'
'l'	'l'
'r'	'r'
'r'	'l'
'r'	'r'
'l'	'l'
'l'	'l'
'r'	'r'
'l'	'l'
'l'	'l'

5.4.1.3 Employing Head-pose Features: roll, yaw and pitch

By applying head-pose features, as seen in Table 5.4, the loss value is presented by a score for true vs. predicted labels of roll, yaw and pitch. Refer to Section 5.4.1.4 for results of employing all features combined. The algorithm, using these features, failed in two cases to predict the correct gaze label.

Table 5.4: True vs predicted for roll, yaw and pitch.

True label	Predicted
'l'	'l'
'l'	'l'
'l'	'l'
'r'	'r'
'l'	'r'
'l'	'l'
'l'	'l'
'l'	'l'

'r'	'r'
'l'	'l'
'r'	'l'
'r'	'r'
'l'	'l'
'l'	'l'
'l'	'l'
'r'	'r'
'l'	'l'
'r'	'r'

5.4.1.4 Employing All Features

Combining all features (eye region, nose COM, jawline, roll, yaw & pitch, along with symmetry axis, 2nd and 3rd order moments.), the t-SNE embedding is illustrated in Fig. 5.13, and the score comparison is presented in Fig. 5.14. Combining multiple features significantly improved the performance of the proposed framework, and the score values for true vs. predicted labels match with no false prediction returned by the algorithm. In feature fusion, sufficient information exists by combining all features, as a result, it can be expected that features fusion can achieve greater performance. However, the processing time and the computational demands of such a system are higher than one-feature system.

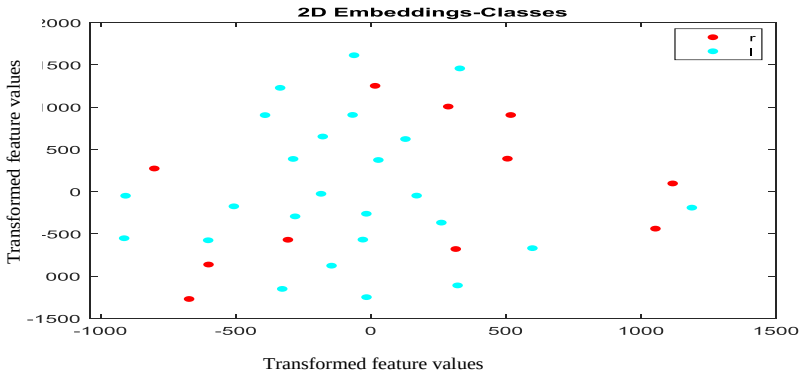


Figure 5.13: t-SNE 2D embedding of right and left gaze points after fusing all features.

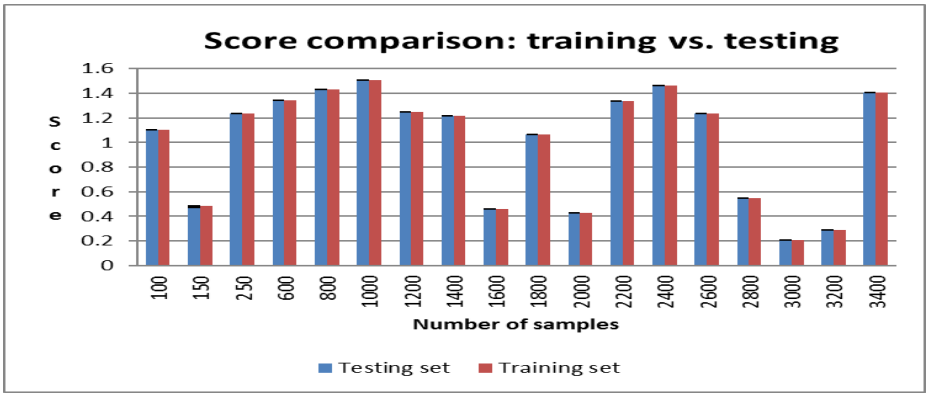


Figure 5.14: Score comparison of training vs. testing of random samples from MPII dataset, using all feature sets. Each sample on the x axis corresponds to a specific gaze class.

5.4.2 Evaluation on CAVE

Using a fit multiclass error-correcting output code (fitecoc), we created a model using the predictors with a posterior probability value assigned by the learner. Using all features sets, Table 5.5 presents two classes; illustrating how the class with the highest posterior probability corresponds to the gaze-class. As seen in Table 5.5 (selection of a handful of samples for illustration purpose, using two gaze labels only), the predictor (using all features) returned a matching result (the highest probability matching the corresponding gaze label) in every case predicting the correct gaze label.

We employed the one-vs-one classifier and the predict ensemble response resubstitution (resubpredict), to get N x K posterior probabilities for all N input feature set, where K is the number of classes [120].

Table 5.5: Predictors with a posterior probability assigned by the learner, using all feature sets.

True Label	Predicted Label	Posterior probability	
		'N15P'	'15P'
'15P'	'15P'	0	1
'15P'	'15P'	0	1
'N15P'	'N15P'	1	2.37E-11
'N15P'	'N15P'	1	2.24E-11
'15P'	'N15P'	0	1
'N15P'	'N15P'	1	2.37E-11
'N15P'	'N15P'	1	2.24E-11
'15P'	'15P'	0	1
'N15P'	'N15P'	1	2.24E-11
'15P'	'15P'	0	1

‘N15P’	'N15P'	1	2.24E-11
'15P'	'15P'	0	1
'N15P'	'N15P'	1	2.24E-11
'N15P'	‘N15P’	1	2.24E-11
'N15P'	'N15P'	1	2.24E-11
'N15P'	'N15P'	1	2.37E-11
'N15P'	'N15P'	1	2.24E-11
'N15P'	'N15P'	1	2.24E-11

The t-SNE embeddings is illustrated in Fig. 5.15 for 4 different classes: class 1: N15P (-15P), class 2: 0P, class 3: 15P and class 4: 30P. Furthermore, Table 5.6 presents the true vs predicted for these classes and the corresponding posterior probabilities, using all feature sets, where the highest posterior probability refers to the corresponding gaze-class. Finally, to simplify the illustration, Fig. 5.16 presents the posterior probability of training vs. testing, for two classes (15P and 30P), using all feature sets. As can be seen in Fig. 5.16, the results of posterior probability of training set are very close to the testing set.

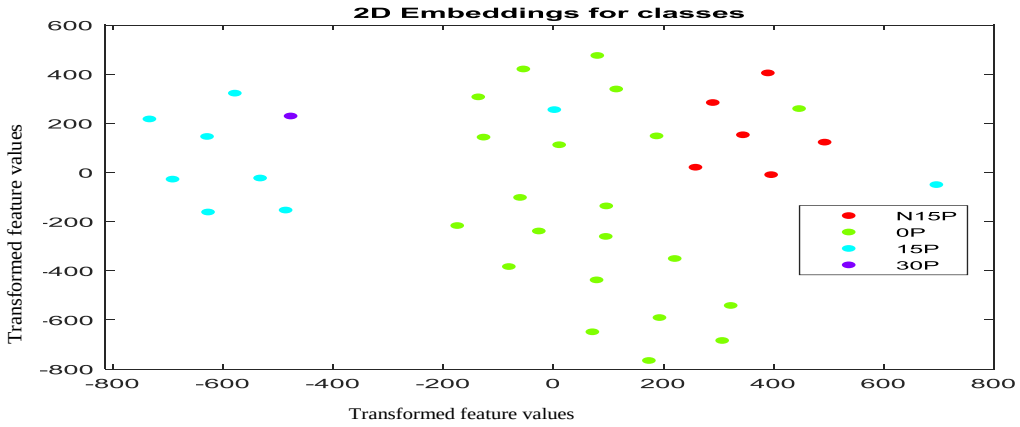


Figure 5.15: t-SNE 2D embeddings for all 4 classes.

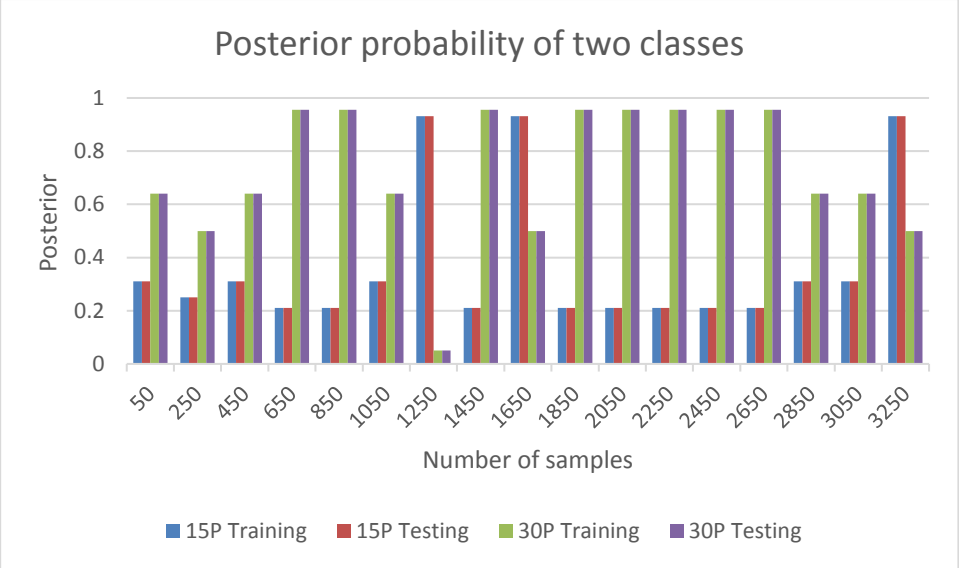


Figure 5.16: Posterior probability of training vs. testing on CAVE dataset, for two classes 15P and 30P, using all feature sets.

The following consecutive sequence of observations were recorded: 7 positive and 21 negative, 7 positive and 9 negative, 7 positive and 1 negative observation etc., were fit to the posterior probability. A total of $\frac{K(K-1)}{2}$ learners with outcomes having as low as one observation to maximum number of input observations. As seen in Table 5.6 (selection of a handful of samples for illustration purpose, using four classes), the predictor (using all features) returned a matching result (the highest probability matching the corresponding gaze label) in every case predicting the correct gaze label.

Table 5.6: True vs predicted and corresponding posterior probabilities, for four classes, using all feature sets. The highest posterior probability refers to the corresponding gaze-class.

True	Predicted	Posterior probability			
		class 1 '-15P'	class 2 '0'	class 3 '15P'	class 4 '30P'
'15P'	'15P'	0.031	0.064	0.905	2.22E-14
'15P'	'15P'	0.025	0.050	0.925	2.22E-14
'15P'	'15P'	0.031	0.064	0.905	2.22E-14
'0P'	'0P'	0.021	0.955	0.023	2.22E-14
'0P'	'0P'	0.021	0.955	0.023	2.22E-14
'15P'	'15P'	0.031	0.064	0.905	2.22E-14
'N15P'	'N15P'	0.931	0.050	0.0184	2.22E-14
'0P'	'0P'	0.021	0.955	0.0233	2.22E-14
'N15P'	'N15P'	0.931	0.050	0.0185	2.22E-14
'0P'	'0P'	0.021	0.955	0.023	2.22E-14
'0P'	'0P'	0.021	0.955	0.023	2.22E-14

'0P'	'0P'	0.021	0.955	0.023	2.22E-14
'0P'	'0P'	0.021	0.955	0.023	2.22E-14
'0P'	'0P'	0.021	0.955	0.0233	2.22E-14
'15P'	'15P'	0.031	0.064	0.905	2.22E-14
'15P'	'15P'	0.031	0.064	0.905	2.22E-14
'N15P'	'N15P'	0.931	0.050	0.0185	2.22E-14

5.4.3 Evaluation on EYEDIAP

The dataset exhibits two classes (FC4=left, and FC1=right), see Fig. 5.17. The illustration of how the class with the highest posterior probability corresponds to the gaze direction label is presented in Table 5.7, while Fig. 5.18 presents the posterior probability of training vs. testing, for two classes FC1 and FC4, using all feature sets. For visual illustration, the t-SNE 2D embeddings for 2 different classes is presented in Fig. 5.17. As seen in Table 5.7 (selection of a handful of samples for illustration purpose), the predictor (using all features) returned a matching result (the highest probability matching the corresponding gaze label) in every case predicting the correct gaze label.

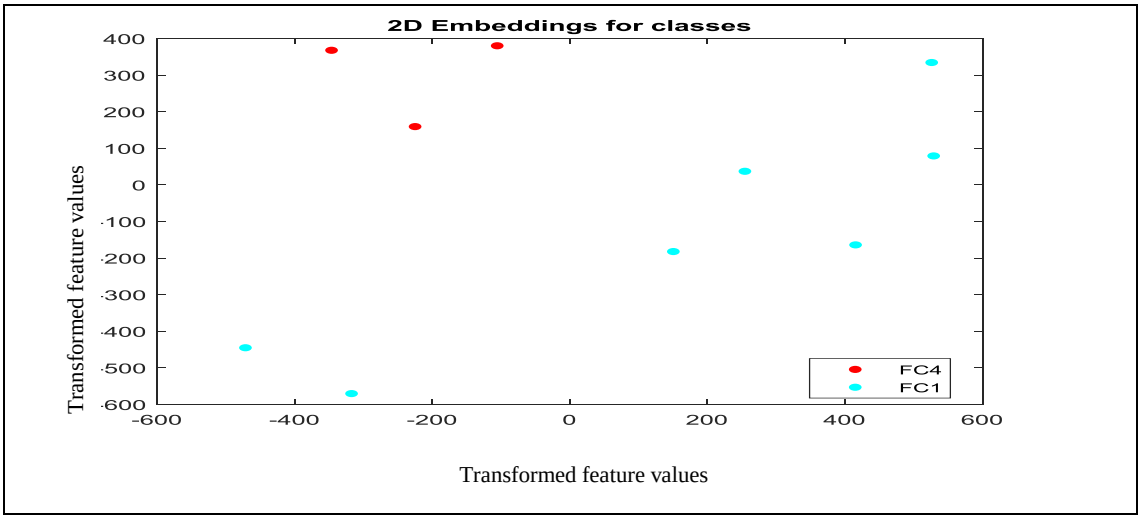


Figure 5.17: t-SNE 2D embeddings for 2 different classes.

Table 5.7: True vs predicted for two classes.

True	Predicted	Posterior probability	
		Class 'FC1'	Class 'FC4'
'FC1'	'FC1'	0.99312	0.0068813
'FC1'	'FC1'	0.99321	0.0067917
'FC4'	'FC4'	6.011e-07	1
'FC1'	'FC1'	0.99321	0.0067917
'FC1'	'FC1'	0.99321	0.0067917

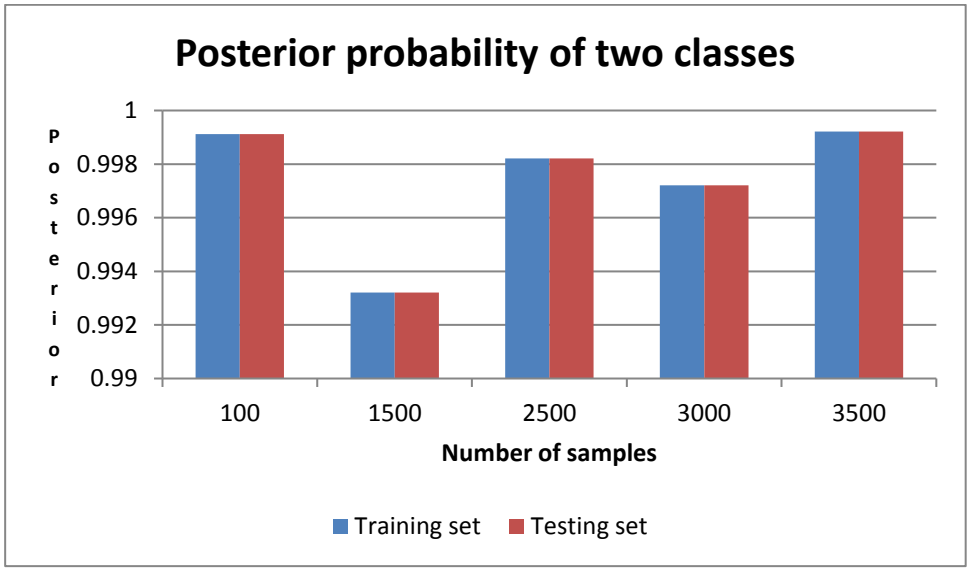


Figure. 5.18: Posterior probability of training vs. testing on EYEDIAP dataset, for two classes FC1 and FC4, using all feature sets.

5.4.4 Evaluation on ACS

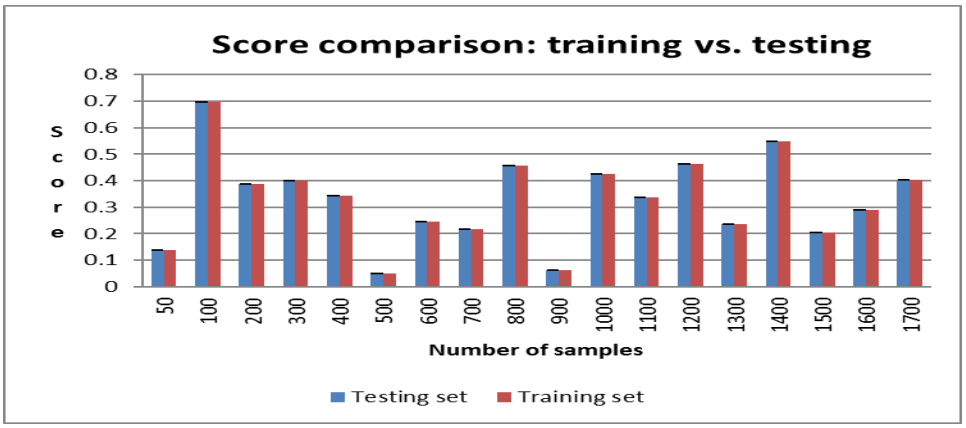


Figure 5.19: Score comparison of training vs. testing of random samples from ACS dataset, using all feature sets. Each sample on the x-axis corresponds to a specific gaze class.

We used ACS dataset (not available to the public) to validate the proposed framework. Fig. 5.19 illustrated the results of score comparison between training vs. testing, using all feature sets. Using multiple features significantly improved the performance of the proposed framework.

A comparison of the proposed framework (using all features) with the methods in [77], [79] [84], [85] and [124] was conducted. Table 5.8 illustrates the performance and accuracy of our method in comparison to other methods. Our method achieved the highest accuracy, which was measured using Eq. 22.

$$\text{Mean Angular Error} = \text{Cos}^{-1}[\text{predicted gaze} \cdot \text{ground truth}] \quad (5.22)$$

Table 5.8: Shows the experimental results validated over MPII, EYEDIAP, Cave and ACS datasets. The accuracy of the predicted gaze and head-pose estimation was indicated in mean angular error.

	MPII	EYEDIAP	CAVE	ACS
Proposed framework	4.5°	4.8°	4.4°	5.0°
Method in [77]	4.8°	6°	N/A	N/A
Method in [79]	5.9°	10.5°	N/A	N/A
Method in [85]	4.6°	7.5°	6.2°	N/A
Method in [124]	4.6°	5.9°	4.8°	5.1°
Method in [84]	5.99° using their own dataset			

5.5 Evaluation on OSLO and UULM

Earlier in the thesis, we validated the proposed framework on MPII, EYEDIAP, CAVE and ACS datasets. We now further evaluate the robustness of the proposed framework by conducting new experiments on two additional datasets: OSLO (3500 male and female face with three gaze directions: left, center and right) [97] and UULM (4000 samples presenting diversity of head-pose and gaze targets) [95]. These datasets offer a wide variety with respect to magnitude, head-pose angles, illumination and facial appearance. We present an extensive comparison with several state-of-the-art head-pose and gaze estimation algorithms on these datasets. The illustration of the experimental results is presented in Fig. 5.20 (showing results from OSLO) and Fig. 5.21 (showing results from UULM).

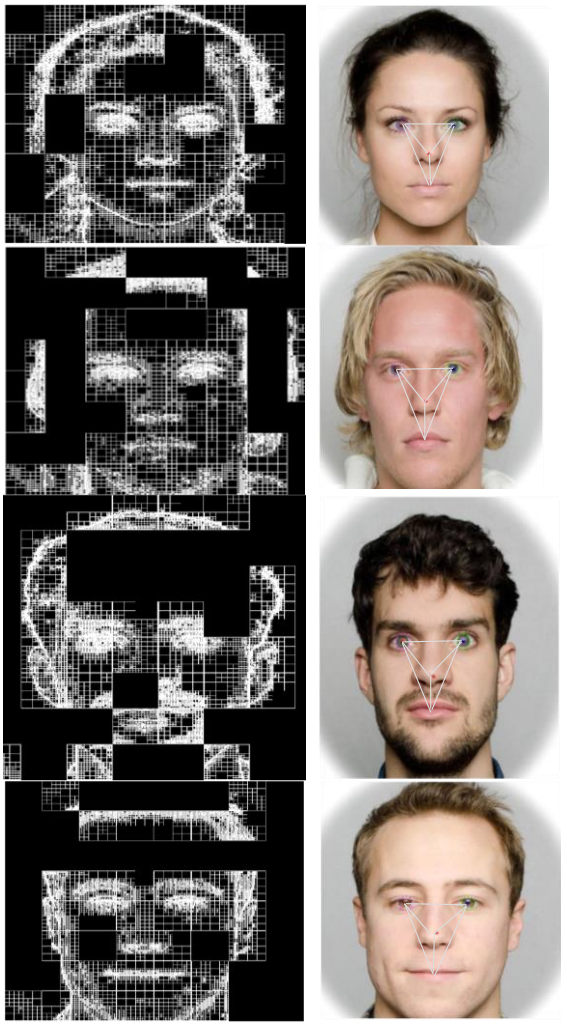


Figure 5.20: Showing samples from OSLO, illustrating QD and selected tiles of the image, identifying the eye region, iris, jaw line (face border) and nose tip.

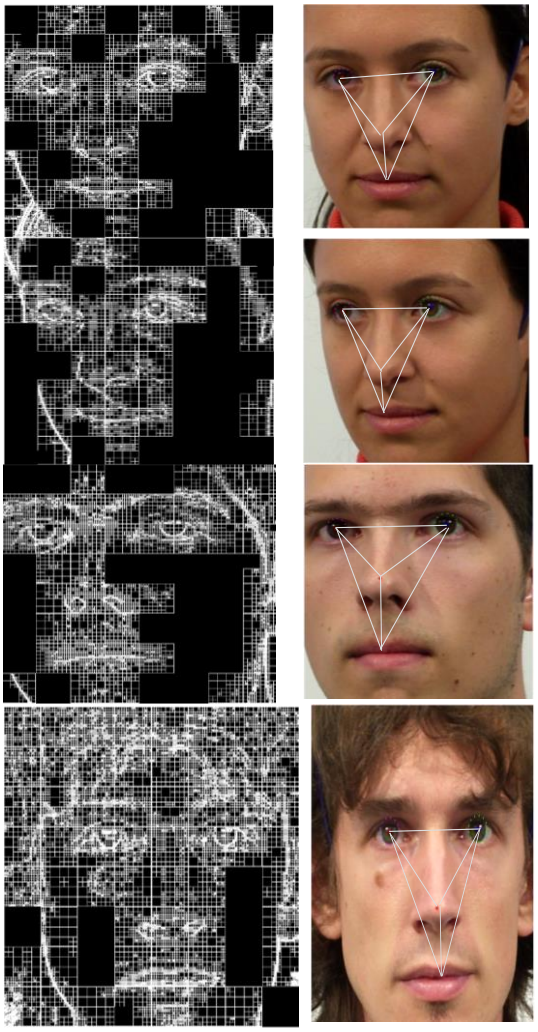


Figure 5.21: Showing samples from UULM, illustrating QD and selected tiles of the image, identifying the eye region, iris, jaw line (face border) and nose tip.

5.5.1 Cross Validation Using Features from the proposed framework in Chapter 4

We applied the framework used in Chapter 4 on MPII, EYEDIAP, CAVE, ACS, OSLO and UULM, which is composed of appearance-based and geometric-based features. The feature extraction process starts with facial landmarks and then narrows focus towards the eye region, along with the region descriptor spatial indexing and the statistical 15D of the eye region. Each feature acts as a mask/label for each sample image. We then employed the kernel-DMCCA to improve the features fusion approach for effective head-pose and gaze estimation.

5.5.2 Cross Validation by Employing Features Using the Proposed Framework (QD and Geometrical Moments) in this chapter

We also applied the proposed framework from this chapter on MPII, EYEDIAP, CAVE, ACS, OSLO and UULM. The framework is based on QD, calculating geometrical moments that define the shape and geometry of the facial landmarks to construct the feature sets: eye region, nose COM, jawline, distance and angles between facial landmarks (distance between the eyes center, distance from the center of each eye to the nose COM, distance from the points on the jawline to the points on the symmetry axis), roll/pitch/yaw, 2^{nd} and 3^{rd} order moments. The extracted features were fused and the training set is constructed using a multiclass one-vs-one SVM model.

5.6 RESULTS AND DISCUSSION

We fused all feature sets presented by the framework in Chapter 4, namely, pupil center, facial landmarks distances and angles, 15D and iris region descriptor. The fusion process is performed by employing K-DMCCA with discriminative correlation criteria based on the 3-σ rule. The training and testing samples were compared using the RBF function, which gives correlations between any two sets of samples. The comparison on OSLO is presented in Fig. 5.22 while Fig. 5.33 presents the comparison on UULM.

One-vs-one SVM multi classification was employed by combining all features presented by the proposed framework in Chapter 5: eye region, nose COM, jawline, roll, yaw and pitch, along with symmetry axis, 2^{nd} and 3^{rd} order moments. Using a fit multiclass error-correcting output code (fitecoc), we created an SVM model using the predictors with a posterior probability value assigned by the learner. The posterior probability comparison between training vs. testing is presented in Fig. 5.24 for OSLO and Fig.5.25 for UULM.

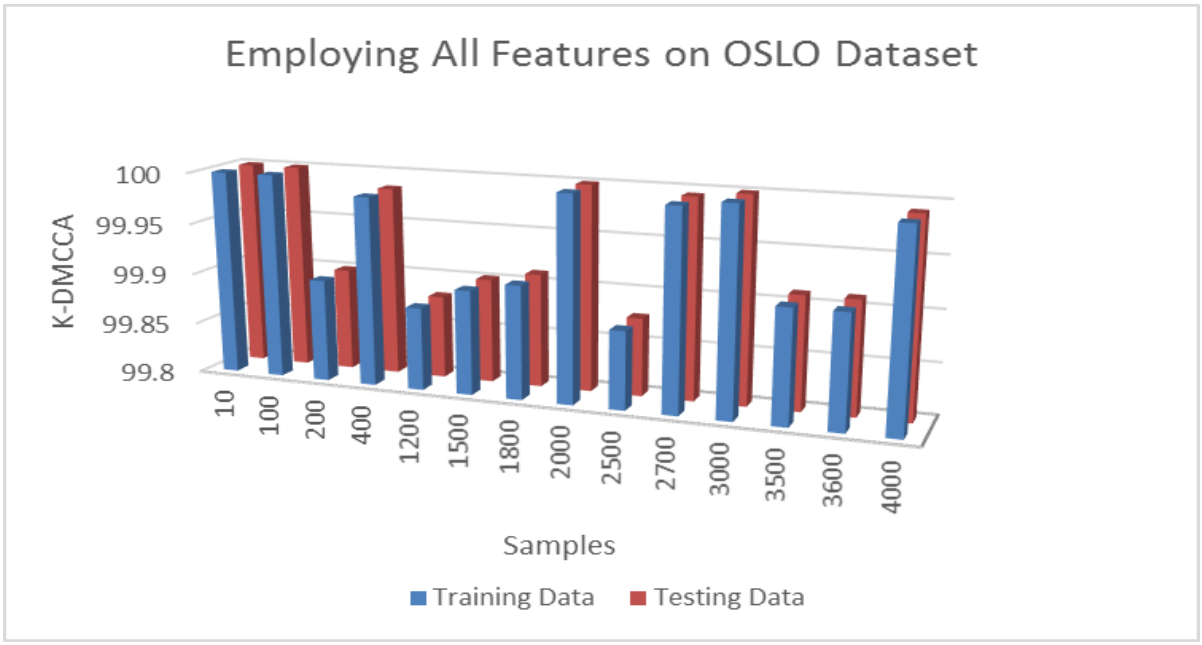


Figure 5.22: Comparison of training vs. testing sets from OSLO, using all features from Chapter 4.

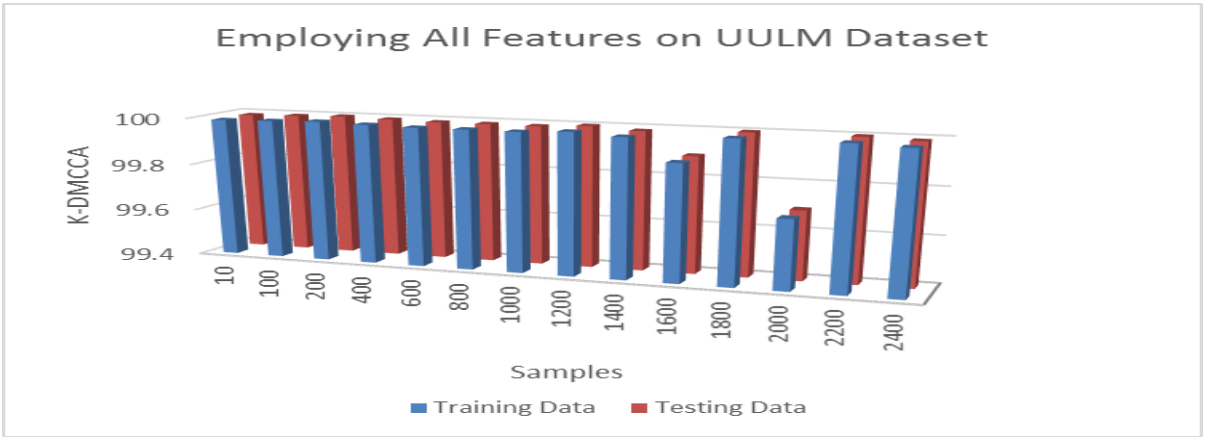


Figure 5.23: Comparison of training vs. testing sets on UULM, using all features from Chapter 4.

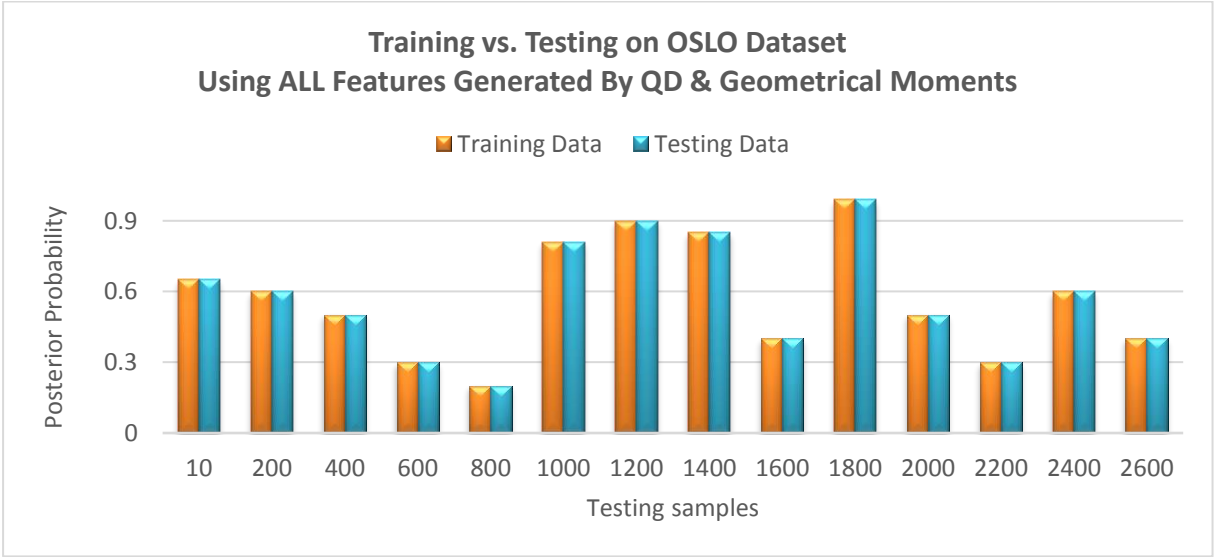


Figure 5.24: Comparison of training vs. testing sets from OSLO, using all features presented by the proposed framework in this chapter.

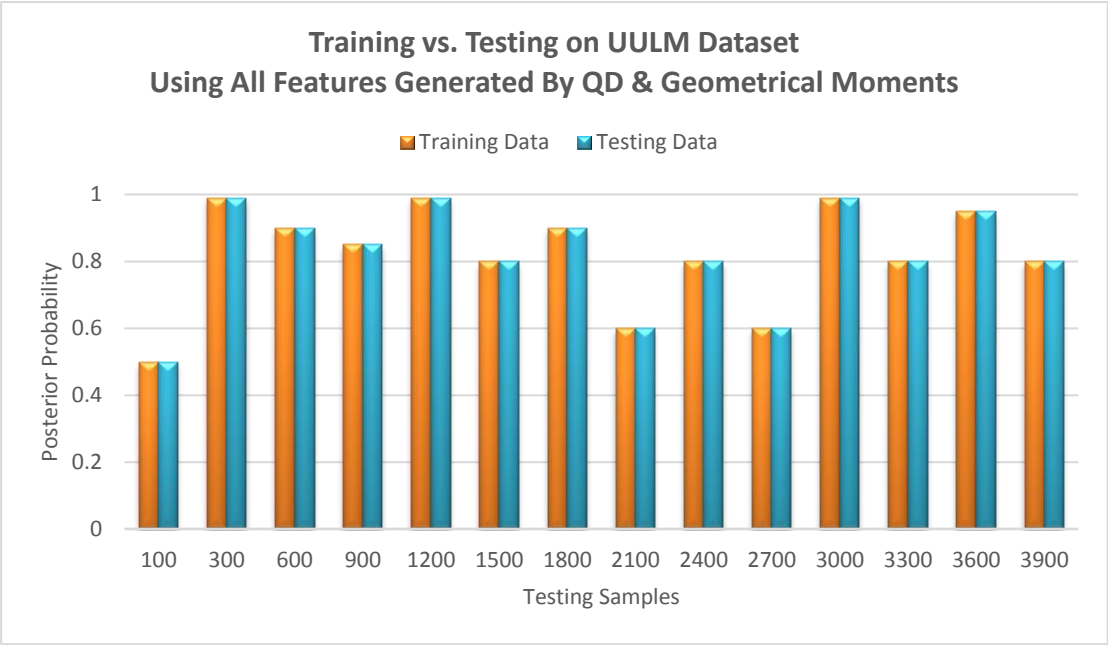


Figure 5.25: Comparison between training vs. testing sets from UULM, using all features presented by the proposed framework in this chapter.

Using all datasets, we measured the mean angular error to measure the accuracy of the algorithm from Chapter 4, see Fig. 5.26.

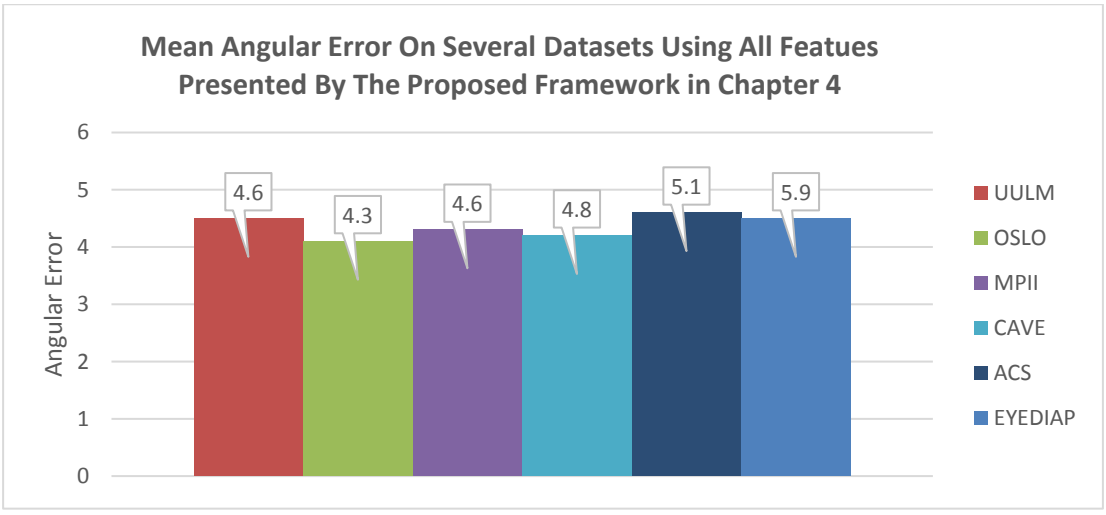


Figure 5.26: Calculating the mean angular error on MPII, EYEDIAP, CAVE, ACS, OSLO and UULM datasets, using all features presented in Chapter 4.

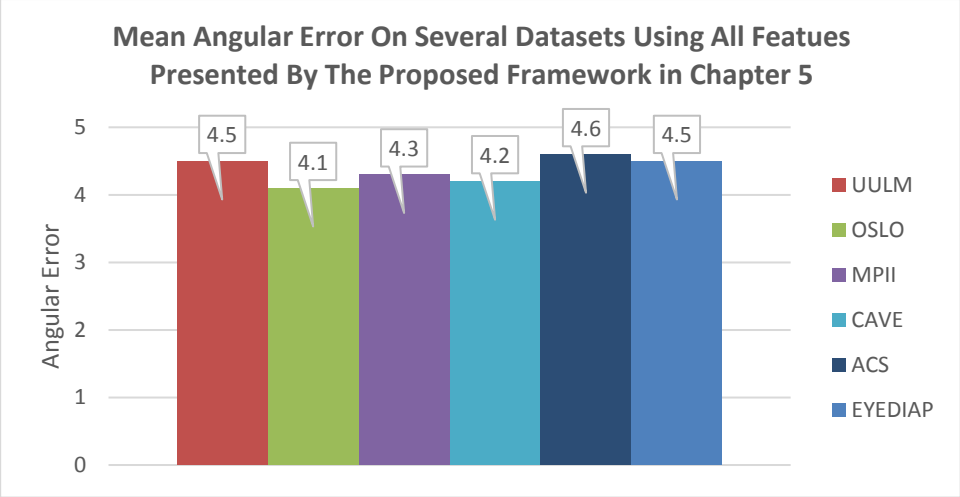


Figure 5.27: Calculating the mean angular error on MPII, EYEDIAP, CAVE, ACS, OSLO and UULM datasets, using all features presented by the proposed framework in this chapter.

Furthermore, we measured the mean angular error to assess accuracy when applying the proposed framework from Chapter 5 on all datasets, see Fig. 5.27 for illustration. We compared the proposed framework against state-of-the-art methods in the literature, the results are recorded in Table 5.9.

Table 5.9: Comparing the proposed framework with recent state of the art methods.

Citation	Dataset	Error
[125] appearance-based features, using CNN, 2017	CAVE	6.7˚
[85] eye region feature localization, then iterative model-fitting (only eye), 2018	CAVE	6.2˚
[Proposed, Chapter 4]	CAVE	**4.8**˚
[Proposed, in this Chapter]	CAVE	**4.2**˚
[126] appearance-based gaze estimation using deep features and random forest regression, 2016	Own dataset	5.0-7.0˚
[127] geometrical based, deep learning, 2017	Own dataset	4.3˚

[128] appearance based, using CNN, 2015	MPII	6˚
[79] Gaussian process regression model combined with a probabilistic filter, 2006	MPII	5.9˚
[85] eye region feature localization, then iterative model-fitting (only eye), 2018	MPII	4.6˚
[129] appearance based on full face, using CNN, 2017	MPII	4.8˚
[130] facial landmark,head pose tracking, face alignment and appearance extraction, **feature fusion**, 2016	MPII	9.96˚
[131] appearance based using generative eye region model, match using nearest-neighbor approach, 2016	MPII	9.58˚
[132] training models of synthetic images, using refiner neural network, 2017	MPII	7.8˚
[133] deep regression Bayesian, Probabilistic deep learning, 2018	MPII	7.1˚
[Proposed, Chapter 4]	MPII	**4.6**˚
[Proposed, in this Chapter]	MPII	**4.3**˚
[134] facial traits extracted from sensory data, from which distance vectors related to gaze derived, 2015	UULM	7.5˚
[135] estimating the gaze direction using Canonical Correlation Analysis (**CCA**), a gaze vector is calculated based on gathered eye properties, 2011	UULM	5.6˚
[136] iris model rotates under the eye hole permitting the synthesis of new gaze directions, using multi-Texture Active Appearance Model, 2013	UULM	7.0˚
[137] 3D Morphable Model (3DMM) of faces is used to obtain a dense 3D reconstruction of the face, then obtain gaze vector. 2014	UULM	9.7˚
[Proposed, Chapter 4]	UULM	4.6˚
[Proposed, in this Chapter]	UULM	4.5˚
[79] Gaussian process regression model combined with a probabilistic filter, 2006	EYEDIAP	10.5˚
[85] eye region feature localization, then iterative model-fitting (only eye), 2018	EYEDIAP	7.5˚
[129] appearance based on full face, using CNN, 2017	EEYDIAP	6.0 ˚
[Proposed, Chapter 4]	EYEDIAP	5.9˚
[Proposed, Chapter 5]	EYEDIAP	4.5˚
[84] initial appearance-based estimation under fixed head pose, then each subsequent stage is solved by either learning-based method or geometric-based calculation, 2015	Own dataset	5.99˚
[Proposed, Chapter 4]	OSLO	**4.3**˚
[Proposed, in this Chapter]	OSLO	**4.1**˚
[Proposed, Chapter 4]	ACS	**5.1**˚
[Proposed, in this Chapter]	ACS	**4.6**˚

It is worth noting that most of the recent methods in Table 5.9 are based on a convolutional neural network (CNN) model [132, 128, 129] with minimal computational requirements. However, the complexity of CNN lies in the multi-layered structure for which the convergence is an essential condition as a performance metric. In addition, some of these methods only used one eye (our method used the full face, both eyes). It would not be wise to say that CNN is less accurate than the proposed framework. CNN does have an inherent advantage, which is

learning hierarchical features, i.e what features are useful and how to compute them, and then use those features to compute the final result. Having said that, we have highlighted how the proposed framework fuses features together using the KDMCCA. The KDMCCA analysis the correlation within one feature set and in between feature sets. The proposed framework can work on large datasets or small datasets, however, CNN required much larger datasets to achieve higher accuracy. At the same time, for real-time processing, CNN will be more adequate to use (in comparison with the proposed framework).

It is also worth to highlight, that although some methods achieved the same accuracy as the proposed framework, our proposed framework differs by the newly developed features and the methodology the features were constructed.

The method in [130] adopted a feature fusion approach by using facial landmark, head pose tracking, face alignment and appearance extraction. The method in [135] also adopted feature fusion and estimated the gaze direction using Canonical Correlation Analysis (CCA); a gaze vector is calculated based on gathered eye properties. It is worth noting that the Kernel-DMCCA method we adopted and improved differs from CCA in the following: 1) the correlation among the samples in multiple channels is taken as the metric of the similarity between the samples; 2) unlike CCA and Multiple CCA, both the within-class similarity and the between-class dissimilarity is considered by K-DMCCA. Experimental results showed that we achieved more accurate results by using K-DMCCA

We studied key challenges including wide range of gaze classes, illumination conditions, and facial appearance variation. Experimental results showed that image resolution, illumination, pupil center localization, the use of both eyes affect the head-pose and gaze estimation performance. We validated the proposed framework on six different datasets, the accuracy of the proposed framework differs based on the condition of each dataset. ACS dataset was collected for simulation of real-life driving settings. The wide range of ambient conditions surrounding the samples along with the unconstrained head movement made it more difficult for the proposed algorithm. As a result, experimental results were the least accurate when validated on ACS, an average error of 4.6˚ was recorded. Moreover, although UULM dataset had consistent good resolution, but also had a wide range of gaze positions, as well as four degrees of freedom of head-pose displacement. As a result, the proposed framework achieved an average error of 4.5˚. Furthermore, MPII and EYEDIAP datasets were collected without

assumptions regarding user, environment, or camera; these two datasets resembled real life scenarios. Many samples from MPII and EYEDIAP had occlusions such as facial hair, bangs, eyelids occluding the iris, accessories like glasses and low-resolution images, which made the gaze estimation computationally complex. However, EYEDIAP and MPII did not have many gaze glasses. As a result, the proposed framework achieved an average error of 4.5˚ on EYEPIAP and 4.3˚ on MPII. The proposed framework achieved the best result on CAVE and OSLO datasets, because although these datasets had a wide range of gaze classes, but they were collected under controlled lab settings. The resolution, ambient environment and illumination were all consistent throughout the datasets. In addition, CAVE dataset was collected while all users had their chin rested on a chinrest, as a result, the proposed algorithm achieved the best result; an average error of 4.2˚ on CAVE and 4.1˚ on OSLO.

5.7 Summary

This chapter investigated new features to replace features used in chapter 4, for the purpose of arriving at more accurate gaze and head-pose estimation. We validated the proposed framework from Chapter 4 and this chapter against recent head-pose and gaze estimation methods in literature. We chose datasets which offered variability with respect to magnitude, head-pose angles, occlusion, illumination and facial appearance. The proposed framework from Chapter 4 and this chapter outperform other methods with respect to accuracy in this extensive cross-validation comparison.

Chapter 6

6.1 Conclusion

Multi-modal information fusion refers to a process which attempts to achieve more reliable and robust analysis performance by integrating a set of multiple data sources, extracted features, and intermediate decisions. Multi-feature fusion (for the purpose of gaze and head-pose estimation) is a special case of multimodal fusion. In multi-feature fusion for head-pose and gaze estimation, different sets of features are extracted from the same modality data but using different extraction methods, and it is likely to carry richer information where each feature acts as a mask/label for each sample. Therefore, we adopted the fusion process of multi-features because it was our intuition that this would lead to more accurate estimation results.

After a comprehensive background study in Chapter 2, we proposed a novel unsupervised pupil localization graph-based method in Chapter 3: an important step towards gaze estimation. We then employed an unsupervised pupil localization graph-based method with a newly developed iris region descriptor based on quadtree and merged them with existing features from the literature in Chapter 4. We then introduced a discriminative robust head-pose and gaze estimation method using Kernel-DMCCA (the extension of DMCCA) fusion.

The following observations are worth noting:

1. The introduction of quadtree as an iris region descriptor was a robust method for detecting the iris boundary, and it is inclusive of statistical and geometrical indexing that are calibration free. The iris region descriptor was then extended as a general region descriptor to define the facial landmark features: the whole eye region (not only the iris region), face boundary by defining the jawline, nose tip, mouth region and the distances and angles between facial landmarks. These features were employed in a new framework for head-pose and gaze estimation, which produced accurate results as illustrated earlier.

2. Enhancing the performance of DMCCA with kernel to transform the features into a higher dimensional space, which established a better correlation between the training and testing

samples. Extracting multiple features increased the size of the features set, hence, Kernel transformation was essential to select the most important for classification.

3. Chapter 5 investigated a new alternative methodology to replace the existing features in literature that were used in Chapter 4. This was achieved by extending the iris region descriptor feature using quadtree (which was presented in Chapter 4), and developing it further to structure new features for the purpose of achieving more accurate head-pose and gaze estimation.

In Chapter 5, we decided to use kernel-SVM in classification which locates a separating hyperplane in the feature space and classify points in that space. It does not need to represent the space explicitly, simply by defining a kernel function, where the kernel function plays the role of the dot product in the feature space.

A significant part of the future work will be dedicated to comparing K-DMCCA (currently using a gaussian kernel) along with a classifier vs. Kernel SVM (currently using a gaussian kernel). The proposed framework employed gaussian kernel, and there is a need to employ different kernel functions and compare each with the results obtained by the proposed framework.

References

[1] A. Sharma and P. Abrol, "Eye Gaze Techniques for Human Computer Interaction: Research Survey," International Journal of Computer Applications, volume 71, issue 9, 2013,

[2] A. Sesin, M Adjouadi, M. Cabrerizo, M. Ayala and A. Barreto, "Adaptive eye gaze tracking using neural-network-based user profiles to assist people with motor disability," Journal of rehabilitation research and development, Volume 45, Issue 6. 2008.

[3] J. S. Agustin, J. P. Hansen and J. Mateo, "Gaze beats mouse: hands-free selection by combining gaze and emg," China, 2008.

[4] A. Doshi, and M. M. Trived, "On the roles of eye gaze and head pose in predicting driver's intent to change lanes. ITS," September 2009.

[5] R. Parte, G. Mundkar, N. Karande, S. Nain, and N. Bhosale, "A Survey on Eye Tracking and Detection,"cInternational Journal of Innovative Researc in Science and Engineering, vol. 4, no. 10, Oct. 2015.

[6] S. R. Langton, H. Honeyman, and E. Tessler, "The influence of head contour and nose angle on the perception of eye-gaze direction," Perception & psychophysics, 66(5), 2004.

[7] Cogain "Communication by gaze interaction, gazing into the future," http://www.cogain.org, September 2006.

[8] X. Liu, N. Krahnstoever, T. Yu, and P. Tu, "What are customers looking at," Advanced Video and Signal Based Surveillance (AVSS), 2007.

[9] N. Robertson and I. Reid, "Estimating gaze direction from low-resolution faces in video," In ECCV, 2006.

[10] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," PAMI, 31(4), 2009.

[11] D.W. 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, 2010

[12] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao. 2d cascaded adaboost for eye localization. In ICPR, 2006

[13] S. Kim, S.-T. Chung, S. Jung, D. Oh, J. Kim and S. Cho, "Multi-scale gabor feature based eye localization," In World Academy of Science, Engineering and Technology, 2007.

[14] B. Kroon, A. Hanjalic and S. M. Maas, "Eye localization for face matching: is it always useful and under what conditions," CIVR, 2008.

[15] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature'"

CVPR, 2008.

[16] L. P. Morency, J. Whitehill and J. Movellan, "Monocular head pose estimation using generalized adaptive view-based appearance model," Image and Vision Computing, Volume 28, Issue 5, 2010.

[17] S. Huang, Kohsia and M. Trivedi, "Robust Real-Time Detection, Tracking, and Pose Estimation of Faces in Video Streams," ICPR, 2004.

[18] Y. Hu, L. Chen, Y. Zhou and H. Zhang, "Estimating face pose by facial asymmetry and geometry," Proc. Autom. Face Gesture Recog., pp. 651-656, 2004.

[19] K. Ho An and M. Jin Chung, "3D Head Tracking and Pose-Robust 2D Texture Map-Based Face Recognition using a Simple Ellipsoid Model," 2010.

[20] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head-pose and gaze direction measurement," in Proc. FG, pp. 499–505, 2000,

[21] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky, "Real-time stereo tracking for head pose and gaze estimation," in Proc. Autom. Face Gesture Recog., pp. 122–128, 2000.

[22] Q. Ji and X. Yang, "Real-time eye, gaze and face pose tracking for monitoring driver vigilance," Real Time Imaging, vol. 8, no. 5, pp. 357–377, Oct. 2002.

[23] A. Picot, S. Charbonnier and A Caplier, "Drowsiness detection based on visual signs: blinking analysis based on high frame rate video," in IEEE Intl. Instrumentation and Measurement Technology Conference, USA, 2010.

[24] G. Xu, Z. Zhang and Y. Ma, "Improving the performance of iris recognition system using eyelids and eyelashes detection and iris image enhancement," In IEEE Intl. Conf. on Cognitive Informatics (ICCI), 2006.

[25] P. Majaranta, and K. J. Raiha, "Twenty Years of Eye Typing: Systems and Design Issues," In Eye Tracking Research & Applications (ETRA) Symposium," LA, 2006.

[26] F Riquier, B. Herbelin, H. Grillon and D. Thalmann, "Use of virtual reality as therapeutic tool for behavioral exposure in the ambit of social anxiety disorder treatment," In Intl. Conf. Series on Disability, Virtual Reality & Associated Technologies, pp. 105-112, 2006.

[27] H. H. Greene and K. Rayner, "Eye Movements and Familiarity Effects," in Visual Search. Vision Research, 41(27), 2001.

[28] J. Goldberg, M. Stimson, M. Lewnstein, N. Scott, and Wichansky, "Eye Tracking in Web Search Tasks: Design Implications," In Eye Tracking Research & Applications (ETRA) Symposium. New Orleans, LA, 2002.

[29] K. Rayner, “Eye Movements in Reading and Information Processing: 20 Years of Research,” Psychological Bulletin, 2008.

[30] B. Pradhan, T. Parikh, R. Makani and M. Sahoo, “Ketamine, transcranial magnetic stimulation, and depression specific yoga and mindfulness based cognitive therapy in management of treatment resistant depression,” Review and some data on efficacy. Depression Research and Treatment, 2015.

[31] M. Reale, T. Hung and L. Yin, “Viewing direction estimation based on 3D eyeball construction for HRI, ” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[32] Hsin-Pei Sun, Cheng-Hsun Yang and Shang-Hong Lai, “A Deep Learning Approach to Appearance-Based Gaze Estimation under Head Pose Variations,” Asian Conference on Pattern Recognition (ACPR), 4th IAPR 2017.

[33] Y. Matsumoto, T. Ogasawara, and Zelinsky, “Behavior recognition based on head pose and gaze direction measurement,” IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.

[34] M. Kumar, D. P. Garg, and R. A. Zachery, “A generalized approach for inconsistency detection in data fusion from multiple sensors,” in American Control Conference, 2006, pp. 6–10, IEEE, 2006.

[35] P. Smets, “Analyzing the combination of conflicting belief functions,” Information Fusion, vol. 8, pp. 387–412, 2007.

[36] T.-K. Sun, S.-C. Chen, Z. Jin, and J.-Y. Yang, “Kernelized discriminative canonical correlation analysis,” in Wavelet Analysis and Pattern Recognition, 2007. ICWAPR’07. International Conference on, vol. 3, pp. 1283–1287, IEEE, 2007.

[37] A. O. Mohamed, M. P. da Silva, and Vincent Courboulay. "A history of eye gaze tracking," Rapport Interne. 2007. Available from: https://hal.archivesouvertes.fr/hal-00215967

[38] A. T. Duchowski, “Behavior Research Methods, Instruments, & Computers,” 2002.

[39] C. H. Morimoto and M. R. M. Mimica, “Eye gaze tracking techniques for interactive applications,” Computer. Vis. Image Underst., vol. 98, no. 1, pp. 4–24, 2005.

[40] R. Parte, G. Mundkar, N. Karande, S. Nain, and N. Bhosale, “A Survey on Eye Tracking and Detection”, International Journal of Innovative Research in Science and Engineering, vol. 4, no. 10, Oct. 2015.

[41] D.W. 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, 2010.

[42] Soelistio, Y. Eko, E. Postma, and A. Maes. "Circle-based eye center localization (CECL)," 14th IAPR International Conference on Machine Vision Applications (MVA), pp. 349-352, 2015.

[43] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. "Adaptive Linear Regression for Appearance-Based Gaze Estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 10, 2014.

[44] M. Leo, D. Cazzato, T. De Marco, and C. Distante, "Unsupervised Eye Pupil Localization through Differential Geometry and Local Self-Similarity Matching," PloS one 9, no. 8, Aug. 2014.

[45] D. W. 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, 2010.

[46] O. Špakov and D. Miniotas, "Gaze-based selection of standard-size menu items," In Proceedings of the 7th international conference on Multimodal interfaces (ICMI), NY, USA, 124-128, 2005.

[47] M. Kumar, A. Paepcke, and T. Winograd, "EyePoint," Proc. SIGCHI Conf. Hum. factors Comput. Syst," CHI, p. 421, 2007.

[48] L. Iannizzotto and F. La Rosa, "Competitive combination of Multiple Eye detection and Tracking Techniques," IEEE Transactions on Industrial Electronics, 58(8): 3151-3159, 2011.

[49] 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, 32(3):478–500, 2011.

[50] H. Yamazoe, A. Utsum, T. Yonezawa, and A. Abe, "Remote Gaze Estimation with a Single Camera Based on Facial-Feature Tracking without Special Calibration Actions," Proceedings of the Eye Tracking Research & Application Symposium ETRA, Savannah, Georgia, 2008.

[51] I. Taba I, "Improving Eye-Gaze Tracking Accuracy Through Personalized Calibration of a User's Aspherical Corneal Model," MS Thesis, University of British Columbia, 2012.

[52] X-H Yang, J-D Sun, J. Liu, X-C Li, C-X Yang and W. Liu, "A Remote Gaze Tracking System Using Gray-Distribution-Based Video, "Processing, Journal of Biomedical Engg: Applns., Basis & Communications, 217_227, 2012.

[53] J. Merchant, R. Morrissette, and J. L. Porterfield, "Remote measurement of eye direction allowing subject motion over one cubic foot of space," IEEE Transactions on Biomedical Engineering, 309-317, 1974.

[54] X. L. C. Brolly, and J. B. Mulligan, "Implicit calibration of a remote gaze tracker," Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), Volume 8, pp. 134, 2004.

[55] J. J. Cerrolaza, A. Villanueva, and R. Caveza, "Study of polynomial mapping functions in videooculography eye trackers," in ACM Trans. Comput.-Hum. Interaction, Article 10, 25, 2012.

[56] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks," In Proceedings of Advances in Neural Information Processing Systems, volume 6, p 753–760, 1994.

[57] E. Demjen, V. Abosi, and Tomori, "Eye Tracking Using Artificial Neural Networks for Human Computer Interaction, Physiological Research, 2011.

[58] D. Torricelli, S. Conforto, M. Schmid, and T. D'Alessio, "A neural-based remote eye gaze tracker under natural head motion," Comput. Methods Programs Biomed., vol. 92, pp. 66–78, 2008.

[59] J. Orozco, F. Xavier Roca and J. Gonzalez, "Real time gaze tracking with appearance based models," Machine Vision and Applications, 20(6), 353-364, 2009.

[60] F. Lu, Y. Sugano, O. Takahiro, and Y Sato, "A head pose-free approach for appearance-based gaze estimation," in Proc. of the 22nd British Machine Vision Conf, 2011.

[61] S.V. Sheela and P.A. Vijaya, "Mapping Functions in Gaze Tracking," International Journal of Computer Applications 26(3):36-42, 2011.

[62] T.D. Rikert and M.J. Jones, "Gaze estimation using morphable models," IEEE International Conference on Automatic Face and Gesture Recognition, 436-441, 1998.

[63] T. Kar-Han, D.J Kriegman and N. Ahuja, "Appearance-based eye gaze estimation," IEEE Workshop on Applications of Computer Vision, pp. 191-195.

[64] Y. Sugano, Y. Matsushita and Y. Sato, "Appearance-based Gaze Estimation using Visual Saliency," IEEE Trans. Pattern Analysis and Machine Intelligence, 2012.

[65] Flavio L. Coutinho and Carlos H. Morimoto, "Improving head movement tolerance of cross-ratio based eye trackers," International Journal of Computer Vision, 2012.

[66] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance based gaze estimation in the wild," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p 4511–4520, 2015.

[67] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2176–2184, 2016.

[68] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," In International Symposium on Eye Tracking Research & Applications (ETRA), p 207–210, 2014.

[69] A. Kar and P. Corcoran, "A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms

and Performance Evaluation Methods in Consumer Platforms," 2017.

[70] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2176–2184, 2016.

[71] Stefania Cristina, Kenneth P. Camilleri, "Model-based head pose-free gaze estimation for assistive communication," University of Malta.

[72] N. M. Coerra, T. Eicjele, T. Adali, Y. Li and V. D. Calhoun, "Multi-set canonical correlation analysis for the fusion of concurrent single trial ERP and functional MRI," 2010.

[73] L. Gao, L. Qi, E. Chen and L. Guan, "Discriminative Multiple Canonical Correlation Analysis for Multi-feature Information Fusion," IEEE International Symposium on Multimedia, CA, 2012.

[74] J. Chen and Q. Ji, "3D gaze estimation with a single camera without ir illumination," In International Conference on Pattern Recognition (ICPR), pages 1–4, 2008.

[75] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," IEEE Transactions on Image Processing, 21(2):802–815, 2012.

[76] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," In International Symposium on Eye Tracking Research & Applications (ETRA), p 207–210, 2014.

[77] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks", In Advances in Neural Information Processing Systems (NIPS), pages 753–760, 1994.

[78] K.-H. Tan, D. J. Kriegman, and N. Ahuja, "Appearance-based eye gaze estimation," In IEEE Workshop on Applications of Computer Vision (WACV), pages 191–195, 2002.

[79] O. Williams, A. Blake, and R. Cipolla, "Sparse and Semisupervised Visual Mapping with the Sˆ 3GP," In IEEE Conference on Computer Vision and Pattern Recognition, 2006.

[80] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Gaze estimation from eye appearance: A head pose-free method via eye image synthesis," IEEE Transactions on Image Processing, 3680–3693, 2015.

[81] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "Appearance based gaze estimation with online calibration from mouse operations," IEEE Transactions on Human-Machine Systems, 2015.

[82] K. A. Funes-Mora and J.-M. Odobez, "Gaze estimation in the 3d space using rgb-d sensors," International Journal of Computer Vision, pages 1–23, 2015.

[83] M. Mansouryar, J. Steil ,Y. Sugano and A. Bulling, "3D Gaze Estimation from 2D Pupil Positions on Monocular Head-Mounted Eye Trackers," 2016.

[84] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, “A Head Pose-free Approach for Appearance-based Gaze Estimation. Image and Vision Computing,” 32(3):169–179, 2014.

[85] S. Park, X. Zhang, A. Bulling, ”Learning to Find Eye Region Landmarks Remote Gaze Estimation in Unconstrained Settings,” In Computer Vision and Pattern Recognition, May 12, 2018.

[86] F. Lu, Y. Sugano, T. Okabe, and Y Sato, “Adaptive Linear Regression for Appearance-Based Gaze Estimation,” IEEE Trans Pattern Anal Mach Intell., 2014.

[87] P. Fitzpatrick “Pose estimation without manual initialization,” AI Lab , MIT Cambridge, USA.

[88] M. A. Muqeet and H. S. Raghunath, “Local binary patterns based directional wavelet transform for expression and pose-invariant face recognition,” Applied Computing and Informatics, 2017.

[89] Chu-Yin Chang, A A. Maciejewski, V Balakrishnan, R G. Roberts and K Saitwal, “Quadtree-based eigen-decomposition for pose estimation,” in the presence of occlusion and background clutter,Pattern Anal Application, 2007.

[90] Ming-Hao YANG, Jian-Hua TAO, “Data fusion methods in multimodal human computer dialog. Virtual Reality & Intelligent Hardware,” 2019.

[91] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” Neural Computation, vol. 16, no. 12, pp. 2639–2664, 2004.

[92] W. K. H¨ardle and L. Simar, “Canonical correlation analysis,” in Applied Multivariate Statistical Analysis, pp. 443–454, Springer, 2015.

[93] W. H¨ardle and L. Simar, “Canonical correlation analysis,” Applied Multivariate Statistical Analysis, pp. 321–330, 2007.

[94] National Laboratory of Pattern Recognition database. Institute of Automation, Chinese Academy of Sciences, Biometric Ideal, www.cbsr.ia.ac.cn. Iris Database, 2010.

[95] U. Weidenbacher, G. Layher, P. M. Strauss, and H. Neumann, “A Comprehensive Head Pose and Gaze Database,” Proceeding of the 3rd IET International Conference on Intelligent Environments (IE07), pp. 455-458, Ulm, Germany, 2007.

[96] K. A. F. Mora, F. Monay, and J.-M. Odobez, “Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras,” In International Symposium on Eye Tracking Research & Applications, pages 255–258, 2014.

[97] O. Chelnokova, B. Laeng, M. Eikemo, J. Riegels, G. Løseth, H. Maurud, F. Willoch, and S. Leknes, “Rewards of Beauty: The Opioid System Mediates Social Motivation in Humans, “Mol Psychiatry 19:746-747, 2014.

[98] https://acs-corp.com, ACS dataset.

[99] Nilanjan Dey, Girish Mishra, Jajnyaseni Kar, Sayan Chakraborty, and Siddhartha Nath, "A survey of image classification methods and techniques," International Conference on Control, Instrumentation, Communication and Computational Technologies, 2014.

[100] D. Sidibe, P. Montesinos, and S. Janaqi, "A simple and efficient eye detection method in color images," International Conference Image and Vision Computing, New Zealand, 2006.

[101] G. Kootstra and L. R. B. Schomaker, "Predicting Eye Fixations on Complex Visual Stimuli using Local Symmetry," Cognitive computation, vol. 3, no. 1, pp. 223-240, 2011.

[102] Soelistio, Y. Eko, E. Postma, and A. Maes, "Circle-based eye center localization (CECL)," 14th IAPR International Conference on Machine Vision Applications (MVA), pp. 349-352, 2015.

[103] M. Leo, D. Cazzato, T. De Marco, and C. Distante, "Unsupervised Eye Pupil Localization through Differential Geometry and Local Self-Similarity Matching,"Plos, no. 8, Aug. 2014.

[104] S. Rabba , Y. He , M. Kyan , L. Guan, "Pupil localization for gaze estimation using unsupervised graph-based model," ISCAS IEEE International Symposium, 2017.

[105] A. Strupczewski, "Commodity Camera Eye Gaze Tracking," Ph.D. thesis, Warsaw University of Technology, 2016.

[106] H. Samet and R. E. Webber, "On Encoding Boundaries with Quadtrees," IEEE transaction on Pattern analysis and machine intelligence , PAMI-6, No 3, 1984.

[107] P. L. rosin, "Techniques for assessing polygonal approximation of curves," IEEE transaction on pattern analysis and machine intelligence, vol 19 , no 6 , 1997.

[108] S. Wang, Feng Ge and T. Liu Hindawi, "Evaluating edge detection through boundary detection," Publishing, Journal of Applied Signal Processing , Vol 2006, article ID 76278 , 2005.

[109] R. Romero, E. Iglesias, and L. Borrajo, "A linear-RBF multi kernel SVM to classify big text corpora," BioMed Research International, 2015.

[110] K. Karsch, C. Liu, S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling & quot," IEEE transactions on pattern analysis and machine intelligence, 2014.

[111] D. Herrera, J. Kannala, and J. Heikkilä, "Joint depth and color camera calibration with distortion correction," IEEE Trans Pattern Anal Mach Intelligence. 2012.

[112] P. Mazumder, "Planar decomposition using quadtree," Computer vision, graphics and Image processing, 1986.

[113] R. J. Prokop and A. P. Reeves, "A Survey of moment based techniques for unoccluded object

representation and recognition," CVGIP ,Graphical models and Image processing, Vol 54, Issue 5, p 438-460, 1992.

[114] Mohamed Ould Djibril and Rachid Oulad Haj Thami, "A New Quadtree-Based Symmetry Transform With Application to Arab-andalusian images Indexing," ISCC, 2006.

[115] M. A. Muqeet and R. A. Holambe, "Local binary patterns based directional wavelet transform for expression and pose-invariant face recognition," In Applied Computing and Informatics Volume 15, Issue 2, P 163-171, 2019.

[116] Chu-Yin Chang, A. A. Maciejewski, V Balakrishnan, R. G. Roberts and K. Saitwal, "Quadtree-based eigendecomposition for pose estimation in the presence of occlusion and background clutter," In Pattern Anal Applic, 2007.

[117] K. Peng, L. Chen, and K Georgy, "A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles," Journal of Computer Science and Technology – JCST, 2005.

[118] Arcoverde Euclides M. Barreto Rafael, M. Duarte Rafael, Paulo Magalhaes Joao, A. C. M. Bastos Carlos, Ing Ren Tsang and Cavalcanti George, "Real-Time Head Pose Estimation for Mobile Devices," Book: A Weightless Neural Network-Based Approach for Stream Data Clustering p 467-474., 2012.

[119] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proceedings of DARPA Image Understanding, pp. 121–130, 1981.

[120] Guangliang Chen, Wilson Florero-Salinas and Dan Li, "Simple, Fast and Accurate Hyper-parameter Tuning in Gaussian-kernel SVM," International Joint Conference on Neural Networks (IJCNN), 2017.

[121] MING-KUEI Hut, "Visual Pattern Recognition by Moment Invariants," IRE TRANSACTIONS ON INFORMATION THEORY.

[122] J. Rodríguez, A. Pérez, and J. A, Lozano, "Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. Pattern Analysis and Machine Intelligence," IEEE Transactions, p 32. 569 – 575, 2010.

[123] L.J.P. Van der Maaten, and G. E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," Journal of Machine Learning Research, 2008.

[124] S. Rabba, M. Kyan, L. Gao, A. Quddus, A. S. Zandi, L. Guan, "Discriminative Robust Gaze Estimation Using Kernel-DMCCA Fusio," in ISM, p 291-298, 2018.

[125] R. Konrad, S. Shrestha, and P. Varma, "Near-eye display gaze tracking via convolutional neural networks." 2016.

[126] Y. Wang, T. Shen, G. Yuan, J. Bian, and X. Fu, "Appearance-based gaze estimation using deep

features and random forest regression," vol. 110, no. C, pp. 293–301, 2016.

[127] H. Deng and W. Zhu, "Monocular free-head 3d gaze tracking with deep learning and geometry constraints," In 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

[128] X. Zhang, Y. Sugano, M. Mora, and A. Bulling, "Appearance-based gaze estimation in the wild," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p 4511–4520, 2015.

[129] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017.

[130] T. Baltrusaitis, P. Robinson, and L. P. Morency, "Openface: An open source facial behavior analysis toolkit," IEEE Winter Conference on Applications of Computer Vision (WACV), 2016.

[131] E. Wood, T. Baltruˇsaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesized images," In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ACM, pp. 131–138. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857492 , 2016.

[132] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p 2242–2251, 2017.

[133] S. Nie, M. Zheng, and Q. Ji, "The deep regression bayesian network and its applications: Probabilistic deep learning for computer vision," IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 101–111, 2018.

[134] Evangelos Skodras n, Vasileios G. Kanas, Nikolaos Fakotakis, " On visual gaze tracking based on a single low cost camera," Department of Electrical and Computer Engineering, University of Patras, Patras, Greece, IEEE Signal Processing: Image Communication, 2015.

[135] T. Heyman,V. Spruyt, A. Ledda, "3D face tracking and gaze estimation using a monocular camera," In The Second International Conference on Positioning and Context-Awareness(POCA) , pp.23–28, 2011.

[136] H. Salam, R. Seguier, and N. Stoiber, "Integrating head-pose to a 3D multi-texture approach for gaze detection," In Int. J. Multimed. Appl., 2013.

[137] B. Egger, S. Schonborn, A. Forster, and T. Vetter, "Pose Normalization for Eye Gaze Estimation and Facial Attribute Description from Still Images," Department for Mathematics and Computer Science, University of Basel, Basel, Switzerland, Conference Paper, 2014.

[138] Andrew C Gallup, Andrew Chong, and Alex Kacelnik, "The influence of emotional facial

expressions on gaze-following in grouped and solitary pedestrians," Scientific reports, Volume 4, Issue 1, 2014.

[139] S. O. Ba and J. M. Odobez, "Multiperson visual focus of attention from head pose and meeting contextual cues," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 1, pp. 101–116, 2011.