REPUBLIC OF TÜRKİYE

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

FIXEIN: A MOBILE AND MODULAR EYE TRACKING SYSTEM

Şerif İNANIR

MASTER OF SCIENCE THESIS

Department of Computer Engineering

Program of Computer Engineering

Supervisor

Assoc. Prof. Dr. Ali Can KARACA

August, 2024

REPUBLIC OF TÜRKİYE

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

FIXEIN: A MOBILE AND MODULAR EYE TRACKING SYSTEM

A thesis submitted by Şerif İNANIR in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 09.08.2024 in Department of Computer Engineering Program.

Assoc. Prof. Dr. Ali Can KARACA
Yıldız Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Ali Can KARACA, Supervisor
Yildiz Technical University _____

Assoc. Prof. Dr. Hamza Osman İLHAN, Member
Yildiz Technical University _____

Assoc. Prof. Dr. Tuğba DALYAN, Member
İstanbul Bilgi University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of "FIXEIN: A Mobile and Modular Eye Tracking System" supervised by my supervisor, Assoc. Prof. Dr. Ali Can KARACA. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Şerif İNANIR

Signature

Dedicated to my beloved family

and my esteemed professors

ACKNOWLEDGEMENTS

To the emergence of this thesis, which brings me so much joy that it causes to extend my graduate education, I would like to sincerely thank my dear family for their motivational contributions, understanding, and guidance. Also, I thank to my spouse Deniz İNANIR for her effects of my academic work.

It was invaluable for me to reflect the perspectives I gained from Pınar ERTAN BİRECİK and Sina BİRECİK, who have enriched my understanding of various engineering disciplines, in this thesis. Additionally, I am deeply grateful to my managers Özlem SAĞLAM and Mehmet GENÇ, who enabled me to return home happy and with the energy to continue writing my thesis after hard working days.

I would also like to thank Dr. Wolfgang FUHL and doctoral candidate Sean Anthony BYRNE for their objective insights and helpful attitudes regarding their valuable and inspiring work on eye tracking technologies.

Finally, I extend my heartfelt and special thanks to Dr. Yılmaz Kemal YÜCE (my academic father), who introduced me to eye tracking technologies and was my advisor during my undergraduate studies, and to my master's advisor Dr. Ali Can KARACA, whose guidance has illuminated my path during my master's studies and whose light I will carry into my future works as well.

Şerif İNANIR

TABLE OF CONTENTS

LIST OF ABBREVIATIONS viii
LIST OF FIGURES ix
LIST OF TABLES xiii
ABSTRACT xiv
ÖZET xvi
1 INTRODUCTION 1
1.1 Eye Tracker Device Literature 3
1.2 Pupil Detection Literature 12
1.2.1 Traditional Methods 13
1.2.2 Machine Learning Models 15
1.3 Objective of the Thesis 18
1.3.1 Hardware Objectives 19
1.3.2 Pupil Detector Objectives 20
1.4 Hypothesis 22
2 METHODOLOGY 23
2.1 Hardware Design 23
2.1.1 3D Solid Design 24
2.1.2 Hardware Components 28
2.1.3 3D Printing Process 32
2.2 Pupil Detection based-on ROM and PETA 35
2.2.1 Traditional Method 36
2.2.2 Learning Model 40
2.2.3 Pupil Ellipse Trend Analysis (PETA) 50
2.3 Calibration Procedure 54
2.4 Software Design & Implementation 57
2.5 Experimental Setup 60
3 RESULTS OF THE PUPIL DETECTOR 64
3.1 Pupil Patch Creation 64
3.2 Learning Model Comparison 68
3.3 ROM & PETA Analysis 72
3.4 Pupil Detection Comparison 77

4 EXPERIMENTS ON THE PRESENT ETS **81**
4.1 Hardware Design....81
4.2 System Spesifications 84
4.3 Examples of Pupil Detection Results....89
5 DISCUSSION **98**
5.1 Discussion on Hardware Prototype....98
5.2 Discussion on Pupil Detection 101
5.3 Discussion on the Present ETS 106
6 CONCLUSION **107**
REFERENCES **110**
PUBLICATIONS FROM THE THESIS **121**

LIST OF SYMBOLS

$E[\varepsilon]$	Expected Error
f	Frequency
Hz	Hertz
ms	Millisecond
nm	Nanometer
ω	Angular velocity
λ	Wavelength

LIST OF ABBREVIATIONS

ASPP	Atrous Spatial Pyramid Pooling
AUTO	Automatic Traditional Configuration Mechanism
CCSDG	Contrastive Single Domain Generalization
CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
CPA	Contained pupil area
D-5px	5-pixel Error Percent for Dikablish
EGE	Efficient Group Enhanced
ETS	Eye tracking system
FE	First and End
FPS	Frame per second
G	Gradient
IMU	Inertial Measurement Unit
IR	Infrared
L-5px	5px Error Percent for LPW
LD	Max of Length and Depth Product
LED	Light Emitting Diode
LM	Learning Model
LPW	Labeled Pupil in the Wild
LPW-AV	Labeled Pupil in the Wild - All Variations
MD	Max Depth
MF	Median First
ML	Max Length
PCC	Post COVID-19 conditions
PD	Pupil detector
PETA	Pupil Ellipse Trend Analysis
PF	Pupil Founder
PFX	Pupil Founder X
PFY	Pupil Founder Y
ReLU	Rectified Linear Unit
RN	ResNet
ROM	Retro-Oriented Mind
SADT	Structural Analysis and Design Technique
SAM	Segment Anything Model
SG	Savitzky-Golay
THM	Thresholding Method
TM	Traditional Method
USB	Universal Serial Bus
VHIT	Video Head Impulse Test
QAT	Quantization Aware Training

LIST OF FIGURES

Figure 1.1 Modern ETS device forms: a remote system (Eyelink) in left [44] and a wearable system (Pupil Labs Core) in right [45] 4

Figure 1.2 General ETS Structure and Relations Between Its Components..... 5

Figure 1.3 Wearable ETS devices-2: (from left to right) Invisible, Dikablisve, EyeSeeCam-Sci, EyeSeeCam-VHIT, ICS Impulse 10

Figure 1.4 Wearable ETS devices-2: (from left to right) ISCAN ETL100H, ETL500A, ETL500B, ETL530A 10

Figure 1.5 Wearable ETS devices-3: (from left to right) Tobii 2, Tobii 3, ETVision and Pupil Labs Neon 10

Figure 2.1 General structure of the proposed hardware components and relations 24

Figure 2.2 Main frame design to be plugged to subjects's glasses 25

Figure 2.3 Eye camera and frame connector ball joint design to be plugged to main frame 25

Figure 2.4 Eye camera holder connector ball joint design to be plugged to frame connector 26

Figure 2.5 Eye camera holder design to be plugged to holder connector 26

Figure 2.6 World camera holder design to be plugged to main frame 27

Figure 2.7 Socket holder design to be plugged to subject's glasses frame and USB hub 27

Figure 2.8 USB hub holder design 27

Figure 2.9 USB hub holder clips to be used for anchorage 28

Figure 2.10 Headset design for users without glasses 28

Figure 2.11 World camera component: C505e HD and its disassembled form.... 29

Figure 2.12 Eye camera component: GC0308 CMOS module..... 30

Figure 2.13 Glasses attachment apparatus from a sunglasses with clips 30

Figure 2.14 USB hub component: CDG-CNV38 Codegen and its disassembled form 31

Figure 2.15 Socket pins as female and male, IR LED, and ball joint pin 31

Figure 2.16 Silicone eyeglass frame anti-slip component 31

Figure 2.17 Printed parts of 3D solid components 32

Figure 2.17 The external headset frame to be used with printed parts for users without glasses 33

Figure 2.18 The general structure of the present detector as the main activity of SADT diagram and the visualized result according to the pupil

patch received from the traditional method and ellipse parameters calculated by the learning model step 35

Figure 2.19 A denoised minima sequences for x-axis and different patches selected by PF functions. Basically, these functions try to select one segment between parts under the threshold value 39

Figure 2.20 An example of the selected patch with randomly defined configurations. 40

Figure 2.21 Learning model to be used for pupil segmentation: UNet architecture [112] 41

Figure 2.22 Learning model to be used for pupil segmentation: UNet++ architecture [113] 42

Figure 2.23 Learning model: SegNet architecture [114] 43

Figure 2.24 Learning model to be used for pupil segmentation: TransUNet architecture [115] 44

Figure 2.25 Learning model: DeepLabv3 architecture [116] 45

Figure 2.26 Learning model: PPMobileSeg architecture [120] 46

Figure 2.27 Learning model: EGE Unet architecture [121] 47

Figure 2.28 Learning model: CCSDG architecture [122] 48

Figure 2.29 General backbone of the present pupil detector pipeline without PETA and automatic traditional configuration mechanism 50

Figure 2.30 The visualization of Entropy and Intensity concepts. Pupil regions in a segmentation map are shown with white color. Pink ellipses represent the fitted parameters using pupil regions 51

Figure 2.31 PETA Upper Bound Threshold Value representation between two consecutive center points for LPW dataset. The value is about 9 pixels, however we defined it as 19 pixels together with 5-pixel errors 53

Figure 2.32 Calibration marker pattern to be shown to user during the calibration procedure: ‘C’ markers for fitting the mapping function and ‘T’ markers for calculation of calibration angular accuracy in accordance with the mapping coefficients, as cocentric cycles without latters 55

Figure 2.33 Screen marker detection flow: largest white area on thresholded world image is assumed as user’s screen and concentric ellipses are found on the screen area 56

Figure 2.34 User interface’s general view. There are totally 6 areas which were shown directly or will be shown via cursor events or movements 58

Figure 2.35 User interface’s plugin menu (bottom) and plugin settings (right) related with hardware operation of an opened plugin 58

Figure 2.36 User interface’s world camera staging in the center region of the main area 59

Figure 2.37 Eye camera staging based on visualizations in the terms of traditional minima curve, pupil patch, fitted ellipse 60

Figure 2.38 Example images from LPW dataset 61

Figure 2.39 Example images from Dikablis dataset 62

Figure 2.40 Example samples from LPW variation: full resolution, ×4 and ×16 respectively. Source: subject=1, video=1, image=1 63

Figure 3.1 Cumulative distribution function for average pupil patch percents of the traditional method proposed in the scope of ROM..... 68

Figure 3.2 Comparison of models trained with LPW-Full and LPW-AV. Accordingly, LPW-AV, containing different resolution of same images, has a data augmentation affect for performance metrics 75

Figure 3.3 Execution times of the pipeline working with two threads for detection and auto configuration processes 77

Figure 3.4 Comparison of our study and selected studies based on full or binary segmentation maps..... 79

Figure 4.1 Eye image examples collected with the desighed IR LED components 82

Figure 4.2 Eye image examples collected with the external IR LED structure 82

Figure 4.3 Final system appearance: view as described in Methodology section (left), view with external IR LED apparatus (right) 83

Figure 4.4 Final system apparance: view with headset frame (left), side and rear view (right and bottom) 84

Figure 4.5 Average running duration of main gaze producing components at 30 Hz frequency..... 87

Figure 4.6 Average running duration of main gaze producing components components at 120 Hz frequency..... 88

Figure 4.6 Pupil detection results in normal usage. The eye is usually looking straight ahead and the pupil is not at the extreme points..... 91

Figure 4.6 Pupil detection results obtained by looking at the edges as much as possible, including top, right, left and bottom 92

Figure 4.7 Pupil detection results during blinking when eye camera is in off-axis position 93

Figure 4.8 Pupil detection results obtained with the eye camera at a random on-axis position..... 94

Figure 4.9 Pupil detection results during blinking with eye camera on-axis position..... 95

Figure 4.10 Pupil detection results for preserving the patch size to be created in a case where the traditional method setting selection will be made well (Median First has been selected for both axes) 96

Figure 4.11 Unsuccessful results obtained in cases where pupil information is insufficient. However, since the pupil estimate is at eyelash level, jumping data problem may be encountered less 97

Figure 4.12 The pupil detection result of an example that is frequently encountered in on-axis usage, where the pupil loses a lot of dark amplitude information due to users' glasseses. The IR source is reflected from the glasses and the recommended black area does not contain the pupil.....97

Figure 4.13 An example of an incorrect pupil detection for size and ellipse angle due to ellipse fitting, where extrapupillary areas are over-detected and intrapupillary areas are under-detected. However, the center point detection may still be sufficient 98

LIST OF TABLES

Table 1.1 ETS device comparison based on their main components and prices. 11

Table 2.1 THM functions to be used for the selection of threshold value. A value calculated by a THM function is used for sequences of x and y pair of an input image. 37

Table 3.1 Analysis of pupil patch creation fulfilled by traditional method in the scope of the present ROM architecture. 64

Table 3.2 Dice metric comparison of selected learning models to be used for pupil segmentation. Training and test have been fulfilled to LPW-3 Hz dataset variation. 69

Table 3.3 GPU inference time and memory requirement comparison of the selected learning models to be used for pupil segmentation. 70

Table 3.4 GPU inference time and memory requirement comparison of the selected learning models to be used for pupil segmentation. 71

Table 3.5 UNet++-S model training with different variations of LPW dataset to display the effect of the input size to the prediction accuracy. 73

Table 3.6 Accuracy metrics for the combination the present pupil detection components. Ellipse error consists of 3 sections: center point error for x and y axes, size error for width and height, ellipse angle error.. 73

Table 3.7 Average GPU prediction times of UNet++-S for input resolutions at different subsampling rates in millisecond. 76

Table 3.8 Average GPU execution times of the study sub-components except for the learning model in millisecond. 76

Table 4.1 Weights of 3D prints and/or hardware component groups 84

Table 4.2 Angular accuracy based on calibration and test procedure in 2D mapping 89

ABSTRACT

FIXEIN: A Mobile and Modular Eye Tracking System

Şerif İNANIR

Department of Computer Engineering Program

Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Ali Can KARACA

Since the early 20th century, eye-trackers have been employed for three purposes: user analysis, object analysis, device control. These systems provide significant benefits, from medical diagnostics to market research, from therapeutic applications to cognitive analyses. Today, wearable trackers are preferred due to their mobility and accuracy. It is crucial for such systems to be equipped with the hardware required by the experimental setup to ensure the collection of reliable data. Additionally, the trade-off between accuracy and frequency in pupil detection (PD), which is fundamental for gaze point production, directly impacts performance. Therefore, selecting a system that aligns with the desired experimental procedure is essential in determining both reliability and cost. Also acquiring different systems for various experimental procedures can present potential challenges in terms of the learning curve and financial burden.

This thesis presents the design of an eye-tracking system named FIXEIN, which includes hardware, software, and a PD module. The originality of the hardware lies in its plug-and-play interchangeable forms and sensors. Additionally, the system can be adapted to the user's own glasses. The novelty of the pupil detection approach is embodied in the Retro-Oriented Mind concept. In this concept, a traditional method initializes a variably sized pupil area, and the learning model operates within this area, which contains less irrelevant information. The second

concept, Pupil Ellipse Trend Analysis, calculates two new metrics to measure accuracy and attempts to correct erroneous ellipses based on metrics. The software performs calibration and generates gaze points, which are presented through the interface.

PD was evaluated using the LPW and Dikablis datasets. Consequently, the detection frequency was increased from 38 to over 150, thanks to the traditional method. Additionally, the learning model, working with less irrelevant information, demonstrated approximately 4% better performance, achieving an overall accuracy of around 95%. As a result, this thesis introduces a wearable, mobile, and modular eye-tracker, capable of operating at 120 Hz with deep learning accuracy, weighing approximately 62 grams, and adaptable to a broader range of experimental procedures due to its interchangeable sensors and form.

Keywords: Eye-tracking system, pupil detection, deep learning, segmentation, 3D printer.

ÖZET

FIXEIN: Mobil ve Modüler Bir Göz Takip Sistemi

Şerif İNANIR

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisan Tezi

Danışman: Doç. Dr. Ali Can KARACA

20. yüzyılın başlarından beridir, göz takip sistemleri kişi analizi, nesne analizi ve cihaz kontrolü olmak üzere üç amaçla kullanılır. Bu sayede tıbbi teşhis çalışmalarından pazar araştırmalarına, terapi uygulamalarından bilişsel analizlere kadar geniş bir spektrumda fayda sağlamaktadır. Günümüzde gözlük formundaki göz takipçiler, sağladıkları mobilite ve doğruluk sayesinde bahsedilen deneylerde tercih edilmektedir. Bu tür sistemlerin deney gereksinimindeki donanıma sahip olması, yeterli ve güvenilir veri toplanması için kritiktir. Ayrıca bakış nokta üretimi için temel işlev olan pupil tespitinin doğruluk ve hız arasındaki ödünleşimi de performansı doğrudan etkilemektedir. Dolayısıyla, istenen deney prosedürüne uygun sistemin seçilmesi güvenirlik ve maliyeti belirler. Farklı deney prosedürleri için farklı sistemlerin edinilebilmesi ise, sistemlerin öğrenilme süresi ve maddi yükü açısından potansiyel sorunlar oluşturabilmektedir.

Bu tezde donanım, yazılım ve pupil tespitçisiyle birlikte FIXEIN isminde bir göz takip sistemi tasarlanmıştır. Donanımın özgünlüğü, form ve sensörlerinin tak-çalıştır olarak değiştirilebilmesidir. Ayrıca sistem, kullanıcıların kendi gözlüklerine adapte edilebilir. Pupil tespit yaklaşımının özgünlüğü, geleneksel yöntemlerle öğrenme modellerinin birleştirildiği Retro-Oriented Mind konseptidir. Bu konseptte, bir geleneksel yöntem değişken boyutlu bir pupil alanı oluşturur. Öğrenme modeli ise görüntünün tamamına göre daha az gereksiz bilgi içeren bir

alanda çalıştırılır. İkinci konsept ise Pupil Elips Trend Analizidir. Trend analizi, hem tespit başarımını ölçen iki yeni metriği hesaplar, hem de metriklere bağlı olarak elips değerlerini düzeltmeye çalışır. Yazılım ise kalibrasyon süreçlerini gerçekleştirir ve bakış noktalarının üretilmesini sağlayarak arayüz üzerinde sunar.

Pupil tespiti LPW ve Dikablis veri setleriyle değerlendirilmiştir. Böylelikle, geleneksel yöntem sayesinde tespit frekansı 38'den 150'nin üzerine çıkarılabilmiştir. Ek olarak, öğrenme modelinin daha az gereksiz bilgiyle çalışması, görüntünün tamamına kıyasla yaklaşık %4 daha başarılıdır. Genel başarım ise %95 seviyelerindedir. Sonuç olarak, bu tez kapsamında, derin öğrenme model doğruluğunda 120 Hz'de çalışabilecek, yaklaşık 62-gram ağırlığında, sensör ve formu ihtiyaca göre değiştirilebildiği için daha fazla deney prosedürüne uygunluk gösterebilen giyilebilir, mobil ve modüler bir göz takip sistemi ortaya konmuştur.

Anahtar Kelimeler: Göz takip sistemi, pupil tespiti, derin öğrenme, segmentasyon, 3B yazıcı.

1
INTRODUCTION

Eye tracking systems (ETS) serve two primary purposes: (*i*) detecting the subject's gaze in the external environment, and (*ii*) understanding the effects of the area the subject observes on oneself [1,2]. Both functions render eye tracking applicable across various domains, which can be categorized into three main areas: subject analysis, object analysis, and device control.

Subject analysis provides insights into the biological, psychological, social, and educational conditions of participants [3]. This category, utilizing diverse and domain-specific experimental designs, allows for the observation of a subject's competence or their method of task execution. For example, applications across various fields include:

- *Medicine*: Diseases and disorders that manifest behavioral issues in individuals can be diagnosed using ETS [4]. These conditions, stemming from the parasympathetic nervous system, vestibular system, or basal ganglia, can affect spatial orientation and may also lead to psychological problems. Examples include schizophrenia, bipolar disorder, Alzheimer's, Parkinson's, autism, ADHD, epilepsy, traumatic brain injuries, vertigo, multiple sclerosis, and dementia [5-14]. Furthermore, with advancements in machine learning and related fields, classification solutions based on eye movements have reached accuracies of approximately 95%, making them suitable for medical use. These solutions can be applied, and results obtained typically within 5-30 minutes, potentially reducing the need for other expensive diagnostic tools, and decreasing the healthcare burden [15-18].
- *Cognitive and Behavioral Sciences*: ETS are employed to establish relationships between individuals' reasons for actions and their gaze direction [18]. For instance, analyzing the reactions of a surgical doctor during surgery can reveal which external signals may require similar responses.
- *Education*: ETS can detect the disparity between what individuals have learned and their performance of a task [19-20]. For example, assessing how

frequently a student looks at the correct locations during a procedure can objectively determine their qualification for a surgical title.

- *Sociology*: ETS are useful in understanding the norms and behavioral patterns adopted by different sample groups during socialization, categorization, or group formation processes [21-22].

Note that ETS-based subject analysis is not limited to humans but is also conducted on animals like mice and monkeys for medical measurements or behavioral studies [23-25]. These animal studies hold significant potential for understanding natural phenomena and events.

Object analysis enables the evaluation of physical or logical systems that a subject interacts with. This evaluation can be examined in three categories: *i*) functionality (the degree to which a function fulfills its designed need), *ii*) performance (the cost associated with executing the function), and *iii*) satisfaction (the emotional output resulting from system use) [26]. Physical suitability specifically targets the exploration of design obstacles encountered during the execution of a task [27]. For example, an investigation into why the visibility of discounted products on an e-commerce website is low could be mentioned. Logical suitability involves assessing the positive and negative aspects of a system's function relative to its target audience [28,29].

Device control through ETS enables users to control a device using eye movements. This functionality is generally associated with two application areas. The first targets the development of accessibility and therapeutic tools. An example is patients with inadequate hand-arm coordination using their eye movements to control a robot or system, thereby facilitating their employment [30]. This example encompasses both accessibility and therapeutic aspects. There are also instances where these interests diverge, focusing solely on accessibility or therapy. The second application is the automatic execution of predefined actions in conjunction with specific eye movement patterns. This can reduce the time required to decide on an action and execute it [31]. For example, controlling flying objects via eye movements has proven more successful in completing courses than manual control [32]. Another example is enhancing gaming or film experiences by making the object under gaze sharp on the screen while blurring the background, and darkening the surroundings when the user looks at a light source [33].

Except for applications handled in relatively up to date reviews [17,34-38, 156], numerous studies have been presented in recent years. For instance, Wang et al. trained a gaze-guided attention network with X-Ray images that both reduces the dataset preparation time and enhances classification accuracy with specific configurations [39]. In addition, Sun et al. achieved high accuracy in identifying perpetrators and distinguishing innocents in eye movement analysis [40]. Issever et al. measured the cognitive load of computer programmers according to their demographic and professional features when solving some object-oriented code tasks [41]. Alternatively, Vinuela-Navarro et al. investigated the effects of post-COVID-19 conditions (PCC) with saccades, fixation, and pupil responses. The participants with PCC prone to longer latencies in some saccadic paradigms, weaker fixation stability and closer eye positions for vergence [42]. In another application, Xu et al. designed a wheelchair controlled by eye movements and based on a learning model for individuals with ALS disease [43]. Despite limitations, these applications offer promising models for the future, where eye movements could serve as auxiliary or co-control mechanisms rather than main control interfaces.

1.1 Eye Tracker Device Literature

In contemporary applications, modern ETS devices are typically available in two forms, namely, remote and wearable (also known as mobile, head-mounted, or headset) [2]. Remote ETS typically features a type of camera positioned in front of the user. This system extracts eye features from the collected images and calculates gaze points on the screen. On the other hand, wearable ETS are in the form of wearable glasses and are equipped with two types of cameras: eye and world camera. The world camera captures images of the direction the user is looking towards, while the eye camera collects images of the user's eyes. These two sets of images are used to calculate the direction of the user’s gaze on world camera images. Figure 1 illustrates examples of both remote (left) and wearable (right) ETS configurations. Although both forms have their distinct application areas, wearable alternatives have recently gained a more significant position [46]. There are two main reasons for this: *i*) wearable devices do not restrict user movements, allowing for their use not only in fixed positions but also while in motion, and *ii*) wearable

devices can be utilized in many experiments where remote systems are used. Therefore, various wearable systems have been developed both academically and industrially.



Figure 1.1 Modern ETS device forms: a remote system (Eyelink) in left [44] and a wearable system (Pupil Labs Core) in right [45]

Figure 2 illustrates the general factors that can be considered when selecting an ETS. In the figure, the arrows signify relation based on the direction of the arrow, indicating "determined by" or "varies according to", and aim to show the primary components that influence the emergence of a particular component. It is crucial to consider these components and their interrelationships, and to choose a system that is suitable for the intended task. Incorrect data correction through various filters can potentially damage otherwise acceptable data and introduce ambiguities into experimental results. On the other hand, understanding the operational dynamics of the systems can provide a good impression of under what circumstances the performance claims made by system manufacturers can be achieved.

Sensors **(***i.e.* accelerometers and gyroscopes**)**, except for the cameras, represent additional measurement instruments in an ETS. Typically, these sensors can calculate head maneuvers or posture. For instance, in a Video Head Impulse Test (VHIT) study for diagnosing of Vertigo disorder, an accelerometer is utilized to obtain the vestibulo-ocular reflex [137,138]. Another impact of sensors is that they have the potential to enhance the performance of feature detection algorithms, such as fixation detection [139].

Pupil detector is central to an ETS due to its key role in producing gaze points. As recent reviews have noted, it may employ either traditional methods or learning models. Understanding the approach used for the pupil detector can help predict

conditions under which the system might produce erroneous outputs. The design of a pupil detector is tailored according to the desired performance level, lighting conditions around the eye area, and the specifications of the eye camera. Structurally, a pupil detector may either identify the ellipse parameters or locate just the pupil center.

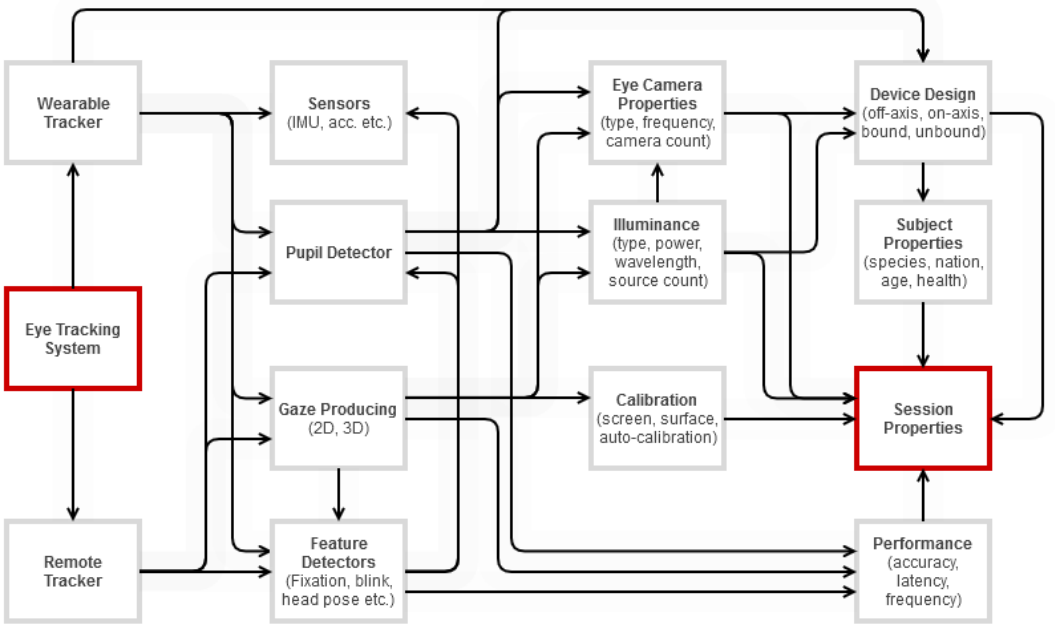


Figure 1.2 General ETS Structure and Relations Between Its Components

Gaze producing is the function where the gaze point is calculated. It involves applying a mapping function established between the eye features, including the pupil center, and the world camera. During its design, desired performance metrics, feature detectors (such as eyeball, eye corners, or glint), calibration method, and lighting conditions of both the eye and the environment, as well as eye camera specifications, should be considered. Generally, gaze production alternatives include 2D-to-2D, 3D-to-2D, and 3D-to-3D mappings [144]. For handling eye information in 3D, a reference point is required, which can be established through detection of the eyeball or glint. Typically, calculating gaze information in 3D can lead to more robust calibration against 2D but might introduce slightly greater angular errors. To reduce this error amount, necessary measures should be taken for the accurate detection of other eye components used in conjunction with the pupil to create a 3D eye model. For instance, rotating the eyeball during detection is one measure that can improve accuracy, although such processes are dependent on the algorithm and can vary from system to system.

Feature detectors, other than the pupil, detect additional eye-related characteristics. These include blink, eye opening amount, eyeball, iris detection, saccade, eye corners, glint, fixation detection, eye angular velocity, head pose for both system forms or simply the eye and head regions for only the remote form. Each feature operates under additional requirements such as environmental conditions, system resources, and sensors [145-149]. During their design, the expected performance can reference the pupil detector and the sensors included in the system. For example, blink detection might depend on the success of pupil detection or be addressed as a separate method [155]. Fixation can be more accurately calculated with head pose; otherwise, solely based on the pupil center, a fixation might be disrupted if the eye angle changes without a change in the viewed object. Moreover, the accuracy of feature detection used in gaze producing is crucial, as these characteristics, along with the pupil information, play a significant role in determining angular error in gaze production.

Eye camera properties are central to the operation of ETS devices, as they support the effective functioning of the pupil detector. Consequently, some structural features are often concealed in commercial systems. Eye cameras come in three types depending on the wavelength they detect: visible (<750 nm λ), infrared (>750 nm λ), and visible+infrared (>400 nm λ and <1050 nm λ). The distinguishing feature between these types relates to the presence or structure of the cut-off filter between the sensor and the lens. Visible types may show significant reflections on the pupil because visible light is more prevalent in daily environments and can reflect off the pupil area. Similar reflections are present in combine visible and infrared types but are somewhat more balanced. Although reflections on the pupil are generally undesirable as they can obscure pupil information, they may be sought after in auto-calibration applications [143,154]. Completely infrared systems typically show less pupil reflection than other types. The frequency of the camera denotes its operational speed. Although it is suggested that cameras in real-time systems should operate at 120 Hz, there must also be compatibility between the camera frequency and the speed of pupil detection. The number of cameras is another factor that influences the pupil detection approach. Some systems use multiple cameras and base their pupil detection method on this criterion, for example, using multiple cameras to build an eyeball model or in methods that utilize

both corneal reflection and black pupil information. Another critical aspect of cameras is their motion capability. It is essential for the eye camera to move as little as possible involuntarily; otherwise, the system may need recalibration.

Illuminance can be applied to both the area of interest and the eye region, although for screen calibration, usually only the eye region needs to be illuminated. In auto-calibration, it might be crucial for the surface to have enough brightness to reflect from the pupil. Depending on the type, visible cameras should be well-lit but not to the extent that it discomforts the user. Infrared and visible+infrared systems generally prefer infrared LEDs between 850-950 nanometers. The choice of infrared LEDs is because this wavelength is invisible to humans, allowing for effective illumination of the eye region. The amount of infrared lighting directly influences the darkness of the pupil while increasing the brightness of other eye areas, enhancing the distinctiveness of the pupil. In most visible+infrared cameras, a single infrared LED might suffice, but this depends on the ambient light conditions. Otherwise, insufficient amplitude information may lead to detection errors. Similarly, the angle between the camera and lighting must also be designed to suit the pupil detection method. If both components are aligned parallel, a corneal reflection is obtained, which is undesirable for approaches that use dark amplitude information [140]. For approaches that continuously require this information, the device design might preferably be on-axis. Correct acquisition of corneal reflection information in non-on-axis wearable devices might be possible with the use of multiple infrared LEDs [142].

Calibration involves estimating the coefficients of the mathematical model to be used for gaze point production. It is fundamentally categorized into screen and surface types. In screen calibration, the user is required to look at or follow markers on the screen for a certain period. Simply put, each pupil center coordinate is matched with the expected coordinate point of the marker being viewed. In wearable systems, marker detection must additionally be conducted. Similarly, in surface calibration, marker detection is performed, but the markers are not on the main screen but rather in a physical form or on an external screen. In both cases, the correct estimation of the coefficients depends on obtaining different gaze vector angles during calibration [141]. Accordingly, various calibration patterns such as 5-point, 9-point, or spiral are applied during calibration to diversify gaze vector

angles. In auto-calibration, this step might not be performed, or it may continuously occur in the background during system use alongside various factors [154]. Additionally, screen or surface-based gaze producing is prone to parallax error. On the other hand, auto-calibration or calibration-free approaches are less affected by parallax but have higher angular error or require stricter environmental conditions for accurate operation.

Device design encompasses several challenges in wearable systems. Issues that may arise include the system's weight, involuntary movement of sensors and cameras, cameras of visible size impairing aesthetic perception and altering user behavior, the necessity of keeping the eyes more open than usual or restricting the gaze vector, requirements for individuals with eye disorders to wear lenses, and camera positioning. Due to the mobility of the human upper eyelid and the desire to collect as much pupil information as possible even during blinking, cameras are typically positioned below the eye, a design known as off-axis. However, due to racial differences or eye disorders, some individuals may have protruding eyes, which can result in insufficient pupil information from upward glances in off-axis designs. Even in more normal eye structures, problems can be observed in upward gaze vectors in off-axis designs. Thus, if the viewed surface in off-axis designs extends downwards starting approximately at eye level, better pupil data can be collected. Using multiple eye cameras or positioning the camera to capture images directly in front of the eye, known as on-axis design, can allow for more freedom in user gazes and minimize problems encountered in upward gazes [142]. A design distinction also exists between bound and unbound systems. In bound designs, the glasses are fixed to the head to prevent movement and disruption of system calibration. In unbound designs, the system resembles normal glasses, and anti-slip components, typically made of silicone, are used only where the frame contacts the skin to prevent slippage. Additionally, an expectation from unbound designs is the concealment of system components to allow natural user behavior. Although in many systems cameras are visibly present, various special designs have been developed in recent years to address this. However, concealing the design generally requires a significant manufacturing process and increases costs. The weight of the device is another concern. Experiments by researchers have shown that a system weighing in such as Pupil Labs Core can cause pain in areas in contact with the

glasses (the nose and tops of the ears) after approximately 3-4 hours of continuous use. Therefore, when working with wearable systems, the session duration might be meaningfully influenced by the weight of the device. Lastly, users with vision impairments typically wear optical lenses in wearable systems. The use of optical lenses can create various comfort issues for those who have not previously used them and may impair the naturalness of subject behavior. Alternatively, if the ETS system has its own lenses, they should be modified to accommodate the user's vision impairment. Systems with lenses might have different prescription lenses available for specific vision impairments or might require fitting the user's own prescription lenses into the system. Additionally, in lensed systems, the lens might be situated between the eye and the camera, which can lead to severe reflections from the lens interfering with pupil data collection or distortions caused by prescription lenses, potentially limiting the success of the gaze-producing model.

Subject properties encompass the characteristics of participants involved in an experiment. Components such as age, nationality, health, and if not human, the species of the animal, have the potential to influence the design of the device used [150]. As mentioned above, reasons such as race or eye structure may necessitate the use of an on-axis configuration to gather adequate pupil information in some cases. Similarly, the system design can be modified for animals like mice and monkeys [151,152].

System performance generally encompasses aspects such as accuracy, frequency, and latency. Accuracy may refer to the individual performance of system components or a general performance measure like angular error. Frequency, similarly, denotes the amount of data generated per second by a component or the general system, and is also related to cost. In some systems, more advanced hardware may be required to achieve the desired computational speed. Latency refers to the time elapsed from the collection of inputs from cameras and sensors to their output as data. Latency is a critical performance indicator, especially in real-time controller applications [153].

Session properties ultimately represent the entirety of the experimental procedures and requirements intended for the use of the ETS. The performance and subject selection should be appropriately aligned with the characteristic aspects of the task at hand, followed by checks for the suitability of the ETS components to the

relevant requirements. Therefore, an ETS must be selected based on its capabilities and the intended experimental procedures, because the constraints of the system and the preparations required for it to function as expected are important to be appropriate for the experiment, depending on the selected ETS [153].

There are numerous ETS developed for both academic and industrial purposes, each featuring distinct characteristics. Examples of wearable systems include: Pupil Labs: Core (Figure 1.1, right) [45,47], Pupil Labs: Invisible (Figure 1.2)[45,48], Ergoneers: Dikablis (Figure 1.3)[46,49], ISCAN: ETL100H, ETL500A, ETL500B, ETL530A (Figure 1.5)[46,50], Argus Science: ETVision (Figure 1.5)[51], Tobii: Glasses 2, Glasses 3 (Figure 1.5)[46,52], Pupil Labs: Neon (Figure 1.5)[57], ASL: H6 Head Mounted [53], EyeSeeTec: EyeSeeCam-Sci, EyeSeeCam-VHIT (Figure 1.3)[54,55], Natus, ICS Impulse (Figure 1.3)[54,56].



Figure 1.3 Wearable ETS devices-2: (from left to right) Invisible, Dikablisve, EyeSeeCam-Sci, EyeSeeCam-VHIT, ICS Impulse



Figure 1.4 Wearable ETS devices-2: (from left to right) ISCAN ETL100H, ETL500A, ETL500B, ETL530A



Figure 1.5 Wearable ETS devices-3: (from left to right) Tobii 2, Tobii 3, ETVision and Pupil Labs Neon

Table 1.1 ETS device comparison based on their main components and prices

	Pupil Core	**Pupil Neon**	**Tobii 3**
Accuracy	0.6-2.5 deg	1.3-1.8 deg	0.6
Frequency	30-200 Hz	30-200 Hz	50-100 Hz
Pupil Detector	Amplitude and Edge	NeonNet	Amplitude
Illuminance	Infrared x1	Infrared x1	Infrared x8
Weight	23 g	45 g	77 g (with cable)
IMU sensor	No	Yes	Yes
Eye Camera Type	Off-axis	On-axis	On-axis
Frame Type	Unbound	Changeable	Unbound
Visual Impairment	Only contact lens	Corrective lenses for a special frame	Corrective lenses for the frame
Calibration	Screen or surface	Free	Screen
Price	3700 $	6400 $ (+2150 $ for lens kit and lens frame)	Only via contact
Included	Software	Software, mobile phone	Software

As is seen from the table, these systems exhibit variations in their applications due to their distinct characteristics. Table 1.1 presents the features of three selected devices as examples. While Pupil Core and Tobii 3 are well-known in this field, Pupil Neon, despite being newer, holds innovative potential. The differences between these systems lie in both hardware and software disparities, as well as in which scenarios their performance metrics can be achieved. For instance, the

systems' ability to operate within expected angular accuracy values depends particularly on the pupil detection approach functioning as anticipated. This might also be significantly influenced by the subject characteristics or environmental factors. Frequency, defined as the maximum number of gaze points that the system can generate, correlates with the frames per second of the eye camera; however, this is related both to the hardware of the system on which the ETS software operates and the computational load of the software. For example, real-time operation of systems supported by deep learning may require more costly computational units. On the other hand, while the sensor types in these systems are fixed and non-interchangeable, Pupil Neon features an integrated camera and sensors setup, allowing for frame modifications suitable for specific tasks. However, the frames in Pupil Core and Tobii 3 are fixed. Considering the facilities provided for participants with vision impairments, users of Pupil Core generally must wear contact lenses, as this and other similar systems cannot accommodate users' own glasses. Yet, Pupil Neon and Tobii 3 offer corrective lenses that can be additionally purchased and replaced with the device's own lenses according to the specific vision impairment. Calibration-wise, Pupil Neon can be calibrated automatically, a relatively easier process in a system where the cameras are integrated and their angles relative to each other are known. In terms of cost, although the price of Tobii 3 can only be obtained via contact, it is generally more expensive than the other two systems.

1.2 Pupil Detection Literature

The pupil, a fundamental feature in the field of eye tracking, forms the cornerstone for designing studies in this domain. This process serves as the foundation for critical eye tracking functions such as gaze point producing, fixation detection, blink detection, and angular velocity calculating, it deeply effects the system performance and accuracy [58]. Also, our experience suggests that smoothing-like filters may not be beneficial for correction of inaccurate pupil data due to effects on accurate data – especially true saccades. Therefore, the selection of a pupil detector (PD) operating within a desired or acceptable error rate is crucial [59-60]. Moreover, system performance is another critical factor to bear in mind. Certain fields, such as medical diagnosis and real-time device control, necessitate high-

speed tracking to fulfill their objectives. In contrast to this, others must prioritize low resource consumption on mobile applications with limited resources. In such scenarios, the choice of a PD should be guided by factors like accuracy rate, computation time, and latency duration required for the task at hand.

Researchers have explored various methods for PD [61-62], classified into two categories: *i*) traditional methods, and *ii*) machine learning models.

1.2.1 Traditional Methods

Traditional methods can also be divided into three categories: amplitude-based, edge-based, and hybrid amplitude-edge methods. Amplitude-based traditional approaches encompass various techniques. For example, Morimoto et al. detected the pupil by subtracting bring and dark areas from each other via corneal reflections [63]. Navaneethan and Nandhagopal applied a morphology closing operations on binarized images [64]. Glabbur et al. utilized a region-coloring approach based on connected components, merging similar regions to identify the pupil [65]. Abbasi and Khosravi tracked the pupil with a genetic algorithm after simple thresholding operation [66]. Bonteanu et al. designed a binarization process based on the first negative slope of cumulative distribution function of grayscale images [67]. Afterwards, they used Convex Hull operation to derive pupil parameters. In addition, Wan et al. proposed a horizontal weighted Haar-like feature less effected by blinking, eyelashes, and eyelids [68]. Timm et al. devised a gradient based algorithm [69], described and evaluated in Ref. [70]. In this study, the image resolution is reduced, and a coarse pupil position is attempted to be determined. Subsequently, the approximate pupil area on the original resolution is extracted, and fine detection is performed. Manuri et al. improved the Starburst algorithm by applying some pre-processing operations and the ray tracking method for fine-tuning of pupil segmentation [71]. Jamaludin et al. employed a Wiener filter to mitigate motion blur and performed amplitude-based methods for eyelid, eyelash, and reflection removal, followed by a morphological closing operation for pupil detection [72]. Among these methods, Wan et al.'s study stands out as one of the most successful in the amplitude-based traditional category [68].

In edge-based approaches, researchers have explored diverse techniques for pupil detection (PD). For instance, ElSe focuses to select pupil curves among edges

detected by Canny detector [73]. In another research, PuRe, combines curves among edges detected by Canny detector after some morphological operations [74]. Additionally, PuRe calculates an accuracy metrics according to integrity of generated ellipse. PuReST is an optimization method designed that defines a ROI to be used the next detection cycle of the PuRe [75]. Furthermore, Susitha utilizes Sobel based method removing eyelids [76]. Indeed, edges are scored based on their connections with other edges, with the pupil selected based on the highest score. Li et al. selected possible pupil ellipse among curves found by Canny detector [77]. This selection requires eyeball information to select suitable pupil curves for their geometrical structure according to center point of the eyeball. Among these edge-based studies, Li et al.'s work is highlighted as one of the most successful in the traditional category [77].

In hybrid traditional approaches, researchers have explored methods that combine amplitude-based and edge-based techniques to detect the pupil. For instance, ExCuSe selects curves found by Canny detector in most darker areas and fit a suitable ellipse [78]. Omran and Alshemmary use gamma correction, smoothing operations, and binary thresholding, followed by the Hough transform to detect pupil and iris areas [79]. Bonteanu et al. converted the image to its binary version, and they fitted ellipses using Convex Hull and evaluated them according to integrity and ellipticity of each ellipse [80]. If the evaluation outcome is insufficient, the same image is subjected to binarization with a lower threshold value. Kassner et al. found the curves with Canny detector on the region with low amplitudes [81]. Lastly, they defined the ellipse parameter with or without a combination of preellipse parameters fitted to these curves. Notably, one of the most successful studies in the hybrid traditional method category was by Bonteanu et al [80].

In brief, amplitude-based methods assume that the pupil is the darkest area in the image, edge-based methods select the most elliptical region as the pupil, and hybrid methods combine both amplitude and edge information. The point to note here is that the reasons of the inaccurate results produced by methods using edge information cannot be foreseen. Underlying this lies the ability of things that can move over time, such as shadows, eyelids, eyelashes, and reflections, to instantaneously form elliptical shapes. Also, considering the possibility of the eye camera moving, these things don't even need to be moving. Amplitude information,

on the other hand, offers precision by eliminating areas darker than the pupil before the experiment or configuring the method based on a relevant subject to obtain accurate pupil information.

1.2.2 Machine Learning Models

Learning models offer a versatile approach in PD, extending beyond merely finding the pupil to encompass the detection of various eye features, including the eyeball, iris, gaze vector, eye corner points, sclera, and eyelid. While learning models can be trained to perform the specific task of PD by directly calculating the center point coordinates or segmenting the pupil region, the latter is more common. Because the size of the pupil ellipse is a crucial factor in certain subject analysis studies [82-83]. Additionally, since a learning model may utilize both amplitude and edge information, precautions may not be taken in the initialization step of the approach to avoid inaccurate results. Indeed, there's a trade-off between traditional methods and learning models, balancing accuracy against computation time. While traditional methods run real-time with some localization lacks, learning models predict accurate with longer execution times.

One of the most popular learning models, Fuhl et al. designed the Pistol model between ResNet-18 and ResNet-34 to detect pupil, iris, eyelids, and other eye features such as eye opening, gaze vector and eyeball [84]. Chinsatit and Saitoh classified images as open, near open and closed with AlexNet, then detected pupil center points with a ConvNet trained separately for each class [85]. Lee et al. suggested a fix-sized patch with 9 cells pointing the pupil area for remote trackers [86]. The median cell should have lowest amplitude averagely in this design to be able to contain the pupil segmented with ResNet. Chen et al. trained a model named as PCR-Net to detect pupil center and radius via 7 points placed between two eye corners [87]. Ates et al. designed a Res-CNN using quantization aware training (QAT) [159]. This approach allowed for a faster architecture by employing lower floating-point bit values and reducing the input image resolution to 40x30. The study also demonstrated that the QAT-enabled model did not adversely affect the loss function at lower resolutions. Gou et al. developed an encoder-decoder network named Multiscale Attention Link for remote trackers to obtain pupil center points [88]. Alternatively, Shi et al. proposed the LVCF model containing V-Net for segmentation and LSTM for tracking [89]. Wang et al. predicted pupil center point

and pupil radius instead of ellipse parameters, using Res-Net and Vision Transformers [90]. Akinlar et al. used EFE-UNet, a UNet with a novel loss function, enforcing more elliptical results [91]. Many studies also use UNet for seg-mentation with different hyperparameter optimizations and dataset preferences [59, 92-95].

However, these models often require many accurately labeled pupil images. This situation can pose challenges in terms of both training durations and labeling costs. Considering the dataset quality is not sufficient, a suitable model cannot be obtained. Although transfer learning is employed to overcome this problem, this approach may not entirely solve the relevant issue. In response to this negativity, Guo et al. proposed a segmentation framework using Swin Transformers [96]. Here, images were randomly masked with patches to achieve higher success with a smaller training set. Afterwards, segmentation was made with a network that combines Swin-Transformer blocks and UNet. Maquiling et al. generated synthetic images based on light and reflection intensities using a Gaussian distribution, testing them with VR-based real images [97-98]. They also leveraged the Segment Anything Model (SAM) for notable results with VR images, showcasing SAM's potential in eye region annotation and segmentation [99].

Some other studies applied various processing methods directly to segmentation maps of a learning model. These methods were generally like operations applied in traditional methods, presenting a rather unconventional way of enhancing model outputs using traditional techniques. As an example, study within this scope, Kim and Lee selected the most elliptical area segmented by DeepLab v3+ as the pupil and used a type of interpolation for ellipse parameters with low segmentation map in-formation using next and previous parameters [100]. Aarti and Kumar proposed to use morphological processing to the segmentation map of UNet to perform a fine-tuning [95]. Moreover, researchers aimed for faster performance by adopting regression-based methods instead of larger deep learning models. For example, Gou et al. used shape augmented cascade regression model initialized with synthetic eye images to find pupil center point [101]. Xiang et al. suggested a classification and regression-based model to calculate pupil center point for scale mapped images addressing different resolution concerns [102].

Another method category used to increase detection speed in pupil detection involves tracking the pupil. For example, EV-Eye segments the pupil with a UNet

and simultaneously tracks the segmentation result to enhance gaze producing speed [158]. The tracking mechanism, referred to as point-to-edge matching, examines the amplitude values of the pupil edges in the subsequent frame. This approach achieved a speed of 38.4kHz in the most successful eye behavior, saccade tracking. While this method produced successful outcomes, it was evaluated by its authors on a small and controlled test set. Consequently, a limitation of the study is the need for testing in more uncontrolled or wild environments, as gaze tracking may exhibit a higher error rate in real-world settings, as described in the study. Additionally, achieving such high speeds required using an event camera and an IR camera-equipped eye tracker. These constraints may impose various usage limitations on the system.

An innovative approach in PD diverges from existing studies, focusing on cropping-based models. Some researchers have employed cropping to enhance the accuracy of models, while others have utilized it for faster inference times. For the accuracy of the model, PupilNet, tries to select 24×24 patches containing pupil area on 16 times down-scaled images with a CNN model, and detect pupil center point with the second CNN [103]. PONet is a classifier CNN designed to localize the pupil area [157]. There are 15 predefined pupil position classes in the image. During training, the image is first binarized using a fixed threshold value. Subsequently, the binarized image, which contains regions with blackness that may include the pupil, is fed into the CNN architecture. The network then classifies the pupil into one of the 15 position classes. However, while this classification method demonstrates suitability for simple applications on mobile devices, it has limited accuracy due to its classification approach. Alternatively, Antonioli et al. trained two UNets (just for pupil) to find a fix-sized pupil patch that is 2.67 times smaller than the original image, and to segment the pupil area on the selected patch [104]. For inference time, Vera-Olmos and Malpica presented two encoder-decoder networks to select fix-sized patch that is 17 times lower than the original image, and segment it [105]. Byrne et al. combined a cropping approach with pretrained UNet based on ResNet34 encoder for synthetic images, presenting Leyes method [106]. The proposed fixed-size cropping method can be performed in two ways: the first is with respect to the image center, and the second is based on the result from the PuRe method. According to their algorithms, if the accuracy value of PuRe is

sufficient, cropping is applied, and the cropped area is given to the model. Then if the model output indicates that the pupil is close to the edges, the model is repredicted based on a crop of the same size around the relevant region. On the other hand, in the case of that PuRe accuracy is not sufficient, cropping is done centrally. Additionally, their synthetic image generation approach operates based on a Gaussian distribution, creating content based on light and reflection intensities.

According to the comparisons they have shared, studies like LEyes and Pistol present remarkable approaches thanks to their performances.

In the literature described above, the most recent studies, published at the end of 2023 and in the first half of 2024, include [72], [84], [87], [88], [97], [99], [106], [157], [158], [159]. These studies reveal that current pupil tracking methods focus on three main directions. The first direction aims to reduce resource requirements and increase system frequency by attempting to minimize input dimensions in various ways. The second direction measures performance in pupil detection by employing different models, model components, and training methods (e.g., synthetic images) developed in the field of image processing. The third direction primarily focuses on increasing the frequency in tracking methods. Additionally, the differences between the most recent studies in the literature and the method proposed in this study will be addressed in the discussion section.

On the other hand, it is important to note that we have refrained from disclosing the accuracy metrics attained in studies addressed in the literature. This is because some built-in datasets yield impressive accuracy results even when subjected to a simple traditional method [92]. Additionally, general datasets analyzed in these studies may contain considerably erroneous annotations [91, 107]. Consequently, there's a need for a comprehensive benchmark in PD studies, possibly using a dataset designed and validated by researchers and professionals from diverse domains such as engineering and medicine.

1.3 Objective of the Thesis

This study focuses on the development of a new wearable and modular ETS. In this context, both hardware and software prototyping have been designed and developed. The software contains some fundamental components essential for the

operation of an ETS, with the pupil detection approach being the most critical addition. Therefore, the developed hardware and pupil detection approach are the primary originalities highlighted in this thesis. The contributions of the study in terms of hardware and pupil detection are detailed below.

1.3.1 Hardware Objectives

There are three hardware distinctions that differentiate existing ETSs: the FPS of the eye camera, the frame structure (standard, head-mounted, etc.), and the presence of a desired sensor in the device. For instance, a ETS with a standard eyeglass frame may be sufficient for device control and usability trials, while medical diagnostics (e.g., vertigo, bipolar disorders) typically use 200 FPS, certain motion sensors, and a head-mounted frame. Therefore, due to their physical properties, a ETS can operate efficiently or inefficiently in different application areas. For example, a device designed for object analysis cannot be used in medical fields, as such ETS might experience slippage effects and low reliability due to FPS. Conversely, a device advanced enough for medical use would not be suitable for prolonged uses such as object analysis and device control, as the product cost is high, and additional measures to prevent involuntary camera movements can restrict user behavior and cause discomfort. Therefore, it is crucial to determine the intended use of the device before purchasing an ETS. If the ETS is to be used for multiple different purposes, multiple ETS products may need to be purchased, which leads to the acquisition, learning, and financial costs of multiple ETS. In this thesis, we are redesigning the ETS hardware in a dynamic and modular structure to make it suitable for a broader range of uses. Consequently, the sensors on the system (e.g., cameras, IMU sensor, etc.) and the frame (a standard eyeglass frame or a head-mounted apparatus) are intended to be interchangeable on a plug-and-play basis. Users can directly attach a camera with different FPS to the system according to their needs and can change the frame. Additionally, the developed system will have unused (idle) connection ports. These ports will allow the direct adaptation of different sensors required by the experimental procedure to be implemented. In other words, for instance, a motion sensor can be attached to the system and used only when needed. The modular design of the ETS hardware aims to create a device that can be efficiently used in different types of experiments, thereby reducing the time and cost incurred from the need to use different ETSs. Furthermore, in the preparation phase before

experiments prepared with existing wearable ETS, contact lenses should be supplied for individuals who wear glasses, or the lenses in the system should be changed according to the person's vision impairment. This situation can create a certain time and cost expenditure and can cause discomfort in individuals not accustomed to contact lenses, leading to unusual eye movements. To prevent this, the modularity of the hardware will be utilized. Thus, it is intended that the backbone mechanism of the system (the connection box containing the camera, sensor, transmission cable, etc.) can be adapted to users' glasses. The contributions of the proposed hardware design are provided below:

- Cameras and sensors can be interchangeable or new sensors can be added to the system on a plug-and-play basis.
- The system provide users' to use their own eyeglasses.
- The frame can be configured as either bound or in a standard eyeglass form (unbound) for different experiments.
- Eye cameras can be utilized in both on-axis and off-axis configurations.

1.3.2 Pupil Detector Objectives

LEyes stands out as the pioneering study combining a learning model with a traditional method [106]. It employs PuRe as an edge-based traditional method and UNet as the learning model, although its strategy isn't solely patch-based. Generally, this combination was specially used to expedite the training of synthetic images with diverse sensor specifications, eye appearances, and environmental conditions. However, the study doesn't thoroughly investigate the impact of the traditional method on the learning model. Also, the images were processed with the central cropping or an edge-based method which is relatively difficult to predict the potential for erroneous output. Eventually, fixed-size cropping may result in the obtained image not consisting solely of the pupil, but rather containing various other objects. Therefore, fixed-size cropping may not be the most effective way to expedite the inference process.

The purposed PD in this thesis focuses on developing a real-time and accurate pupil ellipse detection pipeline for wearable eye tracking systems, addressing challenges and limitations observed in existing approaches. To achieve this goal, a hybrid approach named Retro-Oriented Mind (ROM) combines a novel method in

traditional category and learning model. The traditional method aims to find a dynamically determined rectangle patch potentially containing the pupil, while the learning model segments the pupil region within the patch. Thus, by reducing input size of the learning model through a traditional method, ROM benefit from both real-time and accurate property as in a traditional method and a learning model, respectively. Also, ROM prioritizes an amplitude-based traditional method, allowing for error predetection and adjustment of environmental conditions. Essentially, ROM minimizes the trade-off between frequency and accuracy by combining traditional and state-of-the-art approaches. In other words, the first mechanism of ROM truncates or prunes the problem domain and purify it from unnecessary information as much as possible, and the second mechanism fine-tunes the process to achieve precise localization.

Another objective of the study is to introduce a new concept named as Pupil Ellipse Trend Analysis (PETA). PETA is the way to measure detection accuracy for outputs of the learning model and to correct pupil ellipse parameters using previously parameters if detection accuracy metrics are insufficient. By using PETA as a post-processing step, it is desired to prevent especially inaccurate jumping data which may be observed because of a high relative angular velocity, deficient pupil information, unwanted shakes of the eye camera, and wrong detection maps.

As the contributions of the purposed PD, we present:

- We introduce a new concept ROM that combines traditional methods and learning models based on PD domain knowledge and causality. Also, unlike many existing studies, we don't directly utilize UNet as the learning model for ROM. Instead, we compare the performance of existing segmentation models, selecting the most suitable one.
- Unlike previous studies with fixed input sizes, we enable encoder-decoder architectures in ROM to work with different image sizes using our proposed patching process. Uniquely qualifying the patching for real-time operation, we also arrange learning model parameters for real-time performance in accordance with the average patch size of the proposed traditional method.
- Introducing the PETA concept, we propose two new accuracy metrics, and evaluate the model outputs with these metrics. Unlike past studies using various morphological methods on segmentation maps [95], we evaluate

metrics directly from segmentation maps. We also design a procedure to improve inaccurate results by evaluating segmentation output with these metrics, addressing the common issue of jumping data in PD.

- We introduce an automatic configuration mechanism to be used defining traditional method settings. Thereby, we foresee that unsuitable configurations such as so high or low thresholds that can be defined by users may be prevented using this mechanism. In contrast to our work, previous auto-configuration selection techniques in PD remained at the sensor level and were not addressed at the application level [108].

1.4 Hypothesis

The hypotheses determined for the study in terms of hardware and PD are provided below:

- The patch-based structure for PD can increase inference speed. Moreover, it has the potential to enhance accuracy by allowing the learning model to operate with less redundant information.
- The traditional method undertaking the patch creation task in PD can be manually initialized by the user based on the eye appearance among alternative configurations, which may reduce the entry of erroneous values.
- The auto-configuration mechanism in PD can prevent configurations like thresholds that users might set inadvertently.
- Thanks to the system's modularity, expanding the experimental spectrum by adding different cameras and sensors to the system can reduce the need for different ETS devices.
- The ability of the system to be used with users' own glasses may enable less costly usage of the system.

2
METHODOLOGY

Developing a wearable ETS involves two design processes: hardware and software. Hardware design entails the creation of the system with specific usability considerations and ensuring component compatibility for the software to function as expected. Within the scope of this thesis, these considerations include the modular design of the hardware and the system's capability to be used with users' own glasses. Software design incorporates both usability and fundamental processing and functional sets. Given that the primary function of an ETS is to produce gaze points, the core processing and functions on the software side involve pupil detection and calibration processes. In this regard, the thesis introduces a new pupil detection method to generate gaze points through screen calibration. Additionally, a desktop application software was developed for basic visualization within the framework of specific usability and human-computer interaction concepts. This section sequentially explains the prepared hardware, pupil detection, calibration process, and software design under respective subheadings.

2.1 Hardware Design

The hardware structure proposed in the thesis is a prototype that includes only cameras. Additions of other sensors are beyond the scope of this thesis, and the main structure of the hardware is presented in Figure 2.1. According to the relevant figure, the hardware is divided into two parts: interchangeable parts and fixed parts. Interchangeable parts refer to components that need to be completely replaced along with additions of cameras and sensors. For example, to replace the current world camera, the components marked with blue rectangles need to be changed, and to replace the eye camera, the components marked with green rectangles need to be changed. Therefore, the design for the eye camera components includes both the eye camera and rigid components and infrared LEDs that need to adapt to the main structure. The components marked in red are the backbone structure that will remain unchanged and be used as fixed. Additionally, the diagram includes two types of arrow relationships: solid and dashed arrows. Solid arrows emphasize the assembly

relationship between components, while dashed arrows indicate that two components will be connected by cables.

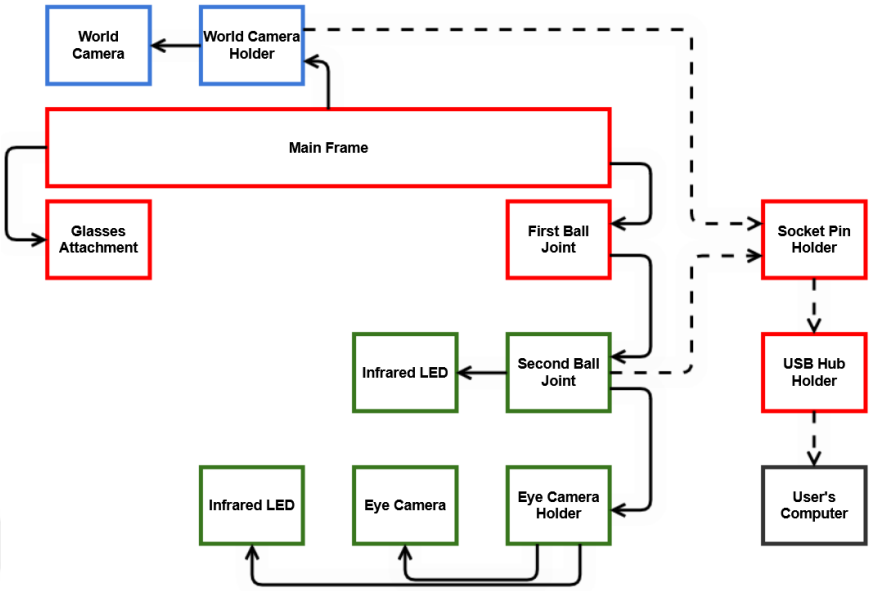


Figure 2.1 General structure of the proposed hardware components and relations

In the hardware proposal of the thesis, the following subsections describe the 3D solid design components, hardware components, and the process implemented for organizing them into a prototype.

2.1.1 3D Solid Design

The solid design efforts have been developed using SolidWorks 2022 software [109] and consist of 8 main components. The first component is the main frame shown in Figure 2.2. This frame is designed to carry cameras and be attached to the user's glasses. The figure in question includes front and rear views of the part. The part is divided into three sections. The right side will contain the apparatus that ensures attachment to the glasses. The middle section, which protrudes upwards, is the area where the world camera will be mounted. The flat bar on the left side is the area where the components related to the eye camera will be added. These three sections are found together in one solid part.

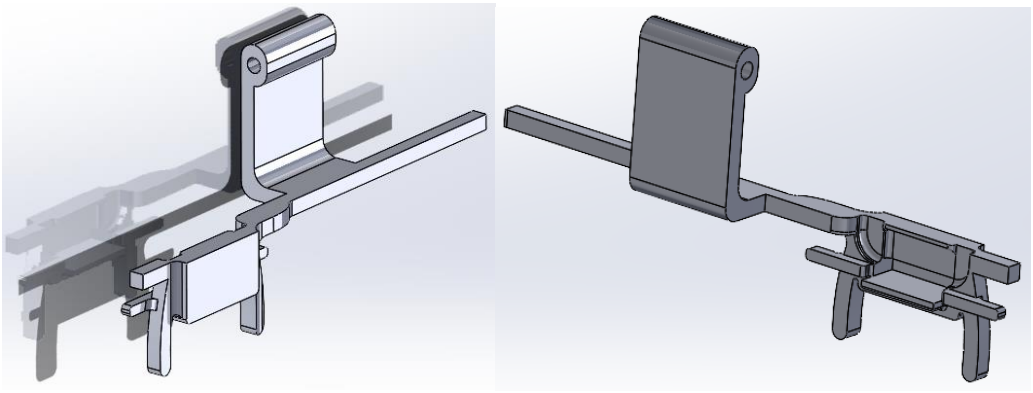


Figure 2.2 Main frame design to be plugged to subjects's glasses

The first component used to incorporate the eye camera into the main frame is a ball joint. This intermediary part is tightly fitted into the right flat bar area of the main frame and is glued during assembly. This part is shown in Figure 2.3.

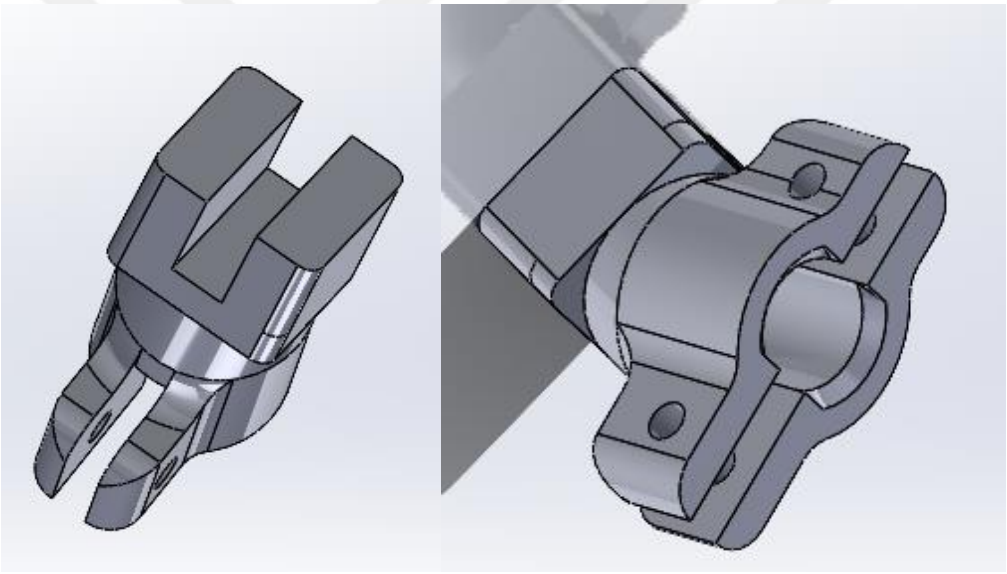


Figure 2.3 Eye camera and frame connector ball joint design to be plugged to main frame

The aforementioned ball joint is connected to a second ball joint, thus employing two ball joints before the assembly of the eye camera. Due to this design featuring two ball joints, the aim is to allow the eye camera to be positioned with considerable flexibility. Consequently, the camera can be positioned in various ways, both off-axis and on-axis, centered around the user's needs. Also, the second ball joint includes a platform that houses an infrared LED. The second part used for eye camera assembly is shown in Figure 2.4.

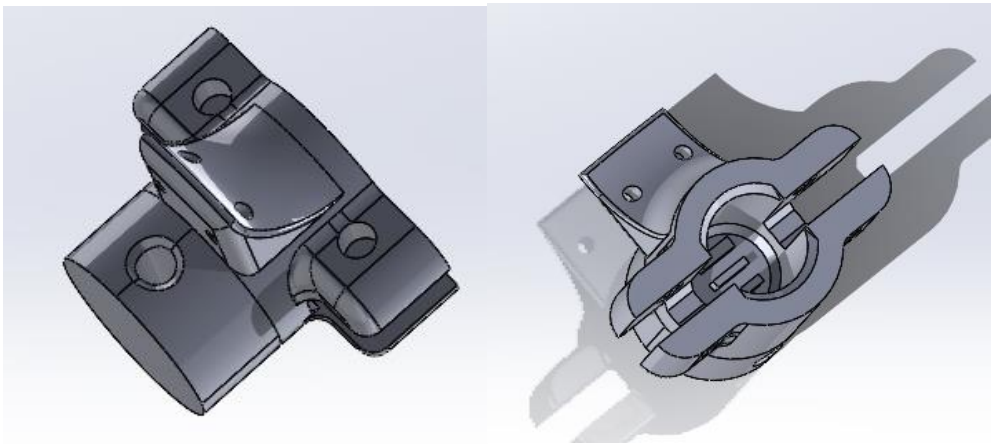


Figure 2.4 Eye camera holder connector ball joint design to be plugged to frame connector

The part where the eye camera will be mounted is shown in Figure 2.5. This component is assembled with the second ball joint. Additionally, the front part of the component features a sloped area. This area is designated for attaching the second infrared LED. The angle of the second infrared LED relative to the camera is designed to minimize the impact of corneal reflection.

The world camera, as a component to be mounted on the main frame, depends on the part shown in Figure 2.6.

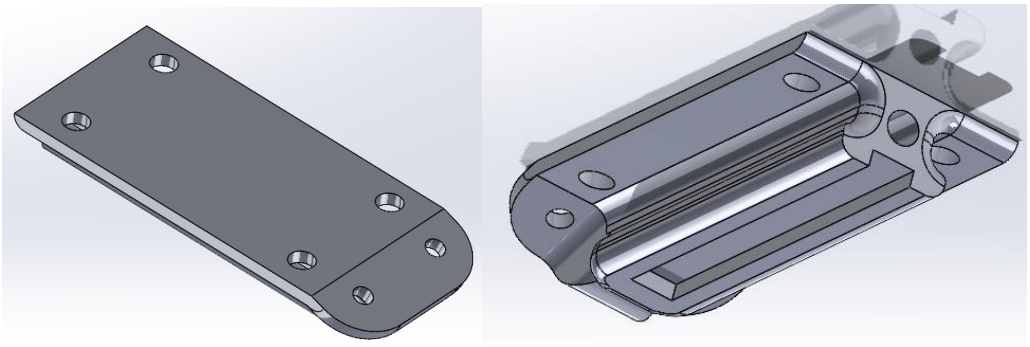


Figure 2.5 Eye camera holder design to be plugged to holder connector

The socket pin holder is the main component that allows for the interchangeability of sensor and camera connections and can be attached to the user's eyeglass frame. It contains several small ports supported by the USB hub, from which any component can be removed, and a new one can be integrated into the system. The relevant part was shown in Figure 2.7.

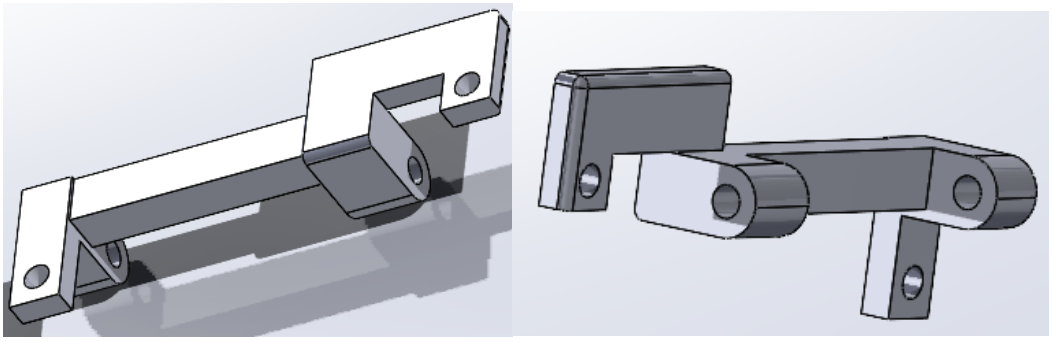


Figure 2.6 World camera holder design to be plugged to main frame

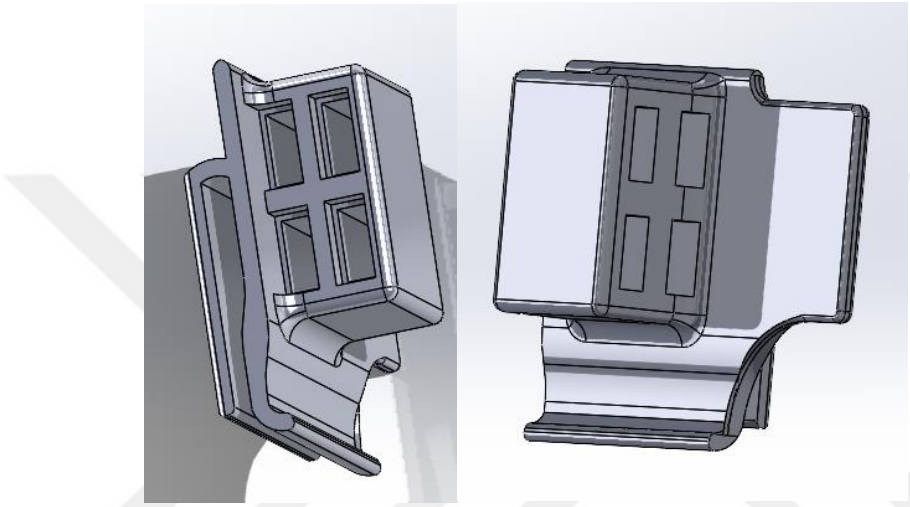


Figure 2.7 Socket holder design to be plugged to subject's glasses frame and USB hub

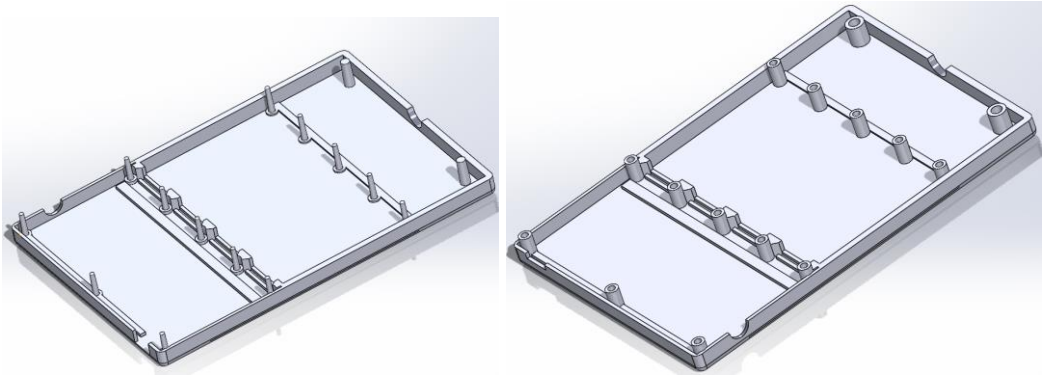


Figure 2.8 USB hub holder design

The USB hub holder contains both the USB hub itself and the necessary space for the cabling. This component is shown in Figure 2.8. Additionally, Figure 2.9 presents a pin structure for the USB holder. This structure will be adhered to the USB holder, and a magnet will be placed inside it. This design allows the USB holder to be secured to a location such as a lapel.

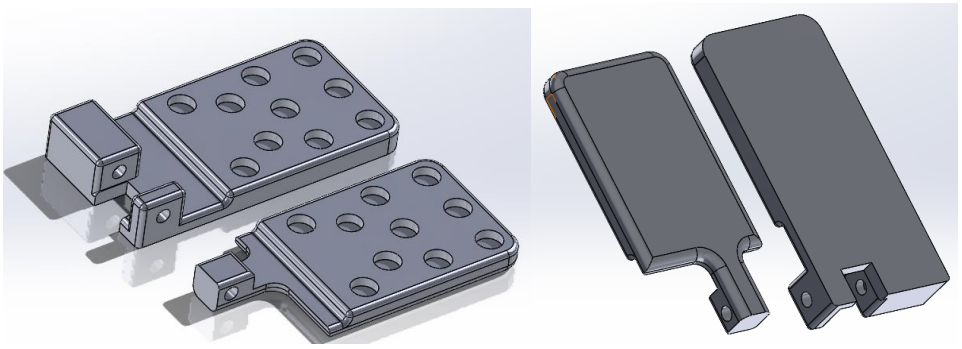


Figure 2.9 USB hub holder clips to be used for anchorage

The thesis also presents a headset onto which system components can be mounted for participants who do not wear glasses. The headset is in the form of a half-frame that is not attached to the head. The relevant headset is shown in Figure 2.10.

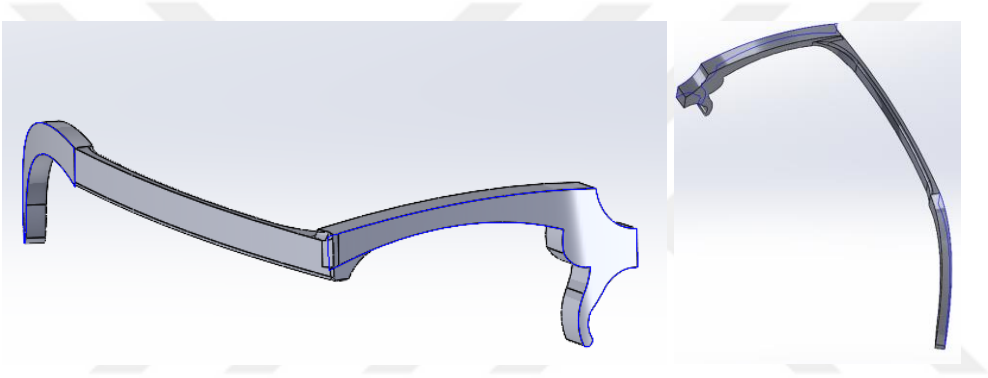


Figure 2.10 Headset design for users without glasses

2.1.2 Hardware Components

The world camera used in the system is the Logitech C505e HD webcam. This camera has a diagonal field of view of 60 degrees, with a resolution of 720p/30Hz. Additionally, it features a built-in microphone. Another important aspect of the camera is the placement of the lens as far to the right as possible. This allows for the field of view to approach the center of the face when the camera is positioned above the left eye. Consequently, the view direction between the subject and the camera can become more similar. To use the C505e HD as a world camera, it has been disassembled, and the camera card inside has been removed. Mounting the camera card onto the main frame requires a specific holder unique to the C505e. Therefore, different holders need to be designed for different cameras as well. The relevant webcam and its camera card can be seen in Figure 2.11.

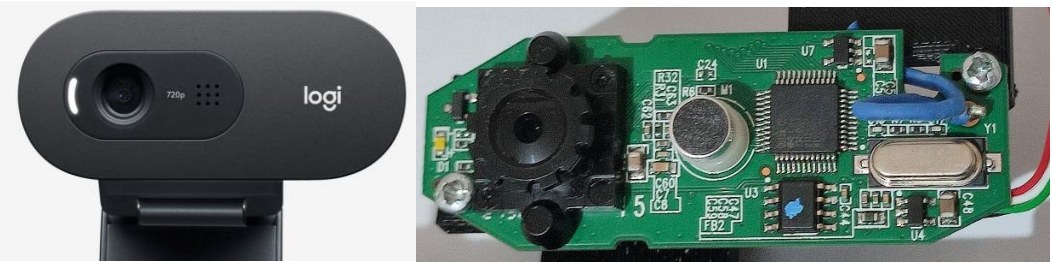


Figure 2.11 World camera component: C505e HD and its disassembled form

The eye camera used in the system employs the Taidacent GC0308 CMOS camera module, as in Figure 2.12. This module features a USB 2.0 cable and has a compact form. Although the camera's detection range is 5 cm, the focal length has been adjusted to be between 3-4 cm. The camera has a diagonal angle of view of 50 degrees and does not require camera calibration due to its lack of image distortion. It supports two different resolutions: 320×240 and 640×480, generally operating at 30 FPS. However, at a resolution of 320×240, it can support up to 128 FPS in MJPEG and YUV2 image formats. In some configurations of this camera group, an IR cutoff filter may be present. In such cases, the filter positioned beneath the lens needs to be removed to allow for proper pupil information capture with illumination from infrared LEDs. Another drawback of the camera is the presence of built-in micro infrared LEDs located to the right and left of the lens. These LEDs, being aligned with the camera angle, create corneal reflections, so the relevant components have been rendered passive by applying high voltage. Similar to the world camera, the eye camera also requires a holder specific to the camera model. If the camera is replaced, the corresponding holder needs to be redesigned accordingly.

The adaptation of the main frame to the user's glasses is a significant factor within the scope of the thesis. Instead of designing a 3D model for this purpose, a selection was made from existing alternatives and utilized in the study. This choice was driven by the necessity for the relevant component to be robust and as compact as possible. Consequently, the polarized sunglasses model capable of fitting over glasses, produced by the Musivon brand, was preferred. This product features a clip structure at its center. The clip structure attaches to the glasses' lenses with silicone tips and can be opened and closed using the built-in lever if desired. The clip, separated from the sunglasses, was adapted to the main frame. The clips are presented in Figure 2.13.

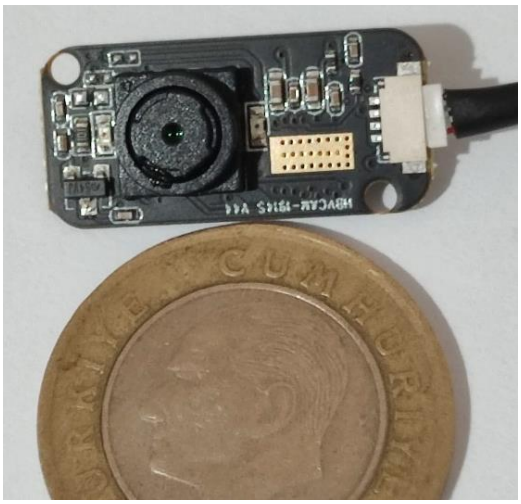


Figure 2.12 Eye camera component: GC0308 CMOS module



Figure 2.13 Glasses attachment apparatus from a sunglasses with clips

A USB hub named Codegen CDG-CNV38 model was utilized to enable the components within the main frame to be connected to the user's computer simultaneously. This hub features USB 3.1 type-C input and four USB 3.0 outputs. Consequently, it allows for a total of four sensors to be connected to the recommended system. Additionally, the hub is equipped with an external power supply socket. However, due to the sufficiency of power for the utilized cameras, the use of this power supply was not required. The relevant component is depicted in Figure 2.14.

In Figure 2.15, the socket pins for connecting the cameras to the USB hub, IR LED, and ball joint pin are illustrated. The socket pins are 4-pin JST connectors with a 1.25mm pitch. The IR LED has a diameter of 3mm and emits light at a wavelength of 940 nanometers, suitable for detection by the eye camera. The ball joint pin consists of a 2mm thick screw with a nut and a 5mm diameter sphere at its end. This sphere allows for the movement of the component attached to the end of the screw.

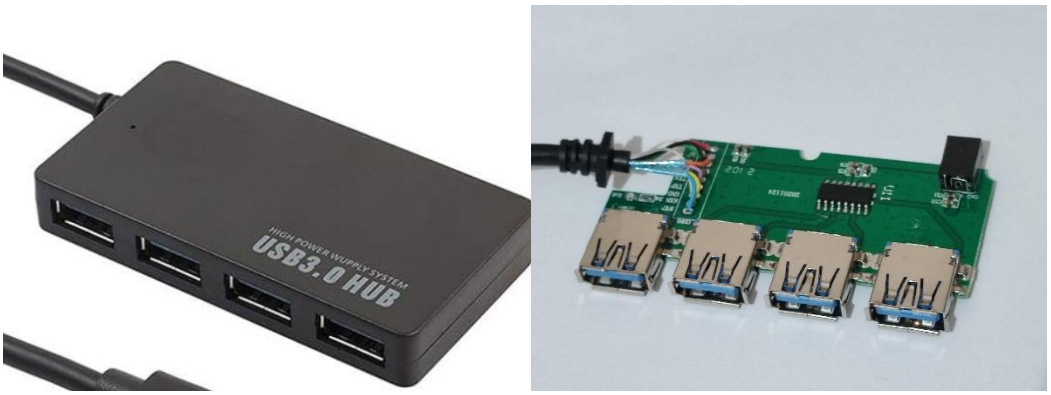


Figure 2.14 USB hub component: CDG-CNV38 Codegen and its disassembled form



Figure 2.15 Socket pins as female and male, IR LED, and ball joint pin

Finally, to prevent the glasses from slipping forward after the installation of the cameras, a silicone component has been utilized to correspond with the user's behind-the-ear area. This component is depicted in Figure 2.16. By increasing the pressure area exerted by the glasses, the amount of discomfort that may occur around the ears can be reduced, and the slipping of the glasses can be prevented.



Figure 2.16 Silicone eyeglass frame anti-slip component

Additionally, 1×5mm neodymium magnets have been used to fill the gaps in the hub cover clips. Furthermore, simple rods have been employed to facilitate the

interlocking of the hub cover clips and the adaptation of the world camera holder to the main frame.

2.1.3 3D Printing Process

During the preparation of the ETS prototype, 3D solid drawings were used with a Creality CR 6 SE brand and model 3D printer, along with Esun PLA+ filament. Prints were obtained using Creality's slicing software. The printing specifications were as follows: an infill density of 60%, a layer height of 0.12mm, a printing speed of 60 mm/s, a cubic infill pattern, support placement in accordance with touching buildplate, a nozzle temperature of 205 celsius, and a bed temperature of 50 Celsius. The components provided in the previous subsection, along with the headset frame that users without glasses will need to use additionally, are presented in Figures 2.16 and 2.17, respectively.



Figure 2.17 Printed parts of 3D solid components



Figure 2.17 The external headset frame to be used with printed parts for users without glasses

The following steps were applied for the assembly of the 3D prints and hardware components:

1. All cameras were attached to the holes in their respective holders using screws.
2. JST female sockets were inserted into the slots in the socket holder and secured with an interference fit.
3. The world camera holder was attached to the main frame with a rod pin for a slightly interference fit, allowing the field of view of the world camera to be adjusted up and down.
4. The ball joints between the main frame and the eye camera holder were first nested together. Then, the edges were tightened with screws through the holes to make the ball part of the joint harder to turn, ensuring that the eye camera could only be positioned deliberately, reducing slip effects.
5. An IR LED was attached to the holes at the end of the eye camera holder and the side of the ball joint to which the eye holder is screwed. This allows the IR LEDs to move dynamically according to the position of the eye camera and better illuminate the eye area. A 330 Ohm resistor fixed under the eye camera holder connects all the LEDs in series, and they are connected to the socket holder via a separate male JST pin.

6. The glasses clip was attached to the relevant place on the main frame.
7. All cameras and sensors have male JST pins to connect to the socket holder.
8. Neodymium magnets (1×5mm) were placed and glued into the circular slots of the USB Hub holder clips. The magnets allow the USB Hub to be attached to the user's clothing. The two clip covers were then stacked and fixed with a rod pin. Finally, the back cover of the USB Hub was glued on.
9. The first part of the three-part USB Hub holder contains the USB Hub. Male USB sockets were placed in the second part and connected to the USB Hub. In the third part, male USB socket cables were gathered and exited the holder as a single cable.
10. The USB Hub Cover has two cables, one going to the computer and one to the socket holder. A 1-meter Type-C extension cable was used due to the short length of the cable for data transfer and power supply to the computer. The cable going to the socket holder was connected to the female sockets in the holder.
11. During the cabling step, JST socket cables were shortened as much as possible and connected to the hardware components' own cables. All other cables used are USB 3.0 compliant to prevent potential data loss compared to some other alternative cables. Additionally, the outer surfaces of all cables between the USB Hub and the main frame were stripped to reduce weight and protected with fabric electrical tape as a lighter alternative. Since no electrical interference was observed in the tests, no additional shielded structure was included in the cables. However, for a more professional production process, it is important that the cables between the socket holder and the USB Hub are shielded.
12. Hot silicone was used to secure the IR LEDs, the cables attached to the 3D print, and the resistor under the eye holder.
13. Hot silicone was applied to the parts of the headset frame that touch the nose and ears to reduce discomfort from the print.
14. The assembly of the headset frame with the main frame and socket holder is similar to that of the user glasses, with no additional assembly steps required.

2.2 Pupil Detection based-on ROM and PETA

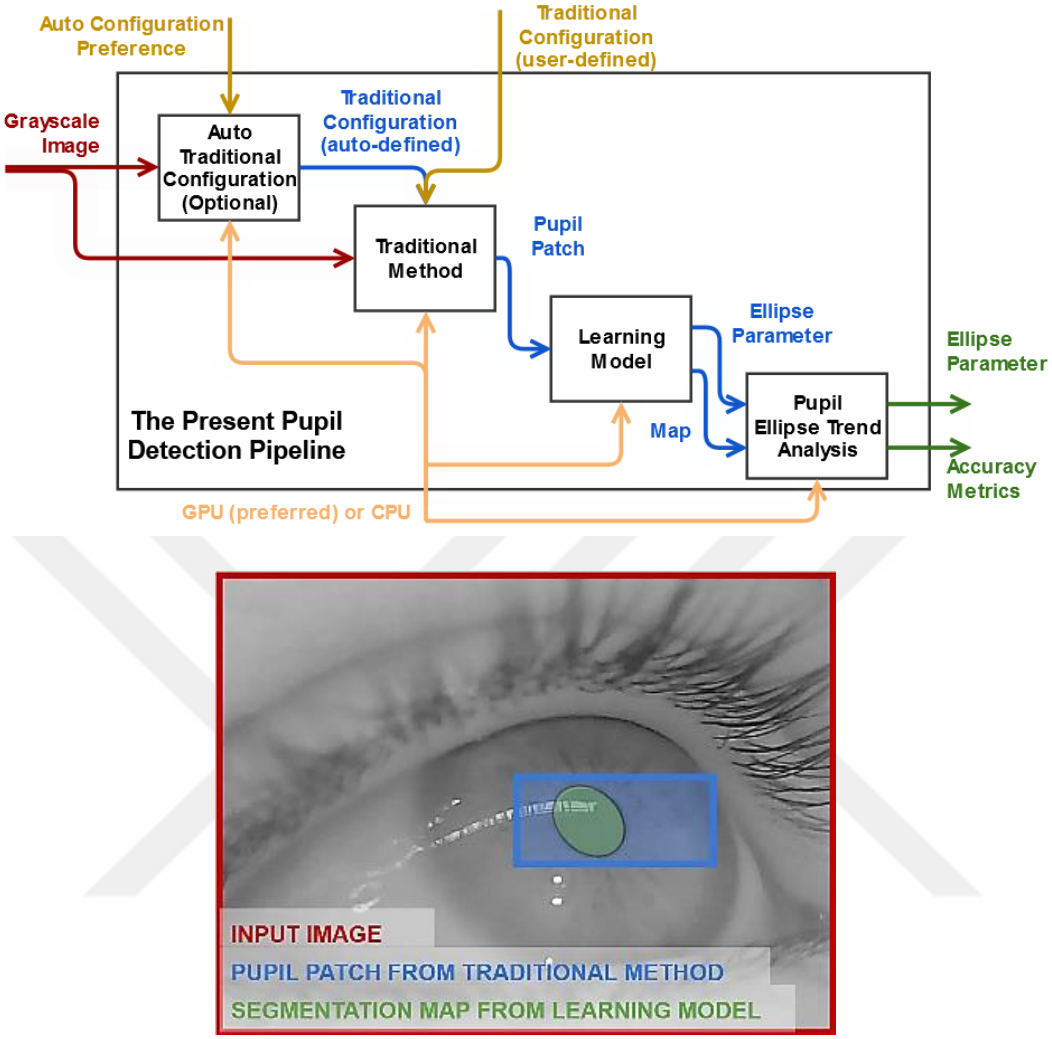


Figure 2.18 The general structure of the present detector as the main activity of SADT diagram and the visualized result according to the pupil patch received from the traditional method and ellipse parameters calculated by the learning model step

The purposed pupil detection architecture comprises three key components: i) the traditional method, ii) the learning model within the scope of ROM, and lastly iii) PETA step. The traditional method process requires a grayscale input image and a configuration (thresh-olding etc.) to be used in finding pupil patches and exports a patch that will be in reduced resolution compared to the input image. Subsequently, learning model receives the patch and uses it to segment pupil area within the patch. Next, fitting an ellipse to segmentation map is actualized here. The PETA component comes into play by calculating accuracy metrics using the ellipse parameters and potentially making corrections if necessary. For an illustrative

depiction of this architecture, given Figure 2.18, which represents the system as a top-level activity within an SADT (Structured Analysis and Design Technique) diagram [110].

It's essential to acknowledge that manually determining the traditional method configuration may lead to the patch containing insufficient or no pupil region when the image contains a pupil. This negatively impacts detection accuracy. To address this issue, an optional mechanism has been added to the system. This mechanism works at a desired time interval (e.g., every second) in a separated thread. It takes the input image which it receives at the end of the time interval and actively seeks to identify the traditional method configuration that optimally suits the given image. Thus, it tries to provide reliable and precise results.

2.2.1 Traditional Method

The causes of inaccurate results in amplitude-based traditional methods are more predictable compared to edge-based and edge-amplitude-hybrid traditional methods. Consequently, amplitude-based traditional methods may allow the user to define configuration or initialization settings in a more controlled manner. Believing in the potential success of amplitude-based methods when used correctly, we propose an amplitude-based method for the traditional method part of the study. The proposed method has three sequential sub-parts:

Finding minima pixel sequences: The lowest amplitudes are obtained in each row for the x-axis and in each column for the y-axis. They are collected in a sequence form as given the expression in the bellow. I_{ij} denotes pixel values of a received image. Since the sequences were created using all rows or columns of the image, each index of these sequences corresponds to a coordinate point and the sequences include a high-variance noise.

$$S_{row} = \begin{bmatrix} \min(I_{1,1}, I_{1,2}, \cdots, I_{1j}) \\ \min(I_{2,1}, I_{2,2}, \cdots, I_{2j}) \\ \vdots \\ \min(I_{i,1}, I_{i,2}, \cdots, I_{ij}) \end{bmatrix}, S_{col} = \begin{bmatrix} \min(I_{1,1}, I_{2,1}, \cdots, I_{i1}) \\ \min(I_{1,2}, I_{2,2}, \cdots, I_{i2}) \\ \vdots \\ \min(I_{1,j}, I_{2,j}, \cdots, I_{ij}) \end{bmatrix} \quad (2.1)$$

Noise reduction: To mitigate the noise present in the minima sequences, the Savitzky-Golay (SG) method was applied as a smoothing filter [111]. The filter fits a local polynomial to a window of adjacent data points and then using the

coefficients of this polynomial to compute the smoothed value at the center of the window. Filtered output of i^{th} sample $\hat{S}_i$ is calculated by a weighted summation of input signal neighbors. The coefficients C_j is determined by solving the least squares polynomial fitting problem for each window. In SG method, window length parameter (2L+1) is individually set to 20% of the length of each sequence.

$$\hat{S}_i = \sum_{j=-L}^{L} C_j \times S_{i+j} \tag{2.2}$$

Patch selection: Patch selection is carried out using the smoothed sequences. The configuration settings to be taken from user or to be defined by auto traditional method configuration mechanism are used in this step. Three parameters, each associated with a function taking a sequence as input, govern this step: THM for thresholding, PFX (Pupil Founder X) for the x axis sequence, and PFY (Pupil Founder Y) for the y axis sequence.

By using a function from THM function list in Table 2.1, threshold values are defined dynamically for each denoised minima sequence. THM functions and equations taking a sequence shown as S are the below, roughly from highest to lower amplitude values.

Table 2.1 THM functions to be used for the selection of threshold value. A value calculated by a THM function is used for sequences of x and y pair of an input image

TH1(S)=max(S)-std(S)	TH2(S)=(MMM×0.66) + min(S)
TH3(S)=max(avg(S), MPM/2)	TH4(S) = avg (S)
TH5(S)=min(avg(S), MPM /2)	TH6(S)=(MMM×0.33) + min(S)
TH7(S)=(avg(S)+min(S))/2	TH8(S)=min(S)+std(S)
where MPM = max*(S)*+min*(S), MMM* = max*(S)*–min*(S)*	

Following the smoothing of the minima sequences, the process continues with the application of PFX and PFY to the *x*-axis and *y*-axis sequences, respectively. There are a total of five distinct functions, collectively referred to as Pupil Founder (PF)

functions, which can be chosen for PFX or PFY. G_k expression is intended for use in relevant functions can be calculated as follows:

$$G_k = \{s_i |\, s_i < \tau, s_{i+1} < \tau, \ldots, s_{i+n} < \tau\} \tag{2.3}$$

where G_k represents the grouping of consecutive sequence elements, and s_i, s_{i+1}, …,s_{i+n} are the elements of the sequence. The condition, $s<\tau$ is applied to each element in the group, ensuring that they are all smaller than the threshold value τ.

1- Median First (MF): It selects the part closest to the middle index under the threshold of the sequence on the relevant axis. It can be calculated as in Eq. (2.4):

$$P_{MF} = \text{argmin}_k(\min(|G_i - M|, |G_{i+1} - M|, \ldots, |G_{i+n} - M|)) \tag{2.4}$$

where *M* represents the median index of the *S* array. Accordingly, MF is useful when the objective is to position the pupil near the center of the sequence along the relevant axis. This option can be particularly useful in places where the experimental procedure is well-defined, such as laboratory environments.

2- Max Depth (MD): MD chooses the segment with the lowest amplitude value from the parts under the threshold on the relevant axis. The equation is given in Eq. (2.5) and it is particularly useful when the pupil is known to exhibit the darkest amplitude. This option carries the potential of obtaining a patch size smaller than expected by preventing the inclusion of other small objects besides the pupil in the patch scope in experimental or in-house environments.

$$P_{MD} = \text{argmax}_k |\min(G_k) - \tau| \tag{2.5}$$

3- Max Length (ML): ML selects the segment with the highest number of elements among those below the threshold on the axis in use. This function given in Eq. (2.6) is suitable when the pupil is known to be larger than other dark areas.

$$P_{ML} = \text{argmax}_k(|G_k|) \tag{2.6}$$

4- Max of Length and Depth Product (LD): LD assigns a score to the parts under the threshold on the axis it is used and selects the part with the highest score. Score to be calculated with the expression is the product of the number of elements of the part and its absolute distance from the threshold level of the lowest amplitude. Hereby, this function given in Eq. (2.7) can be used when the pupil is known to be

darker or larger. Thus, it can ensure that the dark but small areas outside the pupil and the areas that are not as dark as the pupil but occupy a large area are not included in the patch.

$$P_{LD} = \text{argmax}_{\text{k}}(|G_k| * |\min(G_k) - \tau|) \tag{2.7}$$

5- First and End (FE): FE selects the entire area between the first and last indices under the threshold in the axis on which it is used. The equation given in Eq. (2.8) is beneficial in cases where perceiving the pupil based on amplitude information alone is challenging and may result in an enlarged patch size. In general, this should be the last alternative to be chosen, indicating the need for pre-experimental preparation for other options to work effectively. However, in an environment where pupil information is adequately detected, it still has the potential to produce results like the previous functions.

$$P_{FE} = \{s_n, s_{n+1}, \cdots, s_m\} \tag{2.8}$$

$$\text{where } n = \min\{j \mid s_j < \tau\} \text{ and } m = \max\{j \mid s_j < \tau\}\}$$

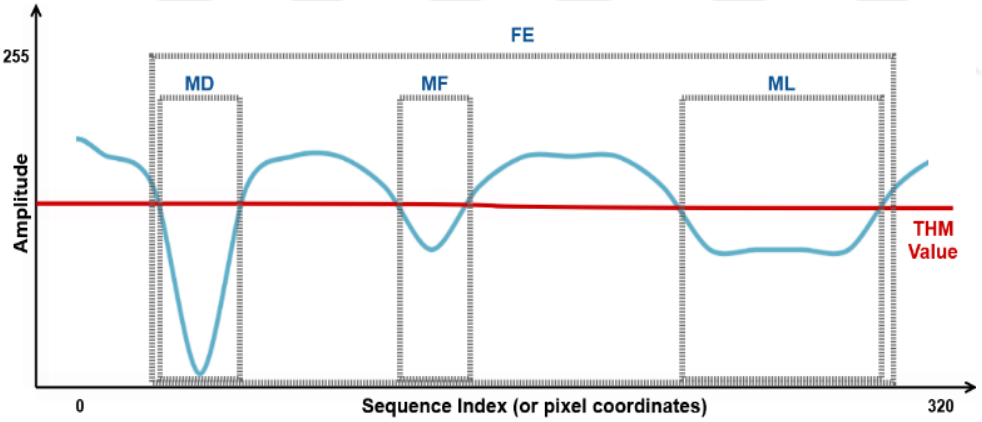


Figure 2.19 A denoised minima sequences for x-axis and different patches selected by PF functions. Basically, these functions try to select one segment between parts under the threshold value

These selections except for LD have been shown via an example minima sequence in Figure 2.19. Additionally, the groups *G*, which remain below the threshold, can also be observed in the relevant example interpretation. To complement these descriptions, a visualization in accordance with the result of the traditional method was given in Figure 2.20. Blue and green lines show the thresholds for *x* and *y* axis, respectively. Blue and green lines indicate the minima sequences for this image.

Top and right edges represent zero values for these graphs and distance from the relevant edge visualize the value of sequence indexes. The turquoise rectangle is the output of the traditional method, defined with minima sequences represented as blue and green curves. Eventually, this rectangle is sent to the learning model step.

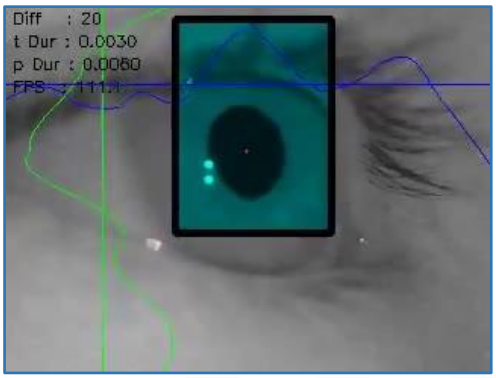


Figure 2.20 An example of the selected patch with randomly defined configurations.

2.2.2 Learning Model

In the learning model segment, an extensive comparison has been conducted among various segmentation models, including UNet [112], UNet++ [113], SegNet-VGG-19 [114,117], TransUNet [115], DeepLabv3-ResNet-50 [116, 118], DeepLabv3-MobileNet-L [119], PPMobileSeg-Tiny [120], CCSGD-ResNet-34 [122] and EGEUNet [121].

In the domain of medical image analysis, the accurate segmentation of biomedical images is pivotal for effective diagnosis and treatment planning. The U-Net architecture represents a significant advancement in this field. Their seminal paper outlines the architecture's unique design tailored specifically for medical image segmentation tasks. The U-Net architecture is structured in a symmetric U-shape, which is instrumental in its performance. The design comprises two main pathways: the downsampling and the upsampling. The downsampling path consists of repeated application of two 3×3 convolutions (each followed by a rectified linear unit (ReLU) and a 2×2 max pooling operation with stride 2 for downsampling). This path is designed to capture the context in the image, which is essential for understanding the overall structure present in the input data. On the other hand, the upsampling involves a series of upsampling steps that aim to enable precise localization. This part of the network uses transposed convolutions to increase the

resolution of the output. Each step in the expansive path also includes a concatenation with the correspondingly cropped feature map from the down sampling. This step is critical as it helps the network to propagate context information to higher resolution layers, allowing for more precise localization. The relevant architecture was shown in Figure 2.21.

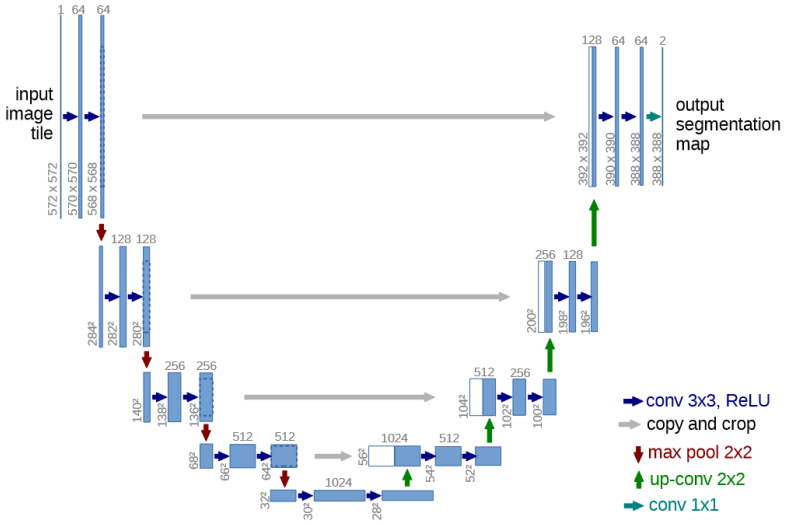


Figure 2.21 Learning model to be used for pupil segmentation: UNet architecture [112]

In the evolving field of biomedical image segmentation, UNet++ has emerged as a notable enhancement of the traditional U-Net architecture, specifically designed to address some of the limitations found in the original model. this architecture refines the standard U-Net by introducing a series of nested, dense skip pathways. UNet++ modifies the classic U-Net structure by incorporating multiple sub-networks at varying depths, which are interconnected through redesigned skip pathways. These pathways are not merely direct connections; instead, they create a dense grid of skip connections that enhance feature propagation and fusion across different layers of the network. The nested architecture allows for an iterative refinement of the segmentation output, which can be particularly beneficial in medical applications where precision is critical. The primary innovation in UNet++ lies in its ability to adaptively adjust the depth of the network during training, enabling it to optimize performance based on the complexity of the segmentation task at hand. This is facilitated by deep supervision, a technique where intermediate layers are also connected to the output, allowing for the direct backpropagation of gradients to earlier layers. This results in faster convergence and potentially higher accuracy, as

each layer learns more directly from the ground truth data. The UNet++ architecture was given in Figure 2.22.

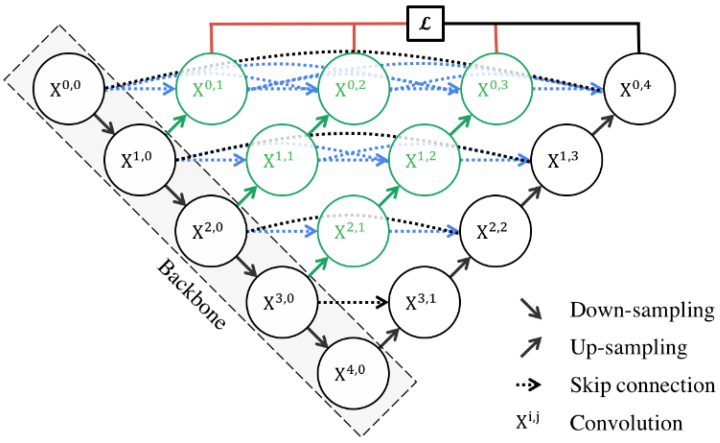


Figure 2.22 Learning model to be used for pupil segmentation: UNet++ architecture [113]

SegNet represents a significant advancement in the field of deep learning for semantic image segmentation, particularly useful in applications such as autonomous driving and urban scene understanding. The architecture of SegNet is distinctively structured as an encoder-decoder network. The encoder consists of a series of convolutional layers derived from the VGG model, which are designed to capture rich feature representations from the input images. The novelty in SegNet comes from its decoder component, which uses a unique method of up-sampling. Unlike conventional methods that use transposed convolutions, SegNet's decoder reconstructs the input image resolution using max-pooling indices recorded during the corresponding encoder's max-pooling operations. This method is both memory efficient and effective in preserving the spatial precision in the output segmentation maps. One of the key strengths of SegNet lies in its efficiency. By reusing pooling indices for up-sampling, SegNet significantly reduces the number of trainable parameters in the decoder, which not only minimizes the computational load but also mitigates the risk of overfitting when training on datasets with limited examples. This aspect makes SegNet particularly appealing for real-time applications, where both accuracy and speed are essential. SegNet structure is in Figure 2.23.

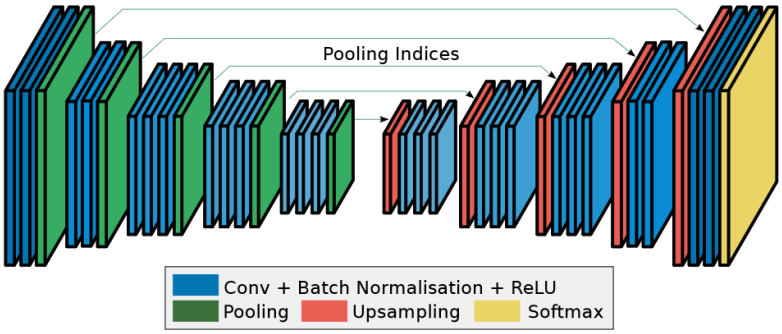


Figure 2.23 Learning model: SegNet architecture [114]

In the rapidly advancing field of medical image segmentation, the integration of Transformer models with U-Net architectures has led to the development of innovative approaches, such as the TransUNet. TransUNet incorporates the Transformer's self-attention mechanisms with the robust spatial encoding capabilities of a U-Net, creating a powerful hybrid model designed for high accuracy in medical imaging tasks. TransUNet uniquely combines the advantages of Transformers and convolutional neural networks (CNNs). It starts with a CNN-based encoder that extracts spatial features from medical images. The Transformer layers then process these features, capitalizing on the self-attention mechanism to model long-range dependencies within the image data. This is critical in medical image segmentation, where understanding global context and subtle details can significantly influence the accuracy of the segmentation results. TransUNet demonstrates considerable improvements over traditional CNN-based segmentation models, particularly in handling complex anatomical structures in medical images. Its ability to integrate detailed contextual information makes it adept at navigating the nuanced variations in medical datasets, which often contain small, yet clinically significant anomalies. TransUNet components are given in Figure 2.24.

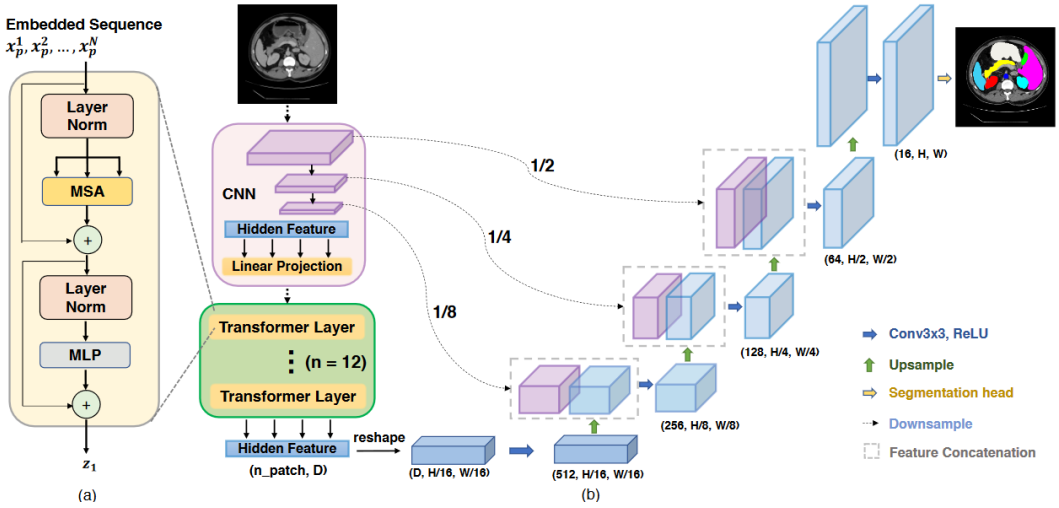


Figure 2.24 Learning model to be used for pupil segmentation: TransUNet architecture [115]

DeepLabv3, an advanced deep learning framework for semantic image segmentation, represents a significant evolution in the DeepLab series, which has been pivotal in addressing the challenges of segmenting complex images. It introduces several enhancements over its predecessors, primarily through the optimized use of atrous convolutions to capture multi-scale context without sacrificing resolution. The core innovation in DeepLabv3 lies in its systematic application of atrous convolutions, which enable the model to manage the scale of context information effectively. Atrous convolutions, also known as dilated convolutions, expand the kernel of traditional convolutions by inserting spaces between the kernel points, thereby increasing the field of view without increasing the number of parameters or the amount of computation. This technique allows DeepLabv3 to gather more extensive contextual information and enhance feature extraction capabilities, especially crucial in handling images with intricate details. Additionally, DeepLabv3 employs an improved atrous spatial pyramid pooling (ASPP) module, which probes an incoming feature map at multiple rates and fields of view, capturing both local and global context efficiently. The ASPP module is equipped with several parallel atrous convolutions with different dilation rates, which helps the network to segment objects at various scales more accurately. This feature is particularly beneficial in medical imaging, where precise delineation of anatomical structures varies significantly in size and shape across different patients. ASPP module of DeepLabv3 was shown in Figure 2.25.

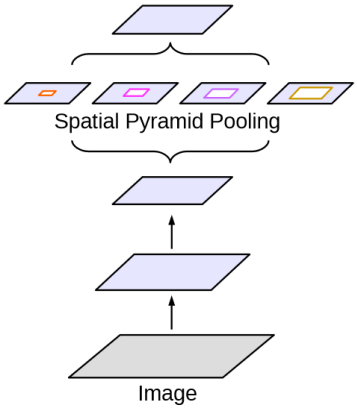


Figure 2.25 Learning model: DeepLabv3 architecture [116]

PPMobileSeg represents a pivotal advancement in the realm of semantic segmentation for mobile and edge devices. Developed to address the stringent constraints of computational resources and energy efficiency required by mobile platforms. This model offers an optimal balance between performance and computational overhead, making it particularly suited for real-time applications in mobile and embedded systems. The core of PPMobileSeg employs depthwise separable convolutions, a technique that significantly reduces the number of parameters and computational complexity compared to traditional convolution operations. These convolutions decompose a standard convolution into a depthwise convolution and a pointwise convolution, drastically lowering computational cost and model size without a substantial sacrifice in performance. In addition to depthwise separable convolutions, PPMobileSeg incorporates an efficient pyramid pooling module to aggregate contextual information from different scales. This module enhances the capability of the network to capture both detailed textures and larger scene contexts, which is crucial for achieving precise segmentation in diverse environmental conditions encountered in mobile applications. Another notable feature of PPMobileSeg is its use of lightweight attention mechanisms, such as squeeze-and-excitation blocks, which further refine the feature maps generated by the network. These attention mechanisms focus on relevant features and suppress less useful information, thereby improving the discriminative power of the network without significant increases in computational demands. PPMobileSeg structure is shown in Figure 2.26.

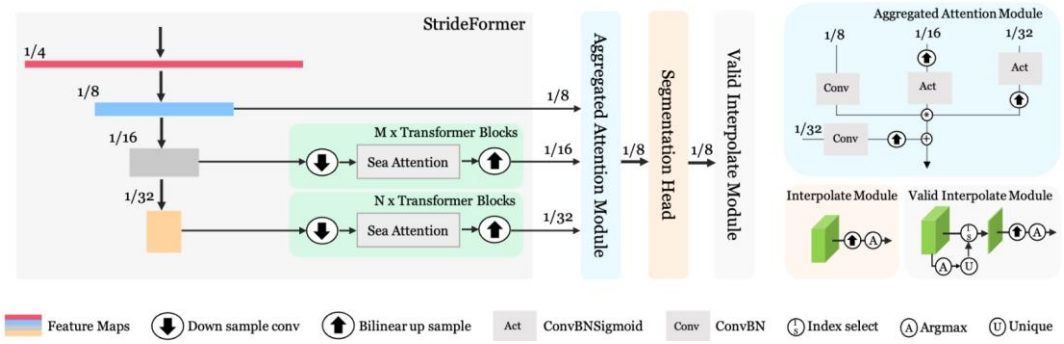


Figure 2.26 Learning model: PPMobileSeg architecture [120]

The ongoing evolution in medical image segmentation has witnessed the introduction of numerous advanced neural network architectures, among which the Efficient Group Enhanced UNet (EGE-UNet) has shown promising results, particularly in the field of dermatological image analysis. It represents a significant advancement in enhancing the accuracy and efficiency of skin lesion segmentation. EGE-UNet modifies the traditional U-Net architecture by integrating group convolution mechanisms, which optimize the network's parameter efficiency and computational speed. The group convolution approach allows the model to handle different feature channels separately, reducing the computational burden while maintaining the model's capacity to learn rich feature representations. This is particularly beneficial for the segmentation of skin lesions, where capturing detailed textural and color variations is crucial for accurate diagnosis. Furthermore, EGE-UNet introduces an enhanced decoding pathway that incorporates attention mechanisms to focus more precisely on areas of interest within the skin images. These attention modules improve the network's ability to distinguish between lesion tissues and normal skin, thereby enhancing the precision of the segmentation output. This focused approach is vital for medical applications where the boundaries of lesions are often subtle and can vary significantly between patients. The introduction of features positions EGE-UNet as a cutting-edge tool in the arsenal of medical imaging technologies. The architecture is shown Figure 2.27.

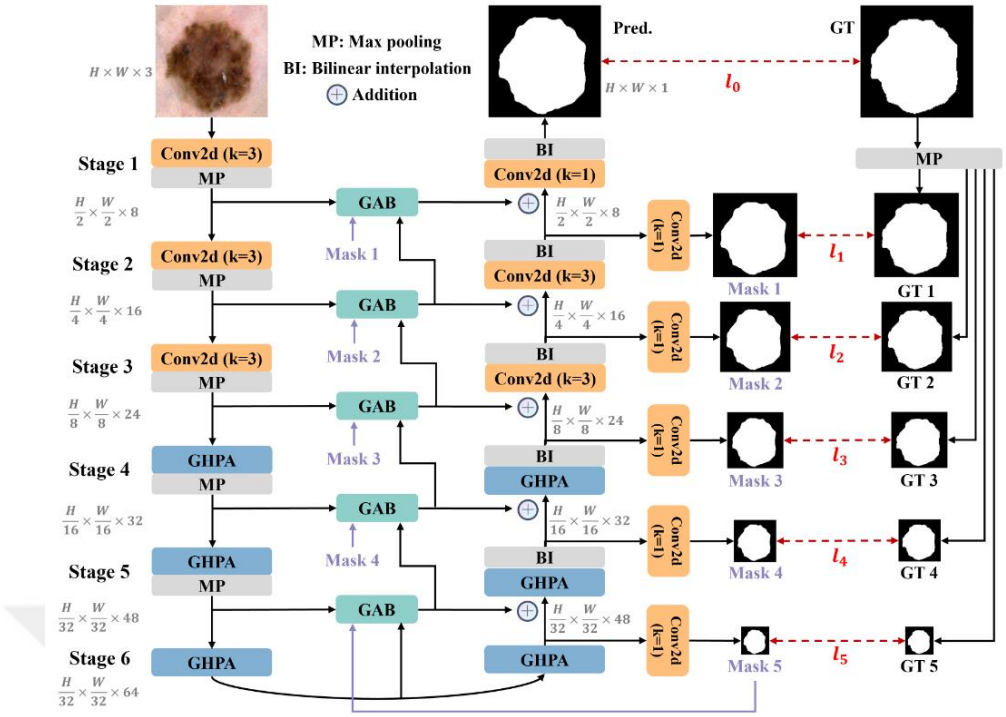


Figure 2.27 Learning model: EGE Unet architecture [121]

In the realm of medical image segmentation, domain generalization remains a formidable challenge due to the inherent variability of medical imaging across different institutions and equipment. Addressing this issue, Hu, S., Liao, Z., and Xia, Y., introduced an innovative framework known as Contrastive Single Domain Generalization (CCSDG). This novel approach aims to enhance the robustness of segmentation models against domain shifts, which is crucial for deploying machine learning models in diverse clinical environments. The core innovation of the CCSDG framework lies in its utilization of contrastive learning to enforce strong discriminative feature learning in medical image datasets. By focusing on channel-wise feature representations, CCSDG effectively captures and amplifies the essential features that are invariant across different domains. The model employs a contrastive loss mechanism that minimizes the distance between similar features and maximizes the distance between dissimilar features across varying imaging conditions. This method significantly improves the generalization capabilities of the model by reducing overfitting to the source domain's specific characteristics. CCSDG incorporates a sophisticated channel-wise attention mechanism that dynamically adjusts the relevance of each channel based on the imaging context. This adaptive feature recalibration enhances the model's sensitivity to pertinent features while suppressing less relevant information, thereby improving the

precision of segmentation outputs. The attention mechanism also contributes to the model's adaptability, enabling it to maintain high performance even when exposed to images from previously unseen domains. Moreover, CCSDG's architecture is designed to be lightweight, making it suitable for real-time applications in medical settings where computational resources might be limited. CCSDG components and their relations was given in Figure 2.28.

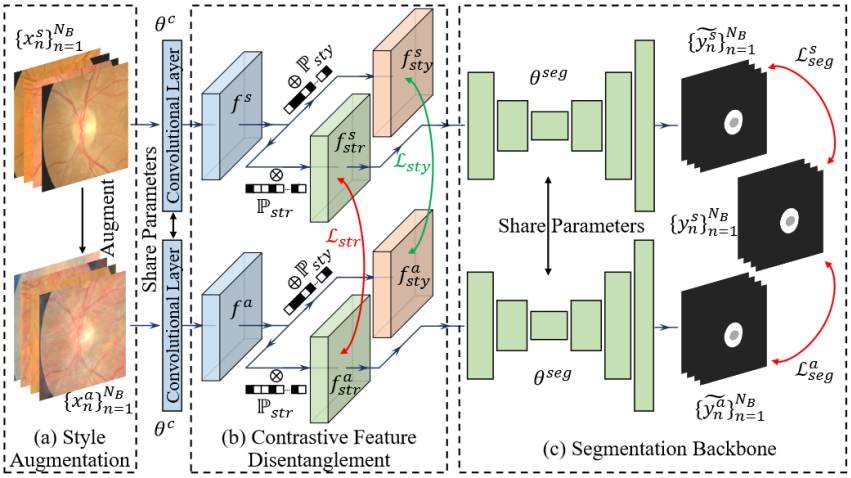


Figure 2.28 Learning model: CCSDG architecture [122]

The performance assessment encompassed evaluating prediction time and memory consumption on GPU and CPU for both full resolution and the average resolution of the results obtained from the traditional method. In addition, the models with satisfactory dice scores have been fine-tuned and brought to a level that can operate in real-time at the average resolution of the traditional method, by decreasing their parameters in various ways. In the study, fine-tuned model names are indicated by the suffix "-S".

During the creation of the S models, the primary factor employed was the reduction of the models' parameters in various ways to achieve operation around 8.3 ms on the setup hardware. The sole procedure applied for parameter reduction was based on the models achieving an operational time of approximately 8.3 ms at the average patch size, and therefore, no additional standard was established.

According to average patch size that will be addressed in Experimental Result section, we have changed model structure as in the below: UNet-S has 3 layers with channels (32,64), (64,128), (128,256) with double convolution. Double convolutions with channel (1,32) are performed before and after from layers. UNet++-S has

3 layers with a channel size as (32,64,128,256). TransUNet-S has 128 hidden size, 128 MLP dimensions, 2 heads with channel 128 and 2 layers for the transformer architecture. Also 16 patch size, decoder channels as (64,64,64,64), skip channels as (512, 256, 64, 32), number of layers (1,1,1) for ResNet and number of skips 3 were performed. SegNet-VGG-19-S has a four VGG stages. Each stages have 6 VGG fea-tures. While first two stages take the features from beginning of the VGG and last stages take from the end. CCSGD-ResNet-34-S has three layers for its UNet block with (128,128,128), (128,64,128), (128,64,128) and a ResNet blocks without last 2 layers. PPMobileSeg-S has channels (8,8,8,16,32), embedding dimensions (16,32), the number of heads 2, and ½ or ¼ MobileNetV3 blocks' channel size compared with its tiny configuration.

The final step within the learning model section involves the fitting of an ellipse into the segmentation map. For this purpose, a method, operating in a least-squares manner, is employed to determine the optimal ellipse placement [123]. With reference to find an optimal ellipse, this implicit equalization is solved using a point set to be used to fit the ellipse:

$$a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6 = 0 \tag{2.9}$$

Using these coefficients shown as a_i, ellipse parameters can be calculated. For example, center point equalization can be given as in the following:

$$(x, y) = \left(\frac{2a_3a_4 - a_2a_5}{{a_2}^2 - 4a_1a_3}, \frac{2a_1a_5 - a_2a_4}{{a_2}^2 - 4a_1a_3}\right) \tag{2.10}$$

The largest ellipse derived from the output of this method is selected and used as the pupil ellipse parameter, thereby finalizing the learning model's contribution to the overall pupil detection process. Ultimately, the calculated ellipse parameters and the segmentation map are sent to the PETA step.

Additionally, the auto traditional configuration mechanism involves running the selected learning model for the entire image and then finding the traditional method parameters that best represent the pupil parameters obtained with entire image segmentation. Eq. (3.1) in the Experimental Results section is employed for the selection of the configuration that represents the pupil area in the smallest size and pupil including rate.

With the determination of the ellipse parameters, the main structure for detecting the pupil in the thesis is completed. The processes from the traditional method to the learning model can generally be summarized as shown in Figure 2.2.

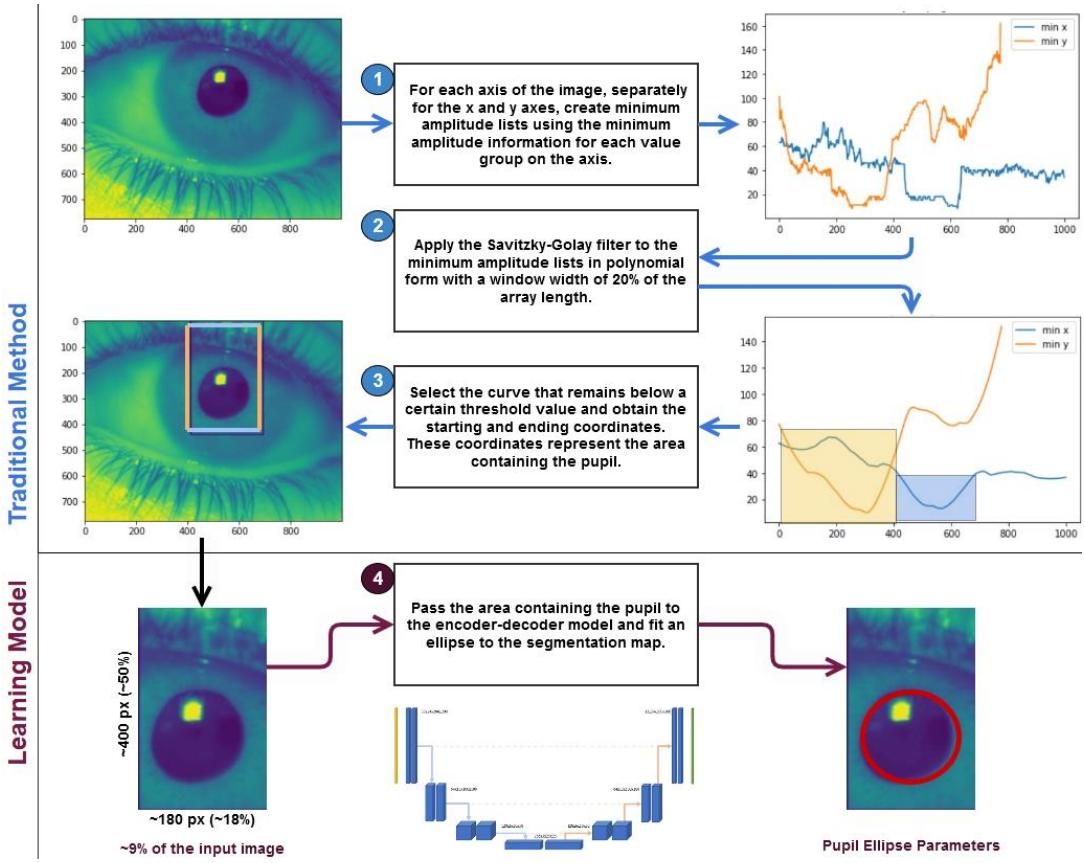


Figure 2.29 General backbone of the present pupil detector pipeline without PETA and automatic traditional configuration mechanism

2.2.3 Pupil Ellipse Trend Analysis (PETA)

PETA approach involves two operations. The first operation computes of metrics designed to assess the outputs generated by the learning model, while the second operation focuses on the development of a correction method to rectify pupil ellipse parameters, should the need arise, based on the metrics. Two metrics named as entropy and intensity have been proposed for metrics of PETA. In order to calculate the entropy, a Sobel edge filter is applied to the segmentation map area only within the pupil ellipse area. To handle the limited area over the model output, map (M) is masked with ellipse parameters (E) as:

$$\widehat{M} = M \odot \mathrm{E} \tag{2.11}$$

After that, gradient (G) is calculated for x and y axes of $\widehat{M}$, using Sobel edge detector. Next, the sum of the gradient amplitudes $G_{i,j}$ obtained at the output of the filter is divided by the total number of pixels, and entropy metric M_{Ent} is calculated as in Eq. (2.12).

$$M_{Ent} = \frac{1}{N}\left(\sum_{i,j} G_{i,j}\right) \tag{2.12}$$

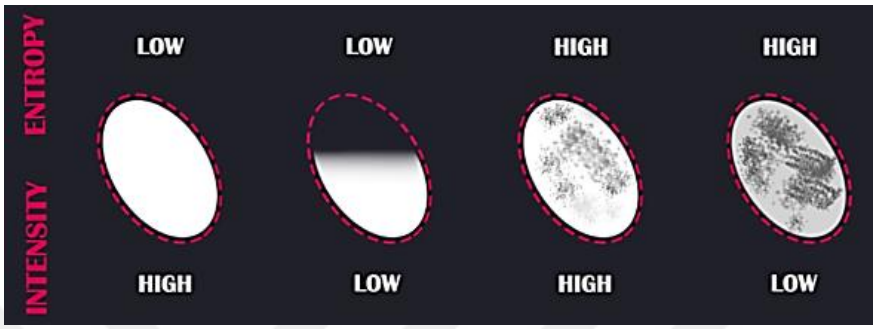


Figure 2.30 The visualization of Entropy and Intensity concepts. Pupil regions in a segmentation map are shown with white color. Pink ellipses represent the fitted parameters using pupil regions

In this respect, the entropy metric can indicate integrity or precision within the segmentation map from which the ellipse is generated. To calculate the intensity M_{Int}, the pixel amplitudes of the segmentation map within only the pupil ellipse area are average of $\widehat{M}$. Therefore, intensity metric can refer to the magnitude of the prediction probability in the segmentation map area from which the ellipse was generated. Using these metrics, an inference has been expressed as follows:

- Low Entropy, Low Intensity: The prediction is considered good. The eye may in the half-closed position or even if the segmentation result is of low probability, it may still better compared to the alternatives.
- Low Entropy, High Intensity: The prediction is considered is very well and the eye is in the open position.
- High Entropy, Low Intensity: The prediction is deemed very poor.
- High Entropy, High Intensity: The prediction is considered good, and the eye is open, but it can be said that there are various obstacles or reflections on the pupil.

The visual representation of the above inferences is provided in Figure 2.30. According to this figure, when the entropy metric is low to medium and intensity is

high, it indicates successful segmentation. On the other hand, the segmentation map can vary depending on the subject's eye appearance and environmental conditions. Therefore, the entropy-intensity values that will distinguish segmentation as successful and unsuccessful will also be specific to each experimental session. To dynamically determine this range and differentiate a successful segmentation, a formula is used based on the last 120 entropy and intensity values (or those within the about last 1 second, to be defined according to the system frequency) with newly calculated values. The comparison simply attempts to detect anomalies in entropy and intensity by using variance. In this way, entropy anomalies that suddenly rise too much or intensity anomalies that suddenly falls too much can be detected. Accordingly, the pupil ellipse is considered successful if $M_{Ent} > \max(\mathrm{L_{Ent}}) + \mathrm{var}(L_{Ent})$ and $M_{Int} < \min(\mathrm{L_{Int}}) + \mathrm{var}(L_{Int})$ otherwise it is unsuccessful. L_{Ent} and L_{Int} refer to last 120 data points for entropy and intensity metrics, respectively.

In the PETA correction step, the last successful ellipse parameter is used instead of the pupil ellipse marked as unsuccessful. However, a condition has been set for this use, and any ellipse parameters that do not meet this condition are not corrected with the previous successful one. The relevant condition is that the difference between the unsuccessful ellipse center point and the last successful ellipse center point is greater than 19 pixels. The threshold value of 19 pixels is the upper band value obtained according to the preparation procedures of the LPW dataset [124]. The threshold value is grounded in the following assumptions and reasoning:

(i) Eyeball Area Within Camera Detection: It is assumed that the area of the eyeball can fit within the area that the eye camera can detect. For example, if the image collected by the eye camera is 320×240 pixels, the eyeball is expressed with roughly 240×240 pixels.

(ii) Maximum Angular Velocity of the Human Eye: A human eye can have a maximum angular velocity of 700 degrees per second [125]. Since this is a peak value allowed to only 25 degrees of visual angle, we decreased it as maximum 500 degrees for LPW dataset [126].

(iii) Frame Rate of LPW Dataset: LPW dataset is 120 Hz. Therefore, the maximum angular velocity that a human eye can achieve between two frames is 4,166 degrees (500/120).

(iv) Upper Coordinate Distance: In a 2D image, the largest difference in pixel coordinates resulting from a change in eyeball angle between two frames occurs when the geometry forms an isosceles triangle, as illustrated in Figure 2.31. Consequently, we computed a maximum shift of 8,723 pixels for an angular change of 4,166 degrees.

(v) Upper Band Threshold Values: Upon identifying that the maximum distance between two center points differed by approximately 9 pixels, we applied the widely accepted 5-pixel margin of error as per the literature for each center point. When we account for this 10-pixel permissible error difference (adding 5 pixels to each center point), the value utilized in our study was became 19 pixels for the upper bound. The relevant equalization rounded to nearest can be seen in the below. ω_{max}, f_{cam}, r_{eye}, $E[\varepsilon]$ parameters denote 500, 120, 120, 5 values, respectively.

$$Thr_{upp} = \left\lfloor \sqrt{2{r_{eye}}^2 \left(1 - \cos\left(\frac{\omega_{max}}{f_{cam}}\right)\right)} + 2E[\varepsilon] + 0.5 \right\rfloor \tag{2.13}$$

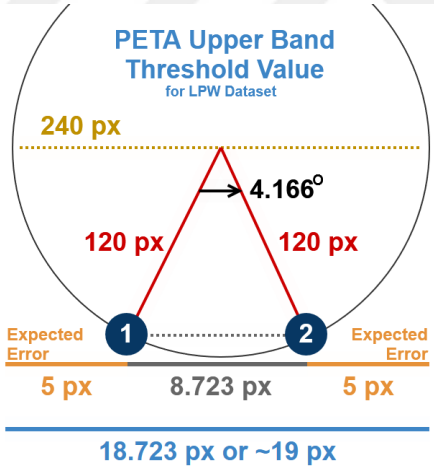


Figure 2.31 PETA Upper Bound Threshold Value representation between two consecutive center points for LPW dataset. The value is about 9 pixels, however we defined it as 19 pixels together with 5-pixel errors

The upper band threshold value, denoted as Thr_{upp} provide a solution to the problem of jumping data, especially during blinking or in cases where insufficient pupil information occurs due to any obstacle. However, the value of 19 pixels is not applied to pupil ellipse parameters that are marked as successfully/normal according to entropy and intensity values. In this way, if it is assumed that the segmentation is fulfilled successfully according to the metrics, the difference

between the calculated and annotated pupil ellipses is prevented in cases such as the movement of the eye camera or the displacement of the pupil center point when the eye is reopened.

2.3 Calibration Procedure

In the thesis, a 2D calibration method is applied to determine the mapping model coefficients. For the mapping model coefficients, two polynomial equations are fitted for each eye using the pupil and marker center points. These two equations are used to separately find the world coordinates for each pair of pupil centers along the x and y axes. The polynomial equation included in the ETS has a total of 8 unknowns and can be represented as follows:

$$W_x = a_1 \times 1 + a_2 \times P_x + a_3 \times P_y + a_4 \times {P_x}^2 + a_5 \times {P_y}^2 + a_6 \times P_x \times P_y + a_7 \times {P_x}^2 \times P_y + a_8 \times P_x \times {P_y}^2 \quad (2.14)$$

$$W_y = b_1 \times 1 + b_2 \times P_x + b_3 \times P_y + b_4 \times {P_x}^2 + b_5 \times {P_y}^2 + b_6 \times P_x \times P_y + b_7 \times {P_x}^2 \times P_y + b_8 \times P_x \times {P_y}^2$$

In the calibration procedure, the instantaneous normalized center coordinates of the pupil are used as P_x and P_y values, and the normalized coordinates of the marker collected from the world camera are used as W_x and W_y values in the equation above. After calibration, the obtained normalized pupil coordinates and the fitted coefficients are inserted into the equation to separately derive the normalized world coordinates W_x and W_y. Once the coefficients in the polynomial function are determined, the newly acquired pupil center coordinates can be entered into the same equation to calculate the region that the subject is looking at on the world camera.

A nine-point calibration session is used to obtain the list of unknowns in this equation. Accordingly, markers are displayed at the corners of the screen, at the midpoint between all adjacent corners, and at the center of the screen. The relevant calibration points are shown in Figure 2.31. From each point, 90 pairs of pupil and marker data are collected to form the list. Thereby, in a calibration procedure, toally 810 pupil center point and marker coordinate pairs are collected. Additionally, if the newly obtained pupil coordinates deviate by more than 5 pixels from the

previous coordinates, the calibration series is cancelled, and the procedure is reapplied for the respective marker. This requires the user to pay careful attention to the markers during calibration. Although a scatter control relative to the first item on the list would prevent saccadic issues, this control has not yet been implemented. Subsequently, the lists are expressed in matrix format and solved using the least squares method. The relevant matrix is shown in Equ. (2.15).

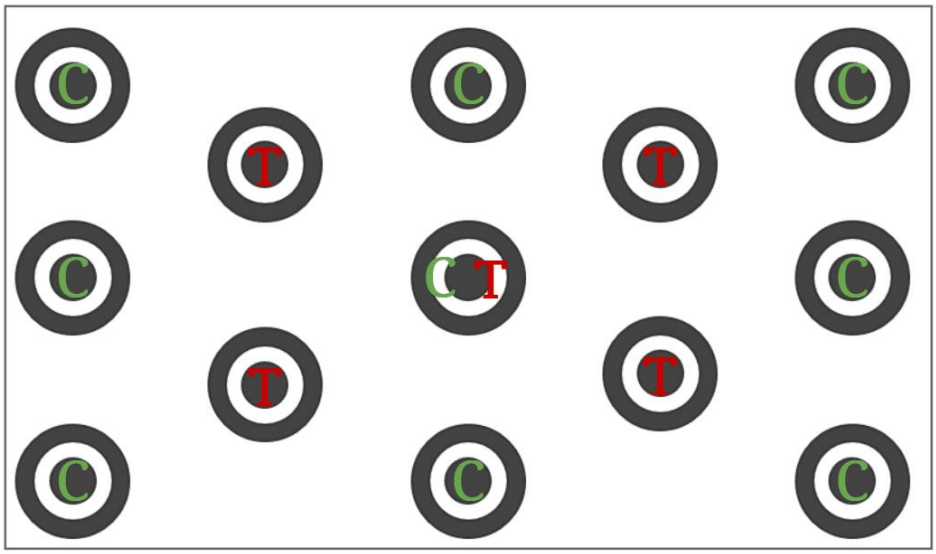


Figure 2.32 Calibration marker pattern to be shown to user during the calibration procedure: 'C' markers for fitting the mapping function and 'T' markers for calculation of calibration angular accuracy in accordance with the mapping coefficients, as cocentric cycles without latters

$$A = \begin{bmatrix} 1 & P_{x(1)} & P_{y(1)} & {P_x}^2{}_{(1)} & {P_y}^2{}_{(1)} & P_xP_{y(1)} & {P_x}^2P_{y(1)} & P_x{P_y}^2{}_{(1)} \\ 1 & P_{x(2)} & P_{y(2)} & {P_x}^2{}_{(2)} & {P_y}^2{}_{(2)} & P_xP_{y(2)} & {P_x}^2P_{y(2)} & P_x{P_y}^2{}_{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & P_{x(90)} & P_{y(90)} & {P_x}^2{}_{(90)} & {P_y}^2{}_{(90)} & P_xP_{y(90)} & {P_x}^2P_{y(90)} & P_x{P_y}^2{}_{(90)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & P_{x(i)} & P_{y(i)} & {P_x}^2{}_{(i)} & {P_y}^2{}_{(i)} & P_xP_{y(i)} & {P_x}^2P_{y(i)} & P_x{P_y}^2{}_{(i)} \end{bmatrix} \tag{2.15}$$

The marker detector is essential for detecting the central point of the screen marker and is therefore used to find world coordinates during the fitting of the polynomial equation in calibration. There are three assumptions in the marker detection procedure. First, as depicted in Figure 2.32, the markers are designed as black circles displayed on a white screen. Second, the markers consist of concentric

circles. Third, the size of the markers must be between 50 px and 400 px. Based on these criteria, concentric ellipses of a certain size on a white background are marked as markers. The algorithm steps developed for detecting these markers is presented in Figure 2.33.

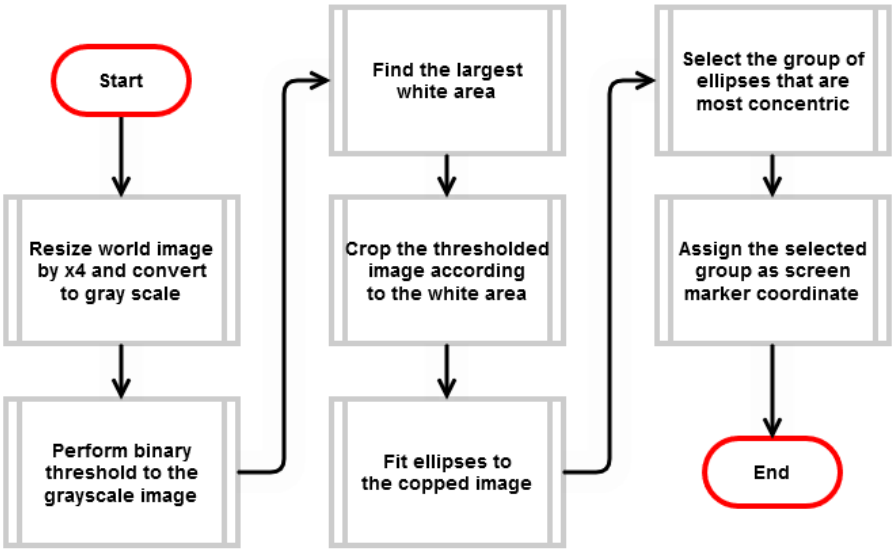


Figure 2.33 Screen marker detection flow: largest white area on thresholded world image is assumed as user's screen and concentric ellipses are found on the screen area

In the marker detection algorithm, the image initially undergoes a downsizing by a factor of four. For instance, if the original resolution of the world camera is 1280×720, the new dimensions would be 640×360. Subsequently, a threshold of 100 is applied to the grayscale version of the image. The thresholded image is in binary form and likely displays the user's screen predominantly in white. This is because during calibration, markers appear over a white screen background. Hence, the widest white area, namely the user's screen, is identified in the thresholded image. Due to the concentric circles of the marker located on the screen, ellipse detection is conducted over this screen area, and the group of ellipses that are most concentric is selected as the marker. As the given downsized image is in a resolution of 640×360, the marker coordinates are also expressed in this frame. To normalize the screen coordinates, the values must be scaled to a range between 0 and 1.

2.4 Software Design & Implementation

The user interface is found within a desktop application written in Python 3.10. The interface is designed as a webpage running on top of the application and has been developed using HTML, CSS, and Javascript. The collecting and display of camera images are performed in Python. Thus, the user interface is a transparent structure overlaid on the camera images. This structure is shown in Figure 2.34.

The interface consists of 6 main sections. The indicator area provides a display related to PETA accuracy metrics and the system's operating speed. When the cursor hovers over a specific area for a period of time, the graph enlarges and becomes more readable. Additionally, the refresh rate per second of the indicator and the total number of measurements a value is dependent on can be set by the user. The Product menu contains links to the developed ETS's help pages, white paper, and website. When the cursor moves to the gradient black area, relevant links appear. The World Camera Area is where images in the direction the user is looking will be displayed, and gaze points visualized. The frequency of World Camera visualization is also user definable. The Left Menu includes frequently used features in a session such as calibration, testing, session recording, and lightweight interface. The lightweight option stops or restarts costly visualization processes in the interface, such as scene image displays. The Plugin Menu is used to adjust ETS features. The menu opens when the cursor moves over the grey stripe. This menu sequentially includes options named Hardware Operation, Eye Camera Staging, User Interface Preferences, Traditional Method, Learning Model, Auto Traditional Conf, PETA Metrics and Correction, Eye Feature Detectors, Surface Tracker, Data Publisher. The Plugin Area indicates where the settings page will open when a plugin icon is clicked. An example visual of the Plugin Menu and Area is provided in Figure 2.35.

The design principle of the interface structure incorporates three key considerations. First, anything that is only occasionally needed over a specific period is discreetly hidden within the interface to a noticeable extent. This design can help balance the cognitive load of the user. Second, anything that is frequently used during a particular period is either prominently displayed or designed to come to the foreground when interacted with. This structure can enhance the user's focus on the task at hand or reduce the number of interface operations. Third, anything that is in

the foreground can be hidden on demand or its display frequency can be reduced. As a result, as the need to check on the task decreases, unnecessary interface display costs can be prevented. These costs are associated with video FPS and real-time visualization processes and may require significant processing capacity for some systems.

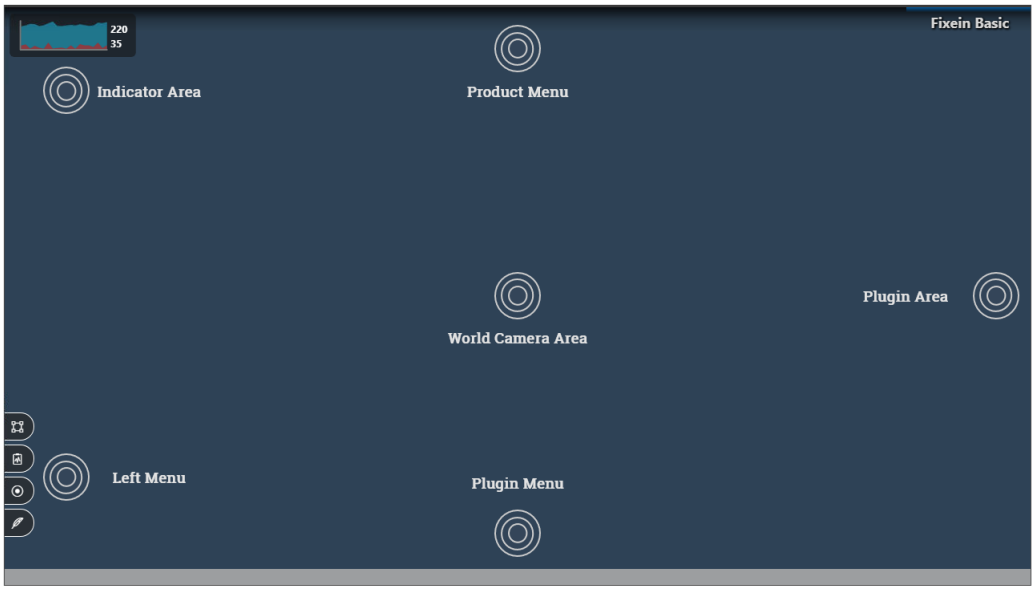


Figure 2.34 User interface's general view. There are totally 6 areas which were shown directly or will be shown via cursor events or movements

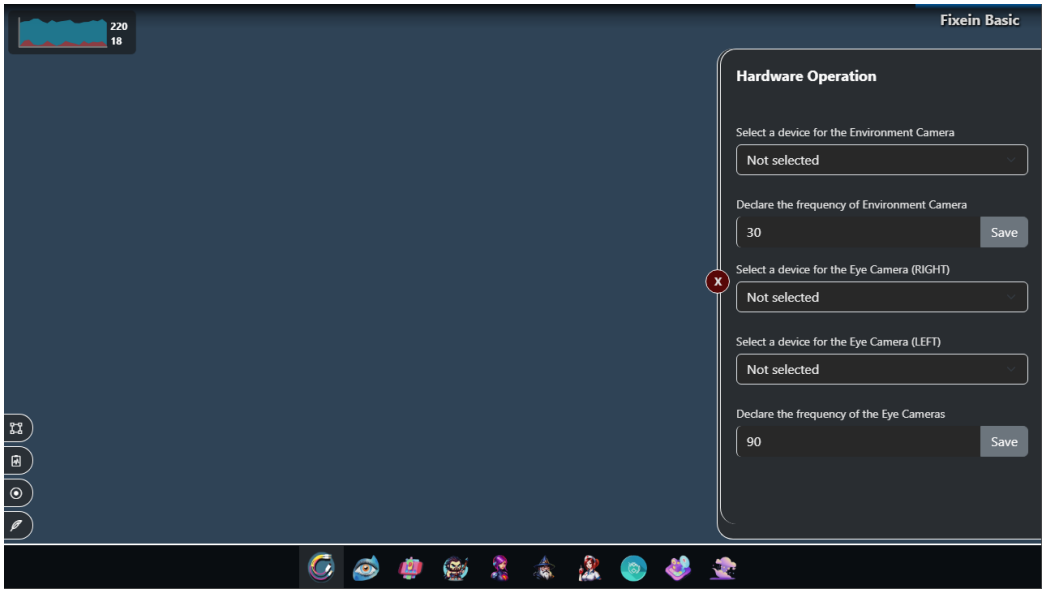


Figure 2.35 User interface's plugin menu (bottom) and plugin settings (right) related with hardware operation of an opened plugin

With these design choices, the screens that users will frequently interact with when the system is in use are shown in Figures 2.36 and 2.37. Figure 2.36 displays the

image collected from the world camera, while Figure 2.37 presents the processed image from the eye camera. On the eye camera display screen, both outputs of the traditional methods and the learning model can be shown. The blue and green curves at the edges of the image represent the minima sequences, and the straight lines denote the thresholding values on the axes. The turquoise rectangle indicates the patch obtained from the traditional method. The red area on the pupil represents the region where pixels with a segmentation result between [0,1] and greater than 0.5 are highlighted in red. Thus, in the example shown, the pupil area is well-detected and accurately highlighted in red. Additionally, the orange circle around the pupil represents the fitted ellipse, and the orange dot in the center indicates the central point of the pupil.

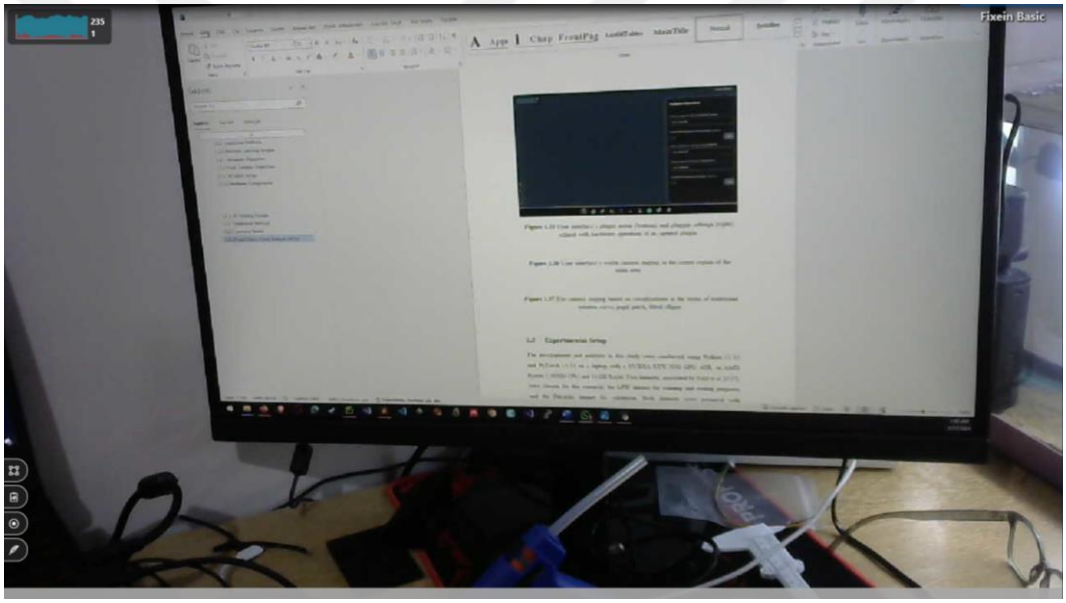


Figure 2.36 User interface's world camera staging in the center region of the main area

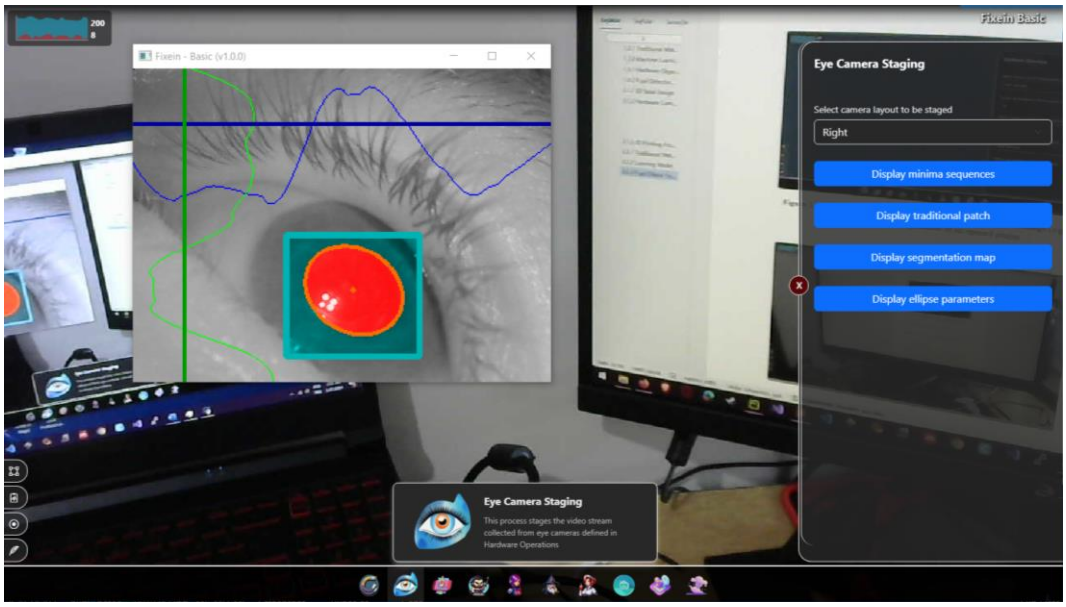


Figure 2.37 Eye camera staging based on visualizations in the terms of traditional minima curve, pupil patch, fitted ellipse

2.5 Experimental Setup

The development and analyses in this study were conducted using Python v3.10 and PyTorch v1.13 on a laptop with a NVIDIA RTX 3050 GPU 4GB, an AMD Ryzen 7 5800H CPU and 16 GB RAM. Two datasets, annotated by Fuhl et al. [127], were chosen for this research: the LPW dataset for training and testing purposes and the Dikablis dataset for validation. Both datasets were prepared with conventional head mounted eye trackers under the IR illumination and with real subjects in different user and environment conditions. While LPW was generated with 66 sessions from 22 subjects and with an eye camera running at 120Hz [124], Dikablis contains various videos, collected at 25 Hz, where the information about which user belongs to which session is unknown. Overall, the most noticeable difference between these two sets lies in the frequency of eye camera and amount of illumination. In LPW, the eye region is well-illuminated, while Dikablis exhibits insufficient illumination. Additionally, both sets predominantly feature content where the pupil is visible. In other words, there are minimal obstacles between the pupil and the camera, resulting in either no or very rare occurrences of blinking during a session. Furthermore, there is often a slight pupil reflection and distinct dark areas aside from the pupil, with this condition being more prevalent in

Dikablis. Therefore, Dikablis proves to be a more challenging dataset for methods relying on the amplitude information. Also, resolutions are 640×480 and 384×288 pixels for LPW and Dikablis respectively. Examples image for LPW and Dikablis were shown Figure 2.38 and Figure 2.39, respectively.



Figure 2.38 Example images from LPW dataset

By the reason of time and physical resource concerns in the study, both dataset resolutions were reduced to 320×240 from 640×480 and 384×288 pixels. Additionally, due to the same concerns, samples that are only below of 100 MB in Dikablish set were used. However, it's worth noting that samples from participants and video pairs with identifiers 1-9, 20-12, 22-7, 22-8, 22-9, 23-3, 23-5, 24-11, 24-5, 24-8, and video pairs of the T series were excluded from the validation set since there were no corresponding annotation files available in the Dikablish dataset. Consequently, 286 videos from Dikablish were analyzed.

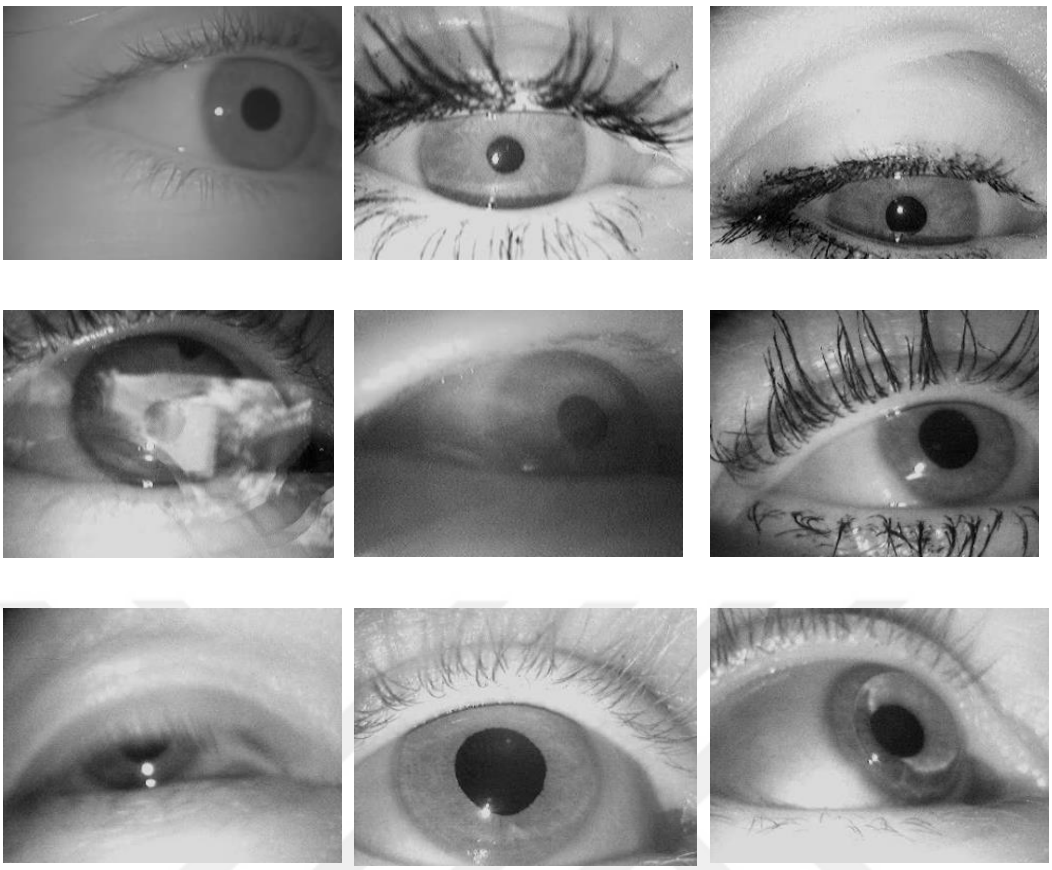


Figure 2.39 Example images from Dikablis dataset

In the training process, videos belonging to the first 5 participants in LPW dataset were allocated for test set, while all videos from the other participants were used for training set. This approach ensured that each sample and participant were exclusively used within only one set. Each training was completed with Adam optimizer with learning rate 1e-3, ReduceRLOnPlateau with patience of 2, dice scoring as loss function and shuffled epochs with batch size of 1 [67]. Augmentation techniques such as horizontal and vertical flips, random rotation, and transpose were applied to the training set, while only horizontal flips were used for the test set. Training was monitored based on dice scores at the end of each epoch, and the process was stopped if there were no changes in dice scores for both the training and test sets over the last 6 epochs. In measurement of pupil detection accuracy, a 5-pixel error rate was performed to accept the results as successfully. In the process of assessment of learning models between each other, we used a LPW variation created as 3 Hz for shorter training durations, not the entire LPW dataset. In other words, the LPW-3Hz set was created by taking the first of every 40 consecutive images. Additionally, LPW variations with different fixed resolutions

have been generated for training of the selected learning model. These variations included resolutions of ×2 (320×120px), ×4 (160×120px), ×6 (107×120px), ×9 (107×80px), ×16 (80×60px), ×25 (64×48px), with each set obtained by cropping the original images to encompass the entire pupil areas. Furthermore, a variation named as LPW-AV containing all images from these variations has been generated. However, the model with AV was trained for only 15 epochs due to its time consumption. For visualization purposes, example samples from the LPW×4 set and LPW×16 set are displayed in Figure 2.40.

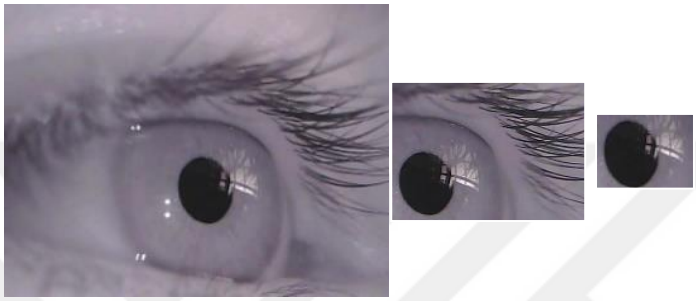


Figure 2.40 Example samples from LPW variation: full resolution, ×4 and ×16 respectively. Source: subject=1, video=1, image=1

As size of these datasets, 130,856 images for each LPW variation are handled without masks- total size for all variations (LPW-AV) is 915,992 images. Also 320,200 images have been used in Dikablish set without masks. Additionally, LPW-3Hz contains about 3,271 images without masks. Thereby, total size of datasets used in this study were generated with 1,239,463 images.

3
RESULTS OF THE PUPIL DETECTOR

3.1 Pupil Patch Creation

The analysis of the present traditional method for LPW dataset are given in Table 3.1 with details. The table presents average measurements in for each video terms of patch size, contained pupil area (CPA) and success. Patch size is the ratio of the generated patch resolution to the original image resolution. Contained pupil area indicates the percentage of the pupil is present in the generated patch. In this analysis, we assume that the ellipse can be fitted if at least 40% of the pupil area is in the patch. Success is obtained according to this assumption. A patch was considered successful if it contained at least 40% of the pupil area and unsuccessful otherwise. Consequently, the success value represents the ratio of successful detections to the total number of detections. To show the maximum positive effects of the traditional method, the PFX, PFY and THM functions used were selected manually. For this, all sessions were analyzed with each function combination and the functions that would give the best performance were selected by the author. Basically, the following equalization has been calculated for all combination and configurations with the highest score have been selected for each different LPW session. Also in the auto configuration mechanism, the same regime was processed to find traditional configurations for the segmentation map generated using full image received every once in a certain period.

$$Score = (100 - PatchSize) \times CPA \tag{3.1}$$

Table 3.1 Analysis of pupil patch creation fulfilled by traditional method in the scope of the present ROM architecture

Subject	Video	PFX	PFY	THM	Patch Size (%)	CPA (%)	Success (%)
1	**1**	FE	FE	TH7	14.51	90.88	100.0
1	**4**	FE	MF	TH7	11.14	95.66	100.0
1	**9**	MF	MD	TH7	3.07	93.08	100.0
2	**4**	ML	ML	TH7	4.19	99.58	100.0

Table 3.1 Analysis of pupil patch creation fulfilled by traditional method in the scope of the present ROM architecture (continued)

2	**10**	MD	ML	TH7	7.65	100.00	100.0
2	**13**	ML	ML	TH7	2.39	99.72	100.0
3	**16**	MD	MF	TH1	44.19	95.38	92.5
3	**19**	FE	ML	TH4	35.67	95.40	100.0
3	**21**	MF	FE	TH6	8.45	95.56	100.0
4	**1**	FE	ML	TH5	31.27	99.54	100.0
4	**2**	MF	ML	TH7	3.99	92.16	99.5
4	**12**	FE	ML	TH7	17.51	94.29	99.7
5	**6**	MD	FE	TH7	5.88	94.38	100.0
5	**10**	ML	MD	TH5	14.45	97.54	100.0
5	**11**	LD	MD	TH6	7.58	99.50	100.0
6	**2**	MD	ML	TH7	4.22	95.77	100.0
6	**5**	MF	ML	TH7	3.76	96.72	99.8
6	**13**	MD	MD	TH7	3.86	98.68	100.0
7	**15**	FE	FE	TH6	26.53	97.70	95.6
7	**18**	FE	MD	TH7	5.29	96.61	100.0
7	**21**	FE	MD	TH7	10.46	99.97	100.0
8	**2**	MF	ML	TH7	3.09	94.88	100.0
8	**7**	LD	ML	TH7	8.42	95.54	100.0
8	**9**	FE	FE	TH5	19.73	99.56	100.0
9	**16**	ML	ML	TH7	8.44	96.81	100.0
9	**17**	ML	ML	TH8	3.09	96.58	100.0
9	**18**	ML	ML	TH8	3.65	97.53	100.0
10	**1**	MF	ML	TH7	4.48	97.08	100.0
10	**8**	MF	ML	TH7	8.47	96.28	100.0
10	**11**	FE	FE	TH3	46.00	99.76	99.9
11	**2**	FE	FE	TH7	12.16	92.98	100.0

Table 3.1 Analysis of pupil patch creation fulfilled by traditional method in the scope of the present ROM architecture (continued)

11	**7**	MF	ML	TH7	10.34	92.44	100.0
11	**13**	MF	ML	TH4	30.66	99.06	100.0
12	**1**	LD	ML	TH7	6.89	93.66	99.4
12	**2**	ML	LD	TH7	3.44	91.83	100.0
12	**9**	MF	FE	TH7	6.58	94.10	100.0
13	**1**	MF	FE	TH5	12.77	99.32	100.0
13	**2**	ML	ML	TH7	7.56	94.38	100.0
13	**9**	MD	ML	TH1	22.22	99.96	99.9
14	**10**	FE	FE	TH7	18.69	92.31	100.0
14	**17**	MF	FE	TH3	27.68	99.54	99.3
14	**22**	FE	FE	TH7	17.61	90.81	100.0
15	**1**	ML	ML	TH7	5.88	97.08	100.0
15	**2**	ML	ML	TH7	7.09	99.08	100.0
15	**7**	ML	ML	TH7	5.29	98.61	100.0
16	**1**	ML	ML	TH7	4.81	99.83	100.0
16	**2**	ML	ML	TH7	3.87	99.94	100.0
16	**13**	ML	ML	TH7	4.44	100.00	100.0
17	**3**	MF	ML	TH4	22.08	99.68	100.0
17	**5**	MF	ML	TH6	17.43	99.02	99.5
17	**12**	FE	MF	TH4	44.55	99.17	100.0
18	**2**	MF	FE	TH7	5.62	91.46	100.0
18	**7**	MF	ML	TH7	8.98	95.19	100.0
18	**11**	FE	ML	TH5	21.73	100.00	100.0
19	**2**	ML	ML	TH7	5.47	83.10	100.0
19	**3**	ML	ML	TH7	3.39	75.79	100.0
19	**6**	MF	ML	TH4	15.58	99.39	99.4
20	**3**	MF	LD	TH7	2.82	91.04	100.0

Table 3.1 Analysis of pupil patch creation fulfilled by traditional method in the scope of the present ROM architecture (continued)

20	**4**	ML	ML	TH7	5.72	95.19	100.0
20	**7**	FE	ML	TH8	3.23	93.32	100.0
21	**4**	LD	MD	TH7	3.26	96.84	100.0
21	**11**	MD	ML	TH7	4.34	91.64	100.0
21	**12**	FE	ML	TH8	2.13	83.12	100.0
22	**1**	MF	MF	TH1	34.80	92.34	91.3
22	**2**	MF	MF	TH1	43.98	94.58	91.3
22	**17**	MF	FE	TH2	36.84	86.16	82.6

On average of all videos belongs to LPW dataset, the patch resolution was 7.66 times smaller than the original image resolution, containing 95.51% of the total pupil area, and achieving a patch generation success rate of 99.23%. However, in the next analyses, we will use the average patch size as 6×, not 7.66×, for ease of calculation. In addition, the Cumulative Distribution Function (CDF) curve of the average pupil patch percentages calculated on a video basis is given in Figure x. Accordingly, although the average patch percentage value is 13.05% and the average magnification factor is 7.66x, within the scope of the CDF, half of the videos have been reduced to a percentage of 7.58% or smaller. Thus, for half of the videos, the magnification factor is equal to or greater than x13.2. On the other hand, the probability related to the intuitive 20% pupil size applied when determining the window size of the Savitzky-Golay filter is also shown in the graph. According to this, a 20% window size on each axis results in a patch that is 4% of the total image. Accordingly, in 23% of the images analyzed within the scope of LPW, a patch size of 4% or smaller, and thus a magnification factor of x25 or larger, has been achieved. The maximum and minimum magnification factors obtained within the scope of LPW are x46.9 and x2.2, respectively. However, the accumulation in the

distribution generally occurs at patch sizes of 2% and 8%, where the graph accelerates rapidly.

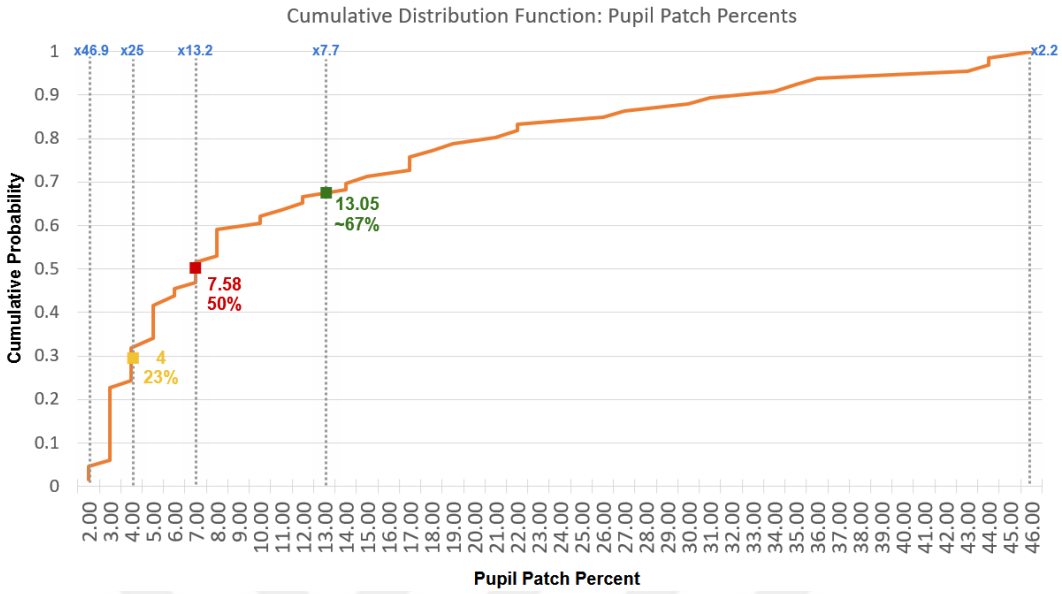


Figure 3.1 Cumulative distribution function for average pupil patch percents of the traditional method proposed in the scope of ROM

3.2 Learning Model Comparison

The examination of the selected learning models within the scope of the study is presented in Table 3.2 for dice metrics, in Table 3.3 for GPU performance measurements, and Table 3.4 for CPU performances. Generally, the models produced similar dice scores for the LPW-3Hz dataset. The source of this similarity is that the LPW dataset contains pupil information that can be well detected by a learning model. On the other hand, small differences in scores distinguish the models from each other. This difference arises from cases that are relatively less common in the LPW dataset. Examples of such cases include blinking, strong reflections, and a large angle of the gaze vector relative to the eye camera. Therefore, small improvements in scores indicate that the model is more robust in inadequate pupil information. Hence, UNet++, UNet++-S, TransUNet, and TransUNet-S models are the best-performing models in terms of both dice and test dice scores. Therefore, it can be said that any of these four models can be used for pupil detection.

Table 3.2 Dice metric comparison of selected learning models to be used for pupil segmentation. Training and test have been fulfilled to LPW-3Hz dataset variation

Learning Model	Param (M)	Dice	Loss	Test Dice
UNet [128]	17.2	0.95	0.032	0.842
UNet-S	1.9	0.954	0.029	0.843
UNet++ [129]	9.1	0.958	0.025	0.868
UNet++-S	2.2	0.956	0.027	0.86
SegNet-VGG-S	0.3	0.933	0.047	0.763
TransUNet [130]	105.1	0.954	0.028	0.876
TransUNet-S	3.2	0.952	0.03	0.867
DeepLabv3-RN-50 [131]	39.6	0.943	0.053	0.838
DeepLabv3-MNv3-L [131]	11	0.902	0.104	0.78
PPMobileSeg-Tiny [132]	0.6	0.899	0.106	0.778
PPMobileSeg-S	0.1	0.889	0.111	0.764
CCSGD-RN-34 [133]	22	0.943	0.035	0.872
CCSGD-RN-34-S	1.4	0.946	0.033	0.868
EGEUNet [134]	0.05	0.93	0.042	0.808

However, another goal of this part of the thesis is to aim for a real-time (at 120 Hz and above) and less resource-consuming system when working on pupil patches. In accordance with this aspect, Table 3.3 presents the GPU performances in the terms

of inference time and memory requirements for both original resolution (320×240px) and the resolution of the average patch size (107×12px). According to the measurements, UNet-S, UNet++-S, SegNet-VGG-S and CCSGD-RN-34-S perform shortest inference time, while UNet-S, SegNet-VGG-S, TransUNet-S, all PPMobileSeg models, and EGEUnet require less resource area.

Table 3.3 GPU inference time and memory requirement comparison of the selected learning models to be used for pupil segmentation

Model and Device		**Inference Time (ms)**		**Memory (GB)**	
		Full	**×6**	**Full**	**×6**
UNet [128]	GPU	28.42	7.13	1.71	0.38
UNet-S	GPU	11.05	4.25	0.93	0.17
UNet++ [129]	GPU	39.77	10.35	2.64	0.51
UNet++-S	GPU	25.41	5.75	1.88	0.34
SegNet-VGG-S	GPU	15.06	3.16	1.13	0.2
TransUNet [130]	GPU	42.46	26.24	2.07	0.46
TransUNet-S	GPU	14.74	8.09	0.88	0.19
DeepLabv3-RN-50 [131]	GPU	25.34	13.8	1.85	0.38
DeepLabv3-MNv3-L [131]	GPU	11.72	11.07	0.39	0.07
PPMobileSeg-Tiny [132]	GPU	20.36	19.72	0.25	0.04
PPMobileSeg-S	GPU	19.98	19.66	0.1	0.01
CCSGD-RN-34 [133]	GPU	11.73	9.8	0.8	0.2
CCSGD-RN-34-S	GPU	5.76	5.24	0.4	0.08

EGEUNet [134]	GPU	15.12	15.27	0.1	0.01

The learning model part of the thesis will be run in GPU. However, investigating models on only CPU may put an insight off related to sufficiencies of models for devices with less capacity such as mobile phones. CPU analysis includes similar measurements like the GPU analysis. The relevant performance metrics are shown in Figure 3.4. Accordingly, CCSGD-RN-34, CCSGD-RN-34-S, EGEUNet and all PPMobileSeg models perform fairly successfully results while compared others.

Table 3.4 CPU inference time and memory requirement comparison of the selected learning models to be used for pupil segmentation

Model and Device		**Inference Time (ms)**		**Memory (GB)**	
		Full	**×6**	**Full**	**×6**
UNet [128]	CPU	521.38	91.25	1.58	0.26
UNet-S	CPU	204.69	27.94	0.89	0.14
UNet++ [129]	CPU	681	102.53	2.3	0.37
UNet++-S	CPU	391.63	56.48	1.68	0.24
SegNet-VGG-S	CPU	234.65	28.71	1.13	0.19
TransUNet [130]	CPU	948.19	225.96	1.8	0.37
TransUNet-S	CPU	199.14	38.72	0.82	0.16
DeepLabv3-RN-50 [131]	CPU	440.27	96.36	1.54	0.27

DeepLabv3-MNv3-L [131]	CPU	64.39	20.36	0.33	0.06
PPMobileSeg-Tiny [132]	CPU	43.34	17.69	0.27	0.04
PPMobileSeg-S	CPU	21.22	15.51	0.1	0.01
CCSGD-RN-34 [133]	CPU	102.57	30.88	0.6	0.1
CCSGD-RN-34-S	CPU	47.47	11.88	0.3	0.06
EGEUNet [134]	CPU	31.96	17.85	0.1	0.02

In the stage of the learning model phase of the ROM concept, the UNet++-S model was selected after reviewing the relevant analyses. This selection was made because the UNet++-S model has produced successful Dice scores and stands out in terms of resource requirements. This model can operate in approximately 6 milliseconds on a GPU using pupil patches and requires 340 MB of space, achieving 95.6% success in training and 86% in testing. Consequently, the UNet++-S model has been used in all other analyses as the learning model. However, it can be said that the CCSGD-RN-34-S model is more prominent in CPU usage due to the balance between speed and performance.

3.3 ROM & PETA Analysis

Table 3.5 provides training sessions of the UNet++-S model with different LPW variations and their corresponding test measurements. According to this, there is a significant correlation between the resolution in the training set and the test dice score of UNet++-S. In general, as the difference between the resolution in the training set and the test input resolution increased, the dice score decreased. On the other hand, LPW-AV has been able to produce successful results in all variations since it was trained with all training variations. Additionally, training UNet++-S with all variations (LPW-AV) has also increased the test dice score of each variation except x6. The inability to significantly increase the x6 test dice score may be related to the fact that the training with LPW-AV was limited to 15 epochs.

However, this situation was not considered, and the UNet++-S model trained with LPW-AV was used in the continuation of the study.

Table 3.5 UNet++-S model training with different variations of LPW dataset to display the effect of the input size to the prediction accuracy

UNet++-S								
Dataset Variation	**LPW-Full**	**LPW ×2**	**LPW ×4**	**LPW ×6**	**LPW ×9**	**LPW ×16**	**LPW ×25**	**LPW-AV**
Dice	0.962	0.96	0.961	0.961	0.96	0.961	0.96	0.963
Loss	0.019	0.02	0.019	0.019	0.019	0.019	0.019	0.018
Test Dice-x1	0.892	0.855	0.798	0.776	0.684	0.553	0.427	**0.896**
Test Dice-x2	0.878	0.894	0.834	0.883	0.799	0.726	0.643	**0.901**
Test Dice-x4	0.775	0.81	0.903	0.785	0.865	0.833	0.728	**0.906**
Test Dice-x6	0.778	0.873	0.816	**0.909**	0.817	0.816	0.739	0.899
Test Dice-x9	0.625	0.75	0.863	0.836	0.896	0.889	0.827	**0.908**
Test Dice-x16	0.353	0.6	0.799	0.832	0.881	0.898	0.854	**0.909**
Test Dice-x25	0.077	0.416	0.597	0.775	0.767	0.854	0.893	**0.909**
Avg Test Dice	**0.625**	**0.742**	**0.801**	**0.828**	**0.815**	**0.795**	**0.730**	**0.904**

Table 3.6 Accuracy metrics for the combination the present pupil detection components. Ellipse error consists of 3 sections: center point error for x and y axes, size error for width and height, ellipse angle error

LM	**TM**	**PETA**	**AUTO**	**L-5px (%)**	**D-5px (%)**	**L-Ellipse Error (Average)**	**D-Ellipse Error (Average)**
	✗	✗	✗	96.18	92.53	(1.35, 1.30), (1.25, 1.53), 6.65	(4.64, 2.60), (2.97, 3.43), 13.67
	✗		✗	96.23	92.56	(1.32, 1.28), (1.24, 1.52), 6.63	(4.60, 2.58), (2.96, 3.42), 13.66
		✗	✗	95.11	92.60	(2.18, 2.03), (1.64, 2.16), 8.03	(4.64, 2.96), (2.52, 3.41), 13.44

		X	**100 ms**	93.41	92.19	(3.29, 2.61), (1.86, 2.64), 8.76	(5.91, 3.27), (2.60, 3.44), 13.87
		X	**333 ms**	91.67	91.34	(4.64, 3.53), (2.29, 3.33), 9.82	(6.54, 3.71), (2.74, 3.64), 14.42
			X	95.15	92.71	(2.10, 1.99), (1.63, 2.15), 8.02	(4.51, 2.89), (2.50, 3.38), 13.40
			100 ms	93.66	92.30	(3.10, 2.48), (1.81, 2.56), 8.61	(5.76, 3.19), (2.57, 3.41), 13.81
			333 ms	92.01	91.38	(4.44, 3.39), (2.19, 3.20), 9.56	(6.43, 3.70), (2.73, 3.62), 14.39

In the next analysis, the impact of the study components on accuracy was assessed, and relevant measurements are presented in Table 3.6. While L and D refer to LPW and Dikablis datasets, LM and TM refer to learning model and traditional method, respectively. AUTO denotes the automatic traditional method configuration mechanism. In addition, measurements were made for two alternatives of AUTO that run every 100ms and 333ms. The results were calculated based on a 5-pixel error and average ellipse parameter error. Ellipse error consists of pixel errors in center point (x, y), size (major, minor) and angle error, respectively. During the analysis, it was seen that the traditional method patch area may not be at a sufficient level for the learning model and ellipse fitting operations. Therefore, the patch taken from the traditional model was analyzed by expanding it 20px from the bottom and left, and 10px from the top and right edges for LPW. If an edge does not have enough space to be expanded, the corresponding edge is expanded to the maximum possible space. As a result of LPW analyses, the extended traditional model patch has a resolution of 19.50% compared to the original image, corresponding to an average resolution of ×5 for the learning model. For Dikablis, average patch size was 10.27% and CPA amount was 94.60%, by using traditional configurations in accordance with scores of each combination. In measurements made with LPW for the average patch size to be similar, the patch area was expanded by 30 pixels from

the bottom and 22 pixels from the left in the vertical direction. Therefore, the results were approximately taken as having x5 resolution for Dikablis as well. Furthermore, since the Dikablish dataset's preparation frequency of 25 Hz, the upper band value was set at 52 pixels for the Dikablis, according to the calculation explained in Section 2.2.3.

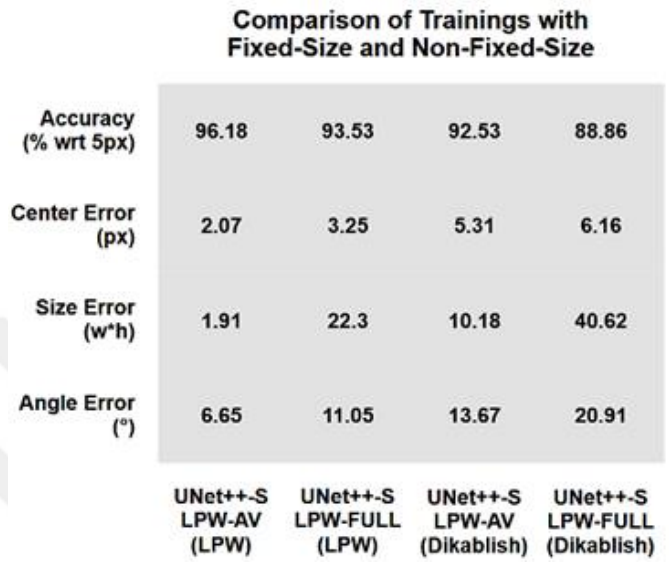


Figure 3.2 Comparison of models trained with LPW-Full and LPW-AV. Accordingly, LPW-AV, containing different resolution of same images, has a data augmentation affect for performance metrics

Also, in the analysis of LPW and Dikablish datasets, three procedures were applied. Firstly, no binarization operation was applied to the segmentation maps when results were obtained. In other words, the maps obtained from the model were used directly for the analyses. Secondly, the model trained with LPW-AV was used in all analyses, including ones without traditional method. Normally, one might consider using the model trained with LPW-Full for the analyses without the traditional method because LPW-AV was generated for models predicting in accordance with different input resolutions. However, as shown in Figure 3.2, we discovered that LPW-AV has a data augmentation effect. Training the model with different resolutions, even though it operates at a fixed resolution, results in a significant improvement in accuracy. In the figure, four performance metrics are shown. Accuracy represents successful detections based on a 5-px error value. Center error is the average error be-tween the estimated and annotated ellipse center coordinates. Size error is presented as the product of the average errors in width and

height between the estimated and annotated ellipses. Angle error represents the average angular errors between the estimated and annotated ellipses. Lastly, if the width or height values of the ellipse parameters calculated in Dikablis are smaller than 5 pixels, the previously calculated ellipse was used. With the use of the 5-px size regime, the UNet++-S trained with LPW-AV achieved an accuracy rate of 92.53%, while in the case where it was not used, the accuracy rate was 92.47%. Although the success difference is quite low, a size-dependent adjustment in models trained with different resolutions, such as LPW-AV, may have the potential to increase success to a certain extent. Consequently, according to Table 5, over 92% was achieved for both LPW and Dikablis datasets. In a situation where traditional model configurations were determined manually, only a 1% 5-px loss of accuracy was observed compared to a flow where only the learning model was used, for LPW dataset. Furthermore, it is observed that the automatic configuration mechanism has relatively little impact to the accuracies. The patch size obtained is the same as the patch rate obtained from AUTO, but the success rate obtained from AUTO may be 2.5% lower. Especially, the fact that the success does not significantly decrease with the use of this mechanism was crucial for the usability of this mechanism. Thereby, this situation may reduce the need for the user to adjust on the traditional method.

Table 3.7 Average GPU prediction times of UNet++-S for input resolutions at different subsampling rates in millisecond

×1 (Full)	**×2**	**×4**	**×5**	**×6**	**×9**	**×16**	**×25**
25.41	13.25	7.51	6.12	5.75	5.62	5.09	4.90

Table 3.8 Average GPU execution times of the study sub-components except for the learning model in millisecond

Code	**Sub-component**	**Time (ms)**	**Code**	**Sub-component**	**Time (ms)**
S1	Finding minima sequences	0.10	**S6**	Fitting ellipse to a map with x5 resolution	0.04

S2	Smoothing with Savitzky-Golay filter	0.72	**S7**	Selecting traditional configuration automatically	8.88
S3	Selecting patch with PFX, PFY, THM functions	0.12	**S8**	Converting matrix to tensor for images with x5	0.04
S4	Calculating entropy & intensity	0.25	**S9**	Converting tensor to matrix for images with x5	0.01
S5	Detecting edges with Sobel filter for 10% area	0.13			

Table 3.7 presents the prediction time of the UNet++-S model at different resolutions. Specifically, the selected model exhibits runtimes of 25.41 milliseconds at full resolution, 5.57 milliseconds with the average patch size (×6), and 6.12 milliseconds with the extended average patch size (×5). Table 3.8 displays the GPU execution times for various subcomponents used in the study. Apart from these, the running time of some mathematical operations and control expressions that will require less compu-ting power has been ignored. Figure 3.3 provides a graphical representation of the execution times for the proposed pipeline on the GPU. Notably, the pupil detection process requires approximately 7.5 milliseconds, while the auto-configuration mecha-nism, which operates on a separate thread, requires 35 milliseconds. Consequently, it can be inferred that pupil detection can be performed at a rate exceeding 120 Hz.



Figure 3.3 Execution times of the pipeline working with two threads for detection and auto configuration processes

3.4 Pupil Detection Comparison

In this section, the performance of the proposed traditional method and learning model in the ROM-PETA concept is compared with previous studies. For this purpose, the comparison includes learning models such as Leyes, along with the study of Wan et al. [128, 135-136]. Other potentially successful edge-based and amplitude-edge based methods could not be included as their code have not been shared before in a public repository. Also, Pistol has not been analyzed since, along with the pupil, it can make many other detections such as iris and sclera. Additionally, the analyses include segmentation models trained with LPW-3Hz and seen to be successful in Experiment section. These models were trained according to the standards specified in the Experimental Setup section with LPW-Full.

In the comparison, LEyes' model initially trained with EDS2019 synthetic images yielded unexpectedly negative results, with 69.20% accuracy for a 5px error. To further explore the potential success of Leyes in pupil detection on real images, the model was retrained on LPW. Our Experiment section findings indicate that an encoder-decoder model trained with fixed resolutions may not accurately detect inputs of different resolutions. Therefore, training LEyes with LPW-Full (320x240), as desired to work at a resolution of 128×128, may not accurately reflect the model's performance. Following these considerations, a new variation of LPW has been created with a resolution of 128×128, with the pupil position located at the center of the cropped area. Except for resolutions, the regime specified in the Experimental Setup section was applied during training. The PuRe and central cropping approach, as applied in LEyes' evaluation, were followed as mentioned in the study published by Byrne at al.88. It's noteworthy that another contribution of LEyes is its ability to generate successful results from a small number of synthetic images created based on parameters determined considering environmental conditions and imaging sensor characteristics during an experimental session. The unexpected result with the EDS2019 model might be due to differences in these parameters. Therefore, due to the differences in dynamics between LPW and Dikablis, we didn’t use LEyes' synthetic image generation approach when retraining, and directly trained the model with the new LPW variation.



Figure 3.4 Comparison of our study and selected studies based on full or binary segmentation maps

The comparison results were presented in Figure 3.4. There are two different analyses: full map and binary map. In full map, segmentation map received the relevant model was used to analyses directly. In binary map, values less than 126 in the segmentation map have been set to 0, while values greater than or equal to 126 have been set to 255. Additionally, there are two different versions of LEyes. In the original LEyes analysis, as mentioned in the literature, the model can make predictions twice for a single image. Suggested by us, LEyes-×1 produces single prediction, and if the cal-culated ellipse center is too close to the edges of the patch, it does not make a second prediction. Models prefixed with "ours" were all been

trained with LPW-AV, while models other than LEyes were trained with LPW-Full. Since the potential for success improvement with LPW-AV was presented in Figure 3.1, this complicates the comparison between "ours" models and those trained with LPW-Full. However, LPW-AV is currently the only alternative for making predictions with variable patches and is a necessity for our proposed models. Therefore, LPW-AV was not used in models not designed to work with variable resolutions. Also, in the Dikablish dataset with 320,200 images, while the LEyes model made 107,907 repetitions in the full map, it made 73,539 repetitions in the binary map. The LEyes model works alone in 8.17 milliseconds, while PuRe works in 3.2 milliseconds. Therefore, the inference time of LEyes was calculated considering both the prediction times of PuRe and the model, as well as the number of repetitions it made on average.

The results reveal that the choice between using a full or binary map is model-dependent. Although the full map is more efficient in some models, especially in our study and LEyes, using the binary map is a better choice when making predictions. Additionally, both our study and the LEyes models work much more efficiently compared to a directly used segmentation models. This indicates that a learning model with a traditional method can operate more successfully with less unnecessary information. Additionally, it can be argued that it may not be necessary for the LEyes model to make repetitions in predictions. Thus, with a slightly shorter inference time, an increase of approximately 17 FPS has been achieved. However, for better results in the repetition-free version of LEyes, using the binary map might be more favorable. In the comparison between our study and LEyes, it is observed that LEyes works slightly more accurately. This level of accuracy holds true for center error, size error, and angle error for the repetitive version of LEyes. On the other hand, our proposed study operates at almost twice the speed. Especially in slightly more controlled experimental environments, it can be said that the small success difference between our study and LEyes will completely close. However, our ap-proximately two times faster prediction speed will remain constant. In the case where PETA metrics and the correction approach are added to our study, predictions can still be made at over 120 FPS, while performance metrics have improved only slightly. Additionally, our automatic configuration mechanism operates at almost similar levels of success. Thus, it has the potential to eliminate

the need for users to make manual settings. When examining the only traditional method in our comparison, as expected, it operates very fast but is much less successful compared to learn-ing models. In general, learning models, on the contrary, are more accurate but slower. This situation more clearly demonstrates the trade-off between learning models and traditional methods.

4
EXPERIMENTS ON THE PRESENT ETS

4.1 Hardware Design

After the assembly of the designed 3D prints and hardware components, eye images of the quality shown in Figure 4.1 were obtained through the eye camera. However, the presence of reflections on the pupil generally increases the rate of incorrect detection, especially in images with insufficient pupil information. Therefore, in situations where the pupil information might be insufficient, the expected benefit

from a deep learning model may not be achieved. To ensure that the learning model performs as expected and resembles the image quality in the training set, efforts were made to eliminate reflections on the pupil. The IR LED structure was temporarily modified to reduce pupil reflections. The IR LEDs used in this study emit at a lower angle, which may not adequately illuminate the eye area. Although increasing the number and emission angles of the LEDs while maintaining their current positions could be a potential solution, this process requires a more professional production setup.



Figure 4.1 Eye image examples collected with the desighed IR LED components



Figure 4.2 Eye image examples collected with the external IR LED structure

Consequently, the existing IR LEDs were temporarily deactivated, and an external IR LED support was added. The temporary IR LED support has the same characteristics as the currently used IR LEDs, but there are four of them, and they can be positioned independently of the eye camera. With the addition of the external IR LEDs, the eye images obtained were as shown in Figure 4.2. According to the image, pupil reflections were significantly reduced, and the pupil information was more prominently highlighted. This improvement intuitively suggests that pupil detection can be performed more accurately. The likely reason for this accuracy is that the collected images resemble the images in the training set, as exemplified in Figure 2.38.

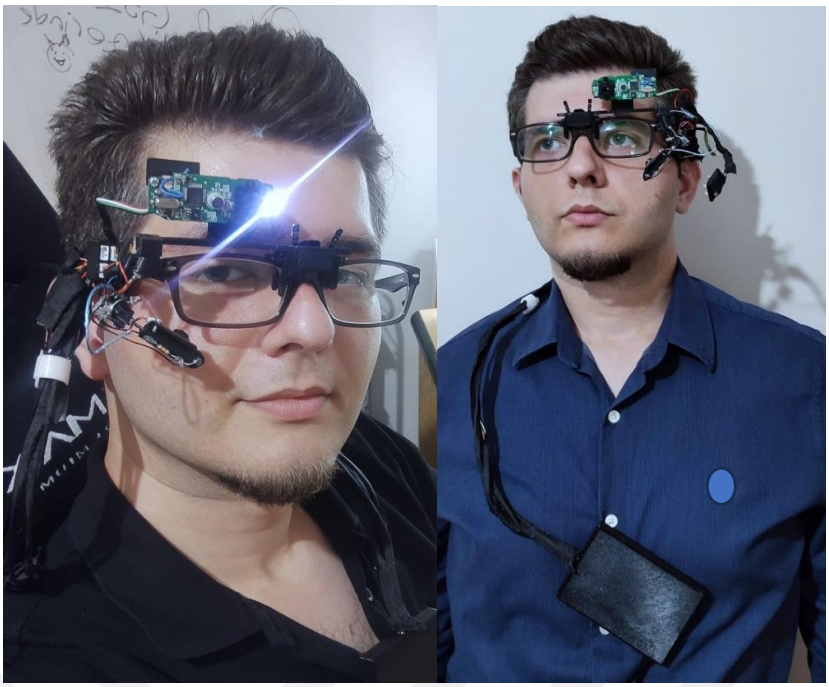


Figure 4.3 Final system appearance: view as described in Methodology section (left), view with external IR LED apparatus (right)

As a result of the hardware design, two different designs were produced: one with external IR LEDs and the main design. In Figure 4.3, the main design is on the left, and the version with external IR LEDs is on the right. The external IR LED apparatus can be repositioned with light force, making it adjustable according to usage. In Figure 4.3, it is shown slightly extended outward to distinguish it from the overall system.

Finally, Figure 4.4 provides the side and rear views of the version developed with the necessary external IR LED apparatus. The cameras are positioned monocularly on the left side of the face, with the connection ports also located on the left. The hub cable wraps around the neck and extends towards the hub from the right side of the face. The advantage of this setup is that the cable between the hub and the socket holder exerts pressure primarily on the shoulder rather than on the parts of the glasses that touch the face.



Figure 4.4 Final system apparance: view with headset frame (left), side and rear view (right and bottom)

4.2 System Spesifications

Table 4.1 Weights of 3D prints and/or hardware component groups

Hardware Component or Group Name	Weight
Main Frame with Clips	11 gr
Eye Camera	1.5 gr
Connectors between Main Frame and Eye Camera	4.5 gr
World Camera	4.5 gr
Connectors between Main Frame and World Camera	2.5 gr
Socket Holder with Sockets	8 gr
System Total Weight	32 gr
Headset Apparatus	14.5 gr
System Total Weight with Headset Frame	46.5 gr
USB Hub with Holders	28 gr
Full System Weight without Cables	74.5 gr
External IR LEDs as a Temporal Component	3.5 gr

Table 4.1 presents the weights of the developed hardware. Accordingly, when the system operates in monocular mode, it has a total weight of 32 grams. Assuming that plastic corrective glasses weigh approximately 30 grams, users would experience a load of about 62 grams on their nose and ears when using the system with their own glasses. Alternatively, the weight of the headset frame is about half that of an average pair of plastic glasses. Using the system with the headset frame results in a load of approximately 46 grams on the nose and ears. Additionally, the total weight of the system is approximately 74 grams. However, the impact of cables was not considered in these and other weight measurements. The cabling is not expected to impose a significant load on the facial area, as the cables between the main frame and the socket holder are very light and short. Although the amount

and weight of the cable between the socket holder and the USB Hub are relatively higher, the weight is borne on the shoulders or the clothing to which the USB Hub is clipped, rather than the face. Furthermore, the duration for which the measured system weights would not cause discomfort to users was not measured. However, the authors believe that a system weighing approximately 50 grams would cause discomfort in the facial area after about 3 hours.

In the developed system, the gaze production rate can be up to 120 Hz due to the use of 120 Hz cameras. However, it is also possible to operate the system at other frequencies, such as 30, 60, or 90 Hz. For example, a performance measurement of the system operating at 30 Hz is provided in Figure 4.5. This measurement aims to show the operating speeds in milliseconds. On the X-axis, the average of every 1000 data points is taken and represented as one set. A total of 10 sets, and thus 10,000 data points for each curve, were collected. These curves represent the speeds of the world camera, eye camera, and pupil detection with and without ROM, respectively.

Accordingly, at 30 Hz, the cameras operate at approximately 33 ms as expected, collecting images without delay. The cameras consistently collect images at approximately 33 ms over 10 sets, which also suggests that there is no electromagnetic interference in the data cables with removed shields. Pupil detection with ROM occurs approximately every 35 ms and does not accelerate over time. Thus, it can be seen that the pupil can be detected consistently in normal use. In the version without ROM, the entire images are given to the learning model instead of pupil patches. In this case, due to latency, the system lacks sufficient capacity to process the images, leading to GPU resource exhaustion. This exhaustion occurs in the third set. Consequently, without creating pupil patches, the current system fails to function properly after approximately 3000 images. This situation necessitates operating the system below 30 Hz for normal pupil detection, although this value was not measured within the scope of this thesis.

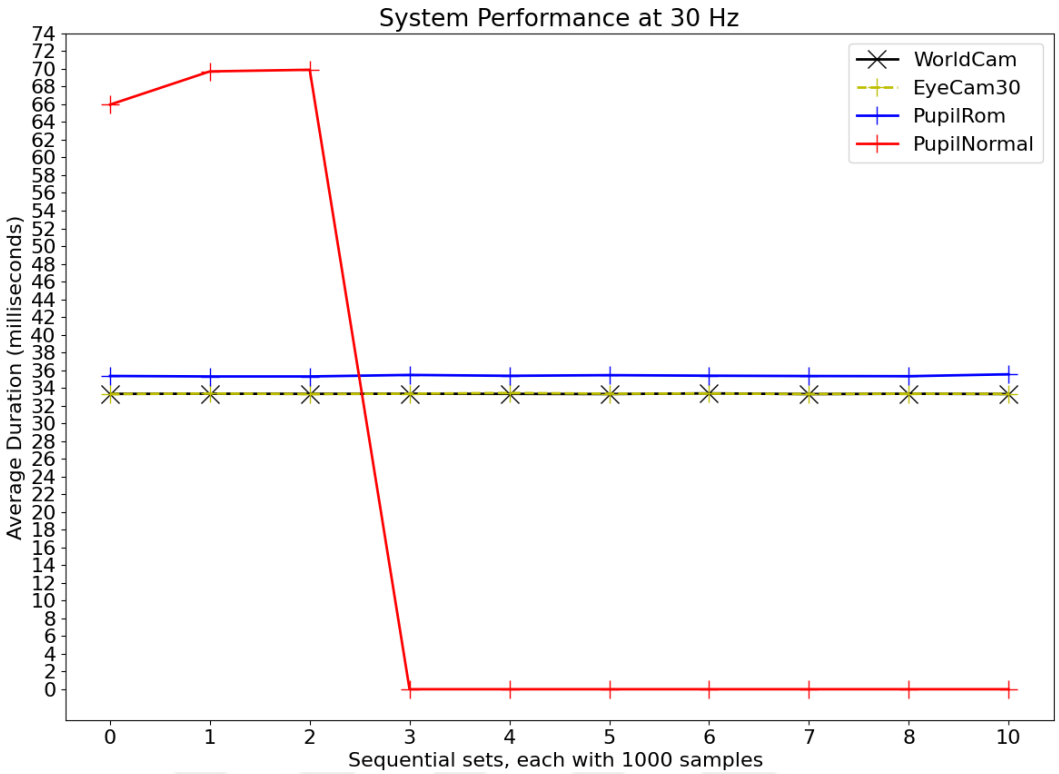


Figure 4.5 Average running duration of main gaze producing components at 30 Hz frequency

The maximum operating frequency of the system is 120 Hz, which is related to the highest speed of the eye camera. Similar curves to those at 30 Hz were obtained for this analysis and are shown in Figure 4.6. In this analysis, the eye camera operates at approximately 8 ms, and the world camera operates at approximately 33 ms. These values are within the expected ranges and do not increase over time. Therefore, it can be said that electromagnetic interference does not create temporal differences for 120 Hz usage. Pupil detection with ROM remains stable at approximately 12 ms without increasing. Thus, the computational load does not fill the GPU resources, and there is no delay during detection. In contrast, the situation is the opposite for normal pupil detection without ROM. Due to the rapid pupil images in the first set showing the first 1000 data points, the GPU resources are immediately consumed, and the system becomes non-functional within the first 1000 images. Consequently, examining both Figure 4.5 and Figure 4.6 reveals the stable performance of ROM in resource utilization through the pupil patch concept.

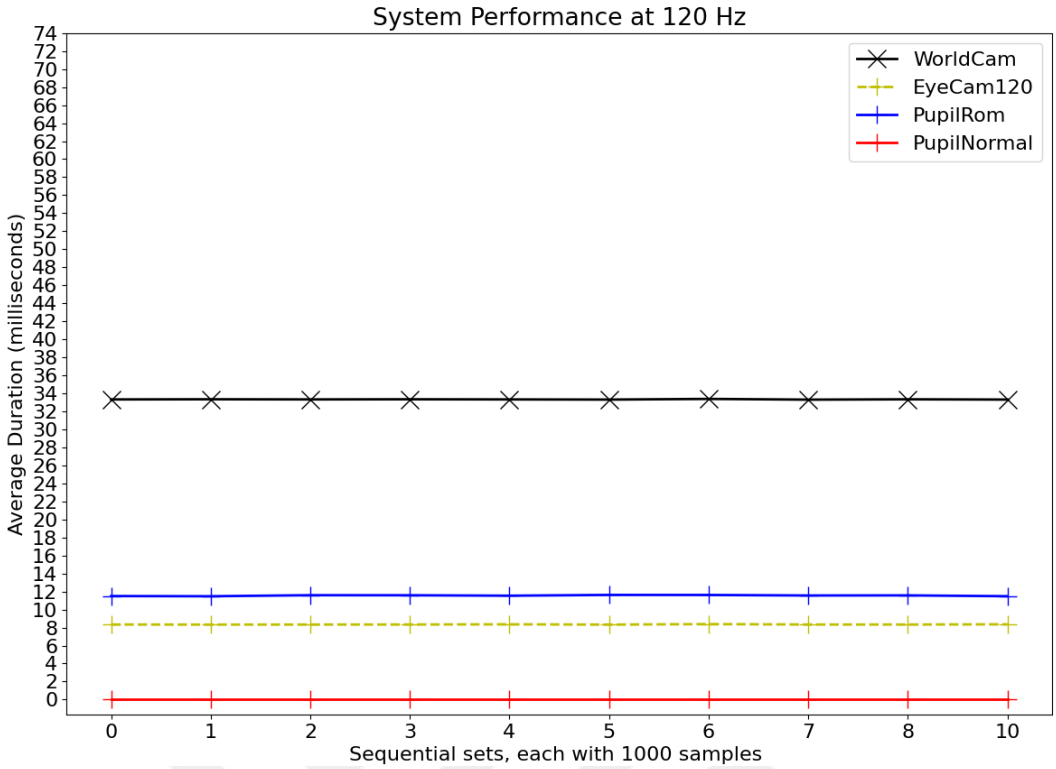


Figure 4.6 Average running duration of main gaze producing components components at 120 Hz frequency

The performance of the system's gaze point accuracy is provided in Table 4.2. To measure performance, both calibration and test results were evaluated. During the data collection process, the system and the calibration surface were kept as stationary as possible, as movement can negatively affect performance in 2D mapping. The calibration screen is 15.6 inches with a resolution of 1920x1080 and is approximately 70 cm away from the world camera. World camera is in 1280x720 resolution settings with 60 diagonal field of view. Based on it, it has 53.13 degree horizontal view and 32.72 degree vertical view. Data was collected by first measuring the calibration performance and then the test performance immediately afterward. After conducting a series of trials, the most successful calibration and test pair was selected and recorded in the table. Similarly, in normal use, a series of calibrations is performed, and the session begins only if the calibration score is sufficient; therefore, only the most successful score is reported instead of all measured scores.

Table 4.2 Angular accuracy based on calibration and test procedure in 2D mapping

2D Gaze Producing Accuracy	Results
Calibration Accuracy (avg, %)	99.55
Calibration Angular Error (avg, º)	0.308
Calibration Pixel Error (avg, px)	29
Test Accuracy (avg, %)	98.68
Test Angular Error (avg, º)	0.839
Test Pixel Error (avg, px)	81

To calculate the accuracy value in the table, the absolute difference between the world coordinates calculated for the given pupil points and the detected marker coordinates in pixels was summed up and averaged by dividing by the total number of observations. Additionally, the calculated value was statistically converted according to the world camera resolution and named as percentage success. In this conversion, both axes of the world camera resolution were expressed in the range of [0-1]. For the calculation of the accuracy value for calibration, the mapping function is first solved, and then world coordinates are generated using the same pupil points with the obtained coefficients. In the test, the score is calculated with the test pupil and test world coordinates using the latest fitted mapping coefficients. According to this, the obtained scores are 99.55% for calibration and 98.68% for the test. For these scores, the calibration error on the world camera is 6 pixels horizontally and 4 pixels vertically, while for the test, it is at the level of 17 pixels horizontally and 10 pixels vertically. The second score in the table, angular error, indicates the average angular deviation of the gaze points. It is obtained by multiplying the respective pixel errors on the world camera with the angular resolution. The Euclidean distance of both horizontal and vertical error amounts is calculated as the system's angular error. The equation for angular resolution used in this calculation is given in Eq 4.1.

$$Angular\ Resolution_H = \frac{FOV_H}{RES_H} = \frac{53.13}{1280} = 0.0415\ degree/pixel$$

$$Angular\ Resolution_V = \frac{FOV_V}{RES_V} = \frac{32.72}{720} = 0.0454\ degree/pixel \quad (4.1)$$

The pixel error value given in the table indicates the pixel error encountered on the screen used for calibration. However, this value is a dependent variable and changes according to the screen resolution. The given scores are calculated for a screen with a 1920x1080 resolution, 70 cm distance, and physical dimensions of 19.5x34x5 cm. For this, the displacement on a surface 70 cm away is calculated from the 0.308 and 0.839 angular errors. The formula in Eq 4.2 is used for this calculation. The obtained scores are 0.37 cm for calibration and 1.03 cm for the test.

$$Linear\ Displacement = \tan(angular\ error) \times 0.70 \quad (4.2)$$

Additionally, the screen's cm/pixel ratio, based on the given horizontal and vertical dimensions, is 0.0177 and 0.0181 cm/pixel, respectively. The pixel equivalent of a 0.37 cm change for calibration is 20.90 pixels horizontally and 20.44 pixels vertically. For the test, it is 58.20 pixels horizontally and 56.91 pixels vertically. The approximate surface pixel error using Euclidean distance will be 29 pixels for calibration and 81 pixels for the test.

4.3 Examples of Pupil Detection Results

In this section, visualized versions of the pupil detection results obtained through the software are shared. The results present all components of the ROM concept over the eye image. These components include the x and y minima sequences curves, threshold lines for x and y, the calculated pupil patch to be sent to the learning model, areas where the learning model's prediction exceeds 50%, shown in red, the fitted ellipse, and the pupil center point. The curves and lines are shown in green and blue according to their axes. Simply put, the potential pupil region is the part below the line in the direction close to the edge. The pupil patch to be sent to the learning model is represented as a turquoise rectangle. The ellipse and center point are shown in orange. Additionally, the influence of PETA is not shown in the results; the focus is solely on ROM. Due to the visualization step of PETA relying on the integrity of approximately 120 images, PETA performance is presented only numerically in the results section.

In the normal use of the system, gazes are usually directed at a narrower angle and close to eye level. Consequently, the pupil region is far from the edges of the eye and can be accurately detected without obstruction. Example detection results related to this are provided in Figure 4.6. Accordingly, the pupil patch and the certainty of the learning model are quite accurate.

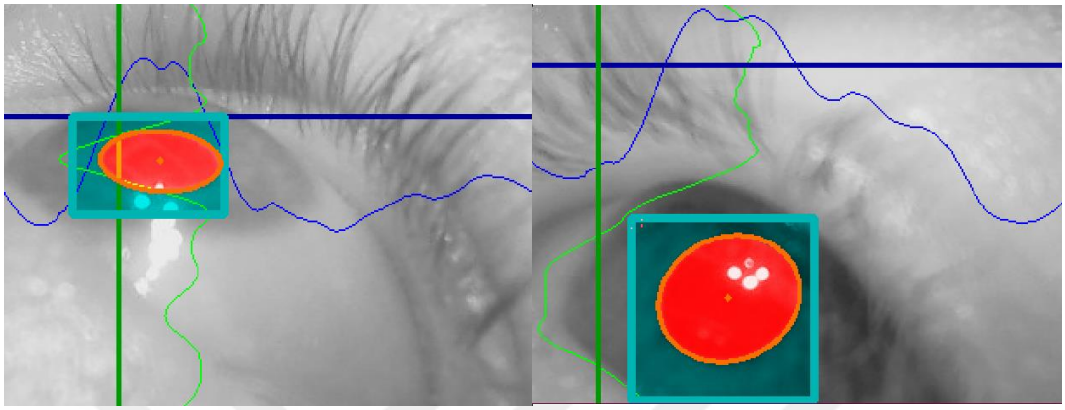


Figure 4.6 Pupil detection results in normal usage. The eye is usually looking straight ahead and the pupil is not at the extreme points

In a different usage scenario, the surface may be significantly mobile, or the user may make various maneuvers during the session. In such cases, the pupil can move closer to the edges of the eye and gradually lose its proper elliptical shape. The deviation of the ellipse's width-to-height ratio from 1 makes detection more difficult. Especially with off-axis camera positioning, this ratio can be significantly distant from 1 at certain pupil angles. In this regard, Figure 4.6 shows the gaze direction as far from eye level as possible in the upward, left, right, and downward directions, respectively. Accordingly, in upward and rightward gazes, where the width-to-height ratio deviates most from 1, the learning model's accuracy value is generally below 50%. However, the ellipse can still be successfully fitted.

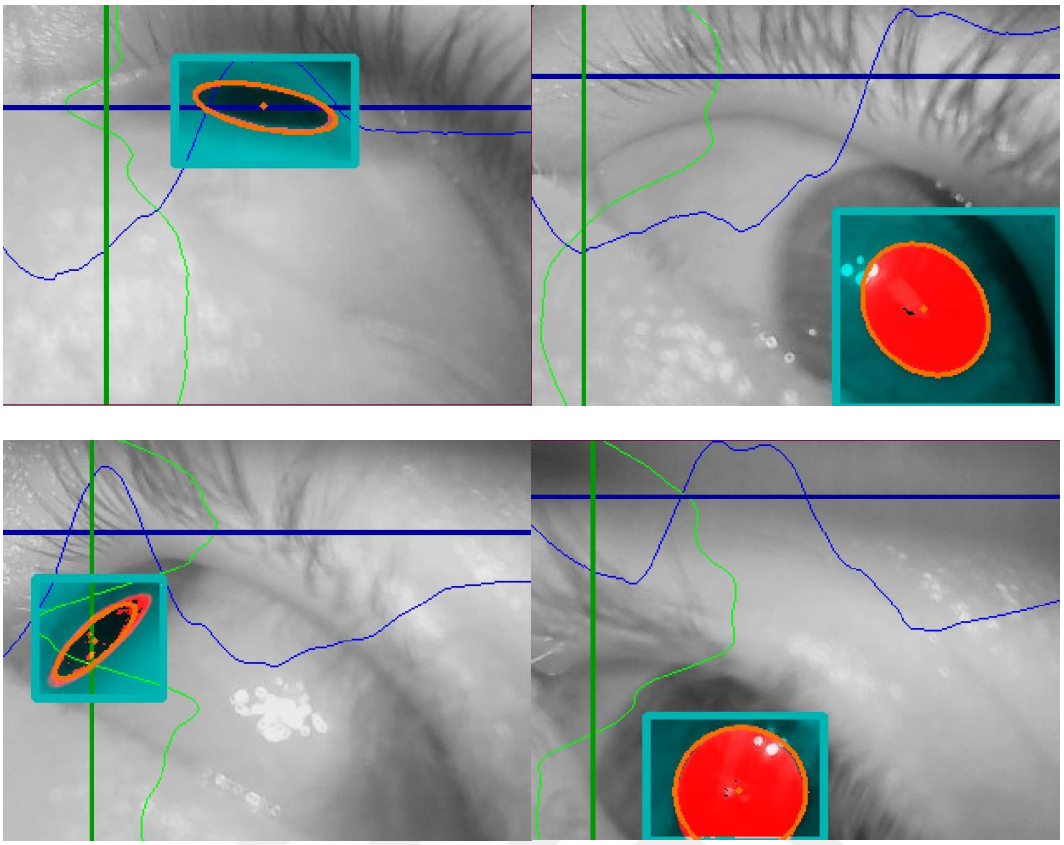


Figure 4.6 Pupil detection results obtained by looking at the edges as much as possible, including top, right, left and bottom

Although many factors negatively affect pupil detection, the most significant one is blinking, as it cannot be mitigated before the experiment. To demonstrate the impact of blinking within the scope of ROM in an off-axis positioning scenario, pupil detection results are presented in Figure 4.7. According to the results, even when the eye is almost closed, the pupil is detected at sufficient levels, and an ellipse is fitted. However, the positive contribution may be due to the traditional model limiting the area where the pupil is searched. As a result, eye components such as eyelashes are not considered in the detection, leading to a more accurate result near the pupil.

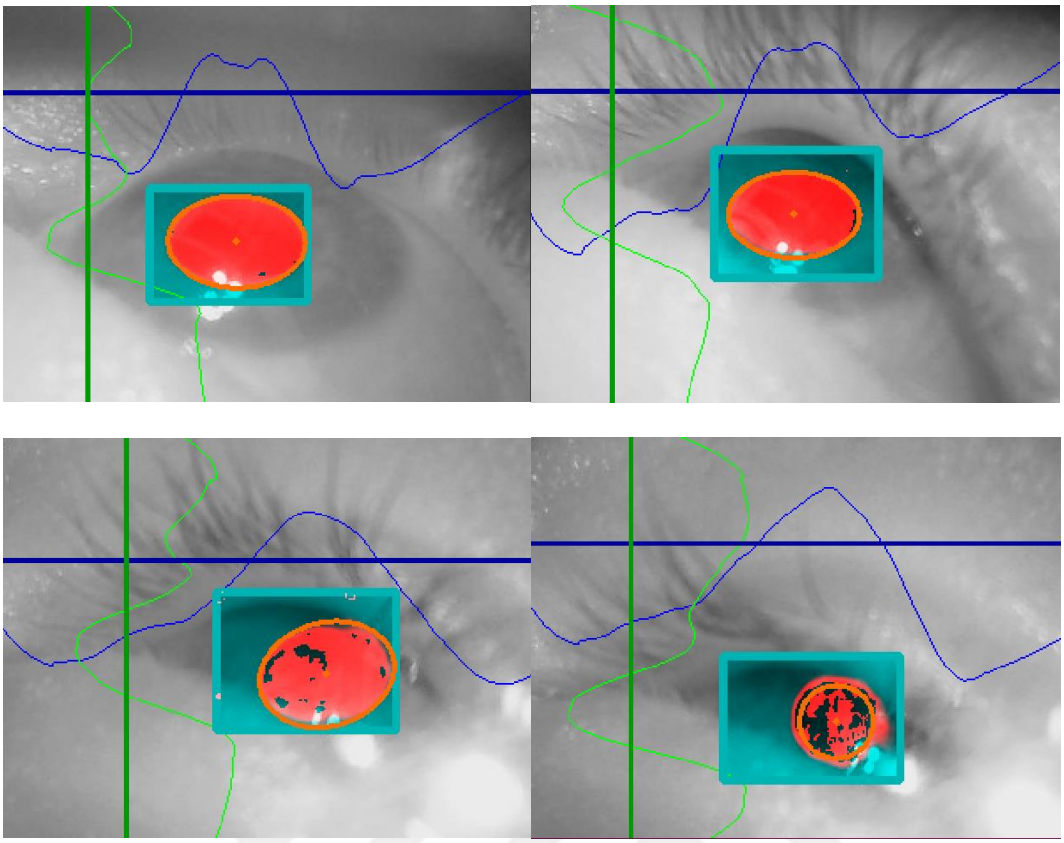


Figure 4.7 Pupil detection results during blinking when eye camera is in off-axis position

The proposed system can be used both off-axis and on-axis. In this usage, the pupil is generally detected more accurately. Results related to random on-axis usages are shared in Figure 4.8.

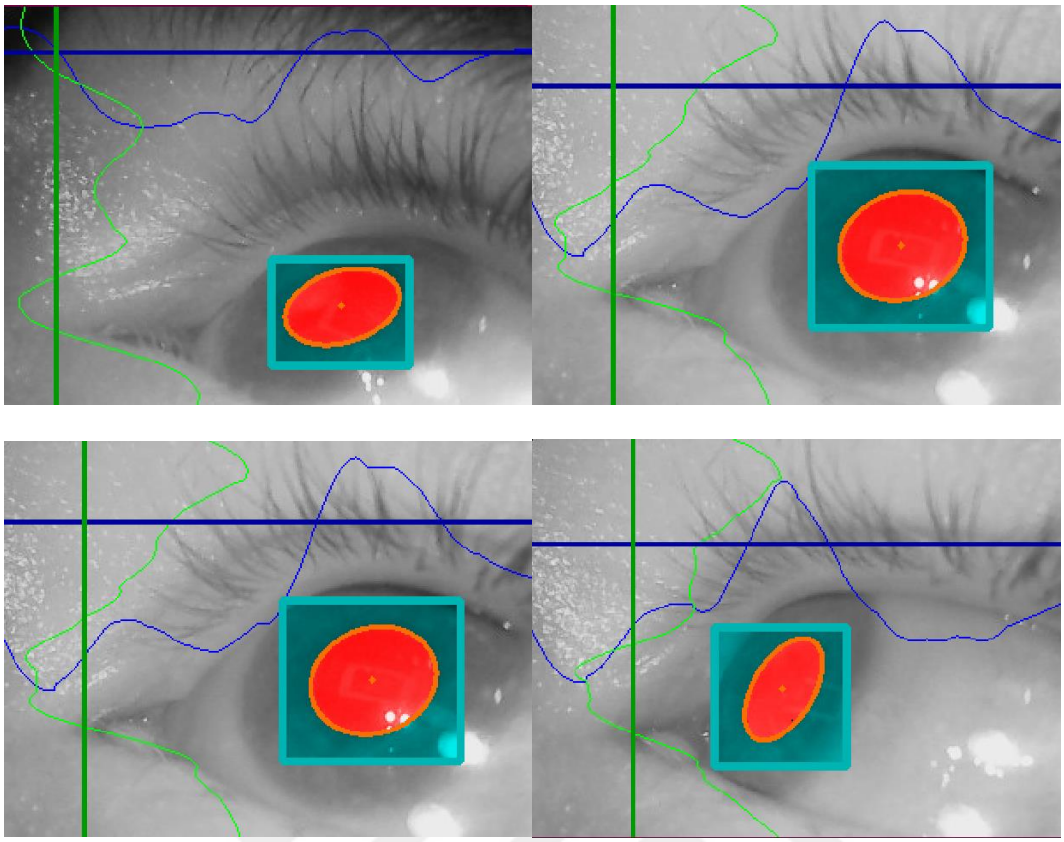


Figure 4.8 Pupil detection results obtained with the eye camera at a random on-axis position

The pupil detection results during blinking in the on-axis position are similar to those in the off-axis position. As presented in Figure 4.9, when the eye begins to close, a decrease in the learning model's segmentation map probabilities is observed. However, at this stage, the ellipse can still be well-fitted to the pupil. As expected, distortions in the ellipse occur due to the significant reduction in the eye opening. However, since the traditional method limits the area where the pupil is searched, other elliptical areas in the eye are not considered. Consequently, the error between the actual and predicted center points is lower. Nevertheless, for ellipse parameters other than the center point, the performance of the learning model remains critically important.

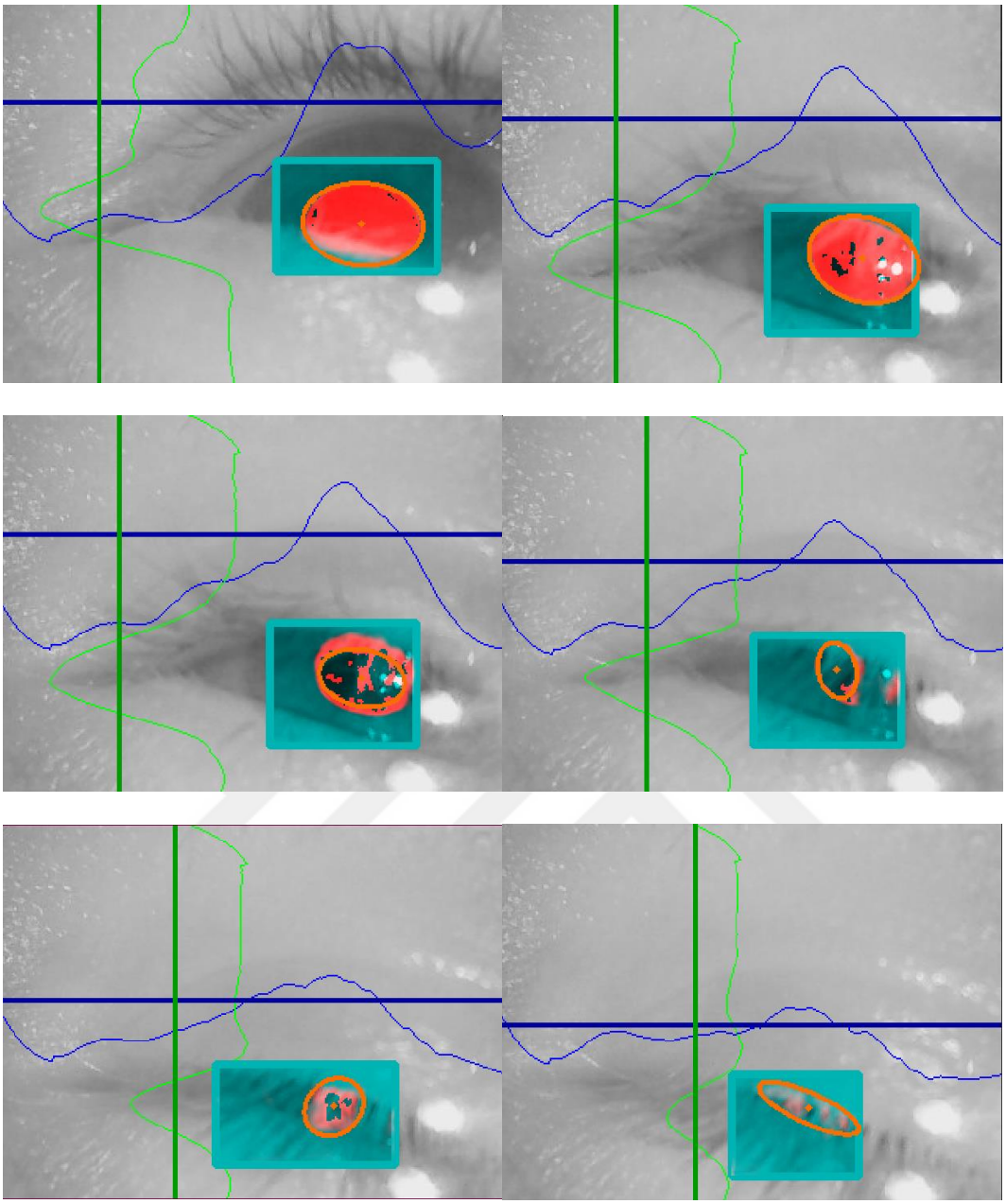


Figure 4.9 Pupil detection results during blinking with eye camera on-axis position

An example where pupil information might be insufficient, but where the pupil is better detected due to traditional methods not detecting areas such as eyelashes, is provided in Figure 4.10. Another notable feature of this figure is that it demonstrates the importance of selecting the correct traditional method. In this case, the Median First and PF methods were chosen for both axes. This selection ensures that the pupil patch remains smaller during moments of reduced eye opening or due to other dark areas in the eye region. As shown in the second image, selecting a PF method such as FE could result in the pupil patch being approximately twice as large as

necessary. In the third image, this could lead to an approximately fourfold increase, unnecessarily prolonging the detection time.

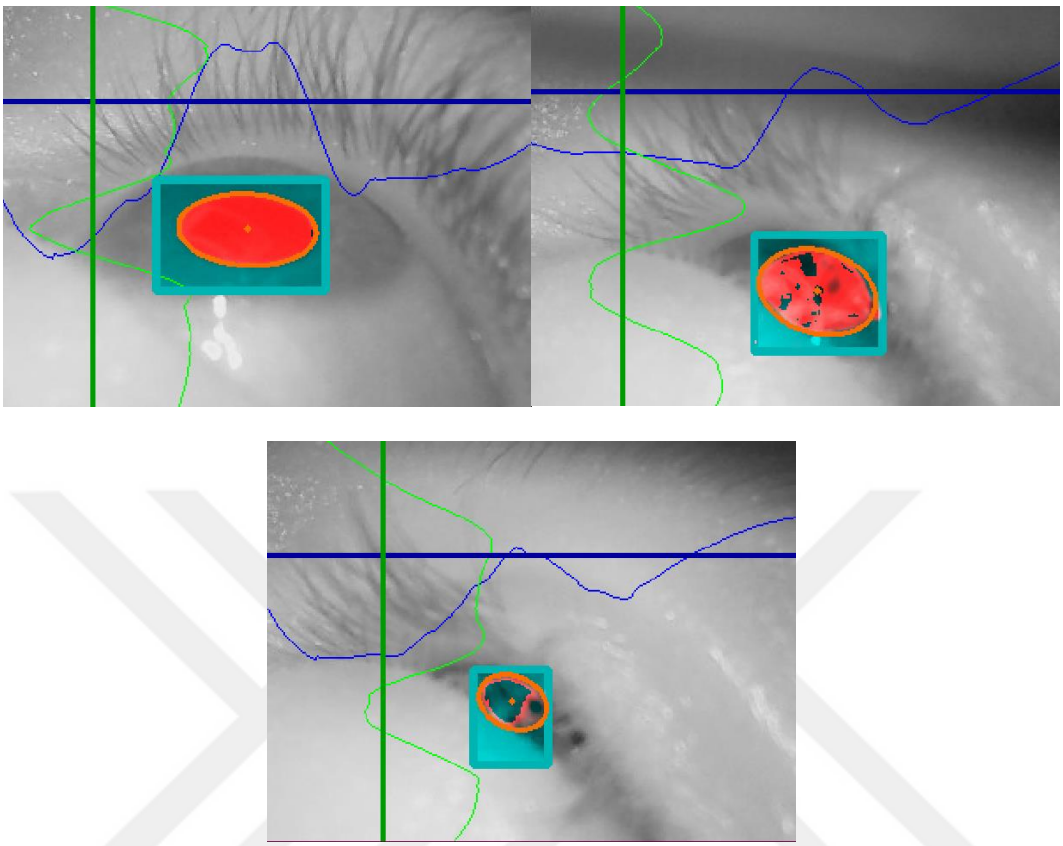


Figure 4.10 Pupil detection results for preserving the patch size to be created in a case where the traditional method setting selection will be made well (Median First has been selected for both axes)

However, if pupil information is absent or very limited within the image, the detections are unsuccessful and do not accurately represent the pupil. Although this traditional method may yield improved results, it does not prevent the learning model from using eyelashes as a reference, resulting in the continued production of erroneous pupil ellipses. An example of this issue is illustrated in Figure 4.11. In the image, eyelashes have been detected at a rate exceeding 50%, and consequently, an incorrect ellipse has been fitted overall.

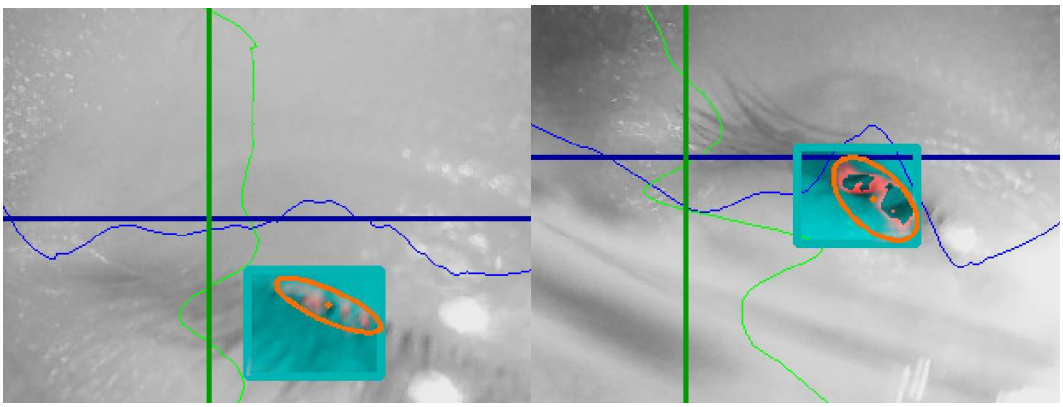


Figure 4.11 Unsuccessful results obtained in cases where pupil information is insufficient. However, since the pupil estimate is at eyelash level, jumping data problem may be encountered less

The primary limitation of the proposed ROM is the potential damage to the blackness information of the pupil. This issue arises in on-axis uses due to reflections on eyeglass lenses. Conversely, situations where areas darker than the pupil are present in the eye region can be easily prevented by rationally selecting PF methods. However, the current system lacks a solution for situations where there are reflections on the pupil. An example of this limitation is shown in Figure 4.12. According to the figure, it is impossible to create a patch that includes the pupil area with any PF function. Consequently, the pupil patch appears in incorrect locations, resulting in entirely unsuccessful detection.

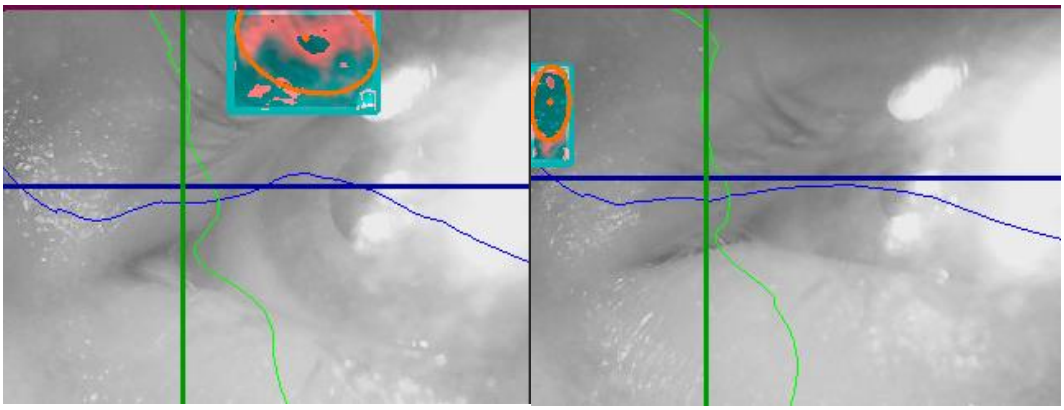


Figure 4.12 The pupil detection result of an example that is frequently encountered in on-axis usage, where the pupil loses a lot of dark amplitude information due to users' glasseses. The IR source is reflected from the glasses and the recommended black area does not contain the pupil

Another erroneous detection of the system occurs when the learning model's result is perceptually adequate, yet a flawed ellipse is fitted. An example of this issue is

illustrated in Figure 4.13. In the image, the segmentation map probabilities are lower in the area between the pupil and the eyelid. Consequently, the ellipse fitted to these probabilities becomes compressed from the sides. However, in some applications, the error margin between the actual and predicted points might still be within a tolerable range.

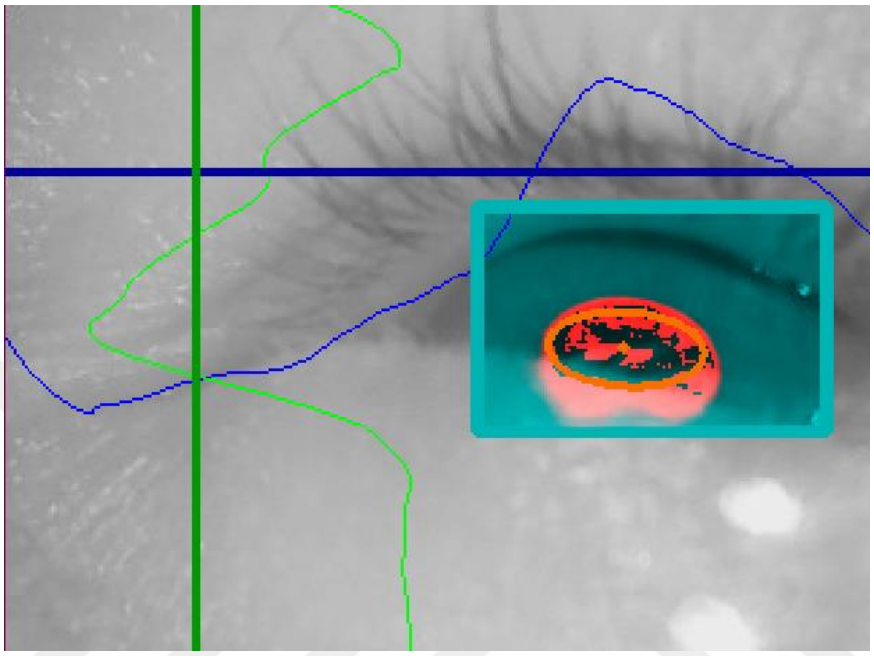


Figure 4.13 An example of an incorrect pupil detection for size and ellipse angle due to ellipse fitting, where extrapupillary areas are over-detected and intrapupillary areas are under-detected. However, the center point detection may still be sufficient

5
DISCUSSION

In this section, the thesis is discussed by dividing it into three parts. The first part addresses the designed hardware prototype. The second part focuses on the proposed pupil detection approach. Lastly, the third part provides a general examination of the overall system as a wearable ETS.

5.1 Discussion on Hardware Prototype

The proposed hardware design is suitable for both a pre-made headset frame and for use with the user's own glasses. This suitability is achieved through the modularity of the main frame and the socket holder. The main frame is necessarily positioned at the user's nose, while the socket holder is placed near the ear to avoid disrupting the balance of the glasses due to its weight. To minimize the weight and effects of the socket holder, it is connected to the main frame only via a cable. Additionally, for the system to be used with the user's own glasses, both the main frame and the socket holder need to be replaced. This situation requires extra care due to the delicate connection between the main frame and the socket holder. The need for extra attention during the system's replacement and storage could potentially reduce user satisfaction.

The modular design of the system aims to facilitate the addition and removal of components. Due to the addition and removal processes, additional supports and adapters have been used to mount the cameras and sensors onto the frame. These choices in the current prototyping preferences increase the visibility and weight of the system due to their connectors. In both cases, this could have a more restrictive impact on user behavior. On the other hand, the impact of visibility and weight can be significantly mitigated by differentiating the production process. On the other hand, the main factor limiting the system's modularity is the number of inputs on the socket holder. This corresponds to the number of ports supported by the USB hub. In the context of this thesis, since the USB hub has 4 ports, the socket holder also has 4 inputs.

Accurate detection of the coordinates of objects captured by both the world camera and the eye camera is crucial for the success of some calculations such as the

mapping process and surface tracking. Therefore, if there is any distortion due to the camera's field of view, camera calibration is necessary. Although there is no image distortion in the eye camera in the current setup, there is minimal barrel distortion in the world camera, which has been corrected through chessboard corner calibration. However, replacing the cameras in the system will require specific calibration for the new cameras. Consequently, this calibration should either be performed by the users, or the system should include predefined cameras that have been calibrated in advance. This procedure will be essential when changing cameras.

Due to corneal reflection issues with the used eye camera, the built-in IR LEDs were deactivated, and external IR LEDs were utilized instead. These external IR LEDs move in conjunction with the eye camera and aim to provide accurate pupil data. However, this integration introduces a power supply issue in a modular system. The IR LEDs must either be powered through the same socket as the camera or via an external port. To ensure optimal camera performance, the LEDs were powered through a separate port. This approach necessitates an additional socket for the LED connections, thereby reducing the system's capacity to support other external components.

The IR LEDs used for illuminating the eye region are invisible to users, ensuring that the eye area is comfortably and adequately illuminated. Although infrared emissions are not perceivable by the user and the LEDs operate at a wavelength of 940 nanometers, intense emissions have the potential to cause irritation to the eyes and skin. The effects of the emission intensity on eye health were not thoroughly examined in this study, and the intensity of the IR LEDs was not measured. However, it was observed that there were no signs of irritation or other adverse effects on the eyes following about 3 hours of use. Consequently, the potential negative effects of infrared emissions on skin and eye health were not considered in this thesis.

Similarly, the effects of the system's weight on users have not been measured. The weight of the developed system is approximately similar to that of Pupil Labs' products. Individual observations indicated that after approximately 3 of use, discomfort was felt in the nose and ears. However, due to the pressure distribution provided by the recommended silicones, the effect on the ears was lighter.

Therefore, increasing the contact areas of the system on the face appropriately can reduce the discomfort experienced.

Wearable ETS hardwares typically use non-slip components made of silicone or similar materials for the nose and ear sections of the frame. These non-slip components help reduce slippage. Similarly, for the proposed ETS to function as expected, it is necessary to minimize the risk of slippage. Although the headset frame includes the necessary silicone components, the user's own glasses may not have a non-slip structure capable of supporting the system's weight. For example, trials have shown that an 11-gram load on the nose bridge can cause the glasses to slip out of the desired position. To mitigate this effect, it is recommended that users use a non-slip component that fits behind the ear. Adapting the non-slip component to the user's own glasses may also incur additional time costs.

The eye camera used in the system, despite being quite small, is still measured in centimeters. This necessitates a more flexible determination of the camera's position to ensure that the camera captures the eye directly rather than the lens of the glasses when the user is wearing their own glasses. Therefore, the system incorporates two ball joints for the eye camera. These ball joints allow the camera to be positioned at the desired level and enable both in-axis and off-axis usage. However, positioning the camera at the required angle using two ball joints can be relatively time-consuming and may require practical adjustments.

During the hardware design phase of the thesis, the positions for placing the IR LEDs were specified. However, it was observed that the IR LEDs used in the results, due to their narrow-angle light dispersion, provided insufficient illumination for the eye region. This occasionally hindered the accurate detection of the pupil, necessitating the use of an additional lighting structure. In the current design, the optimal solution would be either to increase the number of LEDs or to use LEDs capable of wide-angle emission. Both alternatives are considered to require a more professional production process, and thus these enhancements are designated as future work.

Other discussion points on the hardware side include the lack of a head-mounted apparatus and the potentially insufficient field of view of the world camera. Since

neither of these issues poses a problem for the initial use of the developed system, related enhancements and modifications are designated as future work.

5.2 Discussion on Pupil Detection

The definition of the Savitzky-Golay filter as a parabola is due to the characteristic embedding of the pupil area, which is typically oval or rectangular in the curve. However, the window size value has been intuitively determined by the thesis authors and represents the average size of the pupil in the eye region. According to this intuition, for a ratio of 20% on each axis, the pupil area corresponds to approximately 4% of the image. In the experiments conducted, it was observed that lower determined ratios could not correct the noise, while higher ratios reduced the pupil information in the curve. Nevertheless, to achieve a better definition, it may be considered to calculate the average pupil area information for the Savitzky-Golay window size using dataset masks and their annotations.

The average patch size of the traditional method proposed within the scope of ROM is 13.05%. However, according to the CDF curve, the average patch percentage for half of the LPW is equal to or less than 7.58% (approximately x13.2). This suggests that the traditional method can produce much smaller patches for more controlled experimental procedures. As a result, the learning model can operate approximately 1ms faster (based on an x16 magnification factor) and the system frequency can be increased to 184 FPS. Furthermore, by ensuring that the learning model deals with significantly less extraneous information, it can be used solely for localization processes.

When determining dataset variations for LPW, image resolutions were not set uniformly but were instead prepared in accordance with the CDF curve and the intuitively defined 4% pupil size information. Accordingly, variations start from the lowest magnification factor of x2 and end at the intuitive value related to the average pupil size, which is x25, as given in the CDF graph. Intermediate resolution variations are not created at equal intervals but are instead based on regions of accumulation in the CDF. Therefore, variations with resolutions of x25, x16, x9, and x6 are prepared for areas with high accumulation. Those with resolutions of x4, x2, and the original are referenced for areas with less accumulation. However, for

a more comprehensive solution, resolution variations can be diversified, and more frequent resolution choices can be made even in areas with less accumulation.

Unlike edge-based methods and learning models, amplitude-based traditional methods in the literature can enable the implementation of pre-experimental precautions to enhance computational accuracy. These precautions are generally associated with the removal of regions darker than the pupil from the eye area. However, this may not always be possible, and there may be dark areas in the eye region that cannot be removed. Therefore, in such cases, it should not be expected that traditional methods using amplitude information will work accurately. The amplitude-based traditional method proposed in the ROM concept of this thesis does not operate on the assumption that the pupil is the darkest region. Instead, it allows the experimental operator to determine the relationship between the pupil and other dark areas and operates according to PF function selections. PF functions provide a pre-experimental precaution alternative that is more advantageous than other amplitude-based traditional methods by evaluating the darkness, length, and position information of the pupil simultaneously. On the other hand, the First and End function, which ensures the selection of all dark areas, can be used to maximize the likelihood of the pupil being included within the patch, even in extreme situations where pre-experimental precautions cannot be taken. In this regard, the proposed amplitude-based traditional method can inspire measures regarding pupil selection criteria in other methods that operate with traditional processes.

While training and comparing the learning models in the thesis, certain regulations were implemented to reduce the training duration on the fixed hyperparameters and the data set. For a more accurate analysis, training processes can be designed with the all the data set (not with 3 Hz set), cross-validation and hyperparameter optimizations. However, due to our temporal and physical resources, the relevant regulation had to be implemented.

In the literature, pupil detection successes are often reported using metrics such as center error and accuracy based on a 5-pixel error. These metrics may be inadequate for comparing models as the mean value is insufficient to define a stochastic process producing ellipse parameters. Therefore, the comparison section includes not only performance metrics based on ellipse center points but also size and angle errors. Sharing these metrics in future studies might better reflect the performance that

studies can offer. On the other hand, sharing size and angle errors is crucial for understanding the success of a segmentation model working with pupil patches. This is because, in cases where the created patches are small-sized and well-center the pupil, the success of the segmentation model may not be apparent. To better under-stand the performance of the segmentation model, average size error and the average size of the created patches can serve as good indicators.

With the operation of the ROM concept, a learning model with a traditional method can operate more efficiently with less unnecessary information. This is especially prominent in the field of remote eye tracking, as direct pupil information cannot be obtained by capturing the entire scene. Many objects in the scene may be black or circular, like the pupil. Therefore, in remote eye tracking, facial detection is first performed, and the eye region is then extracted, and lastly pupil is segmented. In other words, in remote eye trackers, several procedures are applied to ensure that learning models receive only the most relevant information. This study demonstrates that the "most relevant visual information" principle, which remote eye tracking systems already apply, can also be additionally implemented for wearable eye tracking systems.

In deep learning studies, training sets may be given to training after being cropped. However, in this type of cropping, the resolution of the cropped image is rescaled towards the fixed resolution in the dataset. Although a similar cropping approach has been applied in LPW-AV, resolutions have not been expanded to the original resolution. Additionally, LPW-AV, which includes cropped images at different resolutions, has a data augmentation effect and has been able to significantly improve performance. On the other hand, within the scope of this study, it has not been inves-tigated whether this increase in success is due to different resolutions or only cropping. Furthermore, while exploring the impact of these two cropping approaches on pupil detection could be a different research topic, in almost none of the studies in the literature has fixed-size cropping been used as a data augmentation method.

Models designed to work with pupil patches were trained with LPW-AV, which combines all LPW variations. However, this increases the training duration cost. For example, while one epoch UNet with LPW-Full is completed in approximately 3-4 hours, LPW-AV takes about 10-12 hours for one epoch. To reduce this cost

during training, two regimes can be applied. The first is to select a certain number of samples from each LPW variation rather than using all variations, creating a new mixed LPW variation. The second is to complete this process with synthetic image generators. The working principle of a synthetic image generator could be to produce images at a lower resolution, where the model makes less successful predictions in real-time. We do not explore these two alternatives within the concept of reducing training times and attribute this research as a future work. On the other hand, in the ROM structure, if a model is used whose success is not affected by the input resolution, there is no issue with a variation like LPW-AV and, consequently, no problem of increased training time.

The existing literature has highlighted the utilization of various eye features for diverse applications, emphasizing the potential value of exploring new features. However, our model is designed to find pupil ellipse parameters only. Making the developed model detectable for other eye features can be considered as another future work of this study.

Parameters marked as unsuccessfully according to PETA's metrics are not corrected if they do not meet the condition in the correction approach. It may be considered to improve the current correction approach to correct the mentioned unsuccessfully parameters, but our study aimed to provide a solution for serious jumping data only. In addition, PETA's upper band recommendation is only for the human eye. For the developed method to be used in an animal other than a human, the assumptions will need to be updated according to the relevant animal.

The threshold of 19 pixels, as determined in PETA's upper band correction approach, is notably high, and its impact on improving performance is minute. For a more optimal upper band, the eyeball can be detected and then the angular change between two consecutive frames can be observed. However, since our method does not detect eyeball in its current form, it had to offer a procedure that produces higher upper band values. The integration of an eyeball detection feature or method has the potential to further enhance the accuracy of the PETA correction. However, on the other hand, this correction method with or without eyeball parameters may be useful for traditional pupil detectors.

ROM represents an approach rather than a rigid structure, allowing flexibility in the models and methods employed. According to the naming used in this study, works that combine traditional methods with learning models are referred to as ROM studies. Consequently, ROM presents an opportunity for future research, enabling researchers in the pupil detection field to explore innovative studies based on the fusion of traditional and state-of-the-art techniques. Additionally, from this perspective, we attribute LEyes' pupil detector as a study within the ROM concept.

As mentioned in the literature review, the most recent pupil detection studies are [72], [84], [87], [88], [97], [99], [106], [157], [158], and [159]. In our comparison with PONet, the proposed model not only performs more accurately due to its lack of limitation to 15 positions but also generates pupil area information thanks to its segmentation feature [157]. This is a crucial feature for cognitive studies. While EV-Eye provides a high-level frequency, it is more expensive, non-wearable, and likely less accurate. Therefore, pupil detection frame by frame with CMOS cameras remains a necessity. Res-CNN achieves a significant portion of its frequency increase by reducing resolution [159]. However, as resolution decreases, the error rate increases. Our proposed ROM structure, on the other hand, crops the image instead of reducing the resolution. Although the QAT structure in Res-CNN is not used in our study, it can be adapted. This would provide an additional frequency increase. Furthermore, the SAM-based model [99], Vir-NET [97], and Mal-NET [88] can be directly expressed with the ROM structure. Thus, it is possible to develop new studies that improve the detection speeds of all three models. A similar situation can be said for the studies of Jamaludin et al. [72]. For more accurate results, incorporating noise reduction and segmentation map correction operations into ROM, along with improving learning model accuracy, can be a function to be tested in future studies. However, Jamaludin et al.'s studies work according to black amplitude and lack the accuracy of a learning model. As mentioned in the literature, black amplitude information can easily produce erroneous results due to its insufficiency. On the other hand, the ROM structure is not valid for Pistol [84]. This is because Pistol focuses on detecting multiple components that can spread across the entire image, requiring the model input to be evaluated as a whole. However, as stated in the study, if a single segmentation model is used to improve performance, it is possible to add the Pistol model to the learning model part of ROM solely for

pupil and iris detection. PCR-Net is also unsuitable for ROM for a similar reason as Pistol, as it focuses on detecting non-pupil eye components for remote systems.

5.3 Discussion on the Present ETS

The thesis proposes a new pupil detection method, trained using a dataset that holds significant importance in the literature. However, there are differences in the imaging characteristics between the eye camera used in the dataset and the camera used in the thesis, leading to potential variations in the obtained eye images. These differences might result in the pupil detection method being less accurate in some cases or requiring stricter usage adjustments to maintain accuracy. To enhance accuracy, the learning model used in the thesis could be fine-tuned with images collected from the eye camera used in the system. Due to the time required for planning this process and labeling the images, fine-tuning has been designated as a future work.

The developed desktop application software only supports basic ETS processes. For example, it cannot detect fixations, blinks, or eye movements, and it lacks analytical tools for the collected data. In this regard, the application serves as a foundational software. Integrating additional relevant software features into the system is designated as future work. Another aim of modularity is to enable users to add software plugins. Allowing the system to support external plugin integration is another feature planned for future implementation.

Many current ETS systems support both 2D and 3D gaze point producing. The primary difference between them is that 3D gaze point producing may be more suitable for wearable systems. However, within the scope of this thesis, only 2D calibration is implemented. The use of 2D gaze points can reduce performance due to slippage effects and may only produce gaze points that are adequate within the calibration surface. In the future, the system is planned to include eyeball detection and a 3D calibration process. This enhancement aims to create a system that is less affected by slippage and less dependent on the calibration surface.

6
CONCLUSION

In this thesis, both hardware and software designs have been developed for a mobile and modular eye-tracking technology. The main contributions of the development are the modular design of the hardware and the proposal of a new pupil detection approach. These two advancements have led to the creation of a wearable ETS named FIXEIN. FIXEIN, with its features, focuses on increasing the number of experimental environments in which it can be used. This can reduce the need for acquiring different ETS devices and allow for adaptation of the ETS to meet specific needs.

An important hardware advancement in this study is the system's adaptation to the user's own glasses, which involves a certain initial cost. This cost is associated with adapting the main frame, socket holder, and silicones to the user's glasses. However, it is worth considering that the initial cost can be reduced by designing the hardware components more professionally and custom producing the camera components to be suitable for the ETS.

The pupil ellipse detector proposed in the thesis was designed with the innovative concepts of Retro-Oriented Mind (ROM) and Pupil Ellipse Trend Analysis (PETA). Both ROM and PETA are novel and performance improving approaches in pupil detection field. Benefitting from the proposed methodology, the detector can run real-time and accurate in the conditions of the pupil datasets.

In addition to running at more than 150 Hz in the development environment of the study, the proposed detector was able to show significantly higher accuracy than directly used segmentation models. This situation is an indicator that the learning model can operate more efficiently with the less unnecessary information it receives. Thereby, the detector has the potential to be directly included in general-purpose uses of eye tracking systems. Also, the use of the study in fields that require high accuracy, such as medicine, requires the regularization of some environmental factors and experimental procedures related to eye appearance. It is possible to say that the developed detector with a correct experimental procedure has the accuracy to be used for medical purposes.

Additionally, the proposed automatic configuration mechanism provides almost similar success compared to when the most successful configuration is manually selected. Thus, the need for manually determining settings such as thresholds can be eliminated.

Furthermore, to the best of our knowledge, the proposed pupil detector based on ROM and PETA concepts stands as one of the most high-performing detectors, achieving a harmonious balance between accuracy, latency, and frequency. Compared to LEyes, which is another work that can be evaluated in the ROM concept, our pupil detector in the ROM concept has almost similar accuracies but works about two times faster. This situation arises from both our patch approach not being fixed-sized and the model parameters being downsized to operate in real-time. It has also been observed that the ROM study we proposed works much more efficiently than using an encoder-decoder model directly.

Finally, the limitations of the thesis are addressed as follows:

- Only 2D calibration is used in the tracking software. Therefore, the user must move minimally and ensure no sliding effect occurs within the frame. Additionally, the polynomial equation applied as the mapping function will have higher derivatives outside the surface region, making the calibration surface also the user's main area of interest. This situation increases the need for 3D calibration for more dynamic applications.
- The tracking software is prepared at a basic level. It lacks features and process sets such as fixation, blink, data processing, and visualization. General-purpose features should be added for usability in academic or commercial applications. On the other hand, more specific features such as those required for procedures like VHIT could be integrated as enhancements.
- Parallax error may be more noticeable in surface calibration. Although the effect of parallax error has not been measured within the scope of the thesis, efforts have been made to partially reduce this effect by positioning the world camera lens closer to the eye. To further minimize this effect, automatic calibration processes may need to be integrated into the recommended tracking software. Another consideration is to make the field of view of the world camera more similar to that of the eyes, possibly by aligning it more centrally on the face.

- The hardware presented in the thesis is a prototype in terms of both solid parts and electronic components. Particularly, the eye camera exhibits ghosting effects. Therefore, more accurate pupil detection may be possible by moving the eye more slowly. Increasing the resolution and ROI of the world camera will prevent narrower ranges of user gazes. Additionally, both cameras are relatively large. Smaller cameras could reduce weight and allow for more natural user behavior. The emission angle of infrared LEDs is low and may not sufficiently illuminate the eye area.
- The modularity of sensors and cameras necessitates additional connection parts as they cannot be directly attached to the main frame. Therefore, hardware modularity is a weight-increasing factor in the current hardware design.
- The developed pupil detection approach can work very quickly due to the mechanisms it is equipped with. However, this depends on the performance of the produced patch. If the patch size is close to the original resolution, the speed gain is reduced. Therefore, temporarily reducing the system's frequency when the patch is larger than expected may reduce latency, but data loss may occur.
- The developed system weighs approximately 62 grams. Therefore, based on the experiments conducted by the authors, it is suitable for continuous use of approximately 3 hours; discomfort may be felt in the nose and ears with longer use.

REFERENCES

[1] Ejnestrand, I., Jakobsson, L. 2020. “Object Tracking based on Eye Tracking Data : A comparison with a state-of-the-art video tracker”, Master of Science Thesis, Linköping University. http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-166007 Access date: 20.08.2022

[2] Punde, P. A., Jadhav, M. E., Manza, R. 2017. “A study of eye tracking technology and its applications”, IEEE 1st International Conference on Intelligent Systems and Information Management, Aurangabad, India.

[3] Lewandowska, A., Rejer, A., Bortko, K., Jankowski, J. 2022. "Eye-Tracker Study of Influence of Affective Disruptive Content on User’s Visual Attention and Emotional State", Sensors, *22*(2): 547.

[4] Harezlak, K. and Kasprowski, P. 2018. “Application of eye tracking in medicine: A survey, research issues and challenges”, Computerized Medical Imaging and Graphics, 65, 176-190.

[5] Morita K., Miura K., Kasai K., Hashimoto R. 2020. “Eye movement characteristics in schizophrenia: A recent update with clinical implications”, Neuropsychopharmacol Reports, 40, 2–9.

[6] Madsen, H. Ø., Ba-Ali, S., Heegaard, S., Hageman, I., Knorr, U., Lund-Andersen, H., Martiny, K., Kessing, L. V. 2021. “Melanopsin-mediated pupillary responses in bipolar disorder—a cross-sectional pupillometric investigation”, International Journal of Bipolar Disord, 9, 7, 1-10.

[7] Jang, H., Soroski, T., Rizzo, M., Barral, O., Harisinghani, A., Newton-Mason, S., Granby, S., Stutz da Cunha Vasco, T. M., Lewis, C., Tutt, P., Carenini, G., Conati, C., Field, T. S. 2021. “Classification of alzheimer's disease leveraging multi-task machine learning analysis of speech and eye-movement data”, Frontiers in human neuroscience, 15, 716670.

[8] Grillini, A., Renken, R. J., Vrijling, A. C., Heutink, J., Cornelissen, F. W. 2020. “Eye movement evaluation in multiple sclerosis and Parkinson's disease using a standardized oculomotor and neuro-ophthalmic disorder assessment”, Frontiers in neurology, 11, 971.

[9] Oliveira, J. S., Franco, F. O., Revers, M. C., Silva, A. F., Portolese, J., Brentani, H., Nunes, F. L. 2021. “Computer-aided autism diagnosis based on visual attention models using eye tracking”, Scientific reports, 11, 1-11.

[10] Latifoğlu, F., Esas, M. Y., Demirci, E. 2020. "Diagnosis of attention-deficit hyperactivity disorder using EOG signals: a new approach". Biomedical Engineering / Biomedizinische Technik, 65, 149-164.

[11] Zhu, G., Wang, J., Xiao, L., Yang, K., Huang, K., Li, B., Wang, Q. 2021. “Memory Deficit in Patients With Temporal Lobe Epilepsy: Evidence From Eye Tracking Technology,” Frontiers in Neuroscience, 1160.

[12] Stuart, S., Parrington, L., Martini, D., Peterka, R., Chesnutt, J., King, L. 2020. “The measurement of eye movements in mild traumatic brain injury: a structured review of an emerging area”, Frontiers in sports and active living, 2, 5.

[13] Parker T., M., Farrell N., Otero-Millan J., Kheradmand A., McClenney A., Newman-Toker D., E. 2021. “Proof of Concept for an “eyePhone” App to Measure Video Head Impulses”, Digit Biomark, 5, 1-8.

[14] Carbone F., Ellmerer P., Ritter M., Spielberger S., Mahlknecht P., Hametner E., Hussl A., Hotter A., Granata R,. Seppi K., Boesch S., Poewe W., Djamshidian A. 2021. “Impaired Inhibitory Control of Saccadic Eye Movements in Cervical Dystonia: An Eye-Tracking Study”, Movement Disorders, 36, 1246-1250.

[15] Zammarchi, G., Claudio, C. 2021. "Application of Eye Tracking Technology in Medicine: A Bibliometric Analysis", Vision, 5, 56.

[16] Harezlak, K., Kasprowski, P. 2017. “Application of eye tracking in medicine: A survey, research issues and challenges”, Computerized Medical Imaging and Graphics, 65, 176-190.

[17] Brunyé, T.T., Drew, T., Weaver, D. L. et al. 2019. “A review of eye tracking for understanding and improving diagnostic interpretation”, Cognitive Research: Principles and implications, 4, 1-16.

[18] Zhu, G., Wang, J., Xiao, L., Yang, K., Huang, K., Li, B., Wang, Q. 2021. “Memory Deficit in Patients With Temporal Lobe Epilepsy: Evidence From Eye Tracking Technology”, Frontiers in Neuroscience, 1160.

[19] Dimitriadis, G., Gouleti, K., Kokonis, M. 2022. “Evaluating the Impact of Interactive Cinema in Audiovisual Literacy and Education: An Eye-Tracking Study”, Journal of Media Literacy Education Pre-Prints. https://digitalcommons.uri.edu/jmle-preprints/27 Access date: 20.08.2022.

[20] Schindler, M., Lilienthal, A. J. 2022. “Students’ collaborative creative process and its phases in mathematics: an explorative study using dual eye tracking and stimulated recall interviews”, ZDM Mathematics Education 54, 163–178.

[21] Ristic, J., Capozzi, F. 2022. “Interactive Cognition: An introduction”, Visual Cognition, 30, 1-5.

[22] Rogers, S.L., Speelman, C.P., Guidetti, O. et al. 2018. “Using dual eye tracking to uncover personal gaze patterns during social interaction”, Scientific Reports, 8, 4271.

[23] Gao, J., Adachi, I. Tomonaga, M. 2022. “Chimpanzees (Pan troglodytes) detect strange body parts: an eye-tracking study”, Animal Cognition, 1, 1-13.

[24] Huang, K., Yang, Q., Han, Y., Zhang, Y., Wang, Z., Wang, L., & Wei, P. 2022. “An Easily Compatible Eye-tracking System for Freely-moving Small Animals”, Neuroscience Bulletin, 1-16.

[25] Carbone F., Ellmerer P., Ritter M., Spielberger S., Mahlknecht P., Hametner E., Hussl A., Hotter A., Granata R,. Seppi K., Boesch S., Poewe W., Djamshidian A. 2021. “Impaired Inhibitory Control of Saccadic Eye Movements in Cervical Dystonia: An Eye-Tracking Study”, Movement Disorders, 36, 1246-1250.

[26] Borges, L.R., Naves, E. L. M., Sa, A.A.R. 2021. “Usability evaluation of an electric-powered wheelchair driven by eye tracking”, Univ Access Inf Soc https://doi.org/10.1007/s10209-021-00809-z, Access date: 20.08.2022

[27] Tanış, H. , Akçay, A. , Yılmaz, N. , Yiğit, M. F., Tüzün, H. 2022. “How usable is Coursera? A usability analysis through eye-tracking and authentic tasks”, Participatory Educational Research , 9, 379-395.

[28] Patil, N., Das, R., Dhusia, K., Sanap, V., & Singh, V. K. 2022. “Eye Gaze-Based Student Readability Analysis”, Proceedings of International Conference on Computational Intelligence and Emerging Power System, Singapore.

[29] Harada, H., Nakayama, M. 2021. “Estimation of reading ability of program codes using features of eye movements”, ACM Symposium on Eye Tracking Research and Applications (ETRA '21 Short Papers), New York, USA.

[30] Kazuaki, T., Yoichi, Y., Kentaro, Y. 2020. “Avatar Work: Telework for Disabled People Unable to Go Outside by Using Avatar Robots "OriHime-D" and Its Verification”. https://www.cnet.com/news/paralyzed-people-hired-to-control-robot-waiters-in-tokyo-cafe/ Access date: 20.08.2022.

[31] Wang, Q., He, B., Xun, Z., Xu, C., Gao, F. 2021. “GPA-Teleoperation: Gaze Enhanced Perception-aware Safe Assistive Aerial Teleoperation”, arXiv:2109.04907v1

[32] Groner, R., Kasneci, E. 2021. “Eye movements in real and simulated driving and navigation control”, Journal of Eye Movement Research, 12, 1-3.

[33] Barbuceanu, F., & Antonya, C. 2009. “Eye tracking applications”, Bulletin of the Transilvania University of Brasov. Engineering Sciences. Series I, 2, 17.

[34] Adhanom, I.B., MacNeilage, P. & Folmer, E.: Eye Tracking in Virtual Reality: a Broad Review of Applications and Challenges. Virtual Reality 27, 1481–1505 (2023). https://doi.org/10.1007/s10055-022-00738-z

[35] Efetürk, Ö. A. , Turgut, G. , Asadi Dereshgi, H. & Yılmaz, A.: A review of visual attention research using eye-tracking technologies. Journal of Smart Systems Research, (2022) 3 (2), 128-135. Retrieved from https://dergipark.org.tr/en/pub/joinssr/issue/74246/1208182

[36] Ecenaz Alemdag, Kursat Cagiltay, A systematic review of eye tracking research on multimedia learning, Computers & Education, Volume 125, (2018), https://doi.org/10.1016/j.compedu.2018.06.023

[37] Ahmad F. Klaib, Nawaf O. Alsrehin, Wasen Y. Melhem, Haneen O. Bashtawi, Aws A.,: Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies, Expert Systems with Applications, (2021) https://doi.org/10.1016/j.eswa.2020.114037.

[38] Joseph R.: An introduction to eye tracking in human factors healthcare research and medical device testing, Human Factors in Healthcare, (2023). https://doi.org/10.1016/j.hfh.2022.100031

[39] Wang S, Ouyang X, Liu T, Wang Q, Shen D.: Follow My Eye: Using Gaze to Supervise Computer-Aided Diagnosis. IEEE Trans Med Imaging. (2022). https://doi.org/10.1109/TMI.2022.3146973

[40] Sun, C., Ding, N., Zhuang, D., Liu, X.: Eye movement evidence in investigative identification based on experiments, J. of Safety Science and Resilience, (2023). https://doi.org/10.1016/j.jnlssr.2023.07.003

[41] Issever D, Catalbas MC, Duran F.: Examining Factors Influencing Cognitive Load of Computer Programmers. *Brain Sciences*. (2023). https://doi.org/10.3390/brainsci13081132

[42] Vinuela-Navarro, V., Goset, J., Aldaba, M., Mestre, C., Rovira-Gay, C., Cano, N., Ariza, M., Delàs, B., Garolera, M., Vilaseca, M.: Eye movements in patients with post-COVID condition. Biomed. Opt. Express 14, 3936-3949 (2023)

[43] Xu, J., Huang, Z., Liu, L., Li, X., Wei, K.: Eye-Gaze Controlled Wheelchair Based on Deep Learning. Sensors. (2023). https://doi.org/10.3390/s23136239

[44] Eyelink, https://www.sr-research.com/eyelink-1000-plus/ Access date: 20.08.2022.

[45] Pupil Labs, "Core", https://pupil-labs.com/products/core/ Access date: 20.08.2022.

[46] Cognolato, M., Atzori, M., Müller, H. 2018. "Head-mounted eye gaze tracking devices: An overview of modern devices and recent advances", Journal of rehabilitation and assistive technologies engineering, 5, 1-13.

[47] Kassner, M., Patera, W. 2012. "Pupil: Constructing the space of visual attention, Massachusetts Institute of Technology", Cambridge, MA. http://hdl.handle.net/1721.1/72626 Access date: 20.08.2022

[48] Pupil Lab, "Invisible", https://pupil-labs.com/products/invisible/ Access date: 20.08.2022.

[49] Ergoneers, Eye Tracking Technologies, https://www.ergoneers.com/ Access date: 20.08.2022.

[50] ISCAN, Eye Tracking Technologies, http://iscaninc.com/ Access date: 20.08.2022

[51] Argus Science, Eye Tracking Technologies, http://argusscience.com/ETVision.html Access date: 20.08.2022.

[52] Tobii, Eye Tracking Technologies, https://www.tobiipro.com/ Access date: 20.08.2022

[53] Rakhmatulin, I. 2020, "A review of the low-cost eye-tracking systems for 2010-2020", arXiv:2010.05480

[54] Van Dooren, T. S., Starkov, D., Lucieer, F. M. P., Vermorken, B., Janssen, A. M. L., Guinand, N., van de Berg, R. 2020. "Comparison of three video head impulse test systems for the diagnosis of bilateral vestibulopathy", Journal of Neurology, 267, 256-264

[55] EyeSeeTec, Eye Tracking Technologies, https://eyeseetec.de/ Access date: 20.08.2022

[56] Natus, Eye Tracking Technologies, https://hearing-balance.natus.com/ Access date: 20.08.2022.

[57] Pupil Labs Neon, Eye Tracking Technologies, https://pupil-labs.com/products/neon

[58] Holmqvist, K., Örbom, S.L., Hooge, I.T.C. et al: Eye tracking: empirical foundations for a minimal reporting guideline. Behav (2023). https://doi.org/10.3758/s13428-021-01762-8

[59] Wang S, Ji Y, Bai W, Ji Y, Li J, Yao Y, Zhang Z, Jiang Q and Li K (2023), Advances in artificial intelligence models and algorithms in the field of optometry. Front. Cell Dev. Biol. 11:1170068. doi: 10.3389/fcell.2023.1170068

[60] Kamran Binaee, Christian B. Sinnott, Kaylie J. Capurro, Paul R. MacNeilage, and Mark D. Lescroart. 2021. Pupil Tracking Under Direct Sunlight. In 2021 Symposium on Eye Tracking Research and Applications (ETRA '21 Adjunct), May 25–27, 2021, Virtual Event, Germany. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3450341.3458490

[61] Min, N., Jan, F., Alrashed, S.: Pupil detection schemes in human eye: a review. Multi. Sys. (2021). https://doi.org/10.1007/s00530-021-00806-5

[62] Swathi, A.., Sandeep, K.: Review on Pupil Segmentation Using CNN-Region of Interest. Book, Intelligent Communication and Automation Systems (2021). https://doi.org/10.1201/9781003104599-12

[63] Morimoto, H., Koons, D., Amir, A., Flickner, M.: Pupil detection and tracking using multiple light sources, Image and Vision Computing, (2000). https://doi.org/10.1016/S0262-8856(99)00053-0.

[64] Navaneethan, S., Nandhagopal, N.: RE-PUPIL: resource efficient pupil detection system using the technique of average black pixel density. Sādhanā (2021). https://doi.org/10.1007/s12046-021-01644-x

[65] Gabbur, P., Hua, H., Barnard, K.: A fast connected components labeling algorithm and its application to real-time pupil detection. Machine Vision and App. (2010). https://doi.org/10.1007/s00138-009-0183-1

[66] Abbasi, M., Khosravi, M.R.: A Robust and Accurate Particle Filter-Based Pupil Detection Method for Big Datasets of Eye Video. J Grid Computing (2020). https://doi.org/10.1007/s10723-019-09502-1

[67] Bonteanu, P., Cracan, A., Bozomitu R. G., Bonteanu, G.: A Robust Pupil Detection Algorithm based on a New Adaptive Thresholding Procedure. E-Health and Bioengineering Conference (2019). https://doi.org/10.1109/EHB47216.2019.8970070

[68] Wan, Z., Xiong, C., Chen, W., Zhang, H.: Robust and accurate pupil detection for head-mounted eye tracking, Comp. & Electrical Eng., (2021). https://doi.org/10.1016/j.compeleceng.2021.107193.

[69] Timm, Fabian & Barth, Erhardt. (2011). Accurate Eye Centre Localisation by Means of Gradients.. VISAPP 2011 - Proceedings of the International Conference on Computer Vision Theory and Application. 125-130.

[70] André Frank Krause and Kai Essig. 2019. Boosting Speed- and Accuracy of Gradient based Dark Pupil Tracking using Vectorization and Differential Evolution. In 2019 Symposium on Eye Tracking Research and Applications (ETRA '19), June 25–28, 2019, Denver , CO, USA. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3314111.3319849

[71] Manuri, F., Sanna, A. Petrucci, C: PDIF: Pupil Detection After Isolation and Fitting. IEEE Access, (2020), doi: 10.1109/ACCESS.2020.2973005

[72] Jamaludin, S., Ayob, A., Akhbar, M., Ali, A., Imran, M., Norzeli, S., Mohamed, S.: Efficient, accurate and fast pupil segmentation for pupillary boundary in iris recognition, Advances in Engineering Software, (2023). https://doi.org/10.1016/j.advengsoft.2022.103352 .

[73] Fuhl, W., Santini, T., Kübler, T., Kasneci, E.: ElSe: ellipse selection for robust pupil detection in real-world environments. ACM Symp. on Eye Tracking Res. & App. (2016). https://doi.org/10.1145/2857491.2857505

[74] Santini, T., Fuhl, W., Kasneci, E.: PuRe: Robust pupil detection for real-time pervasive eye tracking, Computer Vision and Image Understanding, (2018). https://doi.org/10.1016/j.cviu.2018.02.002.

[75] Santini, T., Fuhl, W., Kasneci, E.: PuReST: Robust Pupil Tracking for Real-Time Pervasive Eye Tracking. Symp. on Eye Tracking Research and App., (2018). https://doi.org/ 10.1145/3204493.3204578

[76] Susitha, N., Subban, R.: Reliable pupil detection and iris segmentation algorithm based on SPS, Cognitive Systems Research (2019). https://doi.org/10.1016/j.cogsys.2018.09.029.
[77] Li, J., Li, S., Chen T., Liu, Y.: A Geometry-Appearance-Based Pupil Detection Method for Near-Infrared Head-Mounted Cameras. IEEE Access, (2018). https://doi.org/10.1109/ACCESS.2018.2828400
[78] Fuhl, W., Kübler, T., Sippel, K., Rosenstiel, W., Kasneci, E.: ExCuSe: Robust Pupil Detection in Real-World Scenarios. Comp. Anal. of Images and Patterns. (2015). https://doi.org/10.1007/978-3-319-23192-1_4
[79] Omran, M., AlShemmary, N.: Towards Accurate Pupil Detection Based on Morphology and Hough Transform. Baghdad Science Journal, Volume 17, Issue 2, Pages 583-590, (2020).
[80] Bonteanu, P., Cracan, A., Bozomitu, G., Bonteanu, G.: A New Robust Pupil Detection Algorithm for Eye Tracking Based Human-Computer Interface. Inter. Symp. on Signals, Circuits and Sys. (2019) doi: 10.1109/ISSCS.2019.8801818.
[81] Kassner, M., Patera, W., Bulling, A.: Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. Inter. Joint Conf. on Pervasive and Ubiquitous Computing (2014) https://doi.org/10.1145/2638728.2641695
[82] V. Skaramagkas et al., "Review of Eye Tracking Metrics Involved in Emotional and Cognitive Processes," in IEEE Reviews in Biomedical Engineering, vol. 16, pp. 260-277, 2023, doi: 10.1109/RBME.2021.3066072.
[83] Alwin de Rooij, Iris Wijers, and Manon Marinussen. 2021. Emergence of Metacognitive Knowledge via Audible Pupil Size. In European Conference on Cognitive Ergonomics 2021 (ECCE 2021), April 26–29, 2021, Siena, Italy. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3452853.3452870
[84] Fuhl, W., Weber, D., Eivazi, S.: Pistol: Pupil Invisible Supportive Tool to extract Pupil, Iris, Eye Opening, Eye Movements, Pupil and Iris Gaze Vector, and 2D as well as 3D Gaze, (2023), 2201.06799, arXiv
[85] Chinsatit, W., Saitoh, T.: CNN-Based Pupil Center Detection for Wearable Gaze Estimation System, Applied Computational Intelligence and Soft Computing, (2017). https://doi.org/10.1155/2017/8718956
[86] Lee, YW., Kim, KW., Hoang, TM., Arsalan, M., Park, KR.: Deep Residual CNN-Based Ocular Recognition Based on Rough Pupil Detection in the Images by NIR Camera Sensor. Sensors. (2019) 19(4):842. https://doi.org/10.3390/s19040842
[87] Chen, J., Yu, P., Yao, C., Zhao, L., Qiao, Y.: Eye detection and coarse localization of pupil for video-based eye tracking systems, Expert Systems with App., (2024). https://doi.org/10.1016/j.eswa.2023.121316.
[88] Gou, C., Zhong, R., Yu, Y.: MAL-Net: Multiscale Attention Link Network for accurate eye center detection. Computer Vision and Image Understanding, (2023). https://doi.org/10.1016/j.cviu.2023.103750
[89] Shi, L., Wang, C., Tian, F. et al: An integrated neural network model for pupil detection and tracking. Soft Comput 25, 10117–10127 (2021). https://doi.org/10.1007/s00500-021-05984-y

[90] Wang, L., Wang, C., Zhang, Y.: Pupil Detection Using Hybrid Vision Transformer, Interna-tional Journal of Pattern Recognition and Artificial Intelligence (2022). https://doi.org/10.1142/S0218001422550163
[91] Akinlar, C., Kucukkartal, H., Topal, C.: Accurate CNN-based pupil segmentation with an ellipse fit error regularization term, Expert Sys. with Applications, (2022). https://doi.org/10.1016/j.eswa.2021.116004
[92] Rathnayake, Rasanjalee & Madhushan, Nimantha & Jeeva, Ashmini & Darshani, Dhanushika & Subasinghe, Akila & Silva, Bhagya & Wijesingha, Lakshitha & Wijenayake, Udaya. (2023). Current Trends in Human Pupil Localization: A Review. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3325293.
[93] Y. Lee, S. Lee, S. Jang, H. J. Wang, Y. J. Seo and S. Yang, "Pupil Detection and Segmentation for Diagnosis of Nystagmus with U-Net," 2022 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Korea, Republic of, 2022, pp. 1-2, doi: 10.1109/ICEIC54506.2022.9748687.
[94] S. Y. Han, H. J. Kwon, Y. Kim and N. I. Cho, "Noise-Robust Pupil Center Detection Through CNN-Based Segmentation With Shape-Prior Loss," in IEEE Access, vol. 8, pp. 64739-64749, 2020, doi: 10.1109/ACCESS.2020.2985095.
[95] 83. Gowroju, Swathi &., Aarti & Kumar, Sandeep. (2021). Robust Pupil Segmentation using UNET and Morphological Image Processing. 105-109. 10.1109/MIUCC52538.2021.9447658
[96] Y. Guo, X. Yan, L. Ma, L. Tang, J. Xiong and Y. Yan, "A Pupil Segmentation Framework with Masked Image Modeling Enhanced Swin-Transformer," 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Las Vegas, NV, USA, 2022, pp. 2439-2444, doi: 10.1109/BIBM55620.2022.9995137.
[97] Maquiling, Virmarie & Byrne, Sean & Nyström, Marcus & Kasneci, Enkelejda & Niehorster, Diederick. (2023). Vir-Net: A Novel Neural Network for Pupil and Corneal Reflection Detection trained on Simulated Light Distributions. 1-7. 10.1145/3565066.3608690.
[98] Lijinliang Niu, Zhaopeng Gu, Juntao Ye, and Qiulei Dong. 2021. Real-Time Localization and Matching of Corneal Reflections for Eye Gaze Estimation via a Lightweight Network. In The Ninth International Symposium of Chinese CHI. 33–40
[99] Virmarie Maquiling and Sean Anthony Byrne and Diederick C. Niehorster and Marcus Nyström and Enkelejda Kasneci, Zero-Shot Segmentation of Eye Features Using the Segment Anything Model (SAM) (2023) https://arxiv.yezhisheng.me/abs/2311.08077
[100] Kim, T., Lee, EC.: Experimental Verification of Objective Visual Fatigue Measurement Based on Accurate Pupil Detection of Infrared Eye Image and Multi-Feature Analysis. Sensors. (2020). 20(17):4814. https://doi.org/10.3390/s20174814
[101] Gou, C., Zhang, H., Wang, K., Wang, F., Ji, Q.: Cascade learning from adversarial synthetic images for accurate pupil detection, Pattern Recognition, (2019). https://doi.org/10.1016/j.patcog.2018.12.014.

[102] Xiang, Z., Zhao, X. & Fang, A.: Pupil center detection inspired by multi-task auxiliary learning characteristic. Multimed Tools Appl (2022). https://doi.org/10.1007/s11042-022-12278-4
[103] Fuhl, W., Santini, T., Kasneci, G., Kasneci, E.: PupilNet: Convolutional Neural Networks for Robust Pupil Detection, arXiv (2016). https://doi.org/10.48550/arXiv.1601.04902
[104] Antonioli, L., Pella, A., Ricotti, R., Rossi, M., Fiore, MR., Belotti, G., Magro, G., Paganelli, C., Orlandi, E., Ciocca, M, et al: Convolutional Neural Networks Cascade for Automatic Pupil and Iris Detection in Ocular Proton Ther. Sensors. (2021). https://doi.org/10.3390/s21134400
[105] Vera-Olmos, F. J., Malpica, N.: Deconvolutional Neural Network for Pupil Detection in Real-World Environments. Lec. Notes in Computer Science, (2017). https://doi.org/10.1007/978-3-319-59773-7_23
[106] Sean Anthony Byrne and Virmarie Maquiling and Marcus Nyström and Enkelejda Kasneci and Diederick C. Niehorster, LEyes: A Lightweight Framework for Deep Learning-Based Eye Tracking using Synthetic Eye Images (2023) https://arxiv.org/abs/2309.06129
[107] Sabrina, S. Wibirama and I. Ardiyanto, "An Improved Pupil Detection Method under Eye-glass Occlusions," 2019 Asia Pacific Conference on Research in Industrial and Systems Engineering (APCoRISE), Depok, Indonesia, 2019, pp. 1-6, doi: 10.1109/APCoRISE46197.2019.9318871
[108] G. Fiala, Z. Ye and C. Steger, "Framework for Image Sensor Design Parameter Optimiza-tion for Pupil Detection," 2022 8th International Conference on Systems and Informatics (ICSAI), Kunming, China, 2022, pp. 1-6, doi: 10.1109/ICSAI57119.2022.10005532
[109] Solidworks 2022, https://www.solidworks.com/media/solidworks-2022-simulation
[110] Ahmed, F., Robinson, S., Tako, A.: Using the structred analysis and design technique (SADT) in simulation conceptual modeling. Winter Simu. Conf. (2015). 1038-1049. 10.1109/WSC.2014.7019963.
[111] Raju, M., Friedman, L., Bouman, T., Komogortsev, O.: Filtering Eye-Tracking Data From an EyeLink 1000: Comparing Heuristic, Savitzky-Golay, IIR and FIR Digital Filters, (2023), arXiv
[112] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. Med. Image Comp. and Computer-Assis. Interven. (2015). https://doi.org/10.1007/978-3-319-24574-4_28
[113] Zhou, Z., Siddiquee, MMR., Tajbakhsh, N., Liang, J.: UNet++: A Nested U-Net Architecture for Medical Image Segmentation. Deep Learn Med Image Anal Multimodal Learn Clin Decis Support (2018). https://doi.org/10.1007/978-3-030-00889-5_1.
[114] Badrinarayanan, V., Handa, A., Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling (2015). https://doi.org/10.48550/arXiv.1505.07293
[115] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A., Zhou, Y.: TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. (2021). https://doi.org/10.48550/arXiv.2102.04306

[116] Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking Atrous Convolution for Semantic Image Segmentation, (2017), arXiv
[117] Simonyan, K. Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, (2015) arXiv
[118] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, Deep Residual Learning for Image Recognition, 2015, 1512.03385, arXiv
[119] Howard A., Zhu, M. Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, (2017) arXiv
[120] Tang, S. Sun, T. Peng, J., Chen, G., Hao, Y., Lin, M., Xiao, Z., You, J., Liu, Y.: PP-MobileSeg: Explore the Fast and Accurate Semantic Segmentation Model on Mobile Devices. (2023). arXiv
[121] Ruan, J., Xie, M., Gao, J., Liu, T., Fu, Y.: EGE-UNet: an Efficient Group Enhanced UNet for skin lesion segmentation, arXiv:2307.08473v1 (2023)
[122] Hu, S., Liao, Z., Xia, Y.: Devil is in Channels: Contrastive Single Domain Generalization for Medical Image Segmentation (2023). arXiv
[123] OpenCV, Ellipse Fitting in a Least-Squares Sense, https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#gaf259efaad93098103d6c27b9e4900ffa, Access date: 2023-09-12
[124] Tonsen, M, Zhang, X., Sugano, Y., Bulling, A.: Labelled pupils in the wild: A dataset for studying pupil detection in unconstrained environments. Inter. Symp. on Eye Tracking Research & Applications (2016). https://doi.org/10.1145/2857491.2857520
[125] Wikipedia, Saccade, https://en.wikipedia.org/wiki/Saccade, Accessed: 2023-09-12.
[126] Lobão-Neto, R., Brilhault, A., Neuenschwander, S., Rios, R.: Real-time identification of eye fixations and saccades using radial basis function networks and Markov chains, Pattern Recog. Lett., (2022), https://doi.org/10.1016/j.patrec.2022.08.013
[127] Fuhl, W., Kasneci, G., Kasneci, E.: TEyeD: Over 20 million real-world eye images with Pupil, Eyelid, and Iris 2D and 3D Segmentations, 2D and 3D Landmarks, 3D Eyeball, Gaze Vector, and Eye Movement Types, (2021), arXiv, arXiv:2102.02115
[128] cuneytakinlar, UNet implementation, https://github.com/cuneytakinlar/pupil-ellipse-fit-error/blob/master/unet/unet_model.py, Access date: 2023-06.20
[129] ProfessorHuang, UNet++ implementation, https://github.com/ProfessorHuang/2D-UNet-Pytorch/blob/main/models/nested_unet.py Access date: 2023-06.20
[130] Beckschen, TransUNet implementation, https://github.com/Beckschen/TransUNet/blob/main/networks/vit_seg_modeling.py Access date: 2023-06.20
[131] PyTorch, DeepLabv3 implementation, https://github.com/pytorch/vision/tree/main/references/segmentation, Access date: 2023-06.20

[132] PaddleSeg, PP-MobileSeg implementation, https://github.com/PaddlePaddle/PaddleSeg/blob/release/2.8/paddleseg/models/pp_mobileseg.py Access date: 2023-06.20
[133] ShishuaiHu, CCSDG implementation, https://github.com/ShishuaiHu/CCSDG/blob/master/ccsdg/models/unet_ccsdg.py Access date: 2023-06.20
[134] JCruan519, EGEUnet, https://github.com/JCruan519/EGE-UNet/blob/main/models/egeunet.py Access date: 2023-06.20
[135] Amplitude-based traditional method source published by Wan et al, last accessed: 2024-01-13, https://github.com/zhwan934940943/CPRD/tree/master
[136] LEyes learning model source published by Byrne, Access date: 2024-01-13, https://github.com/dcnieho/Byrneetal_LEyes/tree/master
[137] Jaganathan, Niranjana & Mohamed, Mohd Hazmi & Md Pauzi, Ahmad Luqman & Mahayidin, Hasni & Hanapai, Ahmad & Wan Sulaiman, Wan Aliaa & Basri, Hamidon & inche mat, liyana najwa. (2024). Video head impulse test in stroke: a review of published studies. Frontiers in Neurology. 15. 1339039. 10.3389/fneur.2024.1339039.
[138] Ahmed, Z.A.T.; Albalawi, E.; Aldhyani, T.H.H.; Jadhav, M.E.; Janrao, P.; Obeidat, M.R.M. Applying Eye Tracking with Deep Learning Techniques for Early-Stage Detection of Autism Spectrum Disorders. *Data* 2023, *8*, 168. https://doi.org/10.3390/data8110168
[139] Moradizeyveh, Sahar et al. "When Eye-Tracking Meets Machine Learning: A Systematic Review on Applications in Medical Image Analysis." ArXiv abs/2403.07834 (2024): n. pag.
[140] Mestre, Clara & Gautier, Josselin & Pujol, Jaume. (2018). Robust eye tracking based on multiple corneal reflections for clinical applications. Journal of Biomedical Optics. 23. 1. 10.1117/1.JBO.23.3.035001.
[141] Liu J, Chi J and Yang Z (2024) A review on personal calibration issues for video-oculographic-based gaze tracking. Front. Psychol. 15:1309047. doi: 10.3389/fpsyg.2024.1309047
[142] Swirski, Lech. (2015). Gaze estimation on glasses-based stereoscopic displays.
[143] Model, Dmitri & Guestrin, Elias & Eizenman, Moshe. (2009). An Automatic Calibration Procedure for Remote Eye-Gaze Tracking Systems. Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference. 2009. 4751-4. 10.1109/IEMBS.2009.5334183.
[144] Mansourya, Mohsen & Steil, Julian & Sugano, Yusuke & Bulling, Andreas. (2016). 3D gaze estimation from 2D pupil positions on monocular head-mounted eye trackers. 197-200. 10.1145/2857491.2857530.
[145] Fodor, Á.; Fenech, K.; Lőrincz, A. BlinkLinMulT: Transformer-Based Eye Blink Detection. *J. Imaging* 2023, *9*, 196. https://doi.org/10.3390/jimaging9100196
[146] Shakeel, Muhammad & Bajwa, Nabit & Anwaar, Ahmad & Sohail, Anabia & Khan, Asifullah & Ur Rashid, Haroon. (2019). Detecting Driver Drowsiness

in Real Time Through Deep Learning Based Object Detection. 10.1007/978-3-030-20521-8_24.

[147] Zhou J, Li G, Shi F, Guo X, Wan P, Wang M. EM-Gaze: eye context correlation and metric learning for gaze estimation. Vis Comput Ind Biomed Art. 2023 May 5;6(1):8. doi: 10.1186/s42492-023-00135-6. PMID: 37145171; PMCID: PMC10163188.

[148] Coe, B.C.; Huang, J.; Brien, D.C.; White, B.J.; Yep, R.; Munoz, D.P. Automated Analysis Pipeline for Extracting Saccade, Pupil, and Blink Parameters Using Video-Based Eye Tracking. Vision 2024, 8, 14. https://doi.org/10.3390/vision8010014

[149] Jeongmin Hong, Joseph Shin, Juhee Choi, Minsam Ko; Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV),

[150] Pearce, A.J.; Daly, E.; Ryan, L.; King, D. Reliability of a Smooth Pursuit Eye-Tracking System (EyeGuide Focus) in Healthy Adolescents and Adults. J. Funct. Morphol. Kinesiol. 2023, 8, 83. https://doi.org/10.3390/jfmk8020083

[151] Meng Q, Tan X, Jiang C, Xiong Y, Yan B, Zhang J. Tracking Eye Movements During Sleep in Mice. Front Neurosci. 2021 Feb 25;15:616760. doi: 10.3389/fnins.2021.616760. PMID: 33716648; PMCID: PMC7947631.

[152] Hu Y, Wang H, Joshua M, Yang Y. Sensorimotor-linked reward modulates smooth pursuit eye movements in monkeys. Front Neurosci. 2024 Jan 9;17:1297914. doi: 10.3389/fnins.2023.1297914.

[153] Dunn, Matt & Alexander, Robert & Amiebenomo, Onyeka & Arblaster, Gemma & Atan, Denize & Erichsen, Jonathan & Ettinger, Ulrich & Giardini, Mario & Gilchrist, Iain & Hamilton, Ruth & Hessels, Roy & Hodgins, Scott & Hooge, Ignace & Jackson, Brooke & Lee, Helena & Macknik, Stephen & Martinez-Conde, Susana & Mcilreavy, Lee & Muratori, Lisa & Sprenger, Andreas. (2023). Minimal reporting guideline for research involving eye tracking. 10.3758/s13428-023-02187-1.

[154] Mokatren, M.; Kuflik, T.; Shimshoni, I. Calibration-Free Mobile Eye-Tracking Using Corneal Imaging. Sensors 2024, 24, 1237. https://doi.org/10.3390/s24041237

[155] Xun, Z., Baoqing, H., Dian, L. et al. Eye behavior recognition of eye–computer interaction. Multimed Tools Appl 83, 32655–32671 (2024). https://doi.org/10.1007/s11042-023-16763-2

[156] Carmen Bisogni, Michele Nappi, Genoveffa Tortora, Alberto Del Bimbo (2024) Gaze analysis: A survey on its applications, Image and Vision Computing, Volume 144, ISSN 0262-8856, https://doi.org/10.1016/j.imavis.2024.104961.

[157] Xiong, J., Zhang, Z., Wang, C. et al. Pupil localization algorithm based on lightweight convolutional neural network. Vis Comput (2024). https://doi.org/10.1007/s00371-023-03222-0

[158] G. Zhao, Y. Yang, J. Liu, N. Chen, Y. Shen, H. Wen, G. Lan (2023) EV-Eye: Rethinking High-frequency Eye Tracking through the Lenses of Event Cameras, Advances in Neural Information Processing Systems 36

[159] G. C. Ates, C. Coskunpinar, D. Tse, D. Pelaez, E. Celik (2024) Robust residual convolutional neural network based pupil tracking for low-computational power applications, Engineering Applications of Artificial Intelligence, Volume 133, Part A, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2024.108235

PUBLICATIONS FROM THE THESIS

Conference Papers

1. *(Oral Presentation)* İnanır Ş, Karaca, A., C. (2024) Pupil Detection Comparison of Deep Learning Models and Their Parameter-Reduced Versions, 9th World Conference on Computer Engineering, COMENG-2024, Antalya, Türkiye

Papers

1. *(Under Review)* İnanır Ş, Karaca, A., C. (2024) Real-Time and Accurate Pupil Detection Based Retro-Oriented Mind and Ellipse Trend Analysis, Machine Vision and Applications, Springer

2. *(To be Submitted)* İnanır Ş, Karaca, A., C. (2024) FIXEIN: A Real-Time, Accurate, Wearable and Modular Eye Tracking System Based-on ROM and PETA