Eye-tracking Using Image Processing Techniques and Deep Neural Network

Dilhani Thakshila Dodangoda

(20092404)

A thesis submitted to Auckland University of Technology in partial fulfilment of the requirements for the degree of Master of Philosophy Engineering (MPhil)

2024

School of Engineering, Computer & Mathematical Sciences

Primary Supervisor: Professor Hamid GholamHosseini

Abstract

Eye-tracking technology measures pupil dilation, eye movements, and blinking, providing insights into visual attention. It monitors eyes in real time, converting data into gaze points, vectors, and pupil positions. The system includes cameras, lighting sources, and computational power. Advanced image processing algorithms process camera inputs into meaningful data. Eye-tracking has applications in detecting fatigue, biometric verification, assessing attention levels, and improving human-computer interaction. It is widely used in marketing, advertising, assistive technology, usability studies, clinical research, healthcare, education, and automotive sectors.

The study initially considered various methods, and previous research indicated that deep learning (DL) algorithms outperformed other methods. Therefore, a DL approach was chosen due to its ability to analyze and predict complex patterns in images, text, and sound. The study utilized a Dilated Eye Network constructed and trained on eye-tracking datasets using a deep neural network (DNN) architecture. To enhance accuracy, the pre-trained network was fine-tuned with the same datasets. Additionally, the Circular Hough Transform was employed to detect circular shapes, such as eyes identified by edge detection in images. The combined approach significantly improved outcomes, successfully detecting eyes in 17 out of 20 test images, showcasing positive results across the testing dataset.

This study presents an innovative eye-tracking system that combines a Dilated Eye Network with the Circular Hough Transform to detect eye movements. By leveraging DL to analyze complex visual patterns, this approach outperforms traditional methods that often yield inadequate results, particularly in varied conditions. The integration of the Circular Hough Transform enhances robustness by effectively identifying circular shapes in imperfect images. Overall, this research demonstrates superior performance and potential for diverse applications in marketing, healthcare, and human-computer interaction.

This study makes a unique contribution to eye segmentation and tracking by introducing the dilated eye network, which improves accuracy while preserving image resolution and minimizing computational load. Additionally, the integration of semantic segmentation and the Circular Hough Transform enhances performance in noisy and occluded environments. The significance of this research lies in its potential applications for effective eye health monitoring in younger populations, demonstrating a practical advancement that leverages innovative techniques alongside pre-trained networks to address real-world challenges.

Attestation of Authorship

I hereby declaration that submission is my own work and that, to the best of my knowledge and belief, it contains no material previously or written by another person (except where explicitly defined in the acknowledgement), nor material which to substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Dilhani Thakshila Dodangoda

27 September 2024

Acknowledgements

I would like to express my sincere gratitude to several individuals and institutions whose support was indispensable in the successful completion of this project.

First and foremost, I am deeply thankful to Professor Hamid GholamHosseini for his exceptional kindness, patience, and dedicated mentorship. His profound knowledge and expertise have been a constant source of motivation and guidance throughout my academic career.

I am also grateful to Auckland University of Technology (AUT) for providing invaluable opportunities to enhance my educational journey.

I extend special thanks to the technical assistance and ICT help desk team for their crucial support in software installation and for providing an AUT laptop during a critical phase of my research.

I am profoundly appreciative of everyone who contributed their support and dedication, enabling me to overcome challenges and complete this thesis. The kindness and encouragement I received have made my time studying and living in New Zealand immensely fulfilling.

Additionally, I want to express my heartfelt appreciation to my spouse, parents, and friends. Their unwavering support has been a cornerstone throughout my educational pursuits, without which I could not have achieved this milestone.

Table of Contents

List of Abbreviations

Artificial Intelligent	AI
Area of Interest	AOI
Age-related Macular Degenerate	AMD
Autism Spectrum Disorder	ASD
Bayesian Gaussian Model	BGM
Corneal Reflection	CR
Computer Vision	CV
Complementary Metal Oxide	CMO
Convolutional Neural Network	CNN
Deep Learning	DL
Deep Neural Network	DNN
Deep Lab Cut	DLC
Deep learning Processor Unit	DLPU
Deep Learning Vector Quantization	DLVQ
Deep Integrated Neural Network	DINN
Eye Variance Filter	EVF
Electrooculography	EOG
Electroencephalogram	EEG
Fast Fourier Transform	FFT
Field Programmable Logic Array	FPLA
Feedforward Artificial Neural Network	F-FANN
Gated Recurrent Unit	GRC
Grey Level Co-occurrence Matrix	GLCM
Human Computer Interaction	HCI
Infrared	IR
Identification by Random Forest	IRF
Inner Corner Pupil Center Vector	ICPCV
Kanade Lucus Tomasi	KLT
K-Nearest Neighbours	KNN
Local Binary Patterns	LBP
Machine Learning	ML
Multiprocessor System on Chip	MPSOC
Region of Interest	ROI
Recurrent Neural Network	RNN
Random Forest	RF
Support Vector Machine	SVM
Sensor Motoric Instruments	SMI
Stochastic Gradient Descent with Momentum	SGDM
Short-time Fourier Transform	STFT
Typically Developing	TD
Extended Reality System	ERS
You Only Look Once	YOLO

List of Figures

List of Tables

Chapter 1: Introduction

The primary objective of eye movement analysis in eye-tracking research is to discern and categorise various events such as fixations, saccades, post-saccadic oscillations, and smooth pursuits. These events are identified based on established criteria for fixation durations, saccadic amplitudes, and velocities. Eye-tracking refers to the technique of monitoring an individual's gaze movements to precisely determine where and for how long their eyes are fixated. This method is essential for maintaining visual focus on an object by ensuring it remains centrally located on the fovea, optimizing visibility and attention. Technology plays a crucial role in facilitating the observation and measurement of eye movements, pupil dilation, gaze fixation points, and blinking patterns. These measurements provide valuable insights into the allocation of visual attention, identifying stimuli of interest and those disregarded by participants. Eye-tracking finds widespread applications across disciplines such as marketing, engineering, cognitive science, psychology, and neurology. Additionally, it is instrumental in enhancing human-computer interaction, particularly for individuals with disabilities who utilise gaze-based interfaces for communication. The challenge of analyzing eye movements is addressed by classifying raw eye-tracker data into discrete events, a process that benefits from automated algorithms such as those implemented in MATLAB. These algorithms enable objective, efficient, and reproducible classification compared to manual coding methods. By automating this process, researchers can conduct comprehensive analyses of eye movement patterns across diverse populations and experimental condition [1].

Eye-tracking research encounters numerous challenges, particularly in the detection of events during experiments. One significant issue arises from potential disturbances in recorded signals from subjects, which can vary widely and complicate signal analysis. This variability necessitates the development of robust methodologies capable of accommodating diverse signal characteristics and disruptions, as well as managing multiple eye trackers and participants simultaneously. Effective event detection is critical in studies of visual perception and the oculomotor system, offering valuable insights into human cognition. This significance is particularly pronounced in research involving infants, who lack the ability to comply with verbal commands or provide explicit responses. By examining infants' eye movements in response to visual stimuli, researchers can elucidate fundamental aspects of perceptual development, such as color vision processing. Technological advancements have played a pivotal role in enhancing the precision and scope of eye-tracking methodologies. Modern video cameras and hardware capable of digitizing and storing video sequences facilitate sophisticated software analysis, supported by specialised algorithms tailored for this purpose [2].

1.1 What is Eye-tracking?

Eye-tracking devices provide precise measurements of an individual's gaze direction, points of interest, and durations of attention on specific areas. Central to this analysis are the concepts of "fixation" and "saccade." Fixation refers to periods during which the gaze remains relatively stable, facilitating detailed visual processing, interrupted by rapid eye movements called saccades, which facilitate shifts between fixated points. In reading tasks, proficient readers typically exhibit longer saccades as they efficiently traverse text, whereas less skilled readers tend to have shorter saccades and briefer fixations. Eye-tracking software maps these saccades, offering a graphical depiction of the eye's movement trajectory across a page. This technology yields insights into reading behaviors and cognitive processes, illustrating how individuals engage with textual content based on their patterns of eye movement.

Eye tracking typically refers to the measurement of either the point where someone is looking or the movement of an eye relative to the head. An eye tracker is a device designed to measure the positions and movements of the eyes. These measurements are conducted by an eye tracker, which continuously monitors the positions and motions of the eyes. Eye-tracking devices utilise near-infrared (IR) light emitted by high-definition cameras to illuminate and track the eye. By focusing on the eye's position relative to the transverse visual plane of the head, specifically the principal (horizontal) retinal plane, eye trackers achieve precise focus determination. This technology enables the observation and analysis of minute eye movements and visual activities by capturing the eye's location multiple times per second. Eye-tracking devices produce frequent images due to their rapid image capture capabilities, facilitating detailed analysis of eye movements and visual behaviors in research settings [3].

1.2 How Does Eye-tracking Work?

The eye tracker emits near IR light, which is reflected in the eyes. Cameras within the eye trackercapture these reflections. Through the use of filtering and computational techniques, the eye tracker can accurately determine the specific location where an individual is directing their gaze [4].

1.3 Types of Eye-tracking

Eye-tracking devices record the direction and intensity of the participant's gaze, specifically targeting the pupil. There are various types of eye-tracking technologies currently in use. Researchers choose the most appropriate type based on the objectives of their study.

1.3.1 On-screen Records

An effective method for studying how individuals use electronic devices involves on-screen eye-tracking. This technique captures the gaze movements of users as they interact with digital content on phone screens or monitors. Research in digital marketing, usability testing, and user experience design derives substantial benefits from on-screen tracking. These trackers are remote devices capable of connecting to displays or laptops. They utilise cameras, which may be either external or integrated into devices, to monitor eye movements. The data collected is then visualised using graphs and heat maps, revealing patterns of attention.

1.3.2 Transportable

If portability and flexibility are priorities, portable eye-tracking devices are recommended. These devices excel in various contexts, including market research, usability testing, and field research. They feature user-friendly interfaces and lightweight components, making them easy to operate and transport. Beyond laboratory settings, portable devices can be swiftly deployed to capture accurate eye movement data. Examples include virtual reality headsets equipped with integrated eye-tracking and goggles with built-in eye-tracking capabilities.

1.3.3 Webcam

Webcam eye-tracking offers accessibility. It is ideal for straightforward applications and obtaining general insights into user behaviour without requiring specialised equipment. This method finds utility in internet market research, usability studies, and basic research. Unlike traditional eye-tracking systems that use IR illuminators to pinpoint gaze location, webcam tracking relies on a standard webcam for its operations [5].

1.4 Which Light Source is Suitable for Eye-tracking Application?

The preference for an IR light source arises from the necessity to accurately delineate the pupil and detect corneal reflections, essential for precise evaluation of gaze direction. A complete IR light source typically includes a lamp, lamp housing, power supply, and the emission of radiation. Compared to conventional camera light sources, IR-emitting sources offer higher contrast, crucial forachieving the necessary accuracy in eye-tracking.

Examples of IR-emitting sources include the Sun, Earth (IR radiation), electric heaters, remote controls, IR lamps, and thermal imaging cameras. IR light travels directly to the eye's pupil

and reflects off the iris. In contrast, visible light often generates uncontrollable specular reflections. Moreover, since IR light is invisible to humans, it does not interfere with eye-tracking procedures[6].

1.5 Applications

Eye-tracking is a highly valuable tool for researchers across a broad spectrum of behavioural studies. Its applications span diverse fields such as healthcare, psychology, marketing, engineering, education, and gaming. Furthermore, eye-tracking has the potential to enhance human-computer interactions by enabling navigation and control through eye movements.

1.5.1 Market Research

Eye-tracking is highly valued in market research due to its ability to provide comprehensive and unbiased insights into customer behaviour and decision-making processes. It allows researchers to analyse how consumers select products by identifying which product features naturally capture their attention and which ones they disregard. This technology offers precise information on consumer behaviour, which proves invaluable for strategies related to product placement, branding, packaging, and in-store advertising design.

1.5.2 User Experience

A highly effective approach to evaluating user experience is to adopt the user's perspective. Eye-tracking serves as a powerful tool for examining how platforms and services are utilised, revealing both design issues and usage patterns that may not have been apparent during the development phase.

1.5.3 Scientific Investigation

Examining visual activity through methods like eye-tracking can provide valuable insights into various aspects of development, learning styles, and cognitive disorders. This approach is particularly useful for studying patients with conditions such as Alzheimer's disease, Parkinson's disease, schizophrenia, autism, depression, and others. Similarly, eye-tracking can aid in diagnosing and researching dyslexia and other reading or learning disabilities. This technology enables researchers and clinicians to observe and analyse how individuals process visual information, offering insights that contribute to understanding these disorders and improving methods for diagnosis and treatment.

1.5.4 Human Resources

Eye-tracking offers a valuable method to analyse staff behaviour within an organisation. It provides unique insights into rapid, often subconscious activities that can be studied and used to develop training materials for employees. By enhancing training processes and identifying potential risks and operational inefficiencies, businesses can enhance productivity and save time.

1.5.5 Safety Applications

In 2017, the development of the Deep Integrated Neural Network (DINN), which combined CNNs and DNNs, aimed to analyse driver photographs to classify eye states and assess levels of tiredness. The main objective was to monitor blinking patterns—frequency, duration, and timing of blinks—to evaluate driver fatigue through eye-tracking assessments. Trained on data from over 2,400 subjects, DINN achieved accurate identificationof subjects' states between 96% and 99.5% of the time, surpassing the performance of most other AI models which typically operated above 90%. This technology holds potential for detecting driver tiredness and enhancing road safety measures.

1.5.6 Game Theory Applications

In a 2019 study, researchers developed a Convolutional Neural Network (CNN) specifically designed to recognise individual chess pieces. The CNN was trained using eye-tracking data contributed by 30 chess players of varying skill levels. This data enabled the CNN to analyse where on the chessboard a player was directing their gaze. Using gaze estimation, the CNN generated a saliency map that highlighted the specific areas of the chessboard that drew the player's focus. Subsequently, leveraging its understanding of the chessboard layout and the positions of the pieces derived from the saliency map, the CNN made predictions about the players' next moves. Importantly, the study found that the CNN's predictions, guided by the saliency maps, exhibited higher accuracy in predicting the players' next moves compared to random selection. This neural network system represents a significant advancement in using DL techniques to enhance chess analysis. By integrating eye-trackingdata and leveraging CNN capabilities, the study demonstrates how AI can effectively interpret and predict human decision-making processes in strategic contexts such as chess.

1.5.7 Aviation Application

Research has extensively explored the application of eye-tracking technology in enhancing flight safety across various domains. This technology enables the comparison of pilots' eye movement

patterns and the duration of their focus to evaluate candidate performance, assess pilots' capabilities, and measure crew attention and situational awareness. Eye-tracking has been particularly studied in the context of multi-purpose displays in military aircraft and helmet-mounted display systems (HMDS). Investigations have focused on assessing its effectiveness in HMDS for tasks such as locking onto and maintaining focus on head-up targets. Despite its promising potential, some members of the aviation community advocate for further development of both hardware and software components to optimise performance and reliability. Studies conducted in simulator environments have shown that integrating eye-tracking technology can lead to significant reductions in perceived cognitive load and improved response times compared to current systems utilizing multi-functional displays. These findings underscore the potential of eye-tracking to enhance operational efficiency and situational awareness in aviation contexts, paving the way for continued advancements in this field.

1.5.8 Engineering Applications

In recent years, there has been a growing interest in the application of eye-tracking technologies within empirical software engineering research. These technologies are being used alongside advanced data analysis methods to investigate the extent to which subjects comprehend software engineering concepts. Specifically, researchers are examining the ability of individuals to grasp software engineering diagrams and business process models. The research involves the use of various eye-tracking metrics such as fixation duration, scan path analysis, scan path recall, and fixations on areas of interest or relevant regions within diagrams and models. These metrics provide valuable insights into how effectively individuals understand and interpret visual representations in software engineering. The insights gleaned from these studies have significant implications for enhancing various aspects of personnel-related behaviour within software engineering. They contribute insights into topics such as working memory capacity, cognitive load management, learning styles, and the strategies employed by software engineers and modelers to improve the comprehensibility of diagrams and models [7].

1.5.9 Diagnosing Lazy Eye

Amblyopia, commonly known as "lazy eye," causes impaired vision in one or both eyes due to a lack of coordination between the eye and the brain. Children with this condition may experience vision problems that they often become accustomed to and may not communicate to their parents. Consequently, amblyopia can remain undetected for years. Early intervention is essential as treatment can enhance vision and improve communication between the eye and the brain. Delayed or missed diagnosis may result in permanent vision loss later in life. Amblyopia typically affects one eye more

severely, although it can impact both. It is prevalent, affecting approximately two to three out of every 100 children. The key to normal vision development is ensuring equal visual acuity in both eyes, highlighting the importance of identifying and addressing lazy eye early in infancy or childhood. Eye-tracking is a diagnostic technique used by clinicians to assess amblyopia. During this procedure, the clinician moves a target horizontally, vertically, and rotationally, instructing the patient to track it with their eyes. Optometrists may also use quantitative tools to objectively measure eye movement and coordination [8].

1.6 Benefits of Eye-tracking

The capacity of eye-tracking to objectively and comprehensively record and analyse visual behaviour is among its most significant advantages. It is challenging to ask individuals navigating grocery aisles to quantify, let alone recall, the duration they spent looking at each item, where their gaze fell, or which marketing materials captured their attention most effectively. There are additional advantages of precise eye-tracking, some of which are outlined below:

- Unbiased, objective, and quantifiable data is gathered, which eliminates research participants having to remember or explain. This also prevents research participants from making assumptions about details and giving inaccurate information.
- Reveals subconscious behaviour: researchers can learn more about our innate behaviours.
- Allows for natural behaviour: eye trackers are discreet.
- Versatile and mobile: can apply it to practically any situation or place.
- Offers a thorough level of information. The data can offer a detailed level of granularity for in-depth analysis, depending on the equipment and software used.
- Self-explanatory: it may be used to track procedures and actions that are challenging to explain or articulate.
- Provides graphic visualisation: it makes use of heat maps and graphs to display eye-tracking data, making it possible to see how people interact with one another and react to various stimuli.
- Adds value to other biometric data: by offering more details regarding the factors that contributed to physiological reactions [9].

1.7 Research Questions

- How can algorithms in MATLAB be optimised to enhance the accuracy of eye-tracking systems across diverse environmental conditions and demographic groups?

- What are the most effective methodologies for designing and analysing each stage of eye-tracking algorithms, including face detection, face segmentation, and eye detection, to ensure reliable performance and robustness?
- What characteristics define an ideal dataset for training and validating eye-tracking algorithms, incorporating separate subsets for training, testing, and validation, to ensure generalizability and accuracy?
- How can eye-tracking algorithms developed in MATLAB effectively handle images containing background objects around the eyes, ensuring precise eye movement detection and analysis while mitigating interference from extraneous visual elements?

1.8 Objectives of the Study

This study aims to explore and enhance eye-tracking capabilities through the integration of image processing techniques and DNNs. By leveraging advanced computational methods, the research seeks to improve the accuracy and reliability of eye movement detection. The following objectives outline the specific goals of this research:

- To Examine Eye-Tracking Technologies: Investigate the principles and methodologies underlying eye-tracking technology and its applications across various domains such as marketing, healthcare, and human-computer interaction.
- To Develop a DNN for Eye Tracking: Construct a Dilated Eye Network utilizing DL architectures to effectively detect and analyze eye movements from visual data.
- To Evaluate the Performance of the DNN: Assess the accuracy and efficiency of the DL model in comparison to traditional image processing methods in detecting eye movements.
- To Fine-Tune the Pretrained Neural Network: Optimize the performance of the Dilated Eye Network through fine-tuning with comprehensive eye-tracking datasets, aiming to enhance detection accuracy.
- To Integrate Circular Hough Transform with DL: Combine the Circular Hough Transform with the DL model to improve the detection of circular features, such as human eyes, in diverse imaging conditions.
- To Validate the Integrated Approach: Test the effectiveness of the combined methodology on a curated dataset to determine improvements in eye detection accuracy and reliability.
- To Contribute to the Field of Eye Tracking: Provide valuable insights and findings that advance

the understanding of eye-tracking technology, promoting future research and practical applications in this domain.

This study aims to develop eye-tracking algorithms using MATLAB methods capable of accurately tracking eyes across diverse images containing background objects near the eyes. This research seeks to enhance the robustness and applicability of eye-tracking systems, thereby advancing their utility in various contexts, from marketing and usability testing to optimizing digital user experiences [10].

Chapter 2: Literature Review

2.1 Machine Learning (ML) Based Techniques

The referenced study highlighted the significant advantages of real-time gaze monitoring for applications in neuromarketing and psychophysics research. A DL approach utilising videoframes from affordable cameras achieved a median error of approximately one degree of visual angleduring evaluation. This method employed a shallow neural network and DeepLabCut (DLC) to detectfacial landmarks crucial for determining gaze direction towards specific regions. The study also underscored the financial constraints associated with traditional ocular perceiver-tracking technologies, such as appearance-based and model-based techniques. Model-based methods estimate gaze positions using IR light patterns and a three-dimensional ocular perceiver model. In contrast, appearance-based approaches leverage ML to map facial landmarks necessary for gaze estimation using data obtained from video cameras. Moreover, an investigation employing appearance-based ocular perceiver-tracking techniques in conjunction with IR light sources andcameras demonstrated the ability to accurately identify participants' gaze from recorded video frames, even across varied poses. These findings underscore the utility of affordable, ML drivengaze-tracking methodologies in facilitating precise gaze analysis for applications in consumer behaviorresearch and cognitive studies within neuromarketing and psychophysics [11].

Researchers have developed DL methodologies aimed at tackling challenges within computer vision (CV), proposing the implementation of ensemble DNN architecturesspecifically for the categorisation of eye movements. This approach involves a series of developmentalstages, incorporating deep CNNs to capture visual characteristics and recurrent networks, notably Gated Recurrent Units (GRUs), to model temporal features. The studyintroduced a bespoke dataset-driven classifier designed for tracking eye movements. This methodology encompasses several critical stages: data preprocessing, annotation of eye movement patterns, recognition of eyeball positions, and compilation of images containing both eyes. The datasetcomprises frames captured during movement sequences, facilitating the classification of temporal series. The study focused on two distinct types of eye movements: the idle state and specific forms such as the Z and V shapes. To effectively discern these movements, the model utilises CNNs for visual feature extraction and integrates GRUs for predicting temporal patterns within an ensemble DL framework. Following extensive training comprising 1000 sessions, the model achieved a notable accuracy rate of 92%. This high level of accuracy underscores its practical utility for real-world applications requiring precise classification and analysis of eye movements [12].

A previous study introduced a comprehensive training methodology for detecting eye movements using CNNs. The study utilised 64 x 64 resolution images depictingboth closed and open eyes for capturing eye gaze. CNNs employ convolutional processes upon receiving input images. Subsequently, the network constructs layers for categorizing eye movements through convolutional layers and local structures. Each convolutional layer in the CNN is followed byan average pooling layer and batch normalization. The primary objective of the study was real-time detection of eye locations in low-resolution images using CNNs. The methodology extracts structuralproperties from the input data, thereby enhancing the reliability of emotion recognition. Despite achieving a detection accuracy of 0.97, surpassing existing methods, the challenge lies in addressing varying levels of difficulty such as sub-spectrum noise reduction, wrinkles, facial orientations, and postures. Future improvements are expected to focus on refining these aspects to further enhance detection accuracy and robustness in real-world applications [13].

A recent study employed a DL methodology combining directional gradient histogram (HOG), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM) neural networks toidentify eye movement behaviour from front-facing camera video footage. This approach was subsequently applied to develop a practical Human-Computer Interaction (HCI) application. Initially,the system detects and tracks the user's face. It then identifies the binocular area using the coordinatesof four key points located at the corners of the eyes. A SVM model is employed to detect closed eyes.Following this, eye movement is estimated by analyzing the center of the eye position between consecutive frames. Subsequently, a sequence of video frames is input into an LSTM network for prediction. The recognition of eye movement behaviour and subsequent triggering of computer commands conclude the interaction process. During testing with 20,000 samples from a self-generated dataset, approximately 10% of which were negative cases, the accuracy of predicting eye movement behaviour and dynamically identifying blinks exceeded 95%, reaching 99.3% [14].

Eye-tracking systems are crucial for accurately estimating the point-of-gaze in extended reality (XR) applications. These systems typically rely on IR light-emitting diodes and an IR camera to monitor eye movements, though they often encounter challenges such as device slippage. Oneeffective method to determine the point of gaze involves identifying two or more corneal reflections in images of the eye. Most eye trackers incorporate multiple light sources to ensure each gaze positiongenerates at least one pair of reflections. In XR systems, the task of identifying and matching cornealreflections has been tackled by developing a fully CNN based on the UNET architecture. This neural network successfully identified and matched 91% of visual reflections within

a virtual reality headset, achieving processing speeds ten times faster and consuming 33 times less memory compared to previous methods. Our approach yielded an average mean absolute gaze inaccuracy of 1°, signifying a substantial advancement over existing learning-based XR eye-tracking systems [15].

A recent academic study explored techniques for detecting eye movements under challenging conditions such as frontal face rotation, varying lighting conditions, and obstacles like glasses. The research introduces a sophisticated classifier termed the eye variation filter (EVF), which employs a coarse-to-fine approach to identify ocular regions. The study utilises a CNN architecture primarily as a feature extractor. The final classification is executed by a SVM classifier. During the training phase, three separate models are trained using a boosted cascade face detector to initially localise the facial region. Furthermore, the study strategically restricts the eye search area to the upper half of the detected facial region for enhanced efficiency. To expedite the identification process, the study integrates eye variation filters with SVM and CNN, creating a hybrid approach to eye recognition. This methodology aims to improve accuracy and resilience against environmental and physical challenges commonly encountered in practical eye-tracking applications [16].

The study demonstrated the development of a Deep Gradient CNN-DGCNN aimed at classifying track data using eye-tracking signals. Initially, the researchers appliedpreprocessing techniques and standard signal processing tools such as the Kalman filter, Hamming windowing, Short-Time Fourier Transform (STFT), and Fast Fourier Transform (FFT). Subsequently, images were generated from the eye movement and tracking data. The experimental dataset comprisedeye-tracking data collected from 16 individuals, each exposed to four affective stimuli (nervous, calm, joyful, and sad). A training framework based on CNNs was implemented next. Following this, the performance of the DGCNN model was compared with that ofDecision Tree (DT), Bayesian Gaussian model (BGM), and k-nearest neighbour (KNN) classifiers using metrics such as true positive rate (TPR) and false negative rate (FPR). Finally, the training structure of the DNN was optimised by adjusting gradient definition, learning rate, loss function, and mini-batch size. The effectiveness of the proposed approach in analysing eye movementand tracking signals was demonstrated through a predictive classification matrix, achieving anaccuracy rate exceeding 87.2% [17].

The research under discussion focuses on the objectives of tracking eye movements and determining eye position using eye-tracking technology. The study presents a deep-learning-based approach for identifying pupil centres from webcam images. Unlike conventional methods that train object detectors directly on objects of interest, this approach trains the detector using the area surrounding the eyes.

The study highlights significant improvements in overall detection accuracy compared to previous methodologies. It adopts the compact 23-layer YOLOv3 architecture, chosen for its capability to achieve real-time processing without the need for a GPU accelerator. The detectors were trained using the PUPPIE dataset, with optimal detection accuracy settings determined through experimentation. The paper reports that the method achieves a high degree of accuracy, with errors observed in less than 1.76% of images featuring a constricted pupil. The overall accuracy exceeds 98.24% across all images. Furthermore, the study introduces a novel post-processing technique for accurately determining the centre of the pupil based on the identified areas. This approach demonstrates the potential for practical applications in real-time eye-tracking technology [18].

Another recent study in DL has examined visual data pertaining to human eyes, specificallyfocusing on the detection of individuals experiencing emotional distress. Eye-tracking technology hasgained prominence in various technological applications, spanning fields such as virtual reality, psychology, healthcare, and education. However, commercially available eye-tracking devices are often expensive and require users to maintain complete head immobility to accurately track gaze direction. In response to these limitations, this study proposes an eye-tracking system based on DNNs. This approach eliminates the need for costly hardware and allows for operation without the requirement of keeping the user's head still. Comparative evaluations conducted in the study indicate that the proposed method outperforms existing eye-tracking estimation algorithms currently available on the market [19].

This study describes an eye detection system integrated into a Zynq XCZU4EV UltraScale+ multiprocessor system-on-chip (MPSoC). The system is applied within an architecture for remote iris identification, requiring rapid capture of high-resolution photographs as input. A critical requirement is that the detector outputs only images of focused eyes, filtering out those significantly affected by defocus blur, given the high frequency of eye region detections per second. To achieve this, the network is trained exclusively on perfectly focused eye images, aimed at distinguishing them from out-of-focus counterparts. Utilizing the neural network's capability to process multiple input channels, the system inputs both the grey level image and a high-pass filtered version—commonly used to assess iris focus—into the CNN. The CNN is implemented using the DL Processor Unit (DPU) of the MPSoC, which integrates various cores. DPUs facilitate efficient acceleration of CNN inference, offering extensive DL functionalities compared to earlier FPGA-based CNN hardware designs. Experimental validation in real-world scenarios involving subjects in motion has effectively demonstrated the system's ability to accurately detect only in-focuseye images. The prototype module continuously outputs streams of detected eyes as 640 × 480 resolution photographs, utilizing a CMOS digital image sensor capable of 16-pixel imaging.

Impressively, the module successfully discards up to 95% of improperly focused eye images from the input photographs, underscoring its robust performance in practical applications [20].

This research employs feedforward artificial neural network (F-FANN) and DL vector quantization (DLVQ) approaches to classify ocular conditions. DLVQ's capability to learn from a constrained codebook makes it more effective than standard vector quantization (VQ) in classificationtasks. When tested on an EEG-audio retrieval task, DLVQ demonstrates promising performance, whileF-FANN accurately identifies EEG-audio signals indicating whether the eyes are open or closed. In direct comparison, DLVQ surpasses F-FANN in terms of classification accuracy, F-score, precision, recall, and overall performance metrics. To conduct the study, a labelled dataset was essential for training the classifiers and evaluating their effectiveness. This dataset enabled the classifiers to establish relationships between attribute values of each instance and their corresponding labels. The evaluation process involved comparing classifier predictions against the ground truth labels ofinstances used in testing. The EEG data used comprised 150 recordings, each lasting approximately 24 seconds, collected from 27 individuals. These signals were captured using a 16-channel V-AMP amplifier at a sample rate of 1024 samples per second, with impedance maintained below 5 kΩ per channel. The study's focus was to evaluate DLVQ and F-FANN performance specifically for EEG signal classification in predicting the subject's eye condition. The neural network architecture of F- FANN included fully connected layers with varying numbers of neurons across multiple hidden layers. In contrast, DLVQ was structured to incorporate a codebook for learning and classification purposes. Activation functions such as ReLU were used to accelerate learning and mitigate the vanishing gradient issue, while sigmoid activation ensured appropriate output constraints. The study found that while DLVQ generally outperformed F-FANN in overall accuracy metrics for EEG signal classification, F-FANN exhibited higher precision in predicting eyes open states and comparable recall in predicting eyes closed states, resulting in an identical F-score of 91% for both conditions. Furthermore, the average prediction times per data instance were measured: F-FANN recorded 0.009 milliseconds, while DLVQ averaged 0.074 milliseconds. In conclusion, this research highlights DLVQ's superiority over F-FANN in EEG signal classification tasks related to eye state detection, underscoring the effectiveness of DLVQ's codebook learning approach in achieving higher overall performance metrics [21].

A recent study investigated the detection of eye movements using advanced DL algorithms,specifically focusing on recurrent neural network (RNN) architectures. The research aimed to enhancedetection methods and support systems in this domain. The methodology involved a structured approach to data acquisition, encompassing preprocessing, labelling, encoding, feature selection, handling missing values, scaling, partitioning, and addressing data imbalance. Participants were

positioned 60 centimetres away from a 17-inch screen illuminated with IR light reflection. The stimuli used included static items, such as objects and cartoon characters, as well as dynamic stimuli like balloons and animated characters. Among the models assessed, the Long Short-Term Memory (LSTM) network demonstrated the highest accuracy, achieving 98.33%. Following closely was a hybrid CNN-LSTM model, which achieved an accuracy of 97.94%. These findings underscore the potential of sophisticated computational frameworks to significantly improve the diagnosis andclassification of Autism Spectrum Disorder (ASD) through precise eye movement detection methods.In conclusion, the study highlights the effectiveness of RNN-based approaches, particularly LSTM and hybrid CNN-LSTM models, in accurately detecting eye movements across various visual stimuliscenarios. This research contributes towards advancing diagnostic and assistive technologies for ASD[22].

Another proposed method involves a visual saliency model based on DNNs, aimed at enhancing prediction performance by leveraging large-scale training datasets such as eye tracking data.Traditional low-level feature-based models often suffer from issues related to low accuracy and scalability. Moreover, conventional eye-tracking methods are constrained by expensive hardware, labor-intensive procedures, and poor user experiences. To address these challenges, this study introduces a novel approach utilizing crowdsourcing for collecting eye-tracking data. Crowd workersself-report their gaze positions and recall their gaze patterns, enabling the acquisition of significant amounts of data at a lower cost. Parameter optimization techniques were employed to enhance the accuracy of gaze data, achieving a precision of 1° of visual angle, surpassing existing crowdsourcing methods by 3.6%. Using a fully CNN as its core architecture, the visualsaliency model was developed using a dataset gathered from a website employing crowdsourced gazedata. Evaluation results demonstrated a substantial improvement in prediction accuracy, with an increase of 44.8% after training on crowdsourced data compared to previous models of visual saliency.The model not only outperformed existing benchmarks but also offered a practical tool for website designers to assess and enhance visual designs. In practical application, the redesigned website basedon the visual saliency model received significantly higher ratings, showing an improvement of 8.2% compared to its original design. This underscores the effectiveness of leveraging crowdsourced eye- tracking data and DL techniques in advancing visual design assessment and optimization methodologies [23].

Another research initiative developed an eye-movement-based computer control program designed to assist individuals with spinal cord injuries or neuromuscular disorders in using computers through eye movements. The program utilises a mini-USB camera integrated into lightweight eyeglass frames to accurately track the user's pupils, enabling control over activities such as web browsing and video

viewing. The camera, with a diameter of 3.9mm, is securely mounted on standard eyeglass frames using a 120cm wire and a 3D printed camera holder, allowing for easy detachment. The program itself was developed using PyCharm and Python, optimised to run on an AMD processor with 16GB RAM, independent of any operating system installation. Operating at 22-26 frames per second with a resolution of 1280x720 pixels, the plug-and-play USB camera captures frames to locate the pupil accurately using DL techniques. Traditional image processing methods were found inadequate across varying lighting conditions, prompting the adoption of DL for robust pupilrecognition. The program involves initial screen calibration at nine points to ensure precise functionality, activating features after a two-second engagement period and confirming selections witha click. A user-friendly interface allows for easy function customization and continuous monitoring of pupil detection, facilitating natural interaction as the cursor moves according to pupil and cursor movements. In summary, this research introduces an innovative approach using eye-tracking technology and DL for computer interaction, tailored to enhance accessibility for individualswith motor impairments. The system's design ensures reliability and usability, supporting users in performing essential computer tasks effectively through intuitive eye movements [24].

In another research study, 59 school-age participants were exposed to films and images related to social cognition, tailored to their age group. This approach facilitated a simplified and highly accurate diagnostic process through visual representation. Furthermore, the study explored potential correlations between eye movement dynamics and the severity of ASD. The results suggest that ML, visualization techniques, and eye-tracking have the potentialto develop an impartial tool for ASD screening. The study utilised CNN models to train images for classification purposes, achieving notably high levels of accuracy. This streamlined diagnostic procedures, indicating promising applications of ML, visual aids,and eye-tracking in ASD screening and assessment practices [25].

A further experiment was conducted to evaluate event detection using data collected from the SMI HI Speed 1250 system, a high-speed eye tracker developed by Senso Motoric Instruments. The study primarily focused on categorising fixations, saccades, and post-saccadic oscillations. Central to the analysis was the comparison of different algorithms, assessing their alignment with human coders through detailed sample-by-sample comparisons. Researchers identified optimal threshold values and examined the impact of adjusting these values on threshold-based algorithms. The same dataset was used to contrast the performance of human coders with that of the event detection algorithms. Additionally, various event detection algorithms were evaluated and compared, including threshold-based techniques and those integrating ML and DL methods. The findings from the evaluation reveal significant differences in classification outcomes. Notably, algorithms

employing random forests (RF) and CNNs demonstrated superior performance compared to threshold-based techniques [26].

In the fields of neuromarketing and psychophysics, webcam video frames are utilised in a deep-learning-based approach for real-time gaze monitoring. This method employs a shallow neuralnetwork to analyse the point of gaze and accurately locate facial landmarks, achieving a median measurement accuracy of approximately one degree. During experiments, participants track a movingdot on a screen for one minute, and frames with lower likelihood scores are refined or discarded using specific algorithms. The model is trained using the PyTorch GazeNet dataset, which is tailored for gaze estimation tasks. GazeNet, the DL network utilised, fully connects its input and hiddenlayers through a sigmoid activation function. The model optimises its parameters by minimizing the loss function through Stochastic Gradient Descent, a common optimization approach in DL.To enhance the accuracy of gaze estimation, the model is exposed to recordings of subjects followinga circular path through a maze. These recordings provide crucial data on head posture and eye movements, which are essential for refining the model's predictions. Additionally, 50 randomly selected frames with manually labeled facial landmarks contribute to training the model and improvingtracking precision. Overall, continuous training and refinement of DL models on such dataare essential for advancing the accuracy and reliability of real-time gaze monitoring systems in applications like neuromarketing and psychophysics [27].

Another research study proposed a method for detecting strabismus using eye-tracking data and CNNs. In this study, participants were instructed to gaze at nine specific points displayed on a screen, while their eye movements were monitored using specialized equipment. From the recorded data, three gaze deviation (GaDe) maps were generated, whichconsidered the accuracy of fixation and the central points of both eyes. These GaDe maps were then combined to create a comprehensive GaDe image, which was subsequently inputted into a CNN trained on ImageNet. The CNN processed the GaDe image and extracted a feature vector that characterised the gaze patterns of the participant. Based on this feature vector, the subject was classified as either strabismic (having strabismus) or normal (not having strabismus). According to thestudy findings, this approach proved to be successful in detecting strabismus. By leveraging CNNs and eye-tracking technology, the method provided a robust and automated means to assess and classifyindividuals based on their gaze behavior, thereby aiding in the diagnosis of strabismus more effectivelythan traditional methods. This research highlights the potential of combining advanced ML techniques with precise eye-tracking data to enhance diagnostic capabilities in ophthalmologyand related fields [28].

Another researcher has described three distinct artificial intelligence methods aimed at early detection of ASD, employing ML, DL, and a hybrid approach that combines both methodologies. The first method utilises neural networks, specifically feedforward neural networks (FFNNs) and artificial neural networks (ANNs), to classify features extracted through a hybrid approach. This hybrid approach combines the grey level co-occurrence matrix (GLCM) and local binary pattern (LBP) algorithms to extract robust features. This method achieved impressive accuracy rates of 99.8% using FFNNs and ANNs. The second method focuses on deep feature extraction using pre-trained CNN models, specifically GoogleNet andResNet-18. By leveraging these models for feature extraction, the method achieved high accuracies of93.6% with GoogleNet and 97.6% with ResNet-18. The third method, known as GoogleNet + SVM and ResNet-18 + SVM, combines DL with traditional ML techniques (SVM). In this hybrid approach, CNN models (GoogleNet and ResNet-18) are first used to extract deep feature maps. These features are then fed into SVM classifiers for finalclassification. This method demonstrated strong diagnostic capabilities, achieving accuracies of 95.5%for GoogleNet + SVM and 94.5% for ResNet-18 + SVM. Overall, these AI-driven approaches represent significant advancements in the early detection of ASD, showcasing the potential of neural networks, DL models, and hybrid methodologies to accurately classify individuals based ondiagnostic features extracted from various data sources. These methods hold promise for improving diagnostic accuracy and early intervention strategies for ASD [29].

The paper examined the classification of eye-tracking data based on eight primary characteristics: pupil size, saccade, fixation, velocity, blink, pupil position, electrooculogram (EOG), and gaze point.Many studies reviewed utilised multiple attributes of eye behavior, with fixation and pupil size emerging as common features. Some investigations also integrated signals from electroencephalography (EEG) or EOG, where EOG measures the electrical potential between the cornea and retina via electrodes, while EEG detects brain electrical signals. Notably, velocity was theleast explored characteristic among these. The objective of the review was to assess ML methodologies applied to classification tasks using eye tracking data. Researchers predominantly focused on fixation and pupil size, although several studies incorporated additional eye-related metrics.The review underscored the necessity for further exploration into new classification features derived from eye tracking data, emphasizing their potential applicability across diverse domains beyond current applications. In eye tracking, fixation denotes sustained gaze on a particular area, contrasting with saccades, which are rapid shifts between fixations. Pupil size typically ranges from 2-4mm in bright light conditions and expands to 4-8mm in darkness, influenced by ambient light levels [30].

The study focuses on the evolution of event identification in eye movement research, transitioning from manual data analysis to algorithm-based detection methods. It discusses two primary algorithms—velocity-based and dispersion-based—that historically struggled with misclassifications due to noise and sampling variations. Over the last decade, researchers have refined these algorithms by introducing new metrics and enhancing their ability to detect complex eye movements such as smooth pursuit and post-saccadic oscillations, especially in high-quality data acquired at higher sampling rates. Recent advancements have also leveraged RF classifiers to improve the accuracy of identifying events like saccades, fixations, and post-saccadic oscillations. The study underscores how precise classification of these eye movements can enhance biometric applications reliant on eye movement patterns, highlighting ongoing efforts to optimise algorithmic robustness andperformance in diverse research and applied settings [31].

A novel method has been proposed for real-time detection of pupils in real-world scenarios using AI technology. This system utilises parallel architecture in neural networks to classify pupil locations based on eye images. Through training on 40,000 eye pictures drawn from 20 datasets, the algorithm achieved an impressive detection rate of 96.29% accuracy within a margin of five pixels. This system holds promise for applications in real-time settings, computer games, assistive technologies, and the automotive sector. Its capability to accurately and swiftly detect pupils in dynamic, real-world environments underscore its potential to enhance user interfaces, improve human-computer interactions, and advance safety systems in vehicles. The proposed method represents a significant advancement in employing AI for practical applications involving visual recognition, demonstrating robust performance and broad applicability across diverse technological domains [32].

2.2 Other Techniques

Eye-tracking serves as a method to quantify cognitive strain in the interpretation of studies, focusing on both basic visual and oculomotor elements, as well as higher-level cognitive and linguistic aspects. It plays a crucial role in assessing research and providing a clearer understanding of the interpretation process. Unlike analytical, subjective, and performance-based approaches, eye-tracking offers objectivity and allows for a detailed analysis of interpretation step-by-step. Moreover, eye-tracking reveals cognitive representations and adjustments that occur during problem-solving. This technique is employed across diverse groups, using multimodal inputs to capture real-time interpretation processing and identify triggers for difficulties. It provides researchers with insights into how individuals process information and manage cognitive tasks, thereby enhancing understanding of cognitive load and decision-making processes in various contexts [33].

In contemporary research, achieving optimal spatial accuracy in eye tracking techniques involves accurately identifying the center of the pupil and the user's point of gaze across multiple frames. However, it is equally crucial to optimise temporal precision for analyzing rapid eye movements such as fixations and saccades. The resulting data must not only be accurate but also consistent for meaningful analysis. Using webcams as a low-cost eye tracking technology introduces a distinct set of challenges that must be carefully addressed. These challenges, including noise, low resolution, andlow frame rates, contribute to temporal instabilities that can adversely affect the quality of eye tracking outcomes. Researchers have proposed a metric for evaluating the temporal stability of algorithms designed to detect pupils in video streams. This metric aims to mitigate the impact of temporal fluctuations caused by technological limitations inherent in webcam-based eye tracking systems. The utilization of eye tracking technology to collect and analyse data pertaining to users' interactions with computers represents a significant area of contemporary research. Addressing the challenges associated with temporal stability in eye tracking algorithms is crucial for advancing the reliability and applicability of such technology in diverse research and practical applications [34].

According to a recent study, the eye-mind hypothesis, traditionally used to assess human visual attention, posits that both the duration and location of eye fixations generally indicate processing difficulty and the level of attention engaged. Fixation times specifically vary based on the type of information and task demands. Moreover, the focus points and duration of fixations reveal readers' reading strategies linked to their familiarity with information or prior experience. Eye-tracking methods have demonstrated efficacy across diverse fields of study, including emergent literacy, human-computer interactions, reading comprehension, information processing, and problem-solving in mathematics. Recent research employing eye-tracking has focused not only on solving mathematical problems but also on understanding the comprehension process and the strategies used in problem-solving. In particular, the study highlighted that student with higher accuracy in problem-solving allocated more time to addressing challenging problems, utilizing this additional time for integrating information and devising strategies. Conversely, students with lower accuracy tended to rely more frequently on direct translation techniques. In contrast, those with higher accuracy employed problem-model strategies more extensively. These findings underscore the utility of eye-tracking technology in exploring cognitive processes, particularly in educational settings where it offers insights into how individuals’ approach and solve problems based on their attentional focus, reading strategies, and problem-solving methodologies [35].

In a recent article, two key concepts—fixation and saccade—were introduced for calculating metrics related to areas of stimuli. These metrics are crucial for analyzing user behavior based on the Area of Interest (AOI) selected to present stimuli. Historically, integrating eye-tracking into mobile phones

posed challenges due to the complexity of measuring eye movements on small smartphone screens. However, since 2014, the increasing prevalence of mobile users has driven advancements in technology. Traditional eye-tracking glasses, often less accurate for tracking on small screens, have given way to software-only solutions that facilitate direct testing on smartphones. This approach yields reliable data for developing strategies for digital products without requiring additional hardware. Contemporary eye-tracking systems employ near-IR technology coupled with high-resolution cameras to accurately track gaze direction. These systems capture detailed data streams including pupil position, gaze vectors for each eye, and precise gaze points. By decoding eye movements, modern eye-tracking technology converts these data streams into actionable insights. This capability allows smartphones equipped with high-resolution selfie cameras, some reaching up to 50 megapixels, to serve as accurate tools for conducting mobile eye-tracking research. This development marks a significant advancement, enabling comprehensive studies of user interactions with digital interfaces and paving the way for enhanced usability testing and user experience research in mobile contexts [36].

Another study explored the use of eye movements as a primary method for detecting emotions, emphasizing potential future research directions in this area. The study employed direct eye-tracking approaches to classify emotions, achieving a notable accuracy rate of 90% using Artificial Neural Networks (ANN) as the classifier. Key parameters considered in the classification included pupil size, pupil position, and eye motion speed. Successful research demonstrating high accuracy levels,typically above 85%, often involved combining multiple training features. For instance, effective classification results were observed when using at least three features in tandem. Notably, pupil position and Electrooculography (EOG) were the least utilised features across studies that focused on fixation duration. The research article highlighted the significance of eye-tracking technology in identifying and analyzing eye movements for emotion classification tasks. It delineated several criticalaspects of such studies, including the types of emotional stimuli used, participant numbers, the emotions categorised, the specific features and classifiers employed, and the accuracy levels achieved in predictions. Overall, this study underscores the potential of eye-tracking data in enhancing understanding and detection of emotional states, suggesting avenues for future research to further refine methodologies and expand the application of eye-tracking in emotion recognition [37].

The integration of eye-tracking technology in analyzing unconscious behaviors contributes significantly to understanding effective learning principles in video-based learning contexts. Eye-tracking serves as a valuable method for studying information processing during video-based learning. Modern eye trackers utilise IR or near IR light and corneal reflections to track pupil centermovements. Studies indicate that online instructional videos often yield better academic performance

outcomes compared to traditional in-person instruction. Videos are generally more effective than static text and image-based resources in facilitating learning outcomes. However, there is a notable gap in research regarding how specific video design elements impact the learning process. Traditional self-reporting evaluations often miss temporal variations, unconscious responses, and subtle changes in cognitive processes. To comprehensively understand the mechanisms underlying effective video-based learning, empirical studies using eye-tracking are essential. Analyzing eye movements in response to dynamic stimuli such as videos presents unique challenges. Despite the limited research in this area, eye-tracking data can provide valuable insights into students' cognitive activities that are not captured through traditional assessment methods. This investigation emphasises the importance of evidence-based research utilizing specialised information systems and eye-tracking tools to advance our understanding of how individuals engage with and benefit from instructional videos. Continued exploration in this field promises to enhance instructional design strategies and optimise learning experiences through informed analysis of eye-tracking data [38].

Table 1: An overview of the literature on ML.

Reference	Methodology	Advantages	Disadvantages	Results
[11]	DL, DeepLabCut (DLC).	Can use inexpensive webcams.	The median error of approximately one degree of visual angle was achieved by this architectural design.	a median inaccuracy.
[12]	R-CNN	can be used in real time.	High cost-use standard web camera and laptop.	High accuracy.
[13]	CNN	noise reduction, this method increases the precision of emotion recognition	hinder their ability to effectively capture and retain information over extended periods.	high-level performance
[14]	SVM+LSTM neural network approach.	The performance is good on both the test set and the training set.	the calibration process is complex and expensive.	High accuracy
[15]	CNN based UNET architecture	Faster.	Less memory	Good accuracy.
[16]	CNN architecture, SVM.	The CNN can extract different latent eye features, making it an automatic feature extractor for this system.	The CNN's multilayer architecture is highly intricate and comes with a significant computational overhead.	Higher detection accuracy.
[17]	Deep Gradient CNN (DGCNN).	Studying human cognitive processing can be effectively done using a particular method.	The technology is hard to substitute with typical research methods.	an accuracy rate of more than 87.2%.

[18]	YOLOv3 architecture.	can run without a GPU accelerator.	In over 98.24% of images, it is able to achieve pupil center estimation errors smaller than the size of a constricted pupil.	over 98.24%
[19]	Deep Neural Network (DNN).	does not require expensive hardware to operate.	Eye trackers that are available for purchase in the market are too expensive for most people.	the results show that it outperforms these systems.
[20]	CNN	Capable of handling input from multiple channels.	As much as it can, the device detects only images with eyes that are in focus.	95% of accuracy.
[21]	F-FANN	Enhance the training performance.	Sometimes vector quantization causes classification issues.	Higher classification accuracy, and overall performance were high.
[22]	DLTechniques (CNN and RNN).	The capability to differentiate between children with ASD and typically developing (TD) peers.	unparalleled in its efficacy.	the most remarkable accuracy result of 98.33%.
[23]	Deep Neural Network (DNN).	can effectively improve prediction performance.	Time cost, complex operation process.	the model performed better.
[24]	DL.	Easy to handle with a computer, can be used for educational purposes.	Low success rate in pupil detection.	Higher success rate.
[25]	DL and data visualization.	The diagnostic task could be simplified by the visual representation.	The task is difficult and involves a series of cognitive assessments.	High accuracy.
[26]	CNN	The best threshold values were identified.	Differences in classification results.	the classification results differ significantly.
[27]	DL	Use low-cost web cameras.	The median error of the architecture was approximately one degree of visual angle.	achievee a median inaccuracy
[28]	CNN.	The features ofnatural images caneffectively be used to capture eye-tracking data.	A noisy gaze point, a large database are needed.	the technique successfully detects strabismus.

[29]	ML and DL techniques.	A sensitivity system.	The training process requires a significant amount of time.	The accuracy reached a high percentage of 99.8%.
[30]	Biometric ML.	produce 3-5 saccades per second, Saccades are quick.	Spent more time for qualitative synthesis.	offers precise, quantitative data.
[31]	ML	A greater biometric performance comes from more accurate eye movement classification.	Dealing with a specific challenge involves figuring out the change in x and y coordinates for each frame.	More accurate eye movement classification.
[32]	Neural Network (AI-based pupil identification method).	high processing speed, which enables its use in various real-time applications.	Many images are needed.	High accuracy.

Table 2: An overview of the literature on other techniques.

Reference	Methodology	Advantages	Disadvantages	Results
[33]	cognitive strategies (interplay of top-down and bottom-up processing).	Useful for measuring cognitive strain, a more affordable eye tracker.	Complex task.	High accuracy.
[34]	fixations and saccades (eye movement processors).	Can be using Modern smartphones as everyone's use.	Negative impact of outcomes such as noise, low resolution, and low frame rates.	Can be used as very accurately.
[35]	Cognitive techniques.	Can be used variety of study areas-human computer interfaces, math problem-solving solving and information process.	spent more time inspecting relevant factors	High accuracy.
[36]	Fixation, saccade.	can be used Modern smartphones as everyone's use.	Spend more time on the process.	Very accuracy.
[37]	Fixation, electroencephalography (EEG).	Present the classification results in 2 seconds.	pupil diameter achieving highly similar and low accuracies.	The ANN achieved a maximum accuracy of 90%.
[38]	video-based learning.	allows people to learn through watching videos.	There is a poor job of unconscious reactions and latent alterations.	accurately depict people's ongoing learning processes.

Chapter 3: Methodology

The research methodology outlines the methods and approaches used to gather and evaluate data on a particular study topic. Scientists use this framework to plan their research and employ selected research tools to achieve their objectives. In this study, several methods were practically experimented with, such as the KLT algorithm, GrabCut, Viola Jones, and Lazy Snapping. Ultimately, for the specific challenge of eye segmentation, the CNN approach was chosen. This approach is known for its ability to perform complex image analysis tasks effectively.

3.1 KLT (Kanade-Lukas-Tomasi) Algorithm

The KLT tracking method, or Kanade-Lucas-Tomasi, is a widely adopted technique in CV for feature tracking. This algorithm is integral to point tracking systems, offering versatility in applications such as object tracking, camera motion estimation, and video stabilization. It proves particularly effective with objects featuring distinctive visual textures or consistent shapes. While typically used for short-term tracking within larger systems, factors like changes in lighting, out-of-plane rotation, or articulated motion can lead to point loss over time.

To initiate tracking of a group of points, the following steps are typically followed:
- Initialise a PointTracker object and configure its parameters.
- Activate the object by passing necessary arguments, akin to invoking a function.

Selecting the correct warping parameters is crucial for accurately and reliably calculating optical flow between consecutive image frames. In an autonomous driving scenario, where the updated KLT algorithm was compared against several established methods, achieving convergence posed challenges, particularly when dealing with skewed and rotational effects. Among the different setups of warping parameters tested, the model incorporating scale parameters consistently outperformed others in terms of accuracy and reliability [39].

3.2 GrabCut Algorithm

GrabCut is a significant image segmentation technique based on graph-cut methods, first introduced in 2004 by Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. This approach iteratively refines initial foreground and background pixel labels by combining colour and texture features. Utilising graph theory in image processing, GrabCut efficiently segments images by treating pixels as nodes in a graph, where weighted edges denote the likelihood of pixel connectivity. Initially, a

bounding box is defined around the object of interest, guiding the selection of the segment to be isolated. Subsequently, a Gaussian mixture model is employed to predict variations in colour between the target object and the background. In addition to its role in image segmentation, the KLT algorithm is utilised for face detection and tracking within the "Cascade Object Detector". This method distinguishes between foreground and background regions in an image, specifying subregions with a label matrix. A logical mask termed the Region of Interest (ROI) identifies the initial AOI. Beyond its application in image processing, graph theory's effectiveness extends to diverse fields including agriculture, animal husbandry, and medical image analysis [40, 41].

3.3 Lazy Snapping

The objective of Lazy Snapping is to simplify the process of object removal from photographs. It achieves this through a graph cut method, where dynamically generated contours are used to indicate which parts of the image should remain (foreground) or be removed (background). Users can achieve accurate region isolation with minimal user input. Unlike traditional graph cuts, which can be computationally intensive and may take up to a minute for a full-resolution image (approximately 400x600 pixels), Lazy Snapping completes this task in less than a second, ensuring interactive performance.

Lazy Snapping, as a graph-based segmentation algorithm, facilitates the segmentation of images by effectively distinguishing between foreground and background regions. Additionally, MATLAB provides tools for programmatically or interactively segmenting images, where Lazy Snapping is particularly valuable for separating foreground and background elements in an image [42].

3.4 Viola-Jones Algorithm

Paul Viola and Michael Jones introduced the Viola–Jones object detection framework in 2001, which revolutionised MLbased object identification systems. Initially developed for face detection, it has since been adapted for recognizing various other object types. The "vision.CascadeObjectDetector" system is built upon this framework, employing a sliding window approach to scan images for objects. Using a cascade classifier, the detector evaluates whether the object of interest is present within each window. While the window's aspect ratio and vision remain fixed, its size is adjusted to detect objects across different scales.

The "CascadeObjectDetector" system incorporates numerous pre-trained classifiers designed to identify frontal faces, profile faces, noses, eyes, and upper torsos among other objects. However, the

effectiveness of these classifiers may vary depending on the specific application, as they may not always suffice for every scenario.

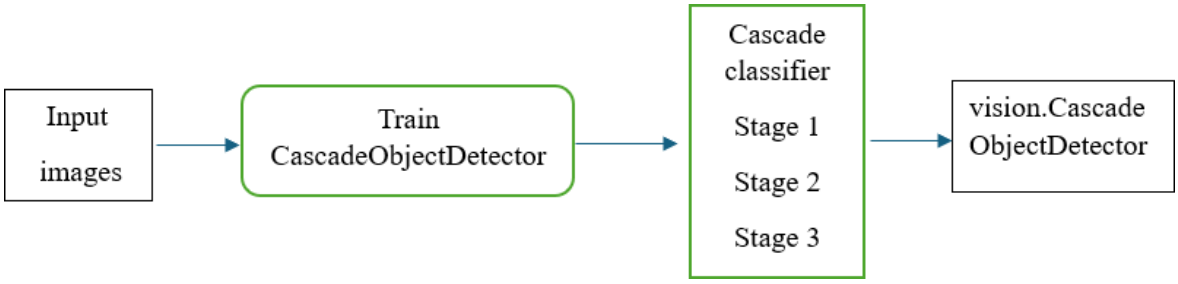

Figure 1: The diagram of flow shows an object detection pathway to apply Viola Jones algorithm.

The cascade object detector from the image recognition toolbox is adept at identifying objectcategories that maintain a relatively consistent aspect ratio. Examples include cars viewed from the side, stop signs, and faces. However, objects that undergo significant out-of-plane rotation pose challenges for most 3D detectors. Therefore, detectors must be trained for each potential object orientation, as a single detector cannot effectively cover all orientations. This ensures accurate detection across varying perspectives and orientations of objects in images.

In the Viola-Jones object detection framework, each stage of the classifier evaluates whether the current position of the sliding window contains an object or not. A positive label indicates the presence of an object, while a negative label signifies its absence. If a negative label is assigned, classification for that region ends, and the detector moves the window to the next position. Conversely, a positive label progresses the region to the next stage. The primary objective of these stages is to swiftly discard regions that do not contain the object of interest, if most windows fall into this category. True positives, where an object is correctly identified, are relatively rare, thus warranting careful validation.

Here are key terms and concepts related to the Viola-Jones approach:

- True Positive: Correctly identifying a region containing the object of interest.
- False Positive: Incorrectly identifying a region as containing the object of interest when it does not.
- False Negative: Incorrectly classifying a region as not containing the object of interest when it does.

Effective operation of each cascade stage requires a low false negative rate to minimise missed detections. If a stage incorrectly dismisses an object as negative, the classification process halts and cannot be rectified. Consequently, stages may tolerate a higher false positive rate since errors can be corrected in subsequent stages. Increasing the number of stages reduces both the overall true positive rate and false positive rate, as more stringent criteria are applied sequentially.

The Viola-Jones approach involves two main stages: identification and preparation. Training results in a neural network capable of distinguishing between faces and non-faces, enhancing the framework's ability to accurately detect objects in images.

3.4.1 Training stage

The Viola-Jones algorithm is designed to extract features from facial photographs through several key steps: it creates an integral representation of the image, computes Haar-like features, and selects the most discriminative features. Adaboost, a boosting algorithm, is then employed to prioritise these features, forming a cascade of classifiers to identify positive instances. This ensemble classifier, composed of multiple weak classifiers, achieves rapid and accurate object detection with high reliability.

However, despite its strengths, the Viola-Jones algorithm faces several challenges in practical applications:

- Non-frontal Faces: Difficulty in accurately detecting faces that are not directly facing the camera.
- Ambiguous Borders: Struggles with images where the boundaries between the object and the background are unclear.
- False Detection of Animal Faces: Occasional misidentification of non-human faces, such as animal faces, as human faces.
- High- or Low-Resolution Images: Reduced performance when processing images with very high or very low resolutions.
- Blurry Images: Faces captured in blurry images may not be accurately detected due to diminished feature clarity.

Addressing these challenges requires ongoing advancements in algorithmic robustness and the adaptation of the Viola-Jones framework to effectively handle diverse and complex image conditions.

3.4.2 Detecting stage.

The Viola-Jones algorithm was initially developed with a primary focus on detecting frontal faces, making it more proficient in accurately identifying faces that are facing directly towards the camera compared to those that are rotated or inclined. Prior to the detection process, images are typically converted to grayscale. This simplifies the colour space and reduces computational demands, enabling more efficient processing. The actual detection of faces is carried out using a trained cascade classifier, which has been taught to recognise specific patterns associated with frontal facial features [43, 44].

3.5 Deep Neural Network (DNN)

DNNs are neural networks characterised by their high complexity, consistingof multiple layers stacked together. These layers typically include two or more input and output layers,with at least one hidden layer in between. DNNs are extensively employed in data-driven modelling due to their ability to capture intricate patterns and relationships in data.

Each layer in a DNN involves nodes (neurons) interconnected by edges (weights), and mathematical operations are performed between these nodes to propagate information through the network. During training, backpropagation is used to adjust these connections based on the error calculated between predicted and actual outputs. These updated connections then serve as the learned relationships that enable the network to predict output variables based on input variables.

One of the key advantages of DNNs is their capability to model complex and nonlinear relationships within data systems. This flexibility allows DNNs to effectively represent and learn from data, regardless of the complexity or nonlinearity inherent in the system. This attribute makes DNNs particularly powerful in various applications ranging from image and speech recognition to natural language processing and beyond.

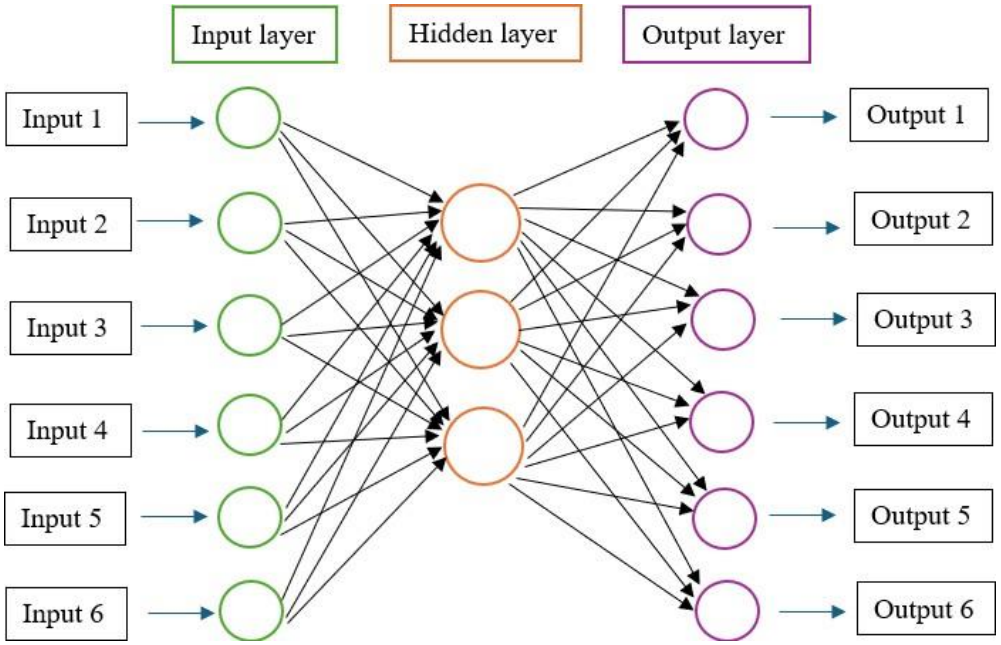

Figure 2: The basic structure of neural network architecture.

In a DNN model, there typically exist three main types of layers: input, hidden,and output. These layers are structured with interconnected nodes, also known as neurons, where eachnode is linked to every node in the subsequent layer.

- Input Layer: This initial layer receives input data features. Each node in the input layer represents a feature or attribute of the input data. For instance, in image recognition tasks, each node might correspond to a pixel value, or a feature extracted from the image.

- Hidden Layer(s): The hidden layers reside between the input and output layers. They play a crucial role in learning and extracting complex patterns from the input data. A DNN can have multiple hidden layers, each consisting of numerous nodes. These layers progressively transform the input data into representations that facilitate the network's ability to learn and generalise from the data.

- Output Layer: The output layer is the final layer of the DNN. It produces the network's predictions or outputs based on the transformed representations learned from the hidden layers. The number of nodes in the output layer typically corresponds to the number of classes or targets in a classification task, or it may represent the dimensions of the output in a regression task.

The interconnected structure of nodes within and between layers enables DNNs to effectively learn complex relationships and patterns in data, making them powerful tools for various ML tasks [45].

3.5.1 Dilation CNN Model

Dilated convolution, also known as atrous convolution, is a technique used in CNNs to increase the receptive field without adding additional parameters. Unlikestandard convolutions where the kernel moves over the input data with a fixed stride, atrous convolutions introduce gaps (dilation) between kernel elements. By varying the dilation rate at each convolutional layer, CNNs can capture a wider range of contextual information. This allows the network to gather more extensive context from the input data while maintaining computational efficiency. Dilated convolutions facilitate a balance between precise feature localization and comprehensive context absorption. This method is particularly effective in tasks such as semantic segmentation and image analysis, where understanding large-scale spatial relationships is crucial. It enables CNNs to effectively expand their receptive fields and enhance performance without significantly increasing computational complexity or model parameters.

In convolutional layers of a neural network, convolution involves sliding a template (filter) across an image to extract features. The size of the receptive field—the area over which the filter operates—is

determined by the dimensions of the filter and the convolution stride. During convolution, a fixed-size filter is applied to analyse a specific region of an input feature map. This process involves multiplying the values of the filter with the corresponding input values within the ROI. These operations generate a single output value, capturing important features from the input data.

Dilated convolutions present several advantages and drawbacks within neural network architectures. They effectively expand the receptive field of filters without increasing parameters, making them beneficial for capturing long-range dependencies while avoiding computational overhead. Moreover, they facilitate multiscale feature learning, crucial in tasks like image segmentation and audio processing. Despite preserving spatial resolution better than alternatives, dilated convolutions still induce some resolution loss and require higher computational resources due to increased operations [46].

3.5.2 Data Normalization

Data normalization is the process of reorganizing data within a computer system to enhance its usability for searches and analysis. It involves cleaning data by removing unnecessary and unstructured material and standardizing its appearance across all fields and records. Normalization also scales a dataset to a uniform range. In ML, the goal often involves scaling and centering data sothat it falls within specific ranges, such as 0 to 1 or -1 to 1, depending on the context. To achieve normal distribution, it's common to compute the mean and standard deviation of a dataset, then transform each data point by subtracting the mean and dividing it by the standard deviation. This normalization stabilises gradient descent steps in ML models, leading to higher learningrates and faster convergence. Additionally, batch normalization, a technique where mini batches of data are normalised within a neural network, further accelerates the learning process by ensuring consistent scaling and centering of inputs at each layer [47].

3.5.3 Activation Function

The Dilated Eye Network (DEN) employs the Leaky Rectified Linear Unit (Leaky ReLU) as its activation function. Unlike the standard ReLU function, Leaky ReLU introduces a small slope for negative inputs, ensuring that there is some gradient flow even for negative values. This small slope, typically a small constant like 0.01, is predefined and remains fixed throughout training. Leaky ReLU is particularly useful in scenarios where sparse gradients might be problematic, such as in training generative adversarial networks (GANs). GANs often benefit from Leaky ReLU because it helps to mitigate the issue of neurons becoming inactive due to negative inputs, thereby maintaining a more stable learning process.

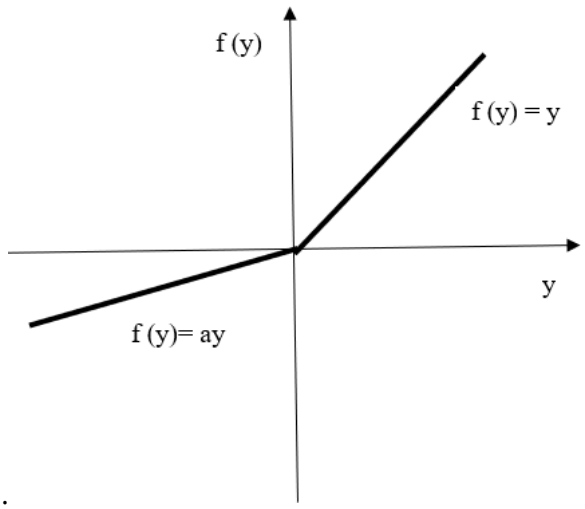

Figure 3: A sample graph for activation function.

values to zero and leaving positive values unchanged, thereby introducing a non-linear threshold operation. Unlike more complex activation functions with learnable parameters, ReLU is simple and effective, promoting abstract feature learning through nonlinear transformations. In shallow networks, using activation functions with learnable parameters can enhance the network's ability to learn intricate features. This capability is well-established within the field. However, a notable drawback of activation functions with multiple learnable parameters is their increased demand for large datasets during training. This requirement arises because more parameters necessitate more data to effectively learn and generalise from the underlying patterns in the data [48].

3.5.4 Semantic Segmentation using Dilated Convolutions

Dilated convolutions are extensively utilised in semantic segmentation networks due to their ability to expand the receptive field of a layer without increasing parameters or computational complexity. Semantic segmentation involves categorizing pixel groups in an image based on various attributes, enabling machines to distinguish between different object classes and background regions. The surge in popularity of ML and artificial intelligence (AI) has underscored the significanceof segmentation mapping and image segmentation as fundamental tools. These techniques are crucialfor training computers to discern context in digital images, spanning applications from landscapes andportraits to medical imaging and beyond. DL algorithms have significantly enhanced computers' ability to interpret images as data, with semantic segmentation enabling precise identification of different types of visual information. In conjunction with image classification, whichidentifies the content within an image, these advancements empower machines to accurately locate objects and understand visual scenes.

Semantic segmentation models generate segmentation maps from input images, where each pixel is color-coded based on its semantic class to create segmentation masks. These masks delineate specific

parts of an image that are distinct from others, facilitating tasks such as object detection and scene understanding. Achieving accurate semantic segmentation involves advanced neural networks that classify each pixel into its real-world semantic category and group related pixels into coherent segmentation masks. ML techniques such as backpropagation and gradient descent are crucial for training these models on large, annotated datasets, which are essential for DL approaches to learn and generalise effectively. Semantic segmentation finds practical applications across diverse fields. For instance, in autonomous driving, it helps identify road segments and obstacles, enabling vehicles to navigate safely. In medical diagnostics, semantic segmentation aids indistinguishing different types of tissues or cells, such as identifying cancerous cells amidst healthy tissue.

In this neural network configuration, a semantic segmentation model employs dilated convolutions to classify each pixel in an image, thereby segmenting it based on predefined classes. Training involves using a dataset of square or rectangular images paired with ground truth data that labels each pixel accordingly. To construct the segmentation network, it is crucial to gather appropriate training data and ensure class balance by considering pixel counts per label. Background pixel labels can skew learning, which is mitigated through techniques like class weighting, such as inverse frequency weighting, to amplify underrepresented classes during training. The network architecture typically begins with an initial image input stage followed by convolutional blocks incorporating ReLU activation layers and batch normalization. Proper adjustment of dilation factors and padding ensures consistent output dimensions across convolutional layers ("same" padding in MATLAB maintains uniform size). Key components include a class-specific convolutional layer followed by a softmax layer for pixel classification. Custom loss functions are integrated for training, with parameters like MaxEpochs, MiniBatchSize, InitialLearnRate, and verbosity settings specified in MATLAB to optimise training efficiency. Post-training, functions handle testing data input and generate ground truth labels for evaluation. The trained network and test data predict pixel classifications, evaluated for accuracy using CV techniques. Semantic segmentation results in a segmented image that visually distinguishes different classes within the test image. In summary, this approach involvesmeticulous data preparation, network configuration, parameter tuning, and rigorous evaluation to effectively implement semantic segmentation for image analysis tasks using MATLAB or similar environments [49].

3.5.5 Model Training

The loss function is used to train and optimise the network by measuring the difference between the predicted score and the target score.

Table 3: The details of the dilated eye network architecture.

Block	Layer	Kernel Size, Feature Maps	Atrous Dilation Rate	Output Size
Input layer	Input	100 x 100 x 1	-	100 x 100 x 1
Block 1	Conv1	3x3, 8	1	100 x 100 x 8
Block 2	Conv2_1	3x3, 16	2	100 x 100 x 16
	Conv2_2	3x3, 16	4	100 x 100 x 16
Block 3	Conv3_1	3x3, 32	4	100 x 100 x 32
	Conv3_2	1x1, 16	6	100 x 100 x 16
	Conv3_3	3x3, 32	8	100 x 100 x 32
Block 4	Conv4_1	3x3, 64	8	100 x 100 x 64
	Conv4_2	1x1, 32	10	100 x 100 x 32
	Conv4_3	3x3, 64	10	100 x 100 x 64
	Conv4_4	3x3, 64	12	100 x 100 x 64
Block 5	Conv5_1	3x3, 128	12	100 x 100 x 128
	Conv5_2	1x1, 64	12	100 x 100 x 64
	Conv5_3	3x3, 128	14	100 x 100 x 128
	Conv5_4	1x1, 64	14	100 x 100 x 64
	Conv5_5	3x3, 128	14	100 x 100 x 128
	Final_Conv	1x1, 2	-	100 x 100 x 2
	SoftMax	-	-	100 x 100 x 2
OutPutMap	Pixel Classification	Cross entropy	Loss function	100 x 100 x 2

3.5.6 Training Options

The following table shows the details used to train the network. The details can be changed according to the layers of the network.

Table 4: Optimised parameters for dilated eye network

Parameter	Value
Input image size	100 x 100 x 1 (pixels)
Momentum	0.9
Initial learning rate	0.001
L2 Regularization	0.005
Mini batch size	8
Verbose frequency	2
Optimiser	SGDM

3.6 Circular Hough Transform (CHT)

Detecting features in images presents a significant challenge in CV, particularly due to the varied sizes and shapes of objects that are not predefined in standard object detection programs. One effective solution to this challenge involves algorithms capable of detecting arbitrary shapes within images and categorizing objects based on specific parameters defining those shapes. The Circular Hough Transform, originally developed by Richard Duda and Peter Hart in 1972, is a widely

used technique for addressing this issue. It identifies various arbitrary shapes in images by representing them using parameterised mathematical forms. For instance, the Circular Hough Transform extends this capability specifically to detect circular objects within images that may have low contrast or noise. In this approach, instead of using traditional slope-intercept parameters, the Circular Hough Transform utilises angle-radius variables to estimate and identify circular shapes more effectively. This technique is particularly useful in applications like pupil detection in images. The effectiveness of the Circular Hough Transform lies in its ability to detect lines and shapes by converting grayscale images into binary formats through edge detection methods such as Sobel or Canny. The Circular HoughTransform employs a voting process across the parameter space to identify irregular shapes within a predefined set. In practical terms, the Circular Hough Transform determines the presence of shapes byidentifying local maxima in an accumulator array, which corresponds to the unknown parameters of the shape being detected. This parameter space is defined by the parametric equations that describe theshapes, such as circles in the image plane. Overall, the Hough Transform, including its circular variant,is a powerful tool in CV for detecting and categorizing shapes within images, enabling applications in various domains where precise object recognition is crucial [50].

Chapter 4: Results

This chapter presents the results obtained from applying various segmentation techniques outlined in the previous chapter. The evaluation includes several approaches such as the KLT algorithm, GrabCut, Lazy Snapping, Viola-Jones, Dilated Eye Network, and Circular Hough Transformation. These techniques are compared to determine which achieves the highest segmentation accuracy.

4.1 Video Reading Results

The "videoReader" function was employed to ingest video data. Once a "VideoReader" object was instantiated, its methods enabled the reading of video frames and querying of its properties. In the context of video data, the term "file format" usually refers to either the container format or the codec. A container format defines the file's structure, while a codec dictates how the video data is encoded and decoded. Different container formats can accommodate data encoded using various codecs. In this research, all video files were consistently read in the "mp4" format.

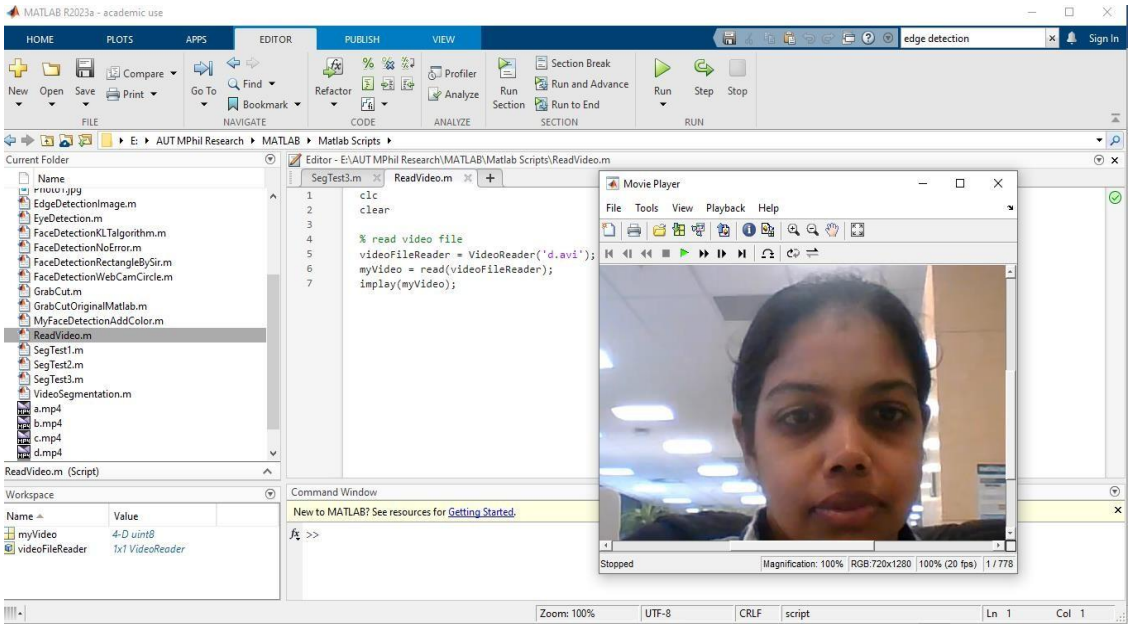

Figure 4: This displays how to read a video in MATLAB.

4.2 Video Face Detection Results

4.2.1 Face Detection using a Rectangle (KLT Algorithm)

Accurate object detection and tracking are essential for various CV applications, including activity recognition, vehicle safety, and surveillance. The following steps are utilised to detect a face using a rectangle shape of frame and the related code displays in Appendix A.

Find a face: The "vision" method uses the "CascadeObjectDetector" object to identify faces in video frames. It uses a trained classification model in conjunction with the Viola-Jones detection approach. The Kanade-Lucas-Tomasi (KLT) method, however computationally expensive, continuously monitors the face. Each frame is tracked by the KLT algorithm.

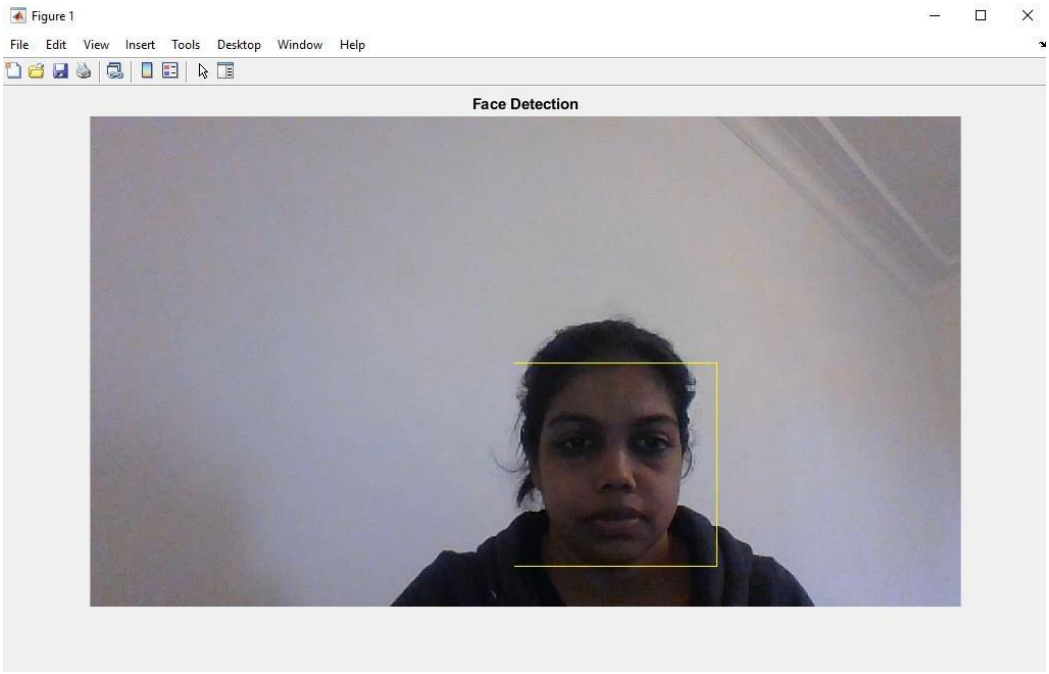

Figure 5: Face detection using KLT algorithm by rectangle shape in MATLAB.

Recognise face characteristics to track: To recognise and track faces, the technique tracks feature points in frames of a video. The "estimate GeometricTransform2D" function is used by the "vision. PointTracker System" object to track these points and then estimate the change in position, rotation, and scale between the original and new coordinates. Performance in noisy situations can be enhanced by implementing a point tracker and putting bidirectional error limits in place.

Follow the face: After tracking points in each frame, the movement of the face is calculated using the "estimateGeometricTransform2D" function. A duplicate of the points is made that will be utilised to ascertain earlier points and provide geometrically altered frames.

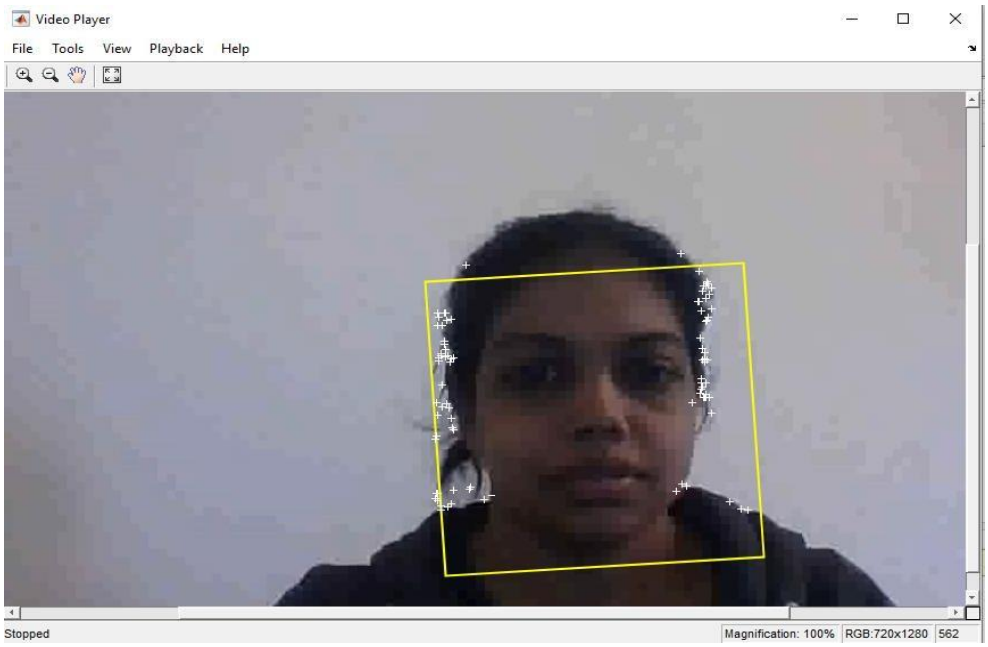

Figure 6: Detect the face using KLT algorithm when moving the face.

4.2.2 Face Detection using a Circle (Webcam)

Accurate detection and tracking of objects, such as faces in live video streams, are essential for various applications in CV, including activity recognition, vehicle safety, and surveillance. The human facial tracking system operates in two modes: detection and tracking. Initially, faces are identified using the "vision.CascadeObjectDetector" in recognition mode, which locates facial features and initialises tracking. Subsequently, a point tracker is activated to maintain continuity by followingspecific points on the detected faces. During tracking, adjustments can be made in real-time using MATLAB's command window to address errors and optimise performance. This method, though morecomplex than straightforward algorithms, enhances robustness and accuracy in object recognition tasks.

When executing code in the "Command Window," the output displays a graphical representation that includes a circular frame. The parameters defining the bounding box are noted as 765, 500, and 110.

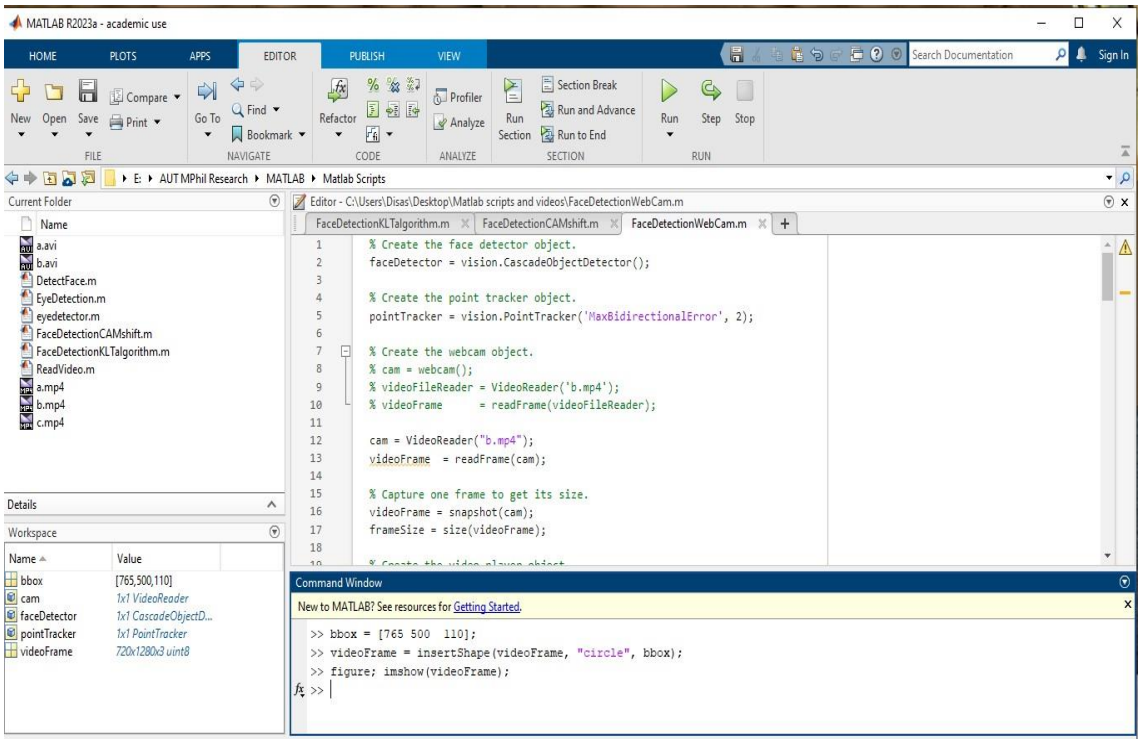

Figure 7: A code for face detection which was run using the command window in MATLAB.

According to the above given value in command window, the eye detection is done by a circle.

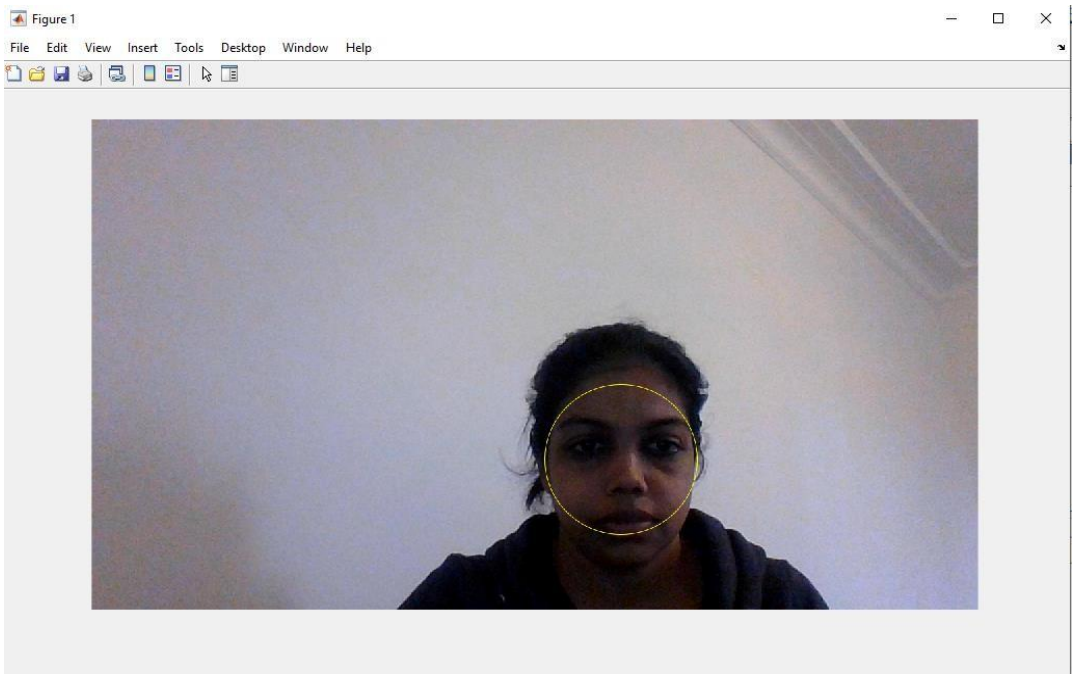

Figure 8: Face detection on webcam using shape of a circle.

4.3 Facial Movement Tracking Results

Below is an academic-style description of the MATLAB code designed to track the movement of a detected face using rectangular shape indicators, including directional movements such as left, right, down, and up:

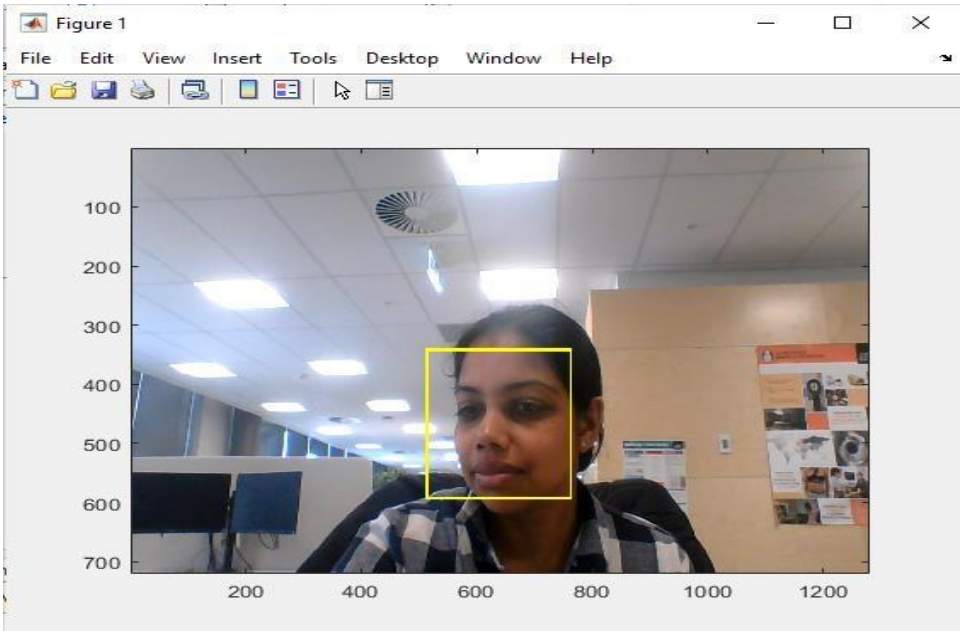

Figure 9: When face is detecting the right movement of a video.

4.3.1 Add Different Colors for the Frame

The figure below illustrates the frame color as red, with the capability to dynamically change its color:

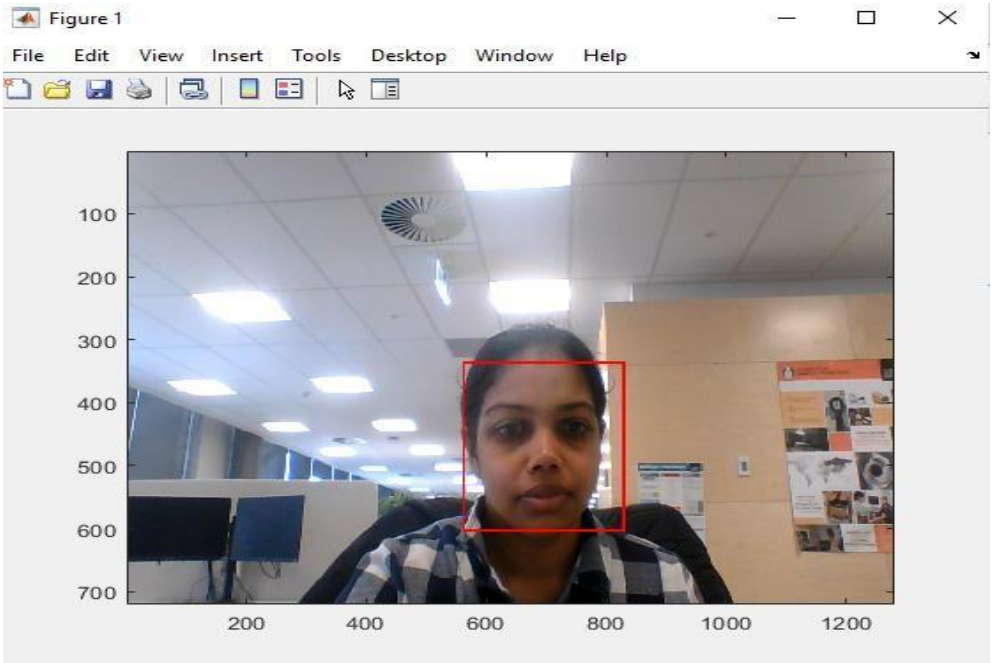

Figure 10: This displays the detected face within a red rectangular frame.

4.4 Face Segmentation

4.4.1 GrabCut

Image processing technology, driven by advancements in computer science and the widespread adoption of CV principles, has become indispensable across diverse fields. The cornerstone of many image-processing applications lies in effective image segmentation techniques, crucial for selecting technologies that meet performance requirements.

One such technique, the GrabCut algorithm, stands out for its ability to accurately extract objects from complex image backgrounds with minimal user intervention. It offers high precision and efficiency in segmentation tasks by requiring only an initial estimate of an object's location, contrasting with methods reliant on manually crafted masks. GrabCut excels in handling intricate object occlusions and boundaries while conserving computational resources.

In the context of face segmentation, GrabCut's straightforward implementation and effectiveness make it a preferred choice over alternative methods. Originally designed for static image segmentation, it can be adapted to process video frames sequentially, leveraging iterative processes such as "for loops." This adaptability underscores its versatility as a robust tool in image processing.

Key features of GrabCut include its interactive polygonal ROI definition, where users specify x and y-axis coordinates. These ROIs can be further customised in terms of shape, position, appearance, and behavior through object attributes and functions. The algorithm operates within the RGB color model, utilizing 8-bit channels for red, green, and blue components to depict varying intensities.

Practically, GrabCut produces binary mask images from ROIs, facilitating precise segmentation of faces detected in video streams. Below are illustrations demonstrating the segmentation of facial movements—up, down, left, and right—accompanied by frames and polygonal overlays outlining the faces.

Here is the figure of segmentation when moving the face up, down, left, and right, with both the frame and polygon drawn around the face.

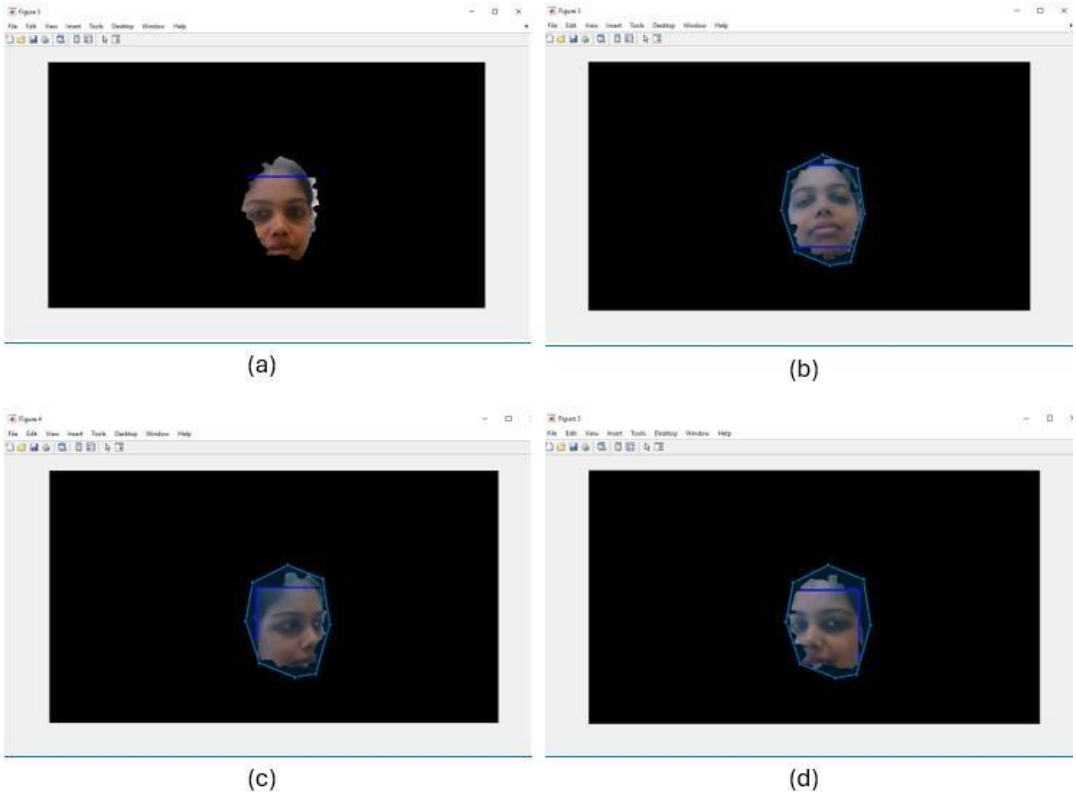

Figure 11: Face segmentation using GrabCut algorithm with a frame and the polygon. (a) The segmentation when face is looking forward. (b) The segmentation when face is moving up. (c) The segmentation when face is moving left. (d) The segmentation when face is moving right.

However, despite its strengths, the GrabCut algorithm does have several limitations. The accuracy of the initial estimation of the object's location heavily influences its effectiveness in segmentation tasks. Achieving precise segmentation often requires iterative adjustments. Moreover, GrabCut is sensitive to variations in color and illumination, which can affect the accuracy of segmentation results. Changes in lighting conditions or color contrasts within the image may pose challenges to maintainingconsistent segmentation quality. Another consideration is the algorithm's computational demands, especially when processing large images or real-time video streams. GrabCut can be resource- intensive, requiring significant processing power to execute efficiently under such circumstances [51].

4.4.2 Edge Segmentation of Figures

In image processing and visual analysis, edge-based segmentation uses the presence of edges to identify and analyse items within an image. This technique is based on the idea that distinct variations in texture, colour, or intensity are usually represented by an object's boundaries in an image.

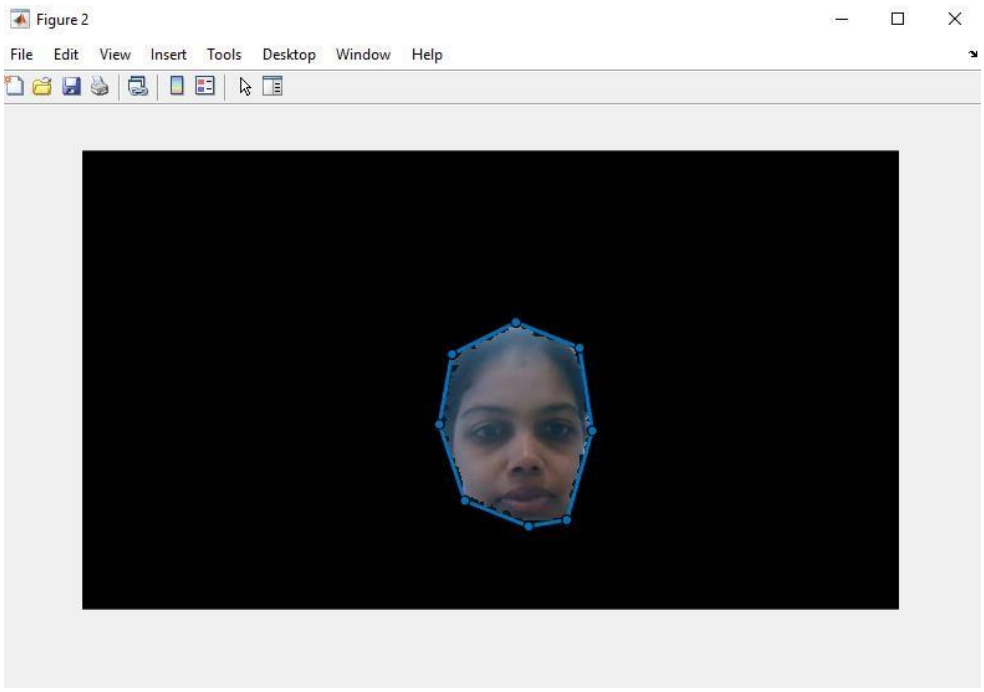

Figure 12: Edge segmentation of face detection when face is looking forward.

4.4.3 Lazy Snapping & GrabCut Together

The Lazy Snapping method for interactive image cutout involves two main stages: a rapid object marking phase and a foundational border adjustment step. It integrates boundary editing with graph-cut segmentation, offering a sophisticated approach to image segmentation. In contrast, GrabCut utilises both regional and boundary information to segment images effectively.

Illustrations below demonstrate the outcomes of applying both Lazy Snapping and GrabCut algorithms:

- Lazy Snapping Algorithm Results: Displaying segmentation achieved through efficient object marking and border refinement.
- GrabCut Algorithm Results: Highlighting precise segmentation using regional and boundary data for accurate object delineation.

These methods exemplify advanced image segmentation techniques, each with distinct strengths in efficiency and precision. Comparative visual examples provide insights into their applicability for complex segmentation tasks.

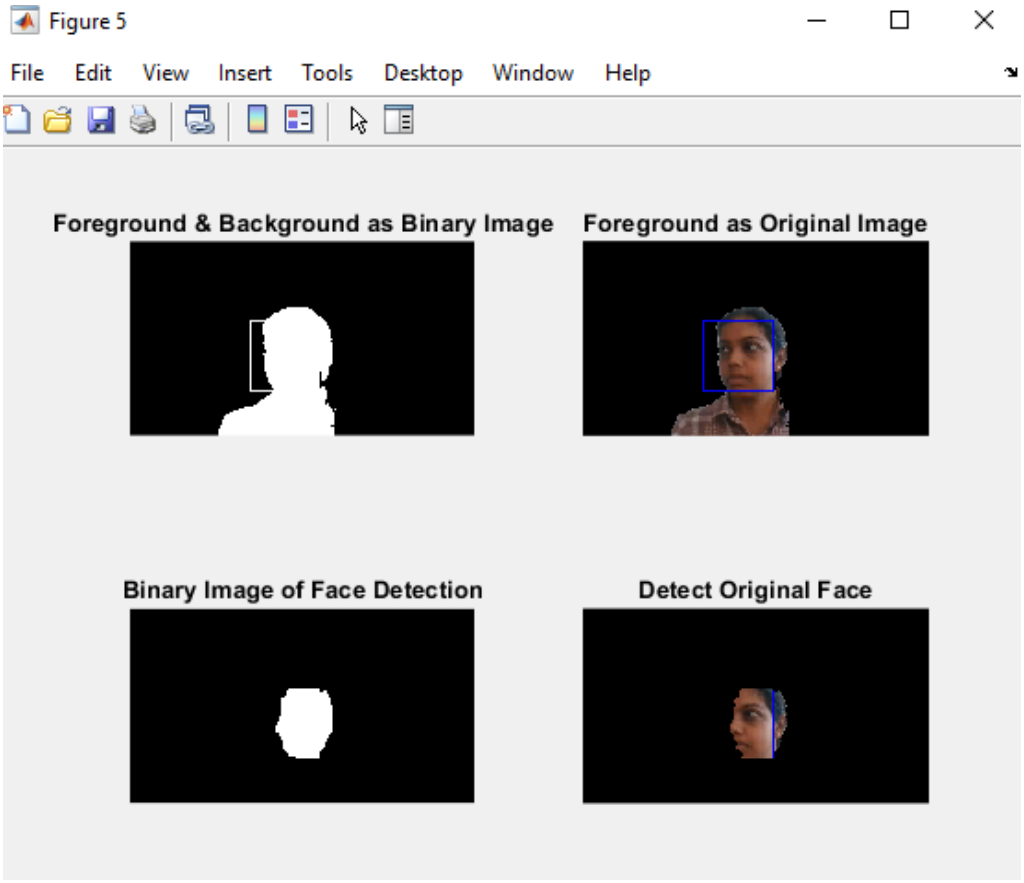

Figure 13: Application of both GrabCut & Lazy Snapping algorithms for face detection.

The initial binary image (first figure) illustrates the segmentation outcome where foreground and background are distinctly separated. Subsequently, the second figure displays the original image with the foreground object isolated, showcasing the segmented area. Following face detection, the third figure presents a binary image used to encircle the face within the original image. However, the final output reveals an incomplete detection of the face, accurately capturing only half of the facial features. This partial detection highlights a limitation in achieving precise and comprehensive facial recognition. Factors contributing to this may include image quality variations, changes in lighting conditions, or challenges inherent in the segmentation and detection algorithms employed.

4.5 Eye Segmentation

4.5.1 Viola-Jones Algorithm

In 2001, Paul Viola and Michael Jones introduced the Viola-Jones algorithm through their paper titled "Rapid object detection using a boosted cascade of simple features." This machine-learning technique was originally devised for facial detection and remains valuable, especially for devices with limited computational resources, despite being less precise than contemporary methods relying on

CNNs. The Viola-Jones algorithm is particularly effective atdetecting frontal faces but may not perform as well with profiles or faces oriented above or below. It begins by converting the image to grayscale, which simplifies processing and reduces computational load. Once a face is detected in the grayscale image, the algorithm determines its position in the original-colored image. Operationally, the algorithm evaluates numerous small sub-regions within animage, searching for specific features that indicate the presence of a face. This approach necessitates examining a broad range of positions and scales within an image, accommodating the potential presence of multiple faces of varying sizes [52].

Next, change the classification model such as EyePairSmall, EyePairBig, RightEye and LeftEye and the results are shown in the following figures.

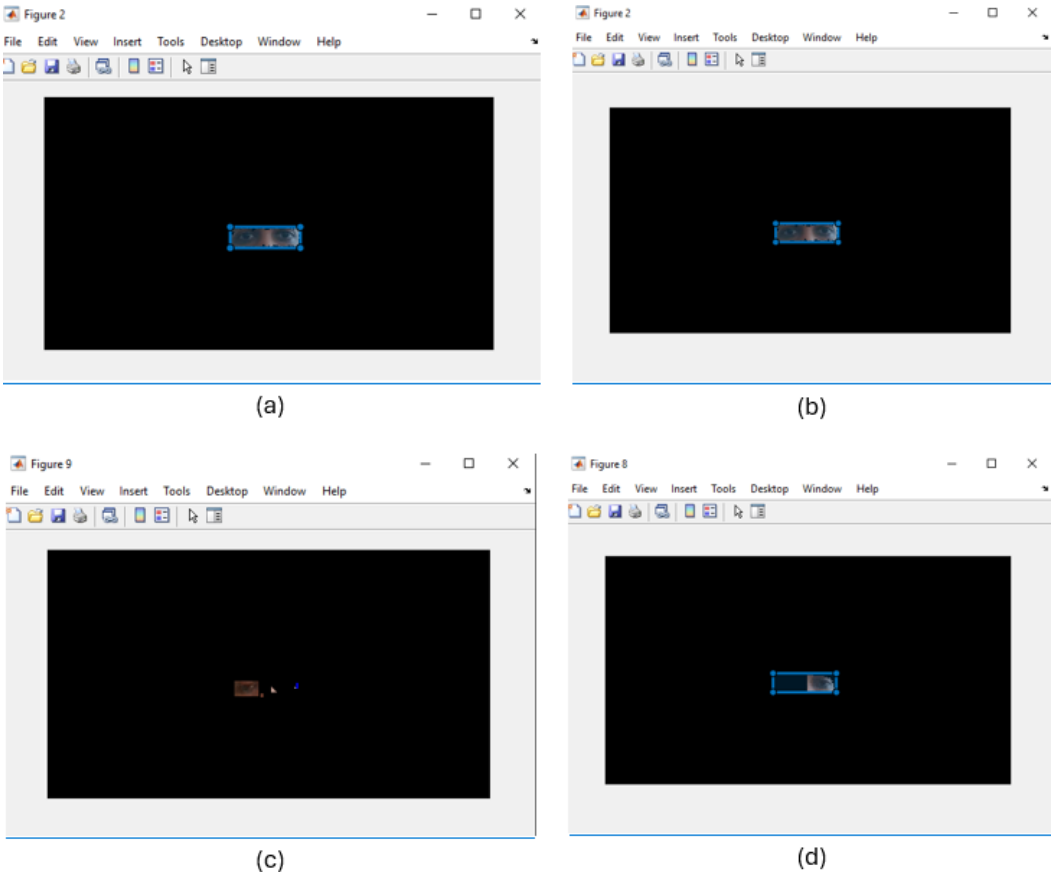

Figure 14: Eye segmentation using Viola Jones algorithm that displays the output according to the classification model. (a) The classification model of "EyePairBig"- [11 45]. (b) The classification model of "EyePairSmall"- [5 22]. (c) The classification model of "RightEye"- [12 18]. (d) The classification model of “LeftEye”- [12 18].

Here, the Viola-Jones segmentation method was implemented to demonstrate segmentation results using a polygon approach. However, its application was limited to a single video, focusing on classification models such as 'EyePairBig', 'EyePairSmall', 'RightEye', and 'LeftEye'. According to the Figure 14 results, despite these efforts, the algorithm did not yield significant outputs according to the classification model, and its accuracy was inadequate. As a result, the effectiveness of this algorithm in providing accurate segmentation results was deemed insufficient and may be overlooked for more reliable alternatives.

Further, there is another option to try out finally; adding threshold values and seeing whether it gives an expected outcome.

- When the threshold value is increased up to 10,11,12….15, then eye detection creates one frame around the eye.
- If the threshold value is decreased to greater than 10, then it displays two frames around the eyes.
- When the threshold value = 0, it displays 4-5 frames around the eyes.

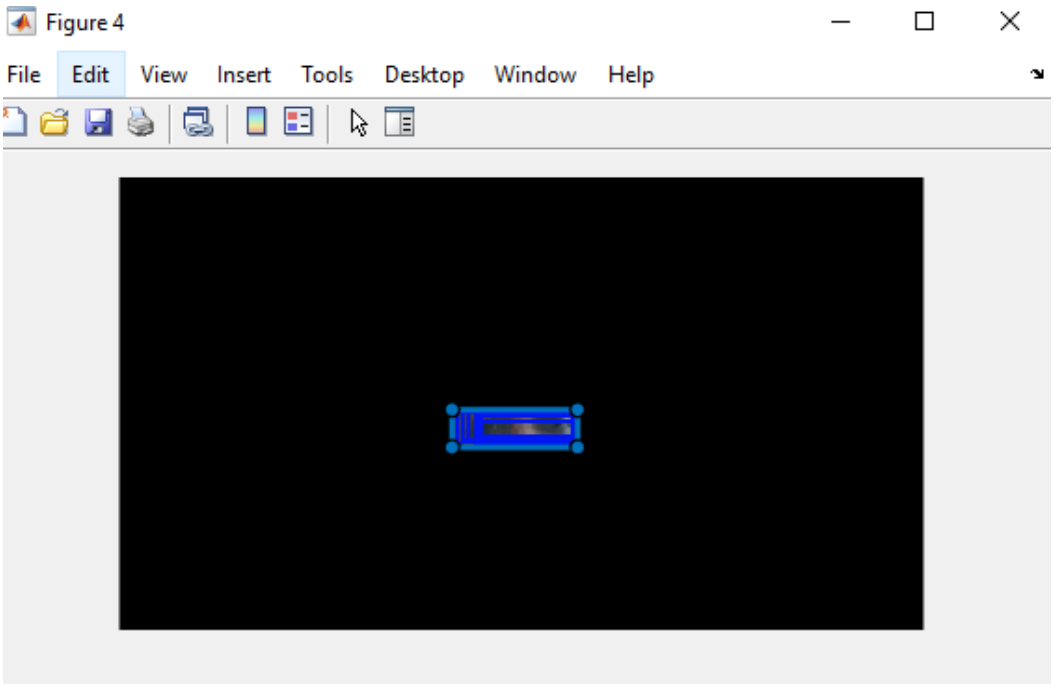

Figure 15: When threshold value is zero, it displays a few frames around the eyes.

4.5.2 Graph-Based Segmentation (Lazy Snapping)

The Lazy Snapping software is designed to extract objects from images by allowing users to draw dynamic boundaries, utilizing graph cuts to distinguish between foreground and background regions. The primary objective of foreground-background separation segmentation is to divide an image into distinct foreground, background, and middle ground areas. In this context, the foreground refers to the closest objects to the camera, while the background represents elements farthest from the image. The middle ground occupies the space between these two areas. The results obtained from using Lazy Snapping to segment background images, while preserving the original image, are documented in Appendix B.

4.5.2.1 Detect Eye Using Lazy Snapping & GrabCut.

As shown in the following figure, first detect the face and then detect the eyes using both Lazy Snapping and GrabCut algorithms.

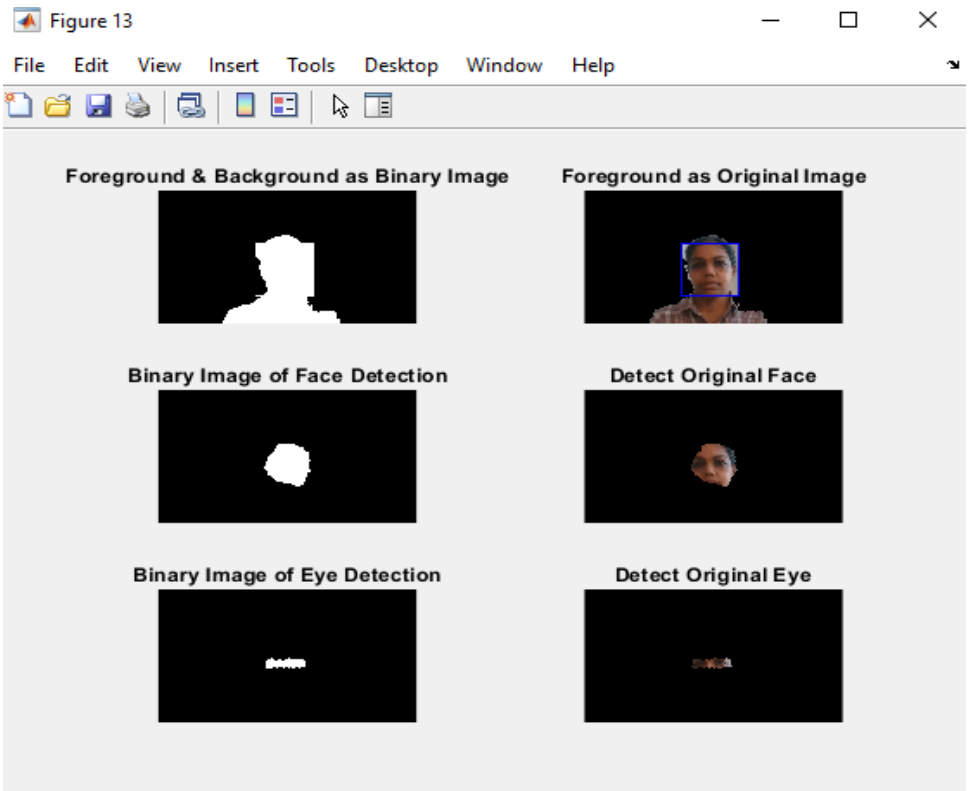

Figure 16: Eye detection using both Lazy Snapping and GrabCut and finally, detected only the eyes with original image.

In some cases, the Lazy Snapping method may not provide accurate segmentation results due to the presence of unwanted noise in the background of binary images. To improve the accuracy of segmentation outcomes, an alternative approach involving DL methods was pursued.

4.6 Dilated Eye Network Segmentation

The process of dilation involves changing the size of an object or shape by either enlarging or reducing its dimensions. Like how opening a door allows light to enter a dark room, dilating pupils enables light to enter the eye. This dilation process is crucial during eye examinations conducted by ophthalmologists to assess various common eye conditions such as diabetic retinopathy, age-related macular degeneration (AMD), and glaucoma.

4.6.1 Datasets

To train the dilated eye network, datasets were sourced from "iStock by Getty Images," a prominent stock photo marketplace known for offering a vast selection of high-quality photographs at affordable prices or for free. Established in 2000, iStock pioneered the crowd-sourced stock images market, providing a platform for user-generated stock images, graphics, and video clips. Today, iStock remains a leading global marketplace for creative content, offering millions of carefully curated images to meet various needs in the digital and creative industries [53].

To train the network the image details of datasets.

Table 5: Details of datasets which categorised by three sections.

Dataset Name	Training	Testing	Validation
iStock by Getty Image	1287	170	78

Image quality is assessed through characteristics like detail, contrast, noise, and distortion, influenced by factors such as production, subject conditions, and technique. A common data splitting strategy involves using 70% for training, 10% for validation, and 20% for testing. The training set teaches the model about quality variations, while the validation set helps tune hyperparameters and prevent overfitting. The separate test set evaluates model performance unbiasedly. Accurate labeling and data augmentation techniques can enhance model robustness, ultimately improving image quality assessment.

The original image format is JPEG (JPG), which was converted to PNG format using MATLAB code because JPEG is a lossy compression format that reduces file size by sacrificing some image quality, making it suitable for photographs, while PNG is a lossless format that retains all image data, which is beneficial for images requiring transparency or sharp edges. The original image size is 40.7 KB, and its dimensions are 612x408 pixels. In this database, the dimensions were reduced to 100x100 pixels, resulting in a file size of 6.25 KB. This reduction aids in data compression, allowing the images to occupy less storage space and decreasing computation times during processing.

4.6.2 Resize Image

The images obtained from "iStock" are typically large. Consequently, these images were resized using MATLAB code for compatibility with the dilated eye network. Upon inspection of the image properties, it was noted that the dimensions were adjusted to 100 x 100 pixels, ensuring uniformity where both width and height measurements are specified in pixels. The MATLAB code responsible for this resizing operation is detailed in Appendix C.

4.6.3 Convert RGB to Gray Scale.

By utilizing the rgb2gray function, RGB images can be converted to grayscale, effectively removing color temperature and saturation information while retaining luminance. This conversion simplifies image processing procedures by reducing computational complexity, making them more accessible for beginners. The corresponding code for this operation can be found in Appendix D.

4.6.4 Convert JPG to PNG

PNG (Portable Network Graphics) employs a lossless compression technology, ensuring that the quality of photos remains excellent even after compression and decompression processes. Unlike many other formats that may degrade image quality when compressed, PNG maintains clarity and detail even at smaller file sizes. In contrast, JPEG (Joint Photographic Experts Group) files utilise "lossy compression," where each time an image is edited and saved, some degree of image quality is sacrificed. PNG, on the other hand, preserves image quality through "lossless" compression, making it ideal for scenarios where maintaining image fidelity is crucial, such as in training, testing, and validating image datasets. Therefore, the choice of PNG format for the images used in training, testing, and validation ensures that the quality of the images remains intact throughout these processes, without any loss of detail or clarity.

4.6.5 Training Progress

Firstly, it displayed a figure for analysis, and it was able to check whether it had errors in the network connections between the layers.

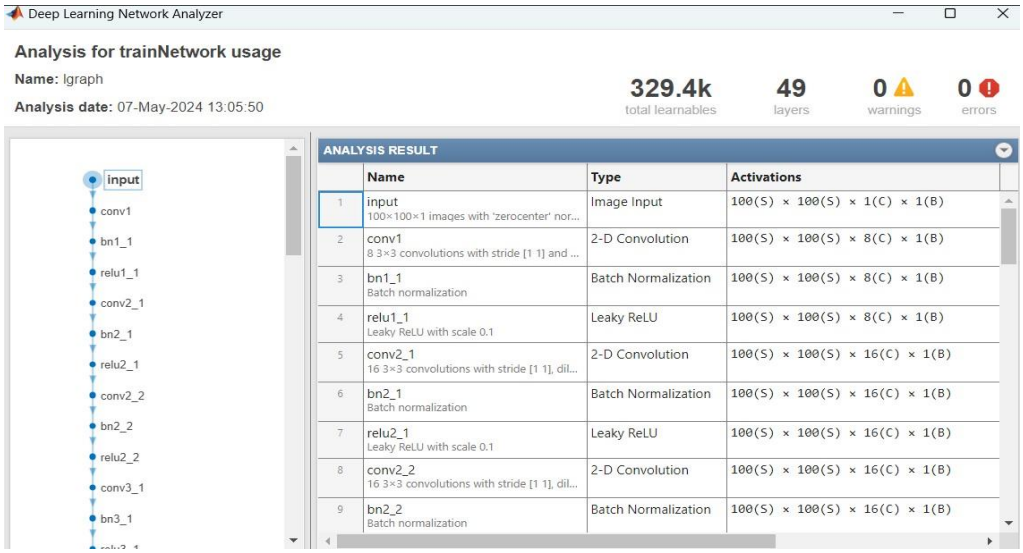

Figure 17: Analysis of train DNN usage and it displays zero errors.

4.6.6 Dilated Eye Network

After resizing the datasets accordingly, a dilated neural network architecture was developed to construct a Dilated Eye Network (DEN) using all categorised images, including those designated for training, testing, and validation.

4.6.7 Normalization

In the context of normalisation within CNNs, researchers utilise techniques like batch normalisation (BN) to ensure stable training and efficient inference. During inference, activations are normalised by linear transformations, maintaining fixed means and variancesestablished during training. BN layers are strategically inserted after subsets of activations to normalisethem across mini batches, enhancing training effectiveness by reducing internal covariate shift.

Batch normalisation layers placed between convolutional layers and nonlinearities such as ReLU help accelerate CNN training and mitigate sensitivity to initialisation. This approach is crucial in networks like the Dilated Eye Network (DEN), where normalisation is applied across separate classes—eyes and background.

In practical application, the network's performance is evaluated based on its ability to classify detected eyes (eye class) and accurately identify instances where eyes were not detected (background class). Performance metrics are typically presented as percentages within a matrix, reflecting the network's predictive accuracy for each class.

For example:

Table 6: Example of normalized confusion matrix during training.

Predicted Class of Eye (%)	Predicted Class of Background (%)
99.26	0.7421
99.68	0.3243

4.6.8 Train the Network

Here are the instructions to write code for training the network.

1. Load the dataset from the folder.
2. Create a function that calculates pixels according to the class label.
3. Import validation data.
4. Create network layers.
5. Define training options.
6. Write code to start network training.
7. Load test data.
8. Evaluate the trained network.
9. Normalisation.
10. Segment an image.

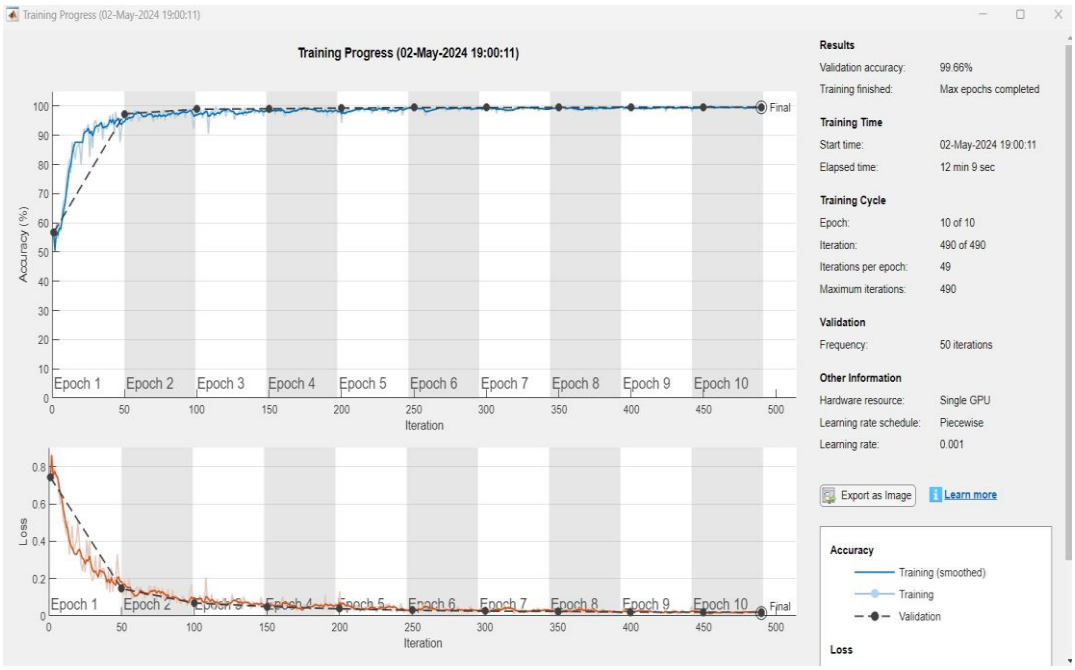

Figure 18: The graph shows the training progress of Dilated Eye Network for 10 epochs which gain the highest accuracy of training phase.

According to the training performance table of the Dilated Eye Network, increasing the number of epochs resulted in higher accuracy. However, this improvement came at the cost of longer training times, indicating a trade-off between training duration and accuracy enhancement.

Table 7: Training Results of dilated eye network

Number of Epochs	Accuracy (%)	Progression Time (Minutes)
1	97.74	6
5	98.19	25
10	99.66	43

4.6.9 Segmented Image

First, begin by applying semantic segmentation to a single image. Semantic segmentation is a deep learning technique that assigns a specific label or category to each pixel in an image. This method is used to distinguish groups of pixels belonging to different categories or classes within the image.

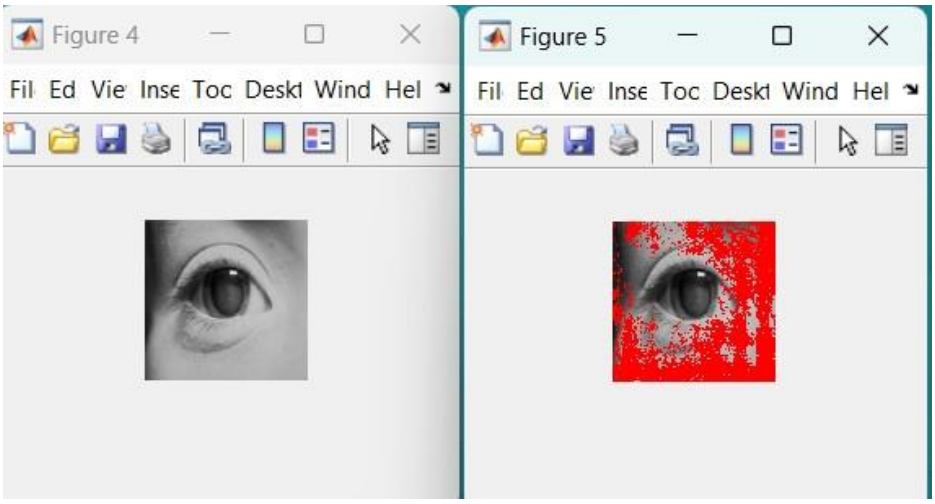

Figure 19: Original image and segmented image.

To achieve more accuracy of segmented image, load pre-trained network with the test datasets.

4.6.10 Load Pre-trained Network

Utilising pre-trained CNNs offers several advantages, notably reducing the time and effort required for training. These networks leverage their learned knowledge of general features and patterns from extensive datasets, making them effective for tasks such as object detection. With pre-trained CNNs, satisfactory results can often be achieved with smaller datasets, shorter training periods, and lower learning rates.

Another significant advantage of pre-trained CNNs is their capability to mitigate issues related to overfitting, where a model becomes overly specialised to the training data and fails to generalise to new data. By employing regularization techniques and minimizing the number of trainable parameters, pre-trained CNNs can effectively adapt to new datasets.

Testing a pre-trained and saved network, such as "myCNN.mat," on new images typically yields satisfactory results. However, discrepancies in the expected outputs may arise, especially when tested images exhibit different pixel values or unexpected variations in appearance and the related code displays in Appendix E.

Figure 20: Testing output of pre-trained network apply for segmented image.

4.6.11 Threshold Value

Image processing plays a crucial role in extracting and isolating pertinent information from images. One fundamental technique within image segmentation is thresholding, which aids in distinguishing significant objects or features from the background by segmenting an image based on pixel intensity or value. This process modifies pixels to convert a grayscale or color image into a binary image, where pixels are either classified as foreground (object) or background based on a specified threshold value.

The corresponding code for applying thresholding to convert an image into a binary format can be found in Appendix F [54].

Figure 21: The testing output of segmented image after adding threshold value.

Based on the results, a higher percentage of eyes have been successfully segmented compared to previous attempts. In the command window, the output for a 100 x 100 matrix appears as depicted in the following figure.

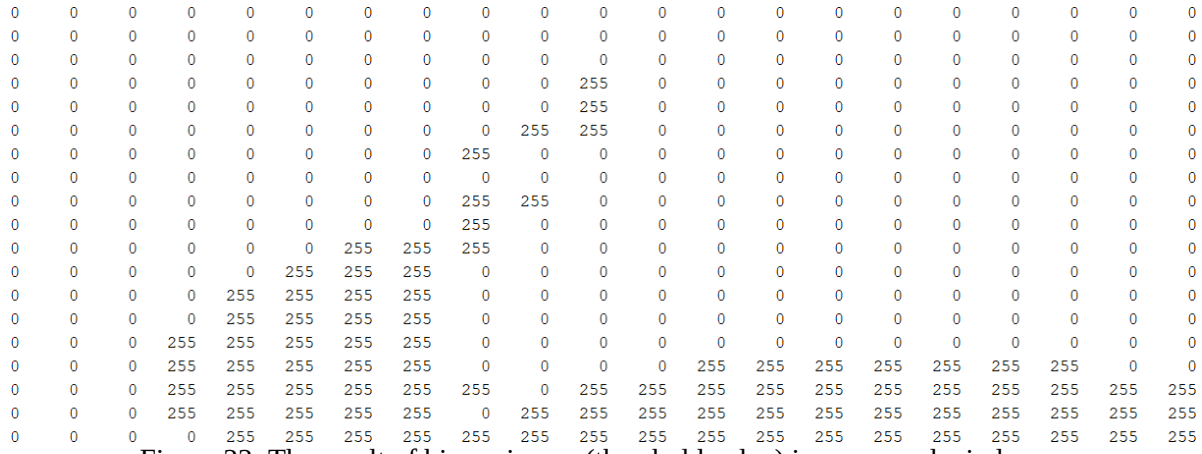

Figure 22: The result of binary image (threshold value) in command window.

The following figure depicts binary images of eyes after applying a threshold value. The image shows the foreground (eye) and background separately, represented in black and white.

Figure 23: The test result of BW of image after applying threshold value.

4.7 Eye-tracking using Circular Hough Transform (CHT)

4.7.1 Apply CHT for an Image

The "imfindcircles" function utilises the Circular Hough Transform (CHT) to detect circles in images. This technique is particularly robust against noise, variations in lighting conditions, and complex backgrounds such as faces and barriers. Implementation of CHT methods can vary, offering flexibility

rather than strict algorithmic specifications. After binarizing an image, the Hough Transform, possibly in conjunction with an edge detector, identifies circular objects. Unlike traditional methods that require a precise radius beforehand, CHT operates efficiently by accepting a range of radius, which enhances its speed compared to iterative radius checks.

Detecting multiple circles is achieved by identifying peaks in the 3D accumulator array. Appendix H displays the relevant code demonstrating how to find circles and curved objects within an image, and subsequently visualizing the detected circles:

Step 1: Display the image.

Step 2: Define the range of radii to search for circles.

Step 3: Perform an initial circle detection attempt.

Step 4: Refine the circle detection sensitivity.

Step 5: Draw the detected circles on the image.

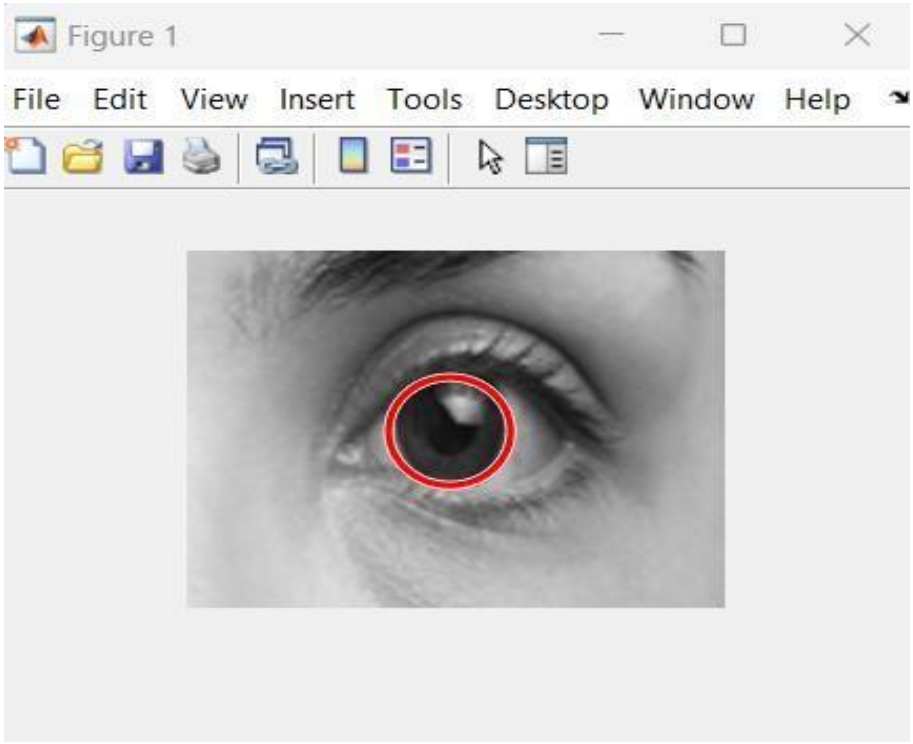

Figure 24: Results of eye detection using the Circular Hough Transform

When the radius is less than or equal to 5, imfindcircles' accuracy is constrained. Depending on the input image, concentric circles might produce different outcomes. Circles with centres outside the image's domain are not found using imfindcircles. Before processing, imfindcircles uses the function rgb2gray to convert truecolor photos to grayscale.

The im2single function is used to convert binary and integer type images to the data type single before processing. As a preprocessing step, imfindcircles additionally uses “imfilter” to apply Gaussian smoothing to binary pictures, improving the accuracy of the result [55].

4.7.2 Apply CHT for a Video

The Circular Hough Transformation was applied to a 15-second video featuring both open and closed eyes, sourced from the website previously mentioned [53]. The objective was to track the movement of the eyes using the algorithm across each frame of the video. The attached image represents one of the frames extracted from the video.

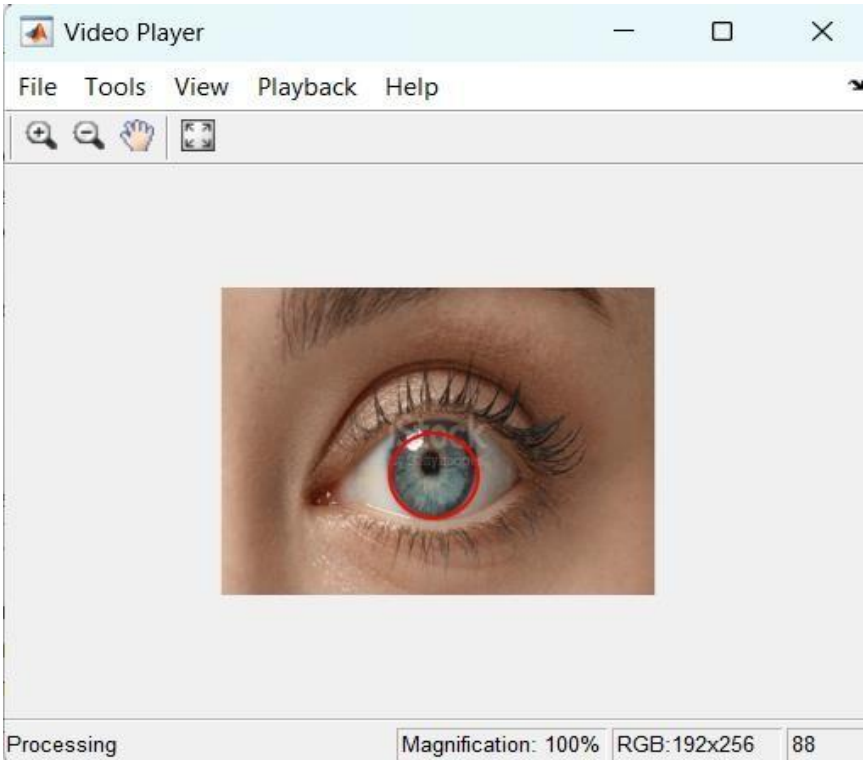

Figure 25: The Circular Hough Transform applied to a video detects eyes frame by frame, drawing a circle around each identified region [53].

4.7.3 Applying CHT for Both Raw Image and Segmented Image

The Circular Hough Transform is a robust method employed to detect basic geometric structures within images, even when these structures are deformed, incomplete, or partially obscured. It proves effective in identifying not only circles but also lines, ellipses, and other fundamental shapes present in images.

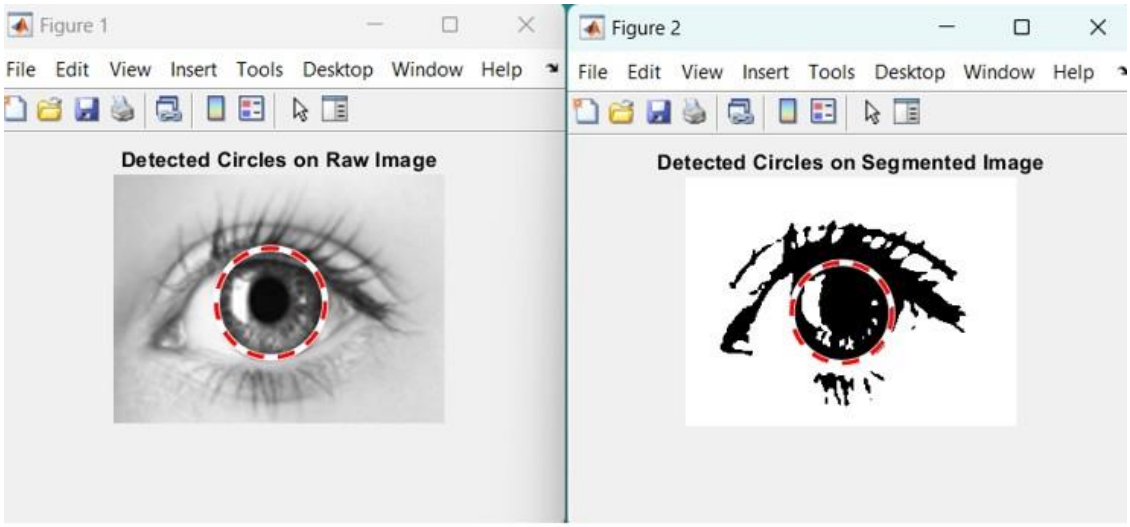

Figure 26: The images show the results of applying CHT algorithm. (a) displays the detected circle on the raw image. (b) displays the detected circles on segmented image of original image.

4.7.4 Apply CHT for a Segmented Video

The provided code processes each frame from the given video by performing thresholding, semantic segmentation, circle detection, and subsequently saves the processed frames into a new video file. Detecting every circle in real-time video playback can be challenging. To achieve slow-motion playback, each frame can be repeated multiple times when writing to the output video file. Alternatively, inserting a "pause (0.8)" command at the end of the while loop allows adjusting the pause duration from 0.5 to 1 second, effectively slowing down the video playback speed.

Slow-motion playback is an effective method to decrease the speed of video playback. The slow-motion factor can be adjusted to any positive integer value to control the playback speed. However, practical considerations such as video duration, frame rate, processing time, and storage capacity should be considered. Generally, slow-motion effects are typically achieved using factors ranging from 2 to 10. Beyond this range, it's important to assess whether the increased resource demands justify the slow-motion effects.

Moreover, the effectiveness of circle detection in the video depends on the parameters set for minimum radius, maximum radius, and sensitivity within the code. The video in question spans 5 seconds of eye movement. The corresponding code can be found in Appendix G.

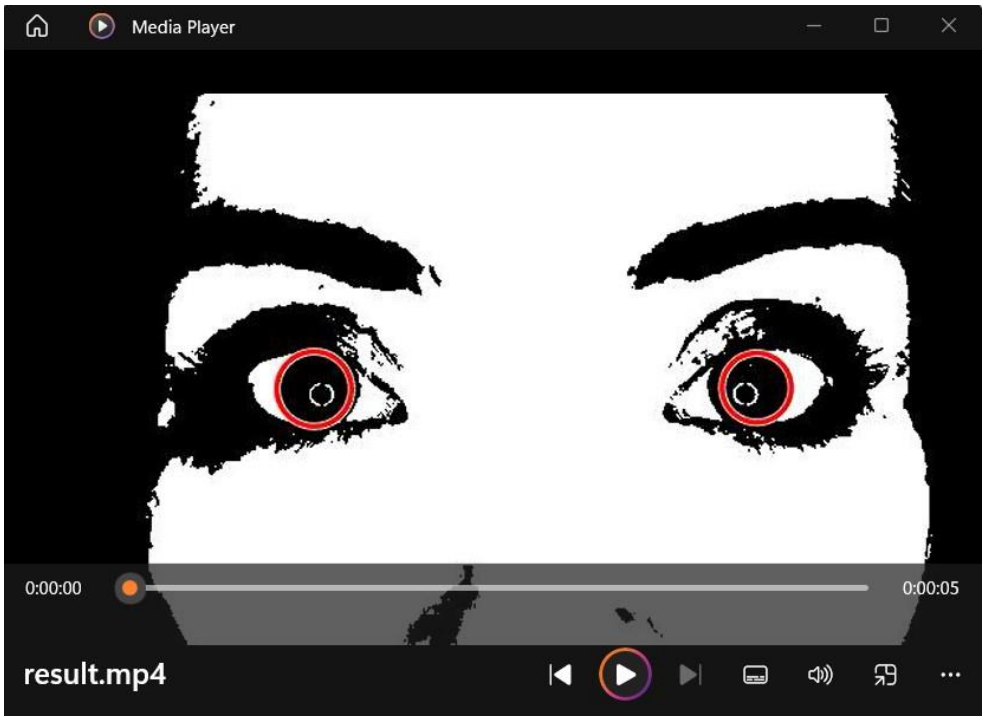

Figure 27: The results of one frame of video player after applying “imfindcircle” or Circular Hough Transformation into a segmented video [53].

4.7.5 Apply CHT for Pre-trained Dilated Eye Network for Images

Firstly, establish a network for the dilated eye network to conduct testing, training, and validation on eye images prior to segmentation. The network provides results such as designing a network by connecting all layers, error analysis, displaying training progress with accuracy percentages, and demonstrating two figures: the original image and the segmented image. Additionally, create a ".mat" file to store trained eye images.

Secondly, load pre-trained data and retest to enhance result accuracy. Thirdly, apply a threshold value to achieve better outcomes than previous results. Furthermore, apply theory to segmented images to improve accuracy rates.

The figure below depicts the output after applying Circular Hough Transformation to results provided by a pre-trained dilated network. Combining the Circular Hough Transform with the pre-trained dilated eye network yielded promising results. Specifically, out of a total of 170 test images, the method proved effective for 158. However, due to the difficulty in presenting all successful detections, a screenshot was created featuring 20 selected test images. Notably, 17 of these images successfully detected eyes according to the algorithms. This focused selection effectively illustrates the outcomes of the approach, highlighting its potential for further applications. The related code can be found in Appendix G.

Figure 28: Eye Detection Results of Pre-trained Dilated Eye Network with Circular Hough Transform

Chapter 5: Conclusion

The study utilized MATLAB's DNN and image processing techniques to develop an eye-tracking system. Initial attempts with various image processing methods were unsuccessful, leading to the adopting of the dilated eye network within a CNN framework for dataset analysis. This approach enables a comprehensive understanding of significant regions and object boundaries through pixel-level categorization, known as semantic segmentation. The study found that the dilated eye network, in conjunction with semantic segmentation, outperformed previous algorithms, effectively minimizing cross-entropy loss and accurately identifying pertinent characteristics. Additionally, the integration of the Circular Hough Transform for feature extraction provided robust identification in noisy and occluded environments, surpassing earlier methods in accuracy. The findings underscore the efficacy of combining a pre-trained network with advanced segmentation and feature extraction techniques for developing a robust eye-tracking system, holding promise for applications in monitoring eye health, particularly among younger populations.

The study introduces an innovative approach by integrating a dilated eye network into a CNN framework for eye-tracking. This significantly improves semantic segmentation compared to traditional methods. The system maintains image resolution and reduces computational load by employing atrous dilations, setting it apart from other algorithms such as GrabCut, Lazy Snapping, and Viola-Jones. The inclusion of the Circular Hough Transform enhances the system's robustness, enabling reliable performance in noisy and occluded environments. Overall, this research offers a more efficient and adaptable solution for eye-tracking, particularly beneficial for monitoring eye health in diverse populations.

The thesis aims to improve understanding of eye conditions by measuring eye movements in four directions and quantifying blink frequency using high-resolution cameras, eye-tracking software, and ML algorithms for accurate data analysis. The study will control for variables such as lighting and participant demographics and implement standardized protocols for consistency. A diverse participant pool will improve generalizability, and statistical methods will help identify correlations between eye movement patterns and specific conditions. Visual feedback mechanisms and a longitudinal approach may provide further insights into how these factors change, contributing to a comprehensive understanding of eye health and potential treatment outcomes.

References

[1] Pauszek, J. R. (2023). An introduction to eye tracking in human factors healthcare research and medical device testing. *Human Factors in Healthcare*, *3*, 100031. https://doi.org/https://doi.org/10.1016/j.hfh.2022.100031

[2] Robert J.K. Jacob, K. S. K. (2002). Eye tracking in human-computer interaction and usability 26. https://doi.org/10.1016/B978-044451020-4/50031-1

[3] Punde, P., Jadhav, M., & Manza, R. (2017). *A study of eye tracking technology and its applications*. https://doi.org/10.1109/ICISIM.2017.8122153

[4] Brunyé, T. T., Drew, T., Weaver, D. L., & Elmore, J. G. (2019). A review of eye tracking for understanding and improving diagnostic interpretation. *Cogn Res Princ Implic*, *4*(1), 7. https://doi.org/10.1186/s41235-019-0159-2.

[5] Santhoshikka R, L. C. R., Harshavarthini C, Preetha R, Saran kumar K. (2021). Eye Tracking and Its Applications. *IARJSET*, *8*(8), 130. https://doi.org/10.17148/IARJSET.2021.8824 (IARJSET International Advanced Research Journal in Science, Engineering and Technology)

[6] Cantero, A., Lebrato, C., Castro, J., Merino Monge, M., Biscarri-Triviño, F., & Escudero, J. I. (2024). A review on visible-light eye-tracking methods based on a low-cost camera. Journal of Ambient Intelligence and Humanized Computing, 15. https://doi.org/10.1007/s12652-024-04760-8

[7] Chen, Y., Guo, Q., Qiao, C., & Wang, J. A systematic review of the application of eye-tracking technology in reading in science studies. *Research in Science & Technological Education*, 1-25. https://doi.org/10.1080/02635143.2023.2285297

[8] Kerr, R., & Fuad, M. M. M. (2019). A Real-Time Lazy Eye Correction Method for Low Cost Webcams. *Procedia Computer Science*, *159*, 281-290. https://doi.org/https://doi.org/10.1016/j.procs.2019.09.183

[9] Burch, M., Haymoz, R., & Lindau, S. (2022). The Benefits and Drawbacks of Eye Tracking for Improving Educational Systems. https://doi.org/10.1145/3517031.3529242

[10] Tahri Sqalli, M., Aslonov, B., Gafurov, M., Mukhammadiev, N., & Sqalli Houssaini, Y. (2023). Eye tracking technology in medical practice: a perspective on its diverse applications. *Front Med Technol*, *5*, 1253001. https://doi.org/10.3389/fmedt.2023.1253001

[11] Zdarsky, N., Treue, S., & Esghaei, M. (2021). A Deep Learning-Based Approach to Video-Based Eye Tracking for Human Psychophysics. *Front Hum Neurosci*, *15*, 685830. https://doi.org/10.3389/fnhum.2021.685830

[12] Arsenovic, M., Sladojevic, S., Stefanovic, D., & Anderla, A. (2018). *Deep neural network ensemble architecture for eye movements classification*.

[13] Alzubaidi LH, Hameed Abdul Hussein A, Ayad Alkhafaji M, Shilpa N, N P T. Efficient Real-Time Eye Gaze Tracking Detection for Human-Computer Integration Using Advanced Techniques2023. 1-6 p.

[14] Yu, M., Tang, X., Lin, Y., Schmidt, D., Wang, X., Guo, Y., & Liang, B. (2018). *An eye detection method based on convolutional neural networks and support vector machines* (Vol. 22).

[15] Robert J. K. Jacob, K. S. K. (2023). Eye Tracking in Human-computer Interaction Recognition. 203-207. https://doi.org/10.1016/B978-044451020-4/50031-1 (Elsevier Science)

[16] Yu M, Tang X, Lin Y, Schmidt D, Wang X, Guo Y, Liang B. An eye detection method based on convolutional neural networks and support vector machines2018. 345-62 p.

[17] Li, Y., Deng, J., Wu, Q., & Wang, Y. (2021). *Eye-Tracking Signals Based Affective Classification Employing Deep Gradient Convolutional Neural Networks* (Vol. 7).

[18] Kurdthongmee, W., Kurdthongmee, P., Suwannarat, K., & Kiplagat, J. K. (2022). *A YOLO Detector Providing Fast and Accurate Pupil Center Estimation using Regions Surrounding a Pupil* (Vol. 6).

[19] Su, M.-C., Tat-Meng, U., Hsieh, Y.-Z., Yeh, Z.-F., Lee, S.-F., & Lin, S.-S. *An eye-tracking system based on inner corner-pupil center vector and deep neural network* (Vol. 20). MDPI AG.

[20] Ruiz-Beltran, C. A., Romero-Garces, A., Gonzalez-Garcia, M., Marfil, R., & Bandera, A. (2023). *FPGA-Based CNN for Eye Detection in an Iris Recognition at a Distance System* (Vol. 12).

[21] Husham Almukhtar, F., Abbas Ajwad, A., Kamil, A. S., Adil Kamil, R., Jaleel, R. A., & Jalal Mosa, S. *Deep Learning Techniques for Pattern Recognition in EEG Audio Signal-Processing-Based Eye-Closed and Eye-Open Cases* (Vol. 11). MDPI.

[22] 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* (Vol. 8). Multidisciplinary Digital Publishing Institute (MDPI).

[23] Cheng, S., Fan, J., & Hu, Y. *Visual saliency model based on crowdsourcing eye tracking data and its application in visual design* (Vol. 27). Springer Science and Business Media Deutschland GmbH.

[24] Yılmaz, H., & Özdem, M. (2024). A Solution for Individuals with Disabilities to Use a Computer Through Eye Movements. *12*(1), 224-232. https://doi.org/10.29109/gujsc.1404305

[25] Cilia, F., Le Driant, B., Carette, R., Elbattah, M., Dequen, G., Guérin, J.-L., Bosche, J., & Vandromme, L. *Computer-aided screening of Autism Spectrum disorder: Eye-tracking study using data visualization and deep learning* (Vol. 8). JMIR Publications Inc.

[26] Birawo, B., & Kasprowski, P. (2022). *Review and Evaluation of Eye Movement Event Detection Algorithms* (Vol. 22).

[27] Zdarsky, N., Treue, S., & Esghaei, M. (2021). A Deep Learning-Based Approach to Video-Based Eye Tracking for Human Psychophysics. *Front Hum Neurosci*, *15*, 685830. https://doi.org/10.3389/fnhum.2021.685830

[28] Chen, Z., Fu, H., Lo, W.-L., & Chi, Z. *Strabismus Recognition Using Eye-Tracking Data and Convolutional Neural Networks* (Vol. 2018). Hindawi Limited. (Journal of Healthcare Engineering)

[29] Ahmed, I. A., Shatnawi, H. S. A., Alwazer, S. M., Alshahrani, M., Senan, E. M., Rassem, T. H., & Ali, M. A. H. (2022). Eye Tracking-Based Diagnosis and Early Detection of Autism Spectrum Disorder Using Machine Learning and Deep Learning Techniques. *11*(4). https://doi.org/10.3390/electronics11040530

[30] Lim, J. Z., Mountstephens, J., & Teo, J. (2021). Eye-Tracking Feature Extraction for Biometric Machine Learning. *15*. https://doi.org/10.3389/fnbot.2021.796895

[31] Zemblys, R., Niehorster, D., & Holmqvist, K. (2018). Correction to: "Using machine learning to detect events in eye-tracking data". *Behavior Research Methods*, *51*. https://doi.org/10.3758/s13428-018-1127-3

[32] Bonteanu, G., Bonteanu, P., Cracan, A., & Bozomitu, R. G. (2024). *Implementation of a High-Accuracy Neural Network-Based Pupil Detection System for Real-Time and Real-World Applications* (Vol. 24).

[33] Hu, T., Wang, X., & Xu, H. (2022). Eye-Tracking in Interpreting Studies: A Review of Four Decades of Empirical Studies. *Front Psychol*, *13*, 872247. https://doi.org/10.3389/fpsyg.2022.872247

[34] Gómez-Poveda, J., & Gaudioso, E. *Evaluation of temporal stability of eye tracking algorithms using webcams* (Vol. 64). Elsevier Ltd.

[35] Tsai, M.-J., Hou, H.-T., Lai, M.-L., Liu, W.-Y., & Yang, F.-Y. (2012). *Visual Attention for Solving Multiple-Choice Science Problem: An Eye-Tracking Analysis* (Vol. 58).

[36] Schneider A, Vollenwyder B, Krueger E, Mühlethaler C, Miller DB, Thurau J, Elfering A. Mobile eye tracking applied as a tool for customer experience research in a crowded train station. J Eye Mov Res. 2023;16(1).

[37] Lim JZ, Mountstephens J, Teo J. Emotion Recognition Using Eye-Tracking: Taxonomy, Review and Current Challenges2020. 2384-404 p.

[38] Deng, R., & Gao, Y. (2023). A review of eye tracking research on video-based learning. *Educ Inf Technol (Dordr)*, *28*(6), 7671-7702. https://doi.org/10.1007/s10639-022-11486-7

[39] Liu, Y., & Tian, L. (2013). An Improved Algorithm on Adaptive KLT Vision Tracking. *Advanced Materials Research*, *631-632*, 1270-1275. https://doi.org/10.4028/www.scientific.net/AMR.631-632.1270

[40] Wang, Z., Lv, Y., Wu, R., & Zhang, Y. (2023). Review of GrabCut in Image Processing. *Mathematics*, *11*(8), 1965. https://www.mdpi.com/2227-7390/11/8/1965

[41] Peng Y, Zhang Z, He G, Wei M. An Improved GrabCut Method Based on a Visual Attention Model for Rare-Earth Ore Mining Area Recognition with High-Resolution Remote Sensing Images. Remote Sensing. 2019;11(8):987.

[42] Li, Y., Sun, J., Tang, C.-K., & Shum, H.-Y. (2004). Lazy snapping. *ACM Trans. Graph.*, *23*, 303-308. https://doi.org/10.1145/1015706.1015719

[43] Paul, T., Shammi, U., Ahmed, M., Rahman, R., Kobashi, S., & Ahad, M. A. R. (2018). A Study on Face Detection Using Viola-Jones Algorithm in Various Backgrounds, Angles and Distances. https://doi.org/10.24466/ijbschs.23.1_27

[44] Kaddouhi, S. E., Saaidi, A., & Abarkan, M. (2017). Eye detection based on the Viola-Jones method and corners points. *Multimedia Tools and Applications*, *76*. https://doi.org/10.1007/s11042-017-4415-5

[45] Arsenovic, M., Sladojevic, S., Stefanovic, D., & Anderla, A. (2018). *Deep neural network ensemble architecture for eye movements classification*.

[46] Kaur R, GholamHosseini H, Sinha R, Lindén M. Automatic lesion segmentation using atrous convolutional deep neural networks in dermoscopic skin cancer images. BMC Medical Imaging. 2022;22(1):103.

[47] Izonin I, Tkachenko R, Shakhovska N, Ilchyshyn B, Singh KK. A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain. Mathematics. 2022;10(11):1942.

[48] Wang Y, Li Y, Song Y, Rong X. The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition. Applied Sciences. 2020;10(5):1897.

[49] Perry, J., & Fernandez, A. (2019). *MinENet: A Dilated CNN for Semantic Segmentation of Eye Features*. https://doi.org/10.1109/ICCVW.2019.00453

[50] Bozomitu RG, Pasarica A, Cehan V, Lupu RG, Rotariu C, Coca E. Implementation of eye-tracking system based on circular Hough transform algorithm. 2015 E-Health and Bioengineering Conference (EHB). 2015:1-4.

[51] Khattab, D., Theobalt, C., Hussein, A. S., & Tolba, M. F. (2014). Modified GrabCut for human face segmentation. *Ain Shams Engineering Journal*, *5*(4), 1083-1091. https://doi.org/https://doi.org/10.1016/j.asej.2014.04.012

[52] Kaddouhi, S. E., Saaidi, A., & Abarkan, M. (2018). Eye Detection based on Viola & Jones Detector, Skin Color, and Eye Template *11*. (International Journal of Control and Automation)

[53] *iStock by Getty Images*. (2024). https://www.istockphoto.com/search/search-by-asset?assetid=1453465854&assettype=image

[54] Pasarica, A., Bozomitu, R., Oana-Diana, H.-E., Tarniceriu, D., & Rotariu, C. (2016). *Analysis of different threshold selection methods for eye image segmentation used in eye tracking applications*. https://doi.org/10.1109/DAAS.2016.7492591

[55] David Young (2024). Hough transform for circles (https://www.mathworks.com/matlabcentral/fileexchange/26978-hough-transform-for-circles), MATLAB Central File Exchange. Retrieved June 24, 2024.

Appendix A

The code explains the face detection using KLT algorithm by visionCascadeObjecDetection and it draws a boundary box around the face for detection.

```
clc
clear

vidObj = VideoReader('d.mp4');
currAxes = axes;

x = vidObj.NumFrames;

faceDetector = vision.CascadeObjectDetector();

       for cnt = 1:5:x
           vidFrame = read(vidObj, cnt);

            bbox = step(faceDetector, vidFrame);
           vidFrame = insertShape(vidFrame, 'rectangle', bbox, ...
               LineWidth=5);
           imagesc(vidFrame, 'Parent', currAxes);
       end
```

Appendix B

The code is used to detect eyes using KLT algorithm by visionCascadeObjectDetector. A video is used to take an output using frame by frame and it draws a boundary box around the eyes.

```
clc
clear

vidObj = VideoReader('g.avi');

currAxes = axes;

x = vidObj.NumFrames;

EyeDetector = vision.CascadeObjectDetector;
       for cnt = 1:150:x
           vidFrame = read(vidObj,cnt);

             bbox = step(EyeDetector, vidFrame);

            vidFrame = insertShape(vidFrame, 'rectangle', bbox, ...
               LineWidth=5,Color='blue',Opacity=0.6,SmoothEdges=true);

           clow = 150;
           chigh = 200;
           clim = [clow chigh];
           imagesc(vidFrame, 'Parent', currAxes,clim);

           EyeDetector = vision.CascadeObjectDetector('EyePairBig');
           EyeDetector.MinSize = [11 45];
           EyeDetector.MergeThreshold = 5000;
           EyeDetector.UseROI = false;

 % ------------------------------------------------------------------------------------------------
    L = superpixels(vidFrame,20000);

    foregroundX = [500 500 700 700];
    foregroundY = [420 370 370 420];


     foregroundInd = sub2ind(size(vidFrame),foregroundY,foregroundX);


    backgroundX = [200 1000];
    backgroundY = [100 100];

    backgroundInd = sub2ind(size(vidFrame),backgroundY,backgroundX);

     BW = lazysnapping(vidFrame,L,foregroundInd,backgroundInd);
 %-----------------------------------------------------------------

           h1 = drawpolygon(gca,'Position',[530,430;530,370;730,370;730,430]);
 %-------------------------------------------------------------

              roi = createMask(h1,vidFrame);
```

```
%-----------------------------------------------------------------
              maskedImage = imfill(vidFrame,"holes");
            maskedImage(repmat(~BW,[1 1 3])) = 0;

    figure;
    % imshow(maskedImage)
 % imshow(BW)
% figure

Array1 = repmat(BW, [1 1 1 2]);
montage(Array1)
Array2 = repmat(maskedImage, [1 1 1 2]);
montage(Array2)

% imshow(vidFrame(:,:,2)-BW);

subplot(1,2,1),imshow(BW);
subplot(1,2,2),imshow(maskedImage);

       end
```

Appendix C

The code for resize input images of the datasets which used to resize all the image using a for loop, in the folder by MATLAB and it automatically save in the given pathway or in a new file.

```
srcFiles = dir('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test jpg\*.jpg');
for i = 1 : length(srcFiles)
filename = strcat('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test jpg\',srcFiles(i).name);
im = imread(filename);
k=imresize(im,[100 100]);
newfilename=strcat('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test jpg\',srcFiles(i).name);
imwrite(k,newfilename,'png');
end
```

Appendix D

The code is used to convert RGB to gray scale and ‘jpg’ to ‘png’. It removes the complexity associated with computational requirements and aids in the simplification of methods. It creates space for more easily learned image processing for beginners. This is so because an image is compressed to the minimal minimum of pixels in grayscale, and it makes visualising easier.

```
% srcFiles = dir('D:\AUT MPhil Research\eye-tracking ROBOFLOW universe
dataset\valid\*.jpg');
RGB = dir('D:\AUT MPhil Research\eye-tracking image datasets and ROBOFLOW\test
jpg\*.jpg');
for i = 1 : length(RGB)
filename = strcat('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test jpg\',RGB(i).name);
im = imread(filename);
I = im2gray(im);
newfilename=strcat('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test jpg\',RGB(i).name);
imwrite(I,newfilename,"png");
end
```

Appendix E

Load the trained network to pre-trained network with tested datasets to get more accuracy of results.

```
net=load ('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\myCNN3.mat');
srcFiles = dir('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test\*.png');

for i = 1 : length(srcFiles)
% testImage = imread('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test\*.png');
filename = strcat('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test\',srcFiles(i).name);
im = imread(filename);
% figure(1), imshow(testImage);
[C,scores] = semanticseg(im,net.net);
B=labeloverlay(im,C,'Colormap','hsv','Transparency',0,'IncludedLabels',"Eye");
newfilename = strcat('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\testSegmentationImage1\',srcFiles(i).name);

imwrite(B,newfilename,"png");
% figure(2),imshow(B);
end
```

Appendix F

Add threshold value for pre-trained network to get more accuracy of segmented images. Newly added nested for loop for columns and rows with the conditions.

```
net = load('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\myCNN3.mat');
net = net.net;

srcFiles = dir('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\test\*.png');

for k = 1:length(srcFiles)

    filename = fullfile(srcFiles(k).folder, srcFiles(k).name);

    im = imread(filename);


    if size(im, 3) == 3
        im = rgb2gray(im);
    end

    threshold = mean(im(:));

    binaryImage = im;
    [rows, cols] = size(im);
    for i = 1:rows % Iterate over rows
        for j = 1:cols % Iterate over columns
            if binaryImage(i, j) < threshold
                binaryImage(i, j) = 0;
            else
                binaryImage(i, j) = 255;
            end
        end
    end

    [C, scores] = semanticseg(binaryImage, net);
    B = labeloverlay(binaryImage, C, 'Colormap', 'hsv', 'Transparency', 0,
'IncludedLabels', "Eye");

    newFilename = fullfile('D:\AUT MPhil Research\eye-tracking image datasets and
ROBOFLOW\testSegmentedimageWithThreshold\', srcFiles(k).name);


    imwrite(B, newFilename, 'png');
end
```

Appendix G

The code displays a result for tracking an eye frame by frame using a video, applying Circular Hough Transformation (CHT) algorithm by “imfindcircle” function in MATLAB.

```
videoFile = 'D:\AUT MPhil Research\eye tracking video apply DilatedEyeNet code\test video\test1.mp4';

vidReader = VideoReader(videoFile);

outputVideo = VideoWriter('D:\AUT MPhil Research\eye tracking video apply DilatedEyeNet code\result1.mp4');

open(outputVideo);

slowMotionFactor = 2;

while hasFrame(vidReader)


    frame = readFrame(vidReader);

    if size(frame, 3) == 3

        grayFrame = rgb2gray(frame);

    else

        grayFrame = frame;

    end

    threshold = mean(grayFrame(:));

    binaryImage = grayFrame;


    [rows, cols] = size(grayFrame);

    for i = 1:rows % Iterate over rows

        for j = 1:cols % Iterate over columns

            if binaryImage(i, j) < threshold

                binaryImage(i, j) = 0;

            else

                binaryImage(i, j) = 255;

            end

        end

    end
```

```
    [C, scores] = semanticseg(binaryImage, net);
    B = labeloverlay(binaryImage, C, 'Colormap', 'hsv', 'Transparency', 1, 'IncludedLabels', "Eye");
    [centers, radii] = imfindcircles(binaryImage, [15 30], 'ObjectPolarity', 'dark', 'Sensitivity', 0.95);
    imshow(B); title('Detected Eyes');
    hold on;
    viscircles(centers, radii, 'EdgeColor', 'r');

    frameWithOverlay = getframe(gca);
    writeVideo(outputVideo, frameWithOverlay.cdata);
    close(gcf);
    pause(0.9)
end
close(outputVideo);
```

Appendix H

The code displays a result for tracking an eye using Circular Hough Transformation (CHT) algorithm by “imfindcircle” function in MATLAB.

```
net = load('D:\AUT MPhil Research\eye tracking images apply DilatedEyeNet code\myCNNtest1.mat');

net = net.net;

srcFiles = dir('D:\AUT MPhil Research\eye tracking images apply DilatedEyeNet code\test\*.png');

for k = 1:length(srcFiles)

    filename = fullfile(srcFiles(k).folder, srcFiles(k).name);

    im = imread(filename);

    if size(im, 3) == 3

        im = rgb2gray(im);

    end

    threshold = mean(im(:));

    binaryImage = im;


    [rows, cols] = size(im);

    for i = 1:rows


        for j = 1:cols % Iterate over columns

          if binaryImage(i, j) < threshold

               binaryImage(i, j) = 0;

           else

              binaryImage(i, j) = 255;

           end

        end

    end

    [C, scores] = semanticseg(binaryImage, net);

    B = labeloverlay(binaryImage, C, 'Colormap', 'hsv', 'Transparency', 1, 'IncludedLabels', "Eye");

    [centers, radii] = imfindcircles (binaryImage, [15 57], 'ObjectPolarity', 'dark', 'Sensitivity', 0.95);
```

```
    figure,
    imshow(B),
    title('Detected Eyes');
    hold on;

    viscircles(centers, radii, 'EdgeColor', 'r');

    newFilename = fullfile('D:\AUT MPhil Research\eye tracking images apply
DilatedEyeNet code\APPLYimfindcircleWITHthreshold\', srcFiles(k).name);
    saveas(gcf, newFilename, 'png');
    close(gcf);
end
```