Ministry of Science and Higher Education of the Republic of Kazakhstan

SDU University

Gaukhar Seitkaliyeva

Implementation of Real-time Focus Tracking in Video Streams through Advanced Computational Methods and Image Analysis

THESIS

Presented in Partial Fulfilment for the

Degree of Master of Technical Science in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Meraryslan Meraliyev**

Kaskelen, June 2024

SDU University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, PhD Akhmedov Ramis

_____

« 04 » 06 2024

Implementation of Real-time Focus Tracking in Video Streams through Advanced Computational Methods and Image Analysis

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science", SDU University

Head of Department _____ Zhanar Mukash

Academic Supervisor _____ Meraryslan Meraliyev

Master student _____ Gaukhar Seitkaliyeva

Kaskelen, 2024

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Gaukhar Seitkaliyeva

June 2024

Acknowledgements

I would like to express my gratitude to my supervisor, Meraryslan Merialiyev, for his help in finding an important topic in my area of interest, guiding me on how to carry out the research, and giving me valuable advice. His support was crucial for the success of this work.

Dedication

This thesis is dedicated to my beloved parents, family and friends. Their support, trust, encouragement, and patience have meant so much to me during this journey. I couldn't have done it without their love and understanding.

Abstract

This research presents the development and implementation of a real-time focus tracking system in video streams, utilizing advanced computational methods and image analysis techniques. The core of the system is based on the L2CS-Net model, a convolutional neural network known for its efficacy in gaze estimation. The objective of this research was to use it for real-world applications, particularly in settings where user engagement and attention monitoring are crucial, such as e-learning and virtual meetings.

To address the challenges of variable video quality and user behavior typically encountered outside laboratory conditions, the L2CS-Net model was refined using a comprehensive dataset, the MPIIGaze, with specific focus on preprocessing techniques that enhance training efficiency and model accuracy. The practical deployment of the model was achieved through a web-based interface, developed to provide real-time feedback on user focus. This interface was hosted locally on a Flask API, ensuring ease of deployment and making changes. A key feature of the system is an alert mechanism that notifies users when the detected gaze deviates beyond set thresholds, indicating potential lapses in attention.

Experimental results demonstrate that the enhanced L2CS-Net model achieves high accuracy in gaze prediction. Furthermore, the web application's real-time processing capabilities and the effectiveness of the alert system were validated under operational conditions. Future works can explore the integration of additional sensor data and improvements in alert accuracy, aiming to further enhance user engagement and monitoring in virtual environments.

Аңдатпа

Бұл зерттеу жетілдірілген есептеу әдістері мен кескін талдау техникасын қолдану арқылы видео ағындарындағы нақты уақыттағы фокусты бақылау жүйесін дамыту және енгізуді көрсетеді. Жүйе көздің қарауын бағалауда тиімділігімен танымал конволюциялық нейрондық желі L2CS-Net моделіне негізделген. Бұл зерттеудің мақсаты - модельдің өнімділігін арттыру және әсіресе оны электронды оқыту және виртуалды кездесулер сияқты пайдаланушының қатысуы мен фокусын бақылау маңызды болып табылатын жағдайларда нақты әлемде қолдану үшін бейімдеу.

Зертханалық жағдайлардан тыс әдетте кездесетін айнымалы бейне сапасы мен пайдаланушы әрекеттерінің қиындықтарын шешу үшін L2CS-Net моделі MPIIGaze жан-жақты деректер жиынын пайдаланып нақтыланды, оқыту тиімділігі мен үлгі дәлдігін арттыратын алдын ала өңдеу әдістеріне ерекше мән берілді. Модельдің практикалық қолданылуына пайдаланушының фокусын нақты уақыт режимінде кері байланысты қамтамасыз ету үшін әзірленген веб-негізделген интерфейс арқылы қол жеткізілді. Бұл интерфейс Flask API интерфейсінде жергілікті түрде орнатылды, бұл орналастыру және өзгертулер енгізудің қарапайымдылығын қамтамасыз етеді. Жүйенің негізгі ерекшелігі - анықталған көздің қарауы белгіленген шектерден ауытқыған кезде пайдаланушыларды хабардар ететін ескерту механизмі, бұл назар аударудың ықтимал ауытқуларын көрсетеді.

Эксперимент нәтижелері жақсартылған L2CS-Net үлгісі көздің қарауын болжауда жоғары дәлдікке қол жеткізетінін көрсетеді. Сонымен қатар, веб-қосымшаның нақты уақыттағы өңдеу мүмкіндіктері мен хабардар ететін ескерту жүйесінің тиімділігі операциялық жағдайларда тексерілді. Болашақ жұмыстар ретінде виртуалды ортада пайдаланушының қатысуын және мониторингін одан әрі жақсартуға бағытталған қосымша сенсор деректерін біріктіруді және ескерту дәлдігін жақсартуды зерттеуге болады.

Аннотация

Данное исследование представляет разработку и внедрение системы отслеживания внимания в режиме реального времени в видеопотоках, используя передовые вычислительные методы и техники анализа изображений. Основой системы является модель L2CS-Net, сверточная нейронная сеть, известная своей эффективностью в оценке взгляда. Целью данного исследования было улучшение производительности модели и адаптация ее для реальных условий, особенно в тех случаях, где важно отслеживание внимания и вовлеченности пользователей, таких как электронное обучение и виртуальные встречи.

Для решения проблем переменного качества видео и поведения пользователей, которые обычно встречаются вне лабораторных условий, модель L2CS-Net была улучшена с использованием комплексного набора данных MPIIGaze, с особым акцентом на методы предварительной обработки, которые улучшают эффективность обучения и точность модели. Практическое развертывание модели было достигнуто через веб-интерфейс, разработанный для обеспечения обратной связи в реальном времени по вниманию пользователя. Этот интерфейс был размещен локально на Flask API, что обеспечило простоту развертывания и внесения изменений. Ключевой особенностью системы является механизм оповещения, который уведомляет пользователей, когда обнаруженный взгляд выходит за установленные пределы, указывая на возможное снижение внимания.

Экспериментальные результаты показывают, что улучшенная модель L2CS-Net достигает превосходной точности в прогнозировании взгляда. Кроме того, возможности обработки в реальном времени веб-приложения и эффективность системы оповещения были подтверждены в условиях эксплуатации. Будущие исследования будут направлены на интеграцию дополнительных данных от сенсоров и улучшение точности оповещений, с целью дальнейшего повышения вовлеченности и мониторинга пользователей в виртуальных средах.

Abbreviations

API — Application Programming Interface
AUC — Area Under the Curve
CNN — Convolutional Neural Network
CORS — Cross-Origin Resource Sharing
CPU — Central Processing Unit
DL — Deep Learning
FPR — False Positive Rate
GAN — Generative Adversarial Network
GPU — Graphics Processing Unit
HMD — Head-Mounted Display
IMU — Inertial Measurement Unit
LSTM — Long Short-Term Memory
MAE — Mean Absolute Error
MPIIGaze — Multi-PIE Gaze Dataset
MSE — Mean Squared Error
NIR — Near-Infrared
NVIDIA — A leading manufacturer of GPUs and AI hardware
ROC — Receiver Operating Characteristic
SLAM — Simultaneous Localization and Mapping
TPR — True Positive Rate
YOLO — You Only Look Once

Table of Contents

Chapter 1

Background and motivations

1.1 Introduction

In educational and professional settings, maintaining attention and focus is essential. Especially during online examinations or e-learning, tracking a participant's focus ensures that the examination's integrity is preserved. Any exam can contain possibilities for cheating, and e-learning is no exception. According to a survey in [1], cheating on online exames was reported by a significant (44.7%) of students in total. Before the COVID-19 pandemic, 29.9% of students reported cheating, but during the pandemic, the percentage increased to 54.7%.

The integrity of online examinations is more than just an academic concern; it underpins the credibility of entire educational institutions. Widespread cheating in online settings can devalue digital certifications and it's essential to combine conventional cheating detection methods with modern digital monitoring and verification techniques to ensure the integrity of assessments in online examinations [2].

As online education surges in all parts of the world, especially post-global shifts like the COVID-19 pandemic, and it's here to stay [3], ensuring authentic assessment becomes essential. Thus, exploring technological solutions, such as real-time focus tracking, is vital to preserving the reputation and efficacy of digital learning platforms.

Additionally, the study by Lei [4] demonstrates a moderate positive correlation between student engagement (emotional, cognitive, behavioral) and academic performance, suggesting that increased levels of engagement are linked to improved academic performance.

It underscores the role of behavioral engagement as the most influential on academic success. These insights suggest that monitoring and enhancing student engagement could be pivotal in improving educational outcomes.

There is an increasing need for efficient methods to ensure that participants in online exams or virtual meetings maintain their attention [5]. Traditional surveillance methods are manpower-intensive, potentially intrusive, and can't scale efficiently. An automated real-time solution using video streams could address this challenge.

However, implementing such a solution is not without its complexities. First

and foremost, accurately detecting and analyzing objects in real-time through video requires advanced computational techniques [6]. Moreover, in order to track an object's focus, algorithms will be needed that are capable of discerning subtle facial and ocular changes. It's essential that these algorithms are not only accurate but also efficient to prevent lag or delay, which could disrupt the online exam or meeting process.

Lastly, the diverse range of hardware and software environments in which such a system would operate poses additional challenges. It's critical to ensure compatibility across various devices and platforms while maintaining a consistent level of accuracy in focus tracking.

1.2 Aim and objectives of the Research

The goal of this research is to create a real-time focus tracking system using the L2CS-Net model, optimized for performance and accuracy in diverse real-world settings. The project focuses on enhancing user engagement in online learning environments and maintaining the integrity of virtual examinations through effective gaze tracking and alert mechanisms.

Objectives of the research are:

1. Train L2CS-Net on MPIIGaze with different batch size.
2. Develop web interface for real-time gaze tracking.
3. Integrate video processing to analyze gaze and alert on deviations.
4. Evaluate system responsiveness in various settings through user testing.

1.3 Research Questions

- How effectively can the L2CS-Net model estimate gaze direction in varying real-world settings?
- How can a web-based interface be used to efficiently process and display real-time gaze data from a video stream?

1.4 Problem Statement

- The credibility of online exams is compromised by widespread cheating.
- Conventional monitoring methods for ensuring focus during online sessions are manpower-intensive.
- There is a significant need for automated, real-time solutions that can efficiently handle gaze tracking.

Chapter 2

Literature Review

2.1 Background and Evolution

The transformation of education in the 21st century has shifted learning from classrooms to online platforms. While early instances of computer-assisted education date back to the 1960s with the University of Illinois using interconnected terminals, the real surge in e-learning began in the 1980s. The University of Toronto introduced the first online course in 1984, and by 1989, the University of Phoenix launched the first all-online academic institution, starting the modern era of e-learning [7]. As technology advanced and global connectivity increased, the importance and reliance on online modes of communication and examination grew exponentially. Digital technologies have significantly increased access to educational resources, allowing learners to access materials anytime and anywhere, which is particularly beneficial for remote and underserved populations. Table 2.1 shows the evolution of Digital Technologies on Education.

Table 2.1 - Evolution and Impact of Digital Technologies on Education

Year	**Technology**	**Impact**	**Challenges Addressed**
1960s	Interconnected terminals	Introduced computer-assisted education	Limited accessibility
1984	First online course (University of Toronto)	Initiated the era of e-learning	Physical classroom limitations
1989	University of Phoenix online institution	Expanded e-learning globally	Scalability of education
2020s	Google Classroom, Google Meet	Facilitated learning during the pandemic	Accessibility, engagement, teacher-student interaction
Various	Gamification, Virtual Labs	Increased student engagement and motivation	Passive learning experiences

The use of digital tools and resources in education has been found to improve learning outcomes by making learning more interactive, engaging, and tailored to individual learners' needs. Technologies like gamification, virtual laboratories, and interactive simulations have been demonstrated to enhance student engagement and motivation. [8]. Today, they are integral to the educational landscape, enabling greater accessibility, flexibility, and opportunities for learners worldwide. The research [9] illustrates the critical role of accessible technology and digital platforms in facilitating learning during the pandemic. The high percentages of students adapting to and proficiently using tools like Google Classroom and Google Meet underscore the digital shift in educational practices.

However, as digital education and virtual meetings became standard, new challenges, like the need for self-discipline, poor time management, and lack of teacher-student, emerged. Realizing its full potential requires addressing existing challenges, including ensuring equitable access to technology and reimagining teaching methodologies to leverage digital tools effectively.[10]. The shift from traditional settings heightened the need for participant engagement and focus. Researchers have found that learners who are attentive and engaged outperform their counterparts. Furthermore, the findings suggest that efforts to increase student engagement could be particularly beneficial for lower-ability students and that institutional practices can significantly influence the effectiveness of engagement strategies in improving academic performance.[11]. In remote environments, distractions can easily compromise the integrity of online exams and reduce the efficacy of virtual meetings. Understanding and addressing these challenges in sustaining attention has become very important.

2.2 Advances in Engagement and Gaze Estimation Technologies

2.2.1 Technological Solutions for Engagement Detection

While traditional methods presented limitations, the advancement of technology introduced new opportunities for monitoring attention in online spaces. Early implementations banked on basic motion detection or rudimentary screen activity analysis. However, as the technology matured, more sophisticated solutions arose that utilized a blend of facial recognition, eye-tracking, and behavioral patterns to assess engagement. Batista's work [12] offered a solution to the crucial problem of real-time monitoring of driver alertness. By accurately detecting and analyzing key visual attention indicators like eyelid movement and head orientation, the system offered a tool for enhancing road safety. These systems not only offered a higher accuracy in detecting distractions but also ensured real-time feedback, making it feasible to prompt users immediately [13].

Eye tracking is a sensor-based technology that tracks where a person's gaze is directed, providing insights into attention and focus by capturing eye movements multiple times per second. This data can evaluate gaze direction, blink duration, and cognitive workload using metrics like the Cognitive Activity Index. Eye track-

ers typically consist of a light source and a camera, with infrared light aimed at the eye to monitor pupil movements and ocular features. There are three main types (Figure 2.1) : screen-based, wearable (including glasses and VR headsets with integrated tracking), and webcam-based, chosen based on research needs. [14].

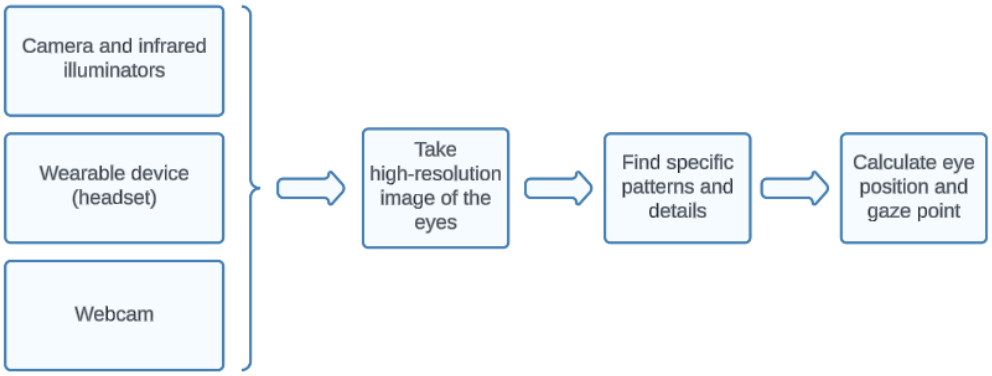


Figure 2.1 - Different types of eye tracking systems

The integration of facial recognition and eye tracking has revolutionized focus detection in online platforms. Facial metrics are powerful indicators of attention, supported by numerous studies. The study found a high level of agreement among human observers on engagement levels and demonstrated that machine learning models could predict engagement with accuracy comparable to human judgments. [15]. Similarly, eye movements - saccades, fixations, and blink rates—offer insights into a learner's engagement. The research by Liu et al. [16] examines the use of synchronized eye movements to evaluate attention engagement in online education. Two experiments with university students revealed that Inter-Subject Correlation (ISC) of eye movements can effectively indicate attention levels. Additionally, more frequent blinking could indicate decreasing attention or tiredness. The review [17] confirmed that eye blinking is a promising physiological measure for assessing fatigue and mental load, supported by the findings from the included studies. Together, these metrics provide a comprehensive view of focus, shaping the future of attentive online learning.

Building on the foundation of eye tracking and facial recognition technologies as vital tools for assessing learner engagement, the Facial Action Coding System (FACS) [18] introduces an additional layer of depth to our understanding. The Facial Action Coding System (FACS) categorizes facial expressions by analyzing the movements of facial muscles, referred to as Action Units (AUs), to theoretically measure specific emotions. According to the study [19], an array of AUs can be mapped to various levels of engagement as seen in Table 2.2 below:

Table 2.2 - List of AUs involved in engagement detection in online learning. Reproduced from Dewan et al [19]

Engagement Levels	**Action Units**
Boredom	AU4, AU7, AU12
Confusion	AU1, AU4, AU7, AU12
Delight	AU4, AU7, AU12, AU25, AU26
Frustration	AU12
Neutral	AU4, AU7, AU12, AU25, AU26
Difficulties in viewing speed	AU1, AU2, AU4, AU5, AU9, AU10, AU12, AU14, AU15, AU17, AU20, AU45
Confusion	AU4, AU7
Frustration	AU1, AU2, AU4
Learning Gain	AU2, AU14
Confusion	AU45, AU1, AU4
Frustration	AU45, head pose feature
Learning Gain	AU2, AU4, AU5, AU12, AU15, AU23

A system for identifying facial expressions and body language linked to emotions during one-on-one learning tasks in educational contexts was presented by Saneiro et al. [20].An specialist in psychoeducational analysis annotated participant video material, taking into account attributes such as AUs, 3D poses of the head, location, type, intensity, and frequency of movements. The study integrated cognitive processes, facial expressions, and body movements analysis to understand emotional responses in educational scenarios.

Yet, the true potential of these metrics is realized when paired with advanced computational methods. In recent years, machine learning (ML), particularly image processing algorithms, has taken center stage in refining and enhancing attention tracking. By training on prepared datasets, Machine learning models can detect patterns and accurately identify signs of drowsiness, identify head pose detect emotions [21]. The integration of ML enhances the precision of focus detection by adapting to individual learning behaviors. By analyzing vast amounts of data, ML models can identify unique patterns of engagement. In [22], authors concluded that using SVM behaved best and showed an accuracy of 76.4% to classify student engagement. Sample images are displayed in Figure 2.2. However, like any evolving technology, the use of ML in attention tracking has had its successes and challenges, which have been explored in the study [23].

Figure 2.2 - Sample photos from the dataset (first line: “Engaged”; second line: “Not Engaged”)

2.3 Machine Learning in Engagement Detection

This study [24] used the Kinect One sensor’s 2D and 3D data to analyze students’ engagement through facial orientation, gaze direction, and body posture, creating a detailed behavioral feature set. Utilizing seven machine learning algorithms, it develops a model to predict students’ attention levels with a notable accuracy of up to 0.753, demonstrating the viability of a Kinect-based system for monitoring classroom engagement. This approach underscores the potential of machine learning for enhancing educational outcomes by accurately assessing and predicting student engagement in real-time settings.

The study [25] uses a combination of ML techniques to analyze student engagement in e-learning environments. It makes use of the Haar-cascade method for eye tracking, Python and Keras for face expression analysis, and a Convolutional Neural Network (CNN) for data processing. While the input layer of the CNN architecture uses 64x64 pixels to collect the raw pixel values of the image, the convolutional layers of the CNN design use 3x3 filters to determine the output of neurons related to local picture regions. During testing with fifteen students, the system demonstrated an average accuracy of 43.98% and a maximum accuracy of 50.77% using a support vector machine (MKL-SVM) model. By combining information on head and eye movements with facial expressions, the technique generates a concentration index with three engagement classes: "very engaged," "nominally engaged," and "not engaged at all."

The study [26] utilized data from camera sensors and eye trackers to automate learner engagement detection. Five types of data were captured throughout the nine pilot sessions in a high school setting: facial motion capture, eye-tracking data, body posture, facial expression, and RGB and Depth stream. Using a recording device equipped with perceptual computing capabilities, these data sets were recorded in real time. The system recorded the user’s interactions with educational content on the screen, enabling the correlation of perceptual data with the learning activities. Learner engagement levels were classified with accuracy ranging from 83% to 89% for known users and sessions using machine learning approaches like Random Forest, Decision Trees, and Naive Bayes.

The paper [27] investigates the effectiveness of Convolutional Neural Networks (CNNs) for classifying students’ engagement levels in e-learning environments. It

compares three popular CNN models (All-CNN, NiN-CNN, VD-CNN) with a proposed model that combines the best characteristics from the three. The proposed CNN model enhances efficiency by combining features from earlier models with innovations like using different block types and keeping the network shallow. It introduces 1x1 convolutions for added complexity and employs batch normalization in 3x3 layers to fine-tune decision-making while substituting fully connected layers with Global Average Pooling (GAP) for better accuracy. Using the Dataset for the Affective States in E-Environments, the proposed model obtains good accuracy in engagement classification (highly-engaged 95.69%, normally-engaged 89.55%, not-engaged 91.74%,).

2.3.1 Gaze Estimation for Focus Tracking

Eye tracking, or gaze tracking, is the practice of monitoring the movements of an eye in relation to the head or the place at which the glance is directed ("gaze point"). At its core, gaze tracking involves detecting and following the visual line of sight—where an individual is looking at any some given time. Eye tracking is an essential technique for comprehending human behavior and interactions because eye movements offer a rich and insightful window into a person's intents and ideas. Gaze tracking systems analyze eye movements to determine what people are thinking based on where they are looking. This process typically involves integrating eye and head position data to compute the location of the gaze in the visual scene. The human gaze is a strong indicator of attention, focus, and intent during social exchanges.

The majority of applications for eye tracking include iris recognition, clinical condition diagnosis, and sleepiness monitoring. These applications are expanded by gaze tracking techniques to include visual search, marketing and advertising, cognitive and behavioral therapy, eye typing for people with physical disabilities, neuroscience, psychology, and human-computer interaction (HCI). In HCI, eye tracking enhances user experience by allowing for more intuitive and natural interactions with technology. While more sophisticated systems, sometimes referred to as non-intrusive or remote systems, are able to track gaze without making physical contact with the user, simpler eye trackers just indicate the direction of the gaze in relation to the head. These systems offer significant advantages in terms of user comfort and ease of use, making them increasingly popular in both research and practical applications [28] [29].

Gaze tracking systems typically utilize cameras and infrared sensors to capture the movements of the eyes. These systems can be integrated into various devices such as desktop computers, mobile devices, or specialized eye-tracking glasses. The captured data is then processed using sophisticated algorithms to determine the gaze direction and the specific point of interest on a screen or in a physical space. Different uses for this data exist, such as performing psychological research, developing better user interfaces, and facilitating accessibility for those with physical limitations.

2.3.1.1 Gaze Estimation Approaches

The two primary types of human gaze estimating techniques are appearance-based and model-based techniques.Model-based methods estimate the 3D gaze direction vector using an explicit geometric model of the eye. These geometric, or 3D model-based, techniques usually work with metric data, which means that a geometric model that includes the locations and orientations of light sources, cameras, and displays, as well as camera calibration, are required. The majority of model-based approaches generally follow a three-step process: first, they reconstruct the eye's optical axis in three dimensions; second, they reconstruct the visual axis; and third, they determine the point of gaze by finding the intersection of the visual axis and the scene geometry[29]. Model-based strategies typically rely on specialized equipment like near-infrared (NIR) cameras to capture eye details and construct a geometric representation These techniques tend to be tailored to individuals and are often confined to controlled settings [30].

Unlike model-based approaches, appearance-based methods offer significant flexibility and adaptability. The integration of CNNs in appearance-based methods has been proven to offer robust capabilities for processing dynamic video input and achieving high accuracy in gaze estimation[5]. The review will continue with a listing of the different gaze estimation models. The Figure 2.3 shows the differences between mentioned approaches.

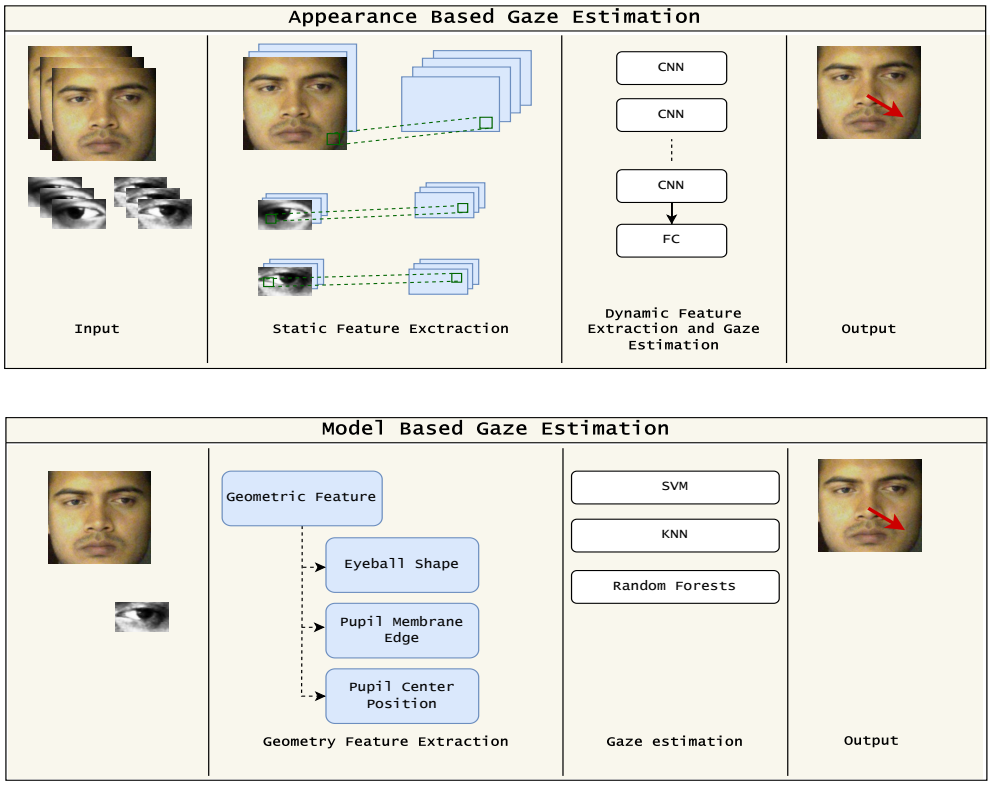


Figure 2.3 - Difference between Appearance Based and Model Based Gaze Estimation

2.3.1.2 Review of Gaze Estimation models

Xucong Zhang et al. [31] introduces an innovative approach in the realm of eye tracking technologies, especially for scenarios outside the controlled settings of a lab. They present the MPIIGaze dataset, which spans many months and includes 213,659 photos of 15 individuals using laptops in regular settings. This dataset is unique because it captures a wide range of real-world conditions, like different lighting and backgrounds, making it a valuable resource for developing more accurate gaze estimation technologies. To analyze this extensive dataset, the authors developed a method using multimodal Convolutional Neural Networks (CNNs).Their method greatly increased gaze estimation accuracy under challenging conditions, where traditional methods often fall short. They evaluated Random Forests (RF), known for clustering training samples by head angles; k-nearest Neighbours (kNN), effective in dense sample scenarios; Adaptive Linear Regression (ALR), initially designed for sparse, person-specific data; Support Vector Regression (SVR), which utilizes a combination of HOG and LBP features; and a shape-based approach known as EyeTab, which estimates gaze by fitting a model to detected iris edges. Their method outperformed others by a notable margin, reducing the average error to 13.9 degrees on their MPIIGaze dataset and even down to 10.5 degrees on another dataset called Eyediap, which focuses on screen-target gaze estimation.

Park, Spurr et. al [32] propose a method that is different from traditional gaze estimation techniques, which typically involve directly predicting gaze angles from eye images. Their methodology introduces a unique intermediary step by employing a novel concept called "gaze maps." Gazemaps are pictorial representations that show the orientation of the eyeball and the location of the iris in a two-dimensional (2D) image in an abstract manner. The first stage uses a deep neural network to transform raw eye images into gaze maps. The network architecture selected for this task is based on the Stacked Hourglass Network, which excels at capturing spatial relationships in images. In the second stage, another neural network, specifically a DenseNet architecture, takes these gaze maps as input and regresses them to final gaze direction outputs. The method demonstrated significant improvements in gaze estimation accuracy across various tests. It reduced the mean error to 4.5 degrees, an 18% improvement over the previous best of 5.5 degrees. On another evaluation, it achieved the lowest mean error of 3.8 degrees, outperforming traditional models. Additionally, in a setting known for its challenging image quality, the method achieved a mean error of 10.3 degrees, showcasing its robustness and effectiveness in diverse conditions.

Fischer, Tobias et al. [33] challenge eye gaze estimation in natural settings and introduce a novel approach containing a unique dataset collection method using eye-tracking glasses and a Kinect camera for precise gaze and head pose annotation. The gaze estimation framework utilizes Multi-Task Cascaded Convolutional Networks (MTCNN) for detecting facial features and extracting eye patches, which are then input into VGG-16 networks for feature extraction. This process involves several fully connected layers, including layers that integrate the head pose vector, resulting in the eye gaze's estimated yaw and pitch angles. In order to enhance robustness, the authors use an ensemble approach, with the mean of predictions from each individual network serving as the final prediction. To bolster the gaze

estimator's robustness further, training images undergo augmentation to adjust for off-centered eye patches, simulate camera blur, vary lighting conditions, and convert color images to grayscale. The training utilizes a combination of loss functions in order to reduce the discrepancy between the expected and actual gaze vectors, with specific learning parameters and weight initialization strategies to optimize network performance. Using the RT-GENE (Figure 2.4) dataset, the suggested technique produced an average gaze estimation error of 7.7 degrees, showcasing superior performance compared to existing techniques. The method achieved an accuracy gain of 18% on the MPIIGaze dataset compared to state-of-the-art techniques, reducing the mean error to 4.3 degrees from the previous best of 5.5 degrees. The UT Multi-view dataset was also used to assess the methods, and the results showed a mean inaccuracy of 5.1 degrees.

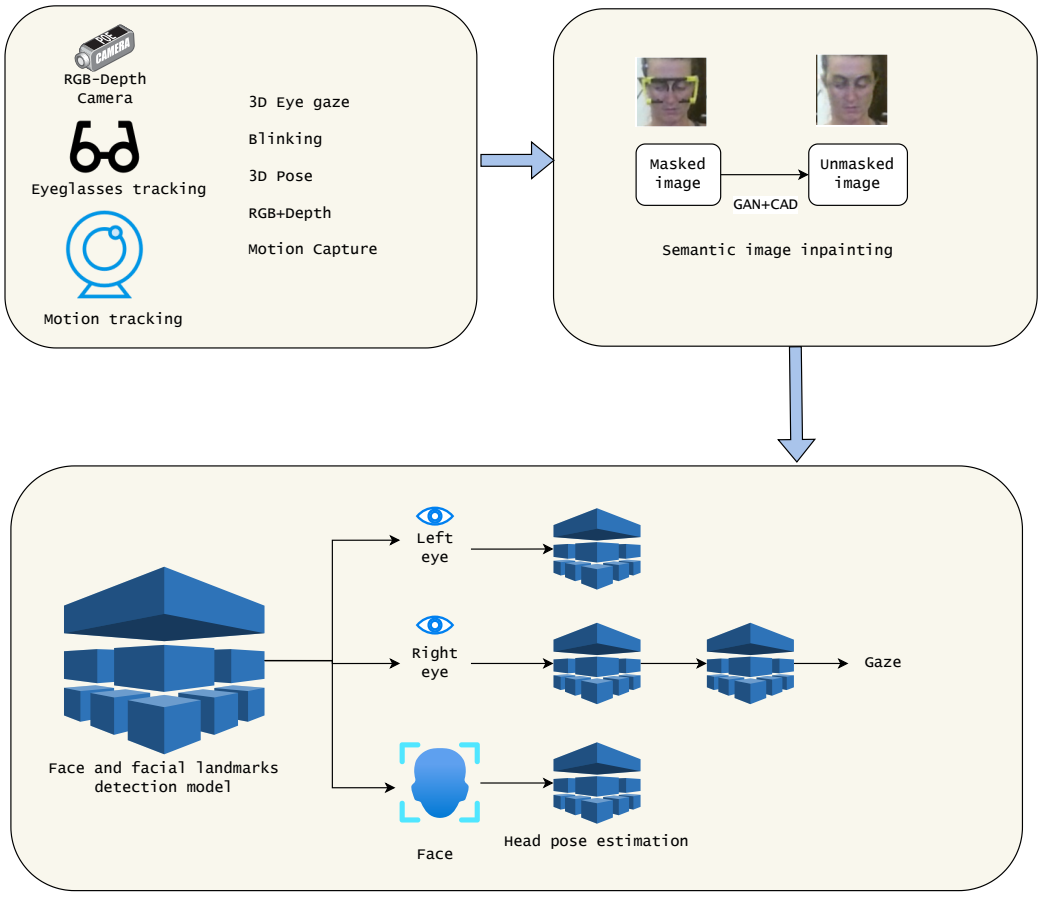


Figure 2.4 - RT-Gene architecture

An innovative approach to appearance-based gaze estimate is presented by Cheng, Lu, et al. [34], highlighting the asymmetry between the left and right eyes. They suggest using this asymmetry to greatly improve gaze estimation performance with their Asymmetric Regression-Evaluation Network (ARE-Net). Their approach consists of two components: the Evaluation Network (E-Net), which adaptively modifies the regression technique by assessing each eye's estimation

performance, and the Asymmetric Regression Network (AR-Net), which uses eye pictures to forecast 3D gaze directions. The ARE-Net operates on the principle that the two eyes, despite their physical consistency in gaze direction, often yield different estimation accuracies when the same regression method is applied. The researchers' discovery of "two-eye asymmetry" inspired them to create a technique that prioritizes the "high-quality eye" above the other eye in order to create a regression model that is more reliable and efficient. To estimate the 3D gaze directions of both eyes, the AR-Net jointly and asymmetrically processes their images. In contrast, the E-Net predicts which eye data should be prioritized in the regression, effectively guiding the AR-Net's learning process. Their experimental evaluation demonstrates significant improvements over existing methods across multiple public datasets. In particular, on the modified MPIIGaze dataset, the ARE-Net achieved a mean gaze estimation error of 5.0 degrees, indicating a state-of-the-art performance for gaze estimation based on eye images.This is an improvement over the Single Eye approach of more than 11% and the RF method of more than 30%. Additionally, the ARE-Net technique performed better in cross-dataset evaluations than the Single Eye method, with gains of 13.5% on the EyeDiap dataset and 25.4% on the MPIIGaze dataset. The authors also conducted a comprehensive assessment of the performance across individual subjects, further validating their approach's effectiveness and robustness. They observed that the ARE-Net and AR-Net consistently outperformed the Single Eye method across nearly all subjects, with the ARE-Net showing consistent improvements over the AR-Net as well.

Cheng et al. [35] present the Coarse-to-Fine Adaptive Network (CA-Net), a gaze estimation framework that employs a hierarchical approach to refine gaze direction predictions progressively. This method utilizes the correlation between facial and eye features, using coarse-grained facial information to approximate the gaze direction initially and then refining this approximation with fine-grained eye information. The process begins with the extraction of coarse-grained features from face images using a convolutional neural network (CNN). The network estimates a basic gaze direction, serving as an approximation that captures the gaze's broader context based on facial orientation and features. After the first approximation, the model switches to extract fine-grained information from eye pictures. These features are used to calculate a gaze residual, which refines the coarse gaze estimation to produce a more precise gaze direction. This refinement accounts for the subtle differences and variabilities in eye appearance that are critical for accurate gaze prediction. A key innovation in CA-Net is its attention mechanism, which weighs the characteristics retrieved from the left and right eyes in an adaptive manner. This component ensures the model prioritizes information from the eye that provides the most reliable cues for gaze direction, addressing the variability in image quality and relevance between the two eyes. The basic gaze direction and gaze residual are linked in the authors' conceptualization of the gaze estimation process, which is a bi-gram model. This model ensures that the refinement step is contextually informed by the initial estimation, allowing for a coherent and accurate final gaze prediction. The CA-Net demonstrates superior performance on standard benchmarks, specifically the MPIIGaze and EyeDiap datasets. In the

MPIIGaze dataset, CA-Net achieves an angular error of 4.1 degrees, outperforming existing methods and setting a new state-of-the-art performance metric. Similarly, on the EyeDiap dataset, CA-Net records a notable angular error of 5.3 degrees, further validating its effectiveness across different settings and conditions.

Petr Kellnhofer et al [36] introduce an innovative dataset and gaze estimation approach capable of handling the complexities of real-world settings. A large-scale gaze-tracking dataset called Gaze360, which includes 238 people in a variety of indoor and outdoor settings, forms the basis of their research, and a robust 3D gaze estimation model that incorporates temporal video information and a novel error estimation technique. The model learns to comprehend gaze continuity by processing sequences of seven frames and using a bidirectional Long Short-Term Memory (LSTM) network to predict the glance direction of the central frame. A significant innovation is the use of pinball regression loss, enabling the model to estimate gaze direction uncertainty, which is crucial for real-world applications where direct eye observation may not always be possible. On the Gaze360 dataset, the proposed model attained a mean gaze estimation error of 13.5 degrees, illustrating its effectiveness over existing methods and setting a new benchmark for in-the-wild gaze estimation. A practical application showcased the model's real-world utility, demonstrating a 51% accuracy in estimating customer attention in a supermarket setting from a standard camera viewpoint. This accuracy improved to 68% when using a smartphone camera positioned for a frontal view of the subjects, highlighting the model's potential in consumer behavior analysis and beyond.

Jindal et al [37] introduce Gaze Contrastive Learning (GazeCLR), a novel framework designed for the gaze estimation task using contrastive representation learning. This work addresses the need for gaze estimation models to be both invariant to appearance changes and equivariant to geometric transformations, a challenge not fully addressed by traditional contrastive learning approaches. The system uses specific data augmentation approaches that do not change gaze directions in order to learn invariance, and it makes use of multi-view data to encourage equivariance with regard to camera perspectives. This two-stage framework first pre-trains an encoder to learn gaze-relevant representations without requiring gaze annotations, using the EVE dataset, which contains video sequences from multiple calibrated and synchronized cameras. GazeCLR uses invariance to ensure gaze direction consistency across appearance changes and equivariance to maintain the relationship between gaze directions across different camera views. For equivariance learning, positive pairings are created from synchronous images taken from several camera angles that are aligned with the same gaze direction, with rotation matrices applied to align them to a common reference system. It demonstrated substantial improvements in gaze estimation accuracy, showcasing the effectiveness of the learned representations. When compared to baselines like SimCLR and BYOL, GazeCLR achieved lower mean angular errors, indicating superior performance in utilizing gaze-relevant features. The study also explored the framework's performance on different domains, such as the MPIIGaze and Columbia datasets, under a few-shot learning setting. GazeCLR outperformed other pre-training methods, indicating strong cross-domain generalization capabilities. Specifically, GazeCLR outperformed SimCLR by a margin of about 17.2% using a mere 20 calibration

samples on the Columbia dataset, highlighting its adaptability to new environments.

In their research into the possibilities of using the full-face image for gaze assessment, Zhang et al. [38] present a unique convolutional neural network (CNN) that uses spatial weights to either increase or suppress information in various facial regions. This approach significantly deviates from traditional methods that rely primarily on eye images, demonstrating that additional facial cues can substantially improve gaze estimation accuracy. The proposed model uses a full-face image as input, applying spatial weights across the feature maps generated by convolutional layers. Through this method, the network may adjust to changes in head attitude, gaze direction, and illumination by dynamically focusing on more informative regions of the face for gaze estimation. Spatial weights are learned through additional convolutional layers that adjust the emphasis on different facial regions based on their relevance to gaze estimation. This results in a weighted activation map that influences the subsequent fully connected layers and the final gaze estimation output. Input face images are pre-processed and resized to fit the network's requirements, ensuring that the spatial weights mechanism has a consistent format across different images. The network outputs either 2D screen coordinates for gaze location or a 3D gaze vector, depending on the task. The MPIIGaze and EYEDIAP gaze datasets are two well-known models used to assess the model's efficacy. Mean Angular Error (MAE) was used to evaluate the gaze estimation performance for both 2D and 3D gaze estimation tasks. The results demonstrate significant improvements over state-of-the-art methods, with the proposed method achieving MAEs of 4.8° on MPIIGaze and 6.0° on EYEDIAP for 3D gaze estimation. These improvements represent a 14.3% enhancement on MPIIGaze and a 27.7% enhancement on EYEDIAP compared to existing techniques. The Figure 2.5 compares the 2D and 3D gaze estimation techniques.

Chong et al. [39] propose a sophisticated model to estimate visual attention by combining inputs from the whole image, a cropped image of the subject's face, and the face's location. This multi-input model is designed to predict the subject's gaze direction, the saliency within the scene related to the subject's gaze, and the likelihood that the subject is fixating on a specific target within the scene. This approach addresses the problem of attention estimation in images, accommodating cases where the subject's gaze target is not only within the image but also when it extends beyond the frame or directly at the camera. The model ingests three primary inputs: the full scene image, a cropped image of the subject's face, and the (x, y) coordinates marking the face's location within the full scene. These inputs feed into two distinct convolutional pathways: one analyzing the full scene and another focusing on the subject's face. This two-path system, modeled after human perceptual strategies, enables detailed interpretation of gaze direction and recognition of important objects along the predicted gaze path. Both the face and scene pathways utilize the ResNet-50 architecture, chosen for its robust feature extraction capabilities. Additional convolutional layers refine the feature maps for gaze direction prediction and saliency map generation. The face pathway is specifically tasked with estimating the gaze angle, represented by yaw and pitch angles, while the scene pathway, in conjunction with face image features,

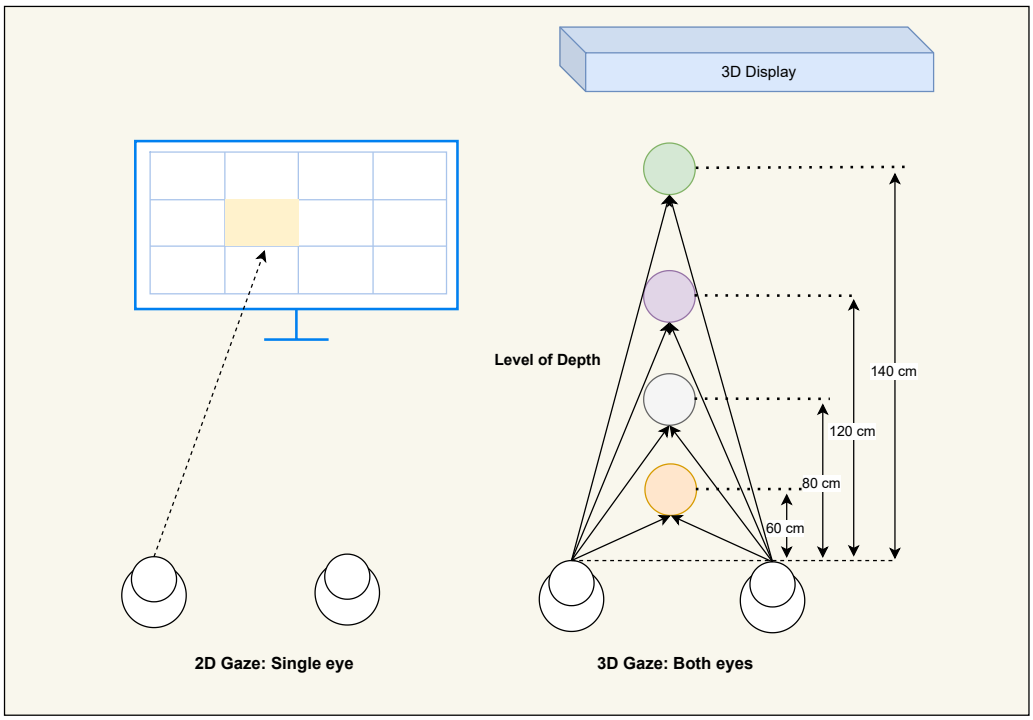


Figure 2.5 - Comparison between 2D and 3D Gaze estimations

aims to predict a person-centric saliency heatmap. A novel aspect of the model is its ability to estimate the likelihood of fixation on a gaze target within the scene. This output provides an important distinction between subjects fixating on points inside versus outside the frame, enhancing the model's applicability across various gaze scenarios. Given the absence of a single dataset encompassing all necessary gaze and scene scenarios, the authors adopt a cross-domain learning strategy. This approach uses three distinct datasets (GazeFollow, EYEDIAP, and SynHead), each contributing different aspects of gaze behavior for a comprehensive training regimen. The model selectively updates its components based on the available supervisory signal from each dataset, ensuring focused learning on specific tasks like gaze angle estimation, scene saliency, and fixation likelihood. The model employs an L1 loss for angle regression and a cross-entropy loss for saliency map and fixation likelihood tasks. An innovative "project and compare" loss encourages consistency between the estimated gaze direction and the actual gaze target, further refining the model's accuracy. The model achieved an Area Under Curve (AUC) score of 0.896, setting a new benchmark for the task and outperforming the previous state-of-the-art method, GazeFollow, which had an AUC of 0.878. It also demonstrated superior precision in gaze-saliency prediction with an L2 distance of 0.187 and a minimum distance metric of 0.112, indicating high accuracy in localizing gaze targets within images. Moreover, the model achieved an angular error of 6.4 degrees, closely competing with state-of-the-art methods in 3D gaze estimation which have errors around 6.0 degrees. This showcases the model's precision in estimating the gaze direction in terms of yaw and pitch angles.

For gaze target grid classification, the model significantly outperformed the GazeFollow baseline across different grid sizes. For instance, in a 2x2 grid, the model achieved a precision of 0.744 and a recall of 0.851, demonstrating its ability to accurately predict gaze targets and differentiate between in-frame and out-of-frame gaze directions.

Recasens et al. [40] introduce a novel approach to gaze tracking in video sequences, going beyond the confines of a single frame to predict where a person is looking, even if the gaze target is present in a different frame. This work marks a significant advancement in understanding human attention and interactions within video content by leveraging both semantic and geometric insights. The model ingests three types of input: the full video frame, a cropped image of the subject's face, and the spatial coordinates of the face within the frame. The model features two convolutional pathways based on the ResNet-50 architecture. One pathway processes the whole scene for saliency prediction, while the other focuses on the face crop for gaze direction estimation. The scene pathway predicts salient locations in the target frame that could attract the subject's gaze. Simultaneously, the gaze pathway estimates the gaze direction in terms of yaw and pitch angles from the face crop. A transformation pathway estimates the affine transformation between the frame containing the subject and potential target frames. This enables the model to map the estimated gaze direction across different frames. Alongside predicting gaze direction and saliency, the model computes the likelihood that a given frame contains the gaze target, facilitating the selection of the most probable target frame. To train and evaluate their model, the authors introduced the VideoGaze dataset, comprising approximately 50,000 annotations across diverse video scenes, annotated with the gaze direction and target locations. The model was tested against various baselines and showed superior performance in predicting gaze locations across frames. It achieved high scores in metrics such as AUC (Area Under Curve), KL divergence, and L2 distance, demonstrating its effectiveness over traditional saliency and static gaze models. For the task of selecting the correct frame containing the gaze target, the model exhibited commendable performance, outperforming baseline methods in accurately identifying frames with the gaze target.

Mahanama et al [41] explore gaze estimation through the novel application of capsule networks, focusing on the ocular region for decoding and representing gaze information. Their method diverges from traditional convolutional neural networks by emphasizing the orientation and pose information inherent to the eyes, which are crucial for determining gaze direction. Gaze-Net employs a capsule network, specifically designed to understand the spatial hierarchies between features in an image. Unlike standard CNNs, which are adept at identifying features but struggle with spatial relationships and orientations due to pooling layers, capsule networks preserve these important spatial hierarchies through dynamic routing mechanisms. This is particularly beneficial for gaze estimation, where the orientation of the eyeball relative to the ocular region provides crucial information. Initially, the network is trained to classify the gaze into one of six directional categories using image patches of the ocular region. This is achieved by passing these patches through a convolutional layer and then into a primary capsule layer, where

dynamic routing helps to understand the orientation and pose information. Subsequently, the network is fine-tuned for the gaze estimation task, focusing on the precise angles of gaze direction in terms of yaw and pitch. The training process incorporates different regularization parameters to balance between reconstruction loss, which encourages the network to accurately reconstruct input images from the capsule representations, and gaze loss, which directly impacts the accuracy of the gaze estimation. This dual focus allows the model to learn both detailed feature representations and specific gaze orientations. The model was evaluated using two publicly available datasets: MPIIGaze, containing over 200,000 "in the wild" images, and Columbia Gaze, with over 5,000 images featuring users at 21 gaze directions observed at five camera angles. The model's performance was measured in terms of classification accuracy (for the initial directional categorization) and Mean Absolute Error (MAE) for the gaze estimation task. On the MPIIGaze dataset, Gaze-Net achieved a Mean Absolute Error (MAE) of 2.84° for combined angle error estimates within the dataset. This result signifies the model's capability to accurately estimate gaze direction from a single ocular image, underlining the effectiveness of capsule networks in preserving and utilizing spatial relationships and orientations. The model was further tested on the Columbia Gaze dataset to assess its transfer learning capabilities. Initially, the model recorded an MAE of 10.04°. After applying transfer learning techniques and retraining the model specifically for the Columbia Gaze dataset, the error was significantly reduced to 5.9°.

Liu, Yu et al [42] introduce a method, distinct from conventional gaze estimation approaches, where they predict the gaze direction by assessing differences between pairs of eye images from the same subject. The fundamental hypothesis underpinning this approach is that by focusing on the differences within a subject's eye images, one can mitigate common errors associated with singular image evaluations, such as alignment issues, eyelid occlusions, and illumination variances, thereby enhancing the overall accuracy of gaze prediction. The core of the proposed method is a differential convolutional neural network (CNN) designed to compute gaze direction differences between two eye images. This architecture comprises two parallel convolutional pathways, each processing an input eye image. The pathways then merge, and through a series of fully connected layers, the network outputs the gaze difference between the input pair. The network is trained on pairs of eye images from the same individual, utilizing a loss function that specifically targets the gaze direction differences. Post-training, the network can adapt to specific subjects by fine-tuning with subject-specific calibration pairs, thereby ensuring that the gaze predictions align more closely with individual gaze behaviors. The model's performance was assessed using three public gaze datasets: MPIIGaze, EYEDIAP, and UT-Multiview. On the MPIIGaze Dataset, the differential approach achieved an MAE of 4.5 degrees, showcasing a substantial improvement over conventional single-image estimation methods, which typically present errors around 5 to 6 degrees. When tested on the EYEDIAP Dataset, the model demonstrated even more pronounced accuracy, with an MAE of 3.8 degrees in gaze estimation tasks across various head movements and illumination scenarios. The model's adaptability and precision were further underscored on the UT-Multiview

Dataset, where it recorded an MAE of 2.1 degrees, significantly outperforming existing models by leveraging the differential analysis of eye image pairs.

Krafka et al. [43] tackle the problem of making eye tracking technology widely available by using the widespread presence of smartphones and tablets.. The study focuses on creating and using GazeCapture, the first large-scale dataset for eye tracking, carefully gathered from more than 1450 participants through thousands of sessions, resulting in almost 2.5 million images. Central to their methodology is the iTracker model, a sophisticated convolutional neural network (CNN) specifically architected for eye tracking. The model's design allows it to process input images from simple camera feeds, extracting critical features necessary for precise gaze estimation. The model architecture combines inputs from both eyes and the face, along with the face grid, to estimate gaze. Specifically, it consists of separate convolutional neural network (CNN) pathways for the left eye, right eye, and face, each extracting relevant features from their respective inputs. An additional input, the face grid, indicates the location and scale of the face within the image frame, providing spatial context to the model. These pathways utilize multiple layers of convolution and max pooling to distill critical features from the input images. The extracted features from each pathway are then concatenated into a unified feature vector, which is processed through fully connected layers to produce the final gaze estimation. Upon evaluation, iTracker demonstrates a notable leap in performance, achieving a mean prediction error of 1.71 cm on mobile phones and 2.53 cm on tablets in a non-calibrated state. Calibration processes further refine accuracy, underscoring the model's potential for real-time application in consumer-grade technology. The Figure 2.6 demonstrates the overall iTracker architecture.

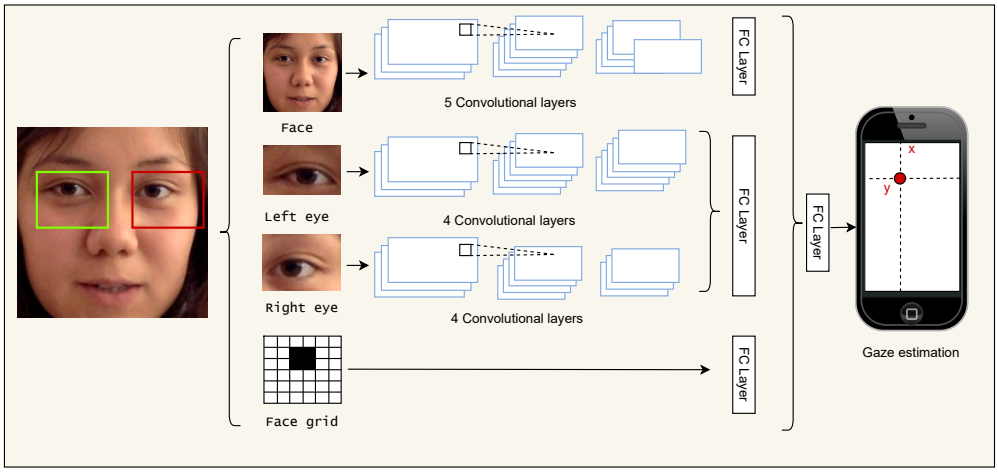


Figure 2.6 - iTracker architecture

Zhaokang Chen et al. [44] introduces an innovative approach to gaze estimation that simplifies the calibration process significantly. Their method, based on gaze decomposition, separates the gaze angle into a subject-independent component extracted from the image and a subject-dependent bias. This approach enables a more efficient calibration, requiring less data and simplifying the setup, poten-

tially to just a single gaze target and head position, thereby enhancing practical applicability. A central part of their research involved the use of the NISLGaze dataset to thoroughly examine the effects of various factors—like the number of gaze targets, the volume of images per target, and head positions during calibration—on the accuracy of gaze estimation. Their findings underscore that even minimalistic calibration setups can yield highly accurate estimations, challenging and streamlining traditional calibration requirements.

The gaze estimation model introduced, GEDDNet, incorporates dilated convolutions and significantly outperforms existing models, achieving an improvement in accuracy by over 6.3%. GEDDNet's performance is notable both with and without calibration, presenting an intriguing finding that deviates from expected norms; under certain conditions, adding a bias during training enhances generalization, contrary to the usual expectation that matched training and testing conditions yield the best results. Their extensive experimental validation shows that a basic calibration using a single gaze target and head position can achieve an angular error reduction of approximately 4.1 degrees, marking a significant advancement in gaze estimation accuracy.

Seonwook Park et al. [45] introduce FAZE, a novel framework designed to personalize gaze estimation models using only a few calibration samples. The FAZE framework addresses the limitations of person-independent gaze estimation networks, which often struggle with accuracy due to inter-personal anatomical differences. The authors propose a solution that combines a disentangling encoder-decoder architecture with meta-learning to create a highly adaptable gaze estimation model.FAZE works by learning a rotation-aware latent representation of gaze through a disentangling encoder-decoder. This architecture separates the factors of appearance, gaze, and head pose in the latent space, which is crucial for accurate gaze estimation. The latent embeddings are then used to train a gaze estimation network using meta-learning, specifically the Model-Agnostic Meta-Learning (MAML) algorithm. This approach allows the model to adapt quickly to new individuals with as few as three calibration samples. The paper presents significant performance improvements, achieving state-of-the-art results with a mean angular error of 3.18 degrees on the GazeCapture dataset, which is a 19% improvement over previous methods. The results demonstrate that FAZE can effectively personalize gaze estimation models with minimal calibration data, making it a practical solution for high-accuracy applications in various fields such as human-computer interaction, virtual reality, and automotive systems.

The paper by Yiran Guan et al. presents a novel method for video gaze estimation called Multi-Clue Gaze (MCGaze). This method leverages the spatial-temporal interaction context among the head, face, and eye in an end-to-end learning framework[46], aiming to enhance the accuracy and efficiency of gaze estimation. The primary innovation of MCGaze is its ability to jointly solve the tasks of clue localization and gaze estimation in a single step, optimizing performance through joint optimization.The MCGaze framework processes video clips by first extracting features to form a video feature tensor. It employs learnable queries for the head, face, and eye to localize and characterize gaze clues. This approach enables robust spatial-temporal feature interaction, ensuring that the model cap-

tures both global and local gaze cues. The method integrates these features using multi-head self-attention and dynamic convolution to update query features iteratively. The model demonstrates superior performance on the Gaze360 dataset, achieving an angular error of 10.02 degrees on the detectable face subset, outperforming several state-of-the-art methods. The model runs efficiently, processing video clips at 70 FPS on an RTX 3090, highlighting its practical applicability in real-world scenarios. The study also includes an ablation analysis, showcasing the effectiveness of incorporating head, face, and eye clues, as well as spatial and temporal interactions.

The Table 2.3 summarizes the existing different Gaze Estimation Models.

Table 2.3 - Summary of Gaze Estimation Models

Study	**Backbone network**	**Dataset**	**Mean angular error in degrees**
[31]	LeNet	MPIIGaze, Eyediap	13.9 and 10.5 degrees
[32]	DenseNet	MPIIGaze, Columbia	4.5 and 3.8 degrees
[33]	MTCNN, VGG-16	RT-GENE, MPIIGaze, UT Multiview	7.7, 4.3 and 5.1 degrees
[34] ARE-Net	AlexNet	MPIIGaze, EyeDiap,	8.8 and 13.5
[35] CA-Net	CNN	MPIIGaze, EyeDiap	4.1 and 5.3
[36] Gaze360	LSTM	Gaze360	13.5
[39]	ResNet-50	EyeDiap, GazeFollow, SynHead	6.4
[41] Gaze-Net	Self developed	MPIIGaze, ColumbiaGaze	2.84 and 10.04
[38]	AlexNet	Own dataset	4.8
[43] iTracker	Caffe	GazeCapture	2.58
[5] L2CS-Net	ResNet-50	MPIIGaze	3.92

2.3.1.3 Publicly Available Gaze Estimation Datasets

Columbia Gaze (2014) The **Columbia Gaze** dataset features 5,880 images from 56 subjects. It includes high-resolution images captured under consistent lighting conditions, with multiple gaze directions and head poses. This dataset's diversity in participant demographics makes it useful for generalizing across different populations.

EyeDiap (2014) The **EyeDiap** dataset includes 94,000 images from 16 subjects. It offers infrared and RGB data with various head poses and interactive settings. This dataset's combination of different data types and environments is valuable for testing gaze estimation models under varied conditions.

MPIIGaze (2015) The **MPIIGaze** dataset includes images of 15 subjects captured over several months, resulting in approximately 213,659 images. It features natural, everyday settings with varying lighting conditions and head poses. This dataset is valuable for training gaze estimation models that can generalize well across different real-world conditions.

GazeCapture (2016) The **GazeCapture** dataset is a large-scale collection with over 2.5 million images from 2,445 subjects. It includes data from various devices and real-world environments, making it essential for developing models that generalize well across different user groups and device types.

GazeFollow (2016) The **GazeFollow** dataset contains over 122,143 images from 130,000 annotated frames. It captures natural gaze behavior in real-world scenes, providing spontaneous gaze directions and diverse environmental contexts. This dataset is crucial for applications requiring interpretation of gaze in unstructured settings.

UTMultiview (2017) The **UTMultiview** dataset comprises 50,000 images from 50 subjects. It features multi-viewpoint images with consistent lighting and various head poses. This dataset helps develop gaze estimation algorithms that account for different perspectives and head orientations.

RT-Gene (2018) The **RT-Gene** dataset contains images of 15 subjects with 122,526 samples. It features multi-view images captured using multiple cameras, which are essential for training models to estimate gaze accurately from various angles. The dataset also includes annotations for head pose and facial landmarks.

TabletGaze (2018) The **TabletGaze** dataset contains 51,000 images from 51 subjects. It features gaze data collected using tablets under various lighting conditions and different gaze targets. This dataset is suitable for developing gaze estimation models for mobile user interfaces and accessibility technologies.

Gaze360 (2019) The **Gaze360** dataset comprises data from 238 subjects, amounting to over 172,000 images. It includes 360-degree gaze directions, which provide a comprehensive range of head poses and eye orientations. This dataset is crucial for developing models that need to perform well in diverse and dynamic environments.

NVGaze (2019) The **NVGaze** dataset includes data from 50 subjects, totaling 10,000 images. It features high-resolution eye images captured under controlled lighting conditions with various gaze directions and head poses. This dataset is particularly useful for fine-tuning gaze estimation models for high accuracy.

ETH-XGaze (2020) The **ETH-XGaze** dataset consists of 1,100,000 images from 80 subjects. It offers high-resolution images with diverse gaze directions and head poses. The controlled lighting conditions and varied appearances make it a valuable resource for developing robust gaze estimation models.

GOOReal (2020) The **GOOReal** dataset includes egocentric video data from 100 subjects, with over 200,000 annotated frames. It captures object-oriented gaze in real-world interactions, making it ideal for studying gaze behavior in naturalistic tasks.

Below listed are the publicly available Gaze Estimation Datasets (Table 2.4) and corresponding best performing models (Table 2.5):

Table 2.4 - Summary of Public Gaze Estimation Datasets

Name	Year	People	Samples
MPIIGaze [31]	2015	15	213,659
Gaze360 [36]	2019	238	172,000
RT-Gene [33]	2018	15	123,000
ETH-XGaze[47]	2020	110	1,100,000
NVGaze [48]	2020	30	4,500,000
Columbia Gaze [49]	2013	56	5880
GazeFollow [40]	2015	130	122,143
GOOReal [50]	2021	100	9552
UTMultiview [51]	2014	50	1,100,000
EyeDiap [52]	2014	16	94 recordings
GazeCapture [53]	2016	1474	2,400,000
TabletGaze [54]	2017	51	816 recordings

Table 2.5 - Summary of Best Models for Gaze Estimation on Different Datasets

Dataset	Best Model
MPII Gaze	FAZE
Gaze360	MCGaze
EYEDIAP (screen target)	RecurrentGaze (Static)
EYEDIAP (floating target)	RecurrentGaze (Temporal)
GazeCapture	Ensemble Calibration
RT-GENE	RT-GENE 4 model ensemble
UT Multi-view	RT-GENE 4 model ensemble
MPSGaze	ResNet18
ETH-XGaze	ETHXGaze

Despite the remarkable advancements in attention tracking, there remain pressing challenges. One of the most significant is the data privacy and ethical concerns. The collection and analysis of facial and eye movement data raise questions about how this information is stored, shared, and utilized, potentially leading to data misuse [55]. Additionally, the very act of continuous monitoring might inadvertently increase stress levels among learners. This could produce a counterproductive effect, where the awareness of being observed could lead to reduced genuine engagement rather than enhancing it [56].

2.3.2 Performance evaluation metrics of gaze estimation models

To assess the effectiveness of 2D and 3D gaze estimation techniques, various performance evaluation metrics are used, each configured to specific aspects of gaze estimation tasks.

2.3.3 Area Under the Curve (AUC)

Definition: Area Under the Curve (AUC) refers to the area under the Receiver Operating Characteristic (ROC) curve [40]. The ROC curve is a graphical representation of a classifier's performance across different threshold settings, plotting the true positive rate (TPR) against the false positive rate (FPR).

Significance:

- **AUC Value**: Ranges from 0 to 1, where a higher AUC indicates better model performance.
  - **AUC = 0.5**: The model performs no better than random guessing.
  - **AUC = 1.0**: The model perfectly distinguishes between classes.
- **Use Cases**: Particularly useful for binary classification problems and provides insight into the trade-offs between sensitivity (TPR) and specificity (1 - FPR).

Calculation:

- Compute TPR and FPR at various threshold levels.
- Plot these values to form the ROC curve.
- Calculate the area under this curve using numerical integration.

Formula:

$$\text{AUC} = \int_0^1 \text{TPR}(FPR)\, d(\text{FPR}) \tag{2.1}$$

2.3.4 Classification Accuracy

Definition: Classification accuracy is the ratio of the number of correct predictions to the total number of predictions[41]. It measures how often the model correctly predicts the class labels.

Significance:

- **Intuitive Metric**: Easy to understand and widely used.
- **Balanced Data**: Works best with balanced datasets where each class is equally represented.

Calculation:

- Count the number of correct predictions.
- Divide by the total number of predictions.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2.2}$$

2.3.5 L2 Distance (Euclidean Distance)

Definition: The straight-line distance in Euclidean space between two places is measured by the L2 distance, which is also referred to as the Euclidean distance. In the context of gaze estimation, it can be used for calculating the distance between estimated and actual gaze points.[50].

Significance:

- **Geometric Interpretation**: Provides a direct measure of the spatial error between predicted and true values.
- **Sensitivity**: More sensitive to large errors due to the squared term.

Calculation:

- Compute the difference between the predicted and actual values for each dimension.
- Square these differences, sum them, and take the square root.

Formula: For two points $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$:

$$L2(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2.3}$$

2.3.6 Mean Squared Error (MSE)

Definition: A popular indicator of prediction error is mean squared error (MSE), which is determined by averaging the squared differences between values that were expected and those that were observed.[36].

Significance:

- **Penalizes Larger Errors**: Squaring the differences places greater weight on larger errors, making MSE sensitive to outliers.
- **Continuous Predictions**: Useful for regression tasks where the goal is to predict continuous values.

Calculation:

- Compute the difference between each predicted and actual value.
- Square these differences.
- Calculate the average of these squared differences.

Formula: For predicted values $\hat{y}_i$ and true values y_i, where n is the number of samples:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{2.4}$$

Table 2.6 - Summary of Gaze Estimation Performance Metrics

Metric	**Summary**
Area Under the Curve (AUC)	Plots the true positive rate (TPR) against the false positive rate (FPR) and calculates the area under the ROC curve. An improved performance is indicated by a higher AUC, which ranges from 0.5 (random guessing) to 1.0 (perfect differentiation). beneficial for classifying binary data.
Classification Accuracy	Ratio of correct predictions to total predictions. Widely used and intuitive, best for balanced datasets.
L2 Distance (Euclidean Distance)	Measures straight-line distance between predicted and actual gaze points. Sensitive to large errors due to the squared term. Provides a direct measure of spatial error.
Mean Squared Error (MSE)	The mean of the squared deviations between the actual and projected values. Bigger errors are penalized, increasing its sensitivity to outliers. Useful for regression tasks

2.3.7 Gaze Tracking Applications

George E. Raptis et al. [57] explores the use of eye-tracking data and visual activities to infer human cognitive styles, proposing a multifactorial model incorporating human cognition, visual behavior, and activity factors. The study addresses the limitations of traditional explicit cognitive style elicitation methods, such as their time-consuming nature and impracticality for real-time integration in interactive systems. By leveraging eye-tracking mechanisms, the authors aim to develop an implicit elicitation framework that can dynamically adapt to users' cognitive needs and preferences, thereby enhancing task performance and user experience. Two feasibility tests were conducted as part of the study, in which participants engaged in various visual tasks. Among people with varying cognitive styles, the eye-tracking research identified quantitative variations in visual behavior. By utilizing these distinctions in categorization ex[eriments, it was possible to achieve gaze-based implicit elicitation of cognitive styles in real time. This approach has the potential to enable interactive systems to adapt to users' cognitive styles, improving assistance and overall user experience.

The paper [58] presents a novel approach to identifying Attention Deficit Hyperactivity Disorder (ADHD) using eye movement data. This study addresses the increasing prevalence of ADHD and its significant impact on individuals and society. The authors propose a rule-based system that utilizes decision tree classifiers to analyze eye movements, including fixations, saccades, gaze positions, and pupil diameters, to identify individuals with ADHD. The paper discusses the experimental setup, which involved recording eye movements of 14 participants using a Tobii Pro X2-60 eye tracker. Among these participants, seven were diagnosed with ADHD. The data collected were processed to extract relevant features, such as the number and duration of fixations and saccades. The authors then applied various decision tree algorithms, including J48, Random Forest, and Hoeffding Tree, as well as classification ruling algorithms like PART and JRip, to classify the data. The results demonstrated high accuracy, with decision tree algorithms achieving up to 85.31% accuracy and classification rules achieving 82The authors highlight the effectiveness of using eye movement data for ADHD identification, noting that this approach is less complex and provides a reduced dimensionality of data compared to other methods like fMRI or EEG. The study also emphasizes the potential of combining eye movement data with other clinical data to improve the accuracy of ADHD diagnosis further. The rule-based system developed in this study offers a promising tool for early detection of ADHD, which is crucial for timely intervention and management of the disorder.

The paper by Haofei Wang et al. [59] presents a system that estimates three-dimensional (3D) gaze locations using a mobile eye tracker integrated with a Simultaneous Localization and Mapping (SLAM) algorithm. This system aims to provide accurate 3D gaze estimates in real-world coordinates, accommodating extensive head movements, and dynamic environmental changes, which is particularly useful for applications in human-robot interaction (HRI). Traditional eye-tracking systems typically estimate gaze in two dimensions (2D) either in head-centric or world-centric coordinates. However, these systems often struggle with accuracy and limited user movement tolerance. The proposed system by Wang

and colleagues addresses these limitations by combining head-centric gaze estimates from a head-mounted eye tracker with head pose estimates obtained from a visual-inertial SLAM algorithm. This combination allows for more accurate and extensive 3D gaze estimation over a larger operational range compared to conventional methods. The system architecture integrates three sensor systems: an inertial measurement unit (IMU) and an RGB camera for SLAM, a head-mounted eye tracker, and an RGB-D sensor for environmental mapping. After analyzing these inputs, the system estimates the user's gaze vector in world-centric coordinates dynamically.By aligning gaze targets with actual object surfaces, initial gaze estimations are improved by using a point cloud representation of the environment. The accuracy and resilience of the system were shown by the experimental findings. The system achieved a mean angular error of 2.9 degrees with a standard deviation of 1.0 degrees over a testing range of 13 square meters. Comparatively, this performance is stable across different positions and less sensitive to head movements than traditional remote eye trackers. Moreover, the system's ability to provide accurate 3D gaze estimates in a natural office environment further underscores its applicability in real-world settings.

Rencheng Zheng et al.'s study [60] examines at how drivers' eye-gaze patterns are affected by different portable navigation system display sizes and locations. The study measures glance frequency, glance time, and overall look time as percentages using a non-intrusive eye-gaze tracking technology. The research involved 20 participants using a driving simulator in various conditions. The findings indicate that while smaller screens result in longer look durations and more visual distraction, convenient display positions with modest visual angles led to shorter glance times but higher gaze frequency. The study also highlights that larger displays at specific positions improve driving safety by reducing the time-to-collision (TTC) index. Subjective evaluations from participants suggest that smaller displays are less acceptable and cause more fatigue compared to larger displays. The findings underscore the importance of display size and position in mitigating visual distraction and enhancing driving safety, suggesting that portable navigation systems should adopt larger displays in optimal positions.

The research [61] introduces a novel system for enabling photo-realistic facial reenactment within virtual reality (VR) environments. This system combines eye tracking and real-time facial performance recording using an infrared (IR) camera integrated into a head-mounted display (HMD) and a common RGB-D sensor. The approach allows for the modification and re-rendering of facial expressions and eye movements in real-time, facilitating applications such as VR teleconferencing where participants' faces are occluded by the HMD. For eye tracking, the system uses a data-driven strategy based on random ferns, capturing high-resolution images of the eye to ensure accurate tracking of eye movements. The combination of facial performance capture with photo-realistic re-rendering enables the synthesis of high-quality facial animations, including the critical eye and mouth regions. The primary application discussed is VR teleconferencing, where the system virtually removes the HMD and renders a photo-realistic version of the user's face, including natural eye movements, thereby enhancing the sense of presence and realism in virtual meetings. The authors' evaluations demonstrate the system's effective-

ness in accurately capturing and reenacting facial expressions and eye movements, offering significant improvements in accuracy and realism over existing methods.

The paper by Velichkovsky et al. [62] addresses the challenge of unintentional activations in eye-movement-based user interfaces, known as the Midas touch problem. Traditional methods, like using dwell times or voluntary blinks, often cause user discomfort and increased visual fatigue. The authors propose a novel approach by differentiating between ambient and focal fixations to identify intentional visual commands. Ambient fixations are short and used for spatial orientation, while focal fixations are longer and associated with conscious visual processing.The study conducted two experiments. The first experiment recorded fixation durations and saccadic amplitudes of ten participants during visual search tasks, showing that focal fixations could be reliably identified. The second experiment, involving twenty participants, compared the dwell duration approach and focal fixations for the purpose of visual command identification accuracy. The results showed similar accuracy rates for both methods, but the focal fixation method resulted in lower visual fatigue.This approach effectively reduces false activations without increasing user discomfort, enhancing the usability of gaze-control interfaces. By leveraging natural fixation characteristics, this method improves the interpretation of user intentions, paving the way for more efficient and user-friendly eye-movement-based interfaces.

The paper [63] explores the integration of eye-tracking technology with virtual reality (VR) in educational settings, tracing its historical development and detailing its technological advancements. It discusses how eye-tracking, initially used in medical and psychological research, has evolved with the incorporation of VR to enhance educational experiences. The review outlines various applications of this technology in education, particularly in medical training, where it enhances the learning process by providing detailed insights into students' focus and engagement. This enables a more personalized and interactive learning experience, potentially improving skills in complex fields such as diagnostics and surgery. The paper also addresses the benefits of using eye-tracking and VR in education, such as increasing student engagement through immersive learning environments and offering personalized learning trajectories. It also discusses significant challenges, including the high costs associated with VR and eye-tracking technologies, the technical complexity of system integration, and user-related issues like motion sickness. Furthermore, the review speculates on the future directions of eye-tracking in educational VR, emphasizing the need for further research to optimize these technologies for broader educational use and to overcome current limitations.

The study [64] provides an extensive review of how eye movements and gaze behavior can predict human intentions, particularly in Human-Robot Interaction (HRI).The review discusses about how gaze-based models can be integrated into human-interactive systems, which can help in understanding and supporting human behavior in various environments, including assistive technologies and teleoperation. The review also articulates the role of gaze as a predictor of human intentions, outlining the psychological and cognitive underpinnings that correlate eye movements with forthcoming actions. It emphasizes the anticipatory nature of eye movements in tasks, demonstrating how these can be early indicators of in-

tention, thus allowing artificial systems to react or assist humans more effectively. The paper spans several applications, from teleoperated robots to assistive devices, illustrating the broad utility of gaze-based intention estimation in improving interaction dynamics between humans and machines.

The research [65] discusses in detail the accuracy, technologies, application statistics, data volume, and future projections associated with driver gaze estimation systems. It mentions that some systems have achieved a mean angular error (MAE) as low as 0.5 to 1.0 degrees under controlled conditions, highlighting their precision in tracking driver gaze direction. The review covers various sensing technologies used, including infrared and RGB cameras, with infrared cameras noted for their effectiveness in low light conditions—critical for night driving. Additionally, the paper points out that understanding driver gaze can significantly reduce accident rates by enhancing the responsiveness of driver assistance systems. Systems that are sensitive to gaze can provide alerts 0.5 to 1.5 seconds earlier than traditional systems, offering drivers more reaction time to avoid accidents. The review also emphasizes the extensive data involved in these studies, typically comprising hundreds of hours of driving data and thousands of gaze instances across various scenarios, aiding in refining the models for improved accuracy and robustness.

See Table 2.7 for summary of Gaze Estimation Applications.

Table 2.7 - Summary of Gaze Estimation Applications

Study	**Application**	**Methodology**	**Key Findings**
George E. Raptis et al. [57]	Cognitive styles	Multifactorial model	Implicit elicitation feasible
Haofei Wang et al. [59]	3D gaze in HRI	3D gaze with SLAM	MAE: 2.9°, robust in dynamic settings
Rencheng Zheng et al. [60]	Driver distraction	Eye-gaze behavior analysis	Display size/position impact on distraction
Krafka et al. [43]	Mobile eye tracking	iTracker model (CNN)	Error: 1.71 cm (phones), 2.53 cm (tablets)
Zhaokang Chen et al. [44]	Simplified calibration	GEDDNet with decomposition	Calibration with single target effective
De Silva et al. [58]	ADHD identification	Rule-based system	High accuracy (up to 85.31%)
Thies et al. [61]	VR facial reenactment	Real-time facial capture	Enhanced VR teleconferencing realism
Maria Mikhailenko et al. [63]	Eye-tracking in VR education	Review	Enhanced engagement, personalized learning
Anna Belardinelli [64]	Predicting human intentions	Differential CNN	Improved gaze estimation accuracy

Chapter 3

Methodology

3.1 Overview of the Research Design

This research aims to explore real-time focus tracking in video streams, utilizing the L2CS-Net convolutional neural network model. The project underscores the importance of gaze estimation technologies for practical applications, such as enhancing engagement in e-learning and virtual meetings. The research will train the L2CS-Net model on an extensive dataset and fine-tune some parameters to enhance its accuracy and efficiency for real-world video stream conditions.

The project's success relies heavily on the creation of a data-driven methodology that can be used to train and validate the model. To achieve this, we will be using the MPIIGaze dataset and focusing on preprocessing techniques that can help increase both training efficiency and model accuracy. Our goal is to address the challenges that arise from variable video quality and user behavior, which are usually encountered outside the controlled conditions of a laboratory.

Furthermore, the thesis will develop a user-friendly web-based interface, facilitating easy access for users to initiate video sessions for gaze tracking. This interface design will prioritize simplicity and minimal setup, aiming to democratize access to advanced gaze-tracking technology. An alert system will form an essential part of the interface, designed to notify users or administrators of significant gaze deviations that suggest lapses in focus. This system will involve setting precise thresholds for deviation, tailoring notifications to user needs, and incorporating feedback mechanisms for continuous improvement.

The research design of this project combines technical model adaptation with practical application development. The objective is to enhance the L2CS-Net model to enable real-time focus tracking and create a web-based platform that is accessible to all. The objective is to make a significant contribution to the use of gaze estimation technology, making them more relevant and beneficial for everyday use. Overall architecture is shown in Figure 3.1.

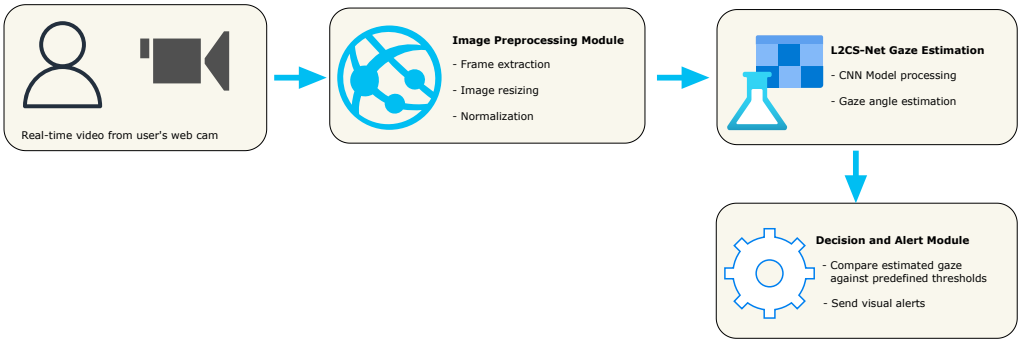


Figure 3.1 - Overall architecture

3.1.1 Model Selection and Rationale

3.1.1.1 Objective

The L2CS-Net model was chosen for the real-time focus tracking project in video streams in order to take advantage of its fine-grained gaze estimation capabilities in unrestricted settings. This choice is informed by the model's innovative approach to gaze prediction, which significantly improves per-angle prediction accuracy by regressing each gaze angle separately and employing a multi-loss strategy to enhance network learning and generalization.

Figure 3.2 provides a visual representation of the network architecture as described in the original publication by Ahmed A. Abdelrahman et al.[66], which helps to explain the L2CS-Net model chosen for this study. This diagram highlights the model's approach to gaze estimation, particularly its use of ResNet-50 for feature extraction and the separate regression of each gaze angle through distinct fully connected layers. Additionally, the figure illustrates the dual-loss strategy employed to optimize angle predictions, combining mean squared error with cross-entropy loss for each angle to enhance accuracy and generalization in unconstrained environments.

A version of the Residual Network (ResNet) architecture known as ResNet-50 was first presented by He et al.[67] in their groundbreaking study from 2015. It consists of 50 layers deep and is one of the most popular choices for deep learning applications in computer vision due to its robust performance across a wide range of tasks. ResNet-50 uses "skip connections," or residual learning connections, to help with the vanishing gradient issue that deep neural networks frequently have.

In the context of gaze estimation, ResNet-50 can extract complex features from input images that are crucial for accurately predicting the direction of gaze. Its ability to learn rich and discriminative features makes it an excellent choice for the backbone of a gaze estimation model. The network can acquire a high degree semantics from pictures of faces thanks to the deep residual layers, which are essential for understanding subtle gaze movements.

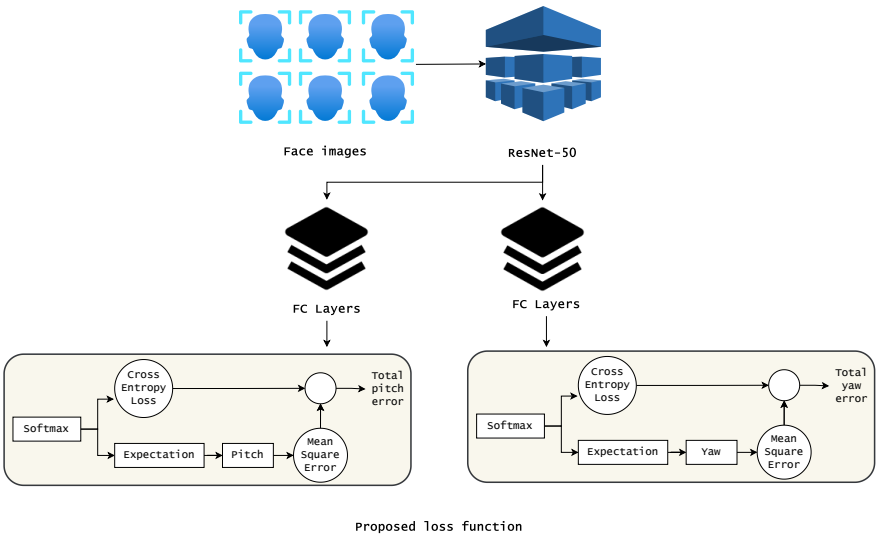


Figure 3.2 - L2CS Architecture

3.1.1.2 Model Adaptation

The architecture of L2CS-Net presents a strong framework for handling the challenges of gaze estimation in real-world scenarios. It extracts spatial gaze information from images by using ResNet-50 as the backbone and two separate loss functions for each gaze angle. Key considerations for adapting L2CS-Net to the project include:

- **Innovative Approach to Accuracy:** The L2CS-Net model achieved remarkable accuracy on the Gaze360 dataset, with errors as low as 10.41° for the front 180° and 9.02° for front-facing setups. These results underscore the model's exceptional capability to capture the nuances of the human gaze within varied and uncontrolled settings.
- **Enhancing Real-time Performance:** Implement optimizations to reduce inference time without compromising accuracy, ensuring the system can provide immediate feedback on user focus during video sessions.
- **Improving Robustness to Head Pose and Eye Appearance Variations:** Integrate additional preprocessing steps or network modifications to better handle the wide range of head poses and eye appearances present in the MPIIGaze dataset, reflecting the model's initial success in unconstrained settings.

3.1.2 Data Collection and Preparation

3.1.2.1 MPIIGaze Dataset

The MPIIGaze Dataset provides a robust dataset collected in natural, everyday settings. Unlike traditional datasets curated in controlled laboratory environments, MPIIGaze collected during three months period from 15 participants 213,659 images, capturing the details of daily laptop use. The dataset has detailed annotations ranging from eye landmarks to 3D head poses. It also includes

calibration data for enhanced model precision. Using pictures from a calibrated RGB camera, sophisticated face and facial landmark detection algorithms were used to begin the data collection process. A 3D facial shape model is then used to predict poses, and space normalization comes next. For gaze direction mapping, a multimodal CNN model including one fully connected layer and two convolutional layers integrated head pose data.

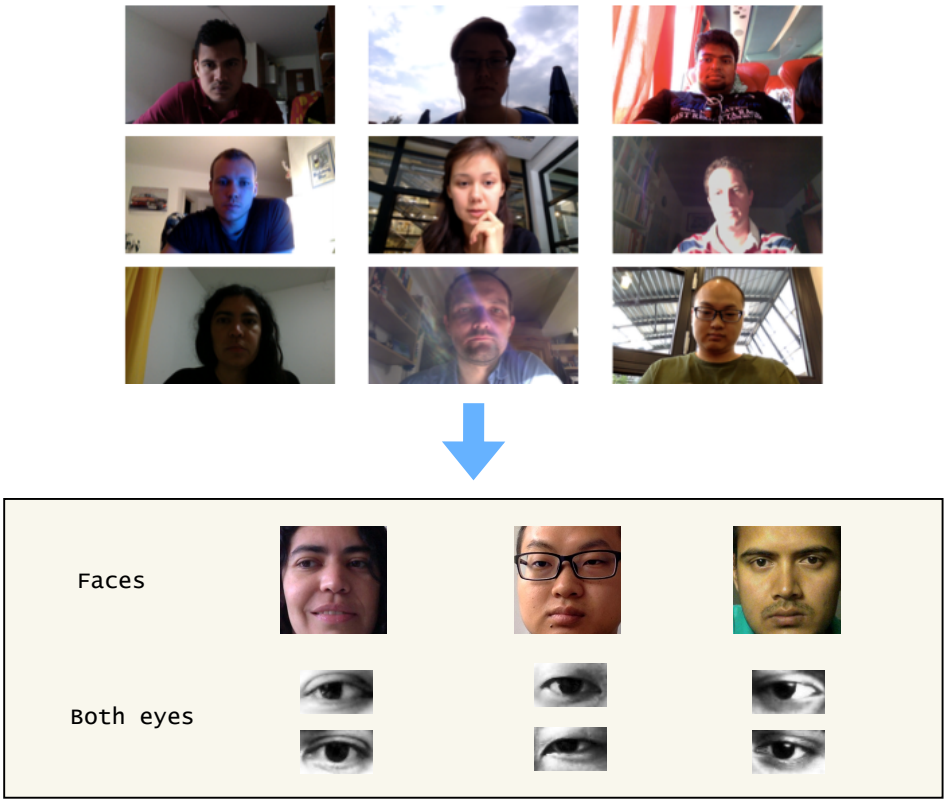


Figure 3.3 - Images prior and after preprocessing

3.1.2.2 Data Preprocessing Steps

The Figure 3.3 above demonstrates the images prior and after the preprocessing. Key preprocessing steps for the MPIIGaze dataset to optimize model training include:

- **Image and Metadata Loading:** The preprocessing begins by loading images along with their associated metadata, such as gaze annotations and camera calibration data. This ensures that all necessary information is available for subsequent normalization and transformation steps.
- **Normalization and Transformation:** Images are normalized to adjust for variations in head pose and camera perspective. This normalization aligns the images to a standard view, making it easier to accurately estimate gaze direction across different subjects and settings.

- **Region of Interest Extraction:** Critical regions within the images, specifically the eyes, are identified and extracted. This step focuses on the most relevant areas for gaze estimation, ensuring that the subsequent analysis is based on precise visual information.
- **Image Enhancement:** To enhance feature visibility, image enhancement techniques, particularly histogram equalization, are applied to the extracted regions of interest. This improves the contrast and clarity of the eye regions, which is vital for effective gaze tracking.
- **Data Annotation and Augmentation:** The processed images are annotated with their corresponding gaze and head pose data. This step also includes any necessary data augmentation to enrich the dataset and enhance the robustness of the gaze estimation model. The structure of the pre-processed dataset is displayed on Figure 3.4 and the content of label files on Table 3.1.

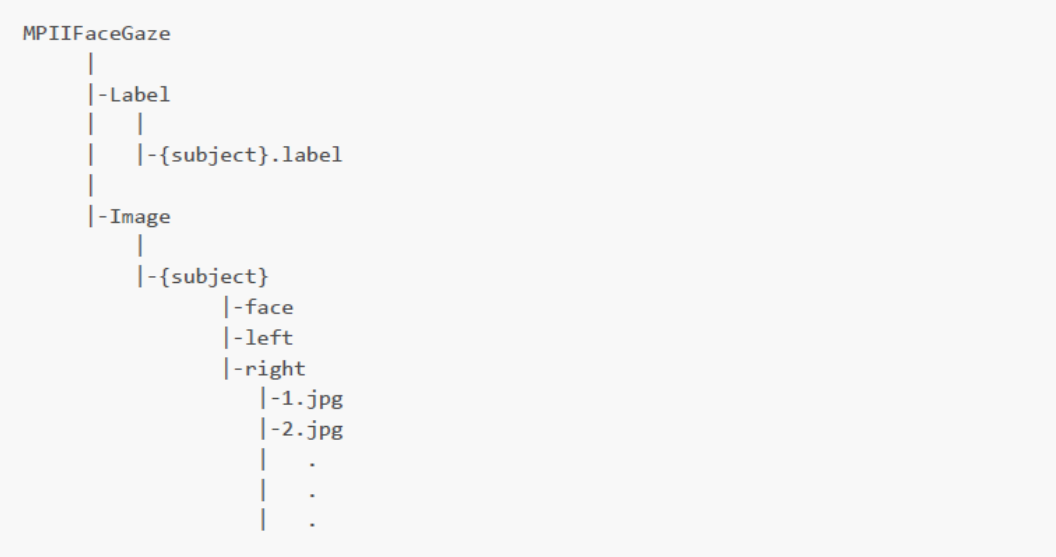


Figure 3.4 - Structure of pre-processed MPIIGaze dataset

3.1.3 Model Training and Evaluation

The training of the L2CS-Net model was conducted using Google Colab Pro, leveraging its powerful computing resources and convenient cloud-based environment. This process utilized Google Colab's advanced GPUs, enabling efficient training and experimentation with various parameters. The platform's integration with popular machine learning libraries and tools facilitated a seamless development and training process.

- **Environment Setup:** The training environment was set up in Google Colab Pro, which provided access to NVIDIA A100 GPUs. The A100 GPU, known for its robust performance in deep learning and high computational capabilities, is part of NVIDIA's latest generation of Tensor Core GPUs. This environment ensured high-efficiency handling of the computationally

Table 3.1 - Description of Variables in the Dataset Label Files

Variable	**Type**	**Description**
Face	string	Path of normalized face image.
Left	string	Path of normalized left eye image.
Right	string	Path of normalized right eye image.
Origin	string	The origin image.
WhichEye	string	Denotes which eye is chosen in standard MPIIGaze evaluation sets.
3DGaze	(3,)	Normalized 3D gaze direction vector.
3DHead	(3,)	Normalized 3D head orientation vector.
2DGaze	(2,)	Normalized 2D gaze direction vector i.e., yaw and pitch.
2DHead	(2,)	Normalized 2D head orientation vector i.e., yaw and pitch.
Rmat	(3,)	Rotation vector from original Camera Coordinate System (CCS) to the normalized CCS.
Smat	(3,)	The diagonal elements of the scale matrix used in normalization procedure.
GazeOrigin	(3,)	Origin of 3D gaze vector in normalized CCS.

intensive operations required for training the L2CS-Net model, allowing for rapid experimentation and optimization.

- **Data Ingestion and Preparation:** Data was uploaded to Google Drive and then mounted directly into the Google Colab environment. This setup facilitated efficient data management and accessibility, leveraging Google Drive's cloud storage capabilities. Custom data loaders were employed within Colab to batch, shuffle, and preprocess images, including resizing and normalization to prepare them for training with the L2CS-Net model.
- **Model Configuration and Versioning:** A total of 14 unique models were trained, each corresponding to a different fold of the dataset. This approach is akin to the "leave-one-person-out" cross-validation method, where the dataset was divided such that each model was trained on all but one specific subset (or "fold") of the data. This method ensures that each model is validated against a unique portion of the dataset that was not seen during its training phase, thereby enhancing the generalization capability of each model across diverse data scenarios.

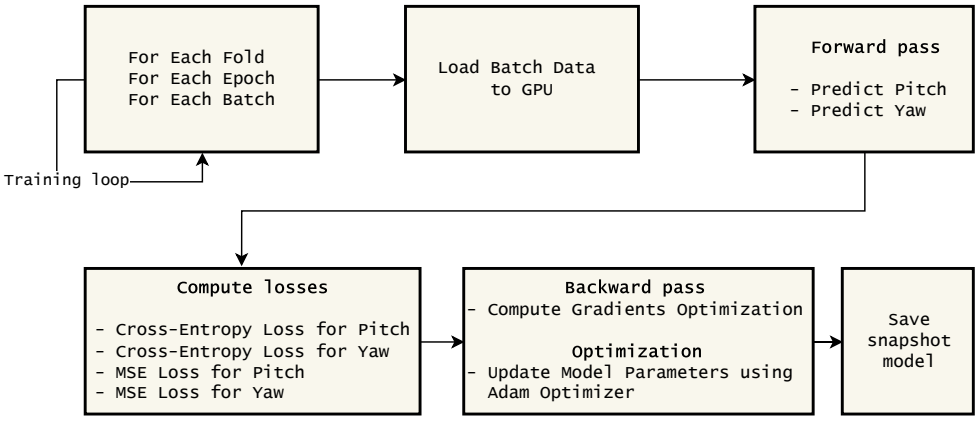


Figure 3.5 - Main steps during training phase

- **Hyperparameter Tuning and Training:** Given the constraints on resources, initially set at 0.001, the learning rate was later changed to 0.0001 when assessment revealed it was more efficient. Due to limited computational resources, batch sizes were set fixed to 20. Each model was trained for only 5 epochs, a decision driven by the need to balance resource availability with the desire to achieve a representative model performance. The training process is demonstrated in Figure 3.5 above.

```
train configuration, gpu_id=0, batch_size=16, model_arch=ResNet50
 Start training dataset=mpiigaze, loader=2607, fold=9--------------

Epoch [1/1], Iter [100/2606] Losses: Gaze Yaw 78.8188,Gaze Pitch 28.5706
Epoch [1/1], Iter [200/2606] Losses: Gaze Yaw 64.1186,Gaze Pitch 23.1180
Epoch [1/1], Iter [300/2606] Losses: Gaze Yaw 49.9005,Gaze Pitch 19.7548
Epoch [1/1], Iter [400/2606] Losses: Gaze Yaw 41.0467,Gaze Pitch 17.4471
Epoch [1/1], Iter [500/2606] Losses: Gaze Yaw 35.6434,Gaze Pitch 15.9941
Epoch [1/1], Iter [600/2606] Losses: Gaze Yaw 31.4879,Gaze Pitch 14.8825
Epoch [1/1], Iter [700/2606] Losses: Gaze Yaw 28.6682,Gaze Pitch 13.9753
Epoch [1/1], Iter [800/2606] Losses: Gaze Yaw 26.4457,Gaze Pitch 13.2788
Epoch [1/1], Iter [900/2606] Losses: Gaze Yaw 24.6599,Gaze Pitch 12.6783
Epoch [1/1], Iter [1000/2606] Losses: Gaze Yaw 23.1409,Gaze Pitch 12.1119
Epoch [1/1], Iter [1100/2606] Losses: Gaze Yaw 21.8186,Gaze Pitch 11.5898
Epoch [1/1], Iter [1200/2606] Losses: Gaze Yaw 20.7650,Gaze Pitch 11.1682
```

Figure 3.6 - Sample output of training process

- **Monitoring and Logging:** Training progress was monitored using basic logging techniques in Google Colab Pro. Key metrics such as loss and accuracy were tracked and displayed in real-time directly in the notebook's output cells. This approach allowed for essential insights into the model's performance. Figure 3.6 above shows the output generated by the training process.
- **Evaluation and Testing:** In the evaluation phase for the MPIIGaze dataset, the model underwent a testing procedure across 15 folds, using a leave-one-person-out cross-validation method (Figure 3.7). Each fold was processed

to assess the mean angular error, indicating the precision of gaze direction predictions. The automated script recorded and visualized the performance across epochs for detailed analysis, ensuring comprehensive validation of the model's generalization capability and performance trends.

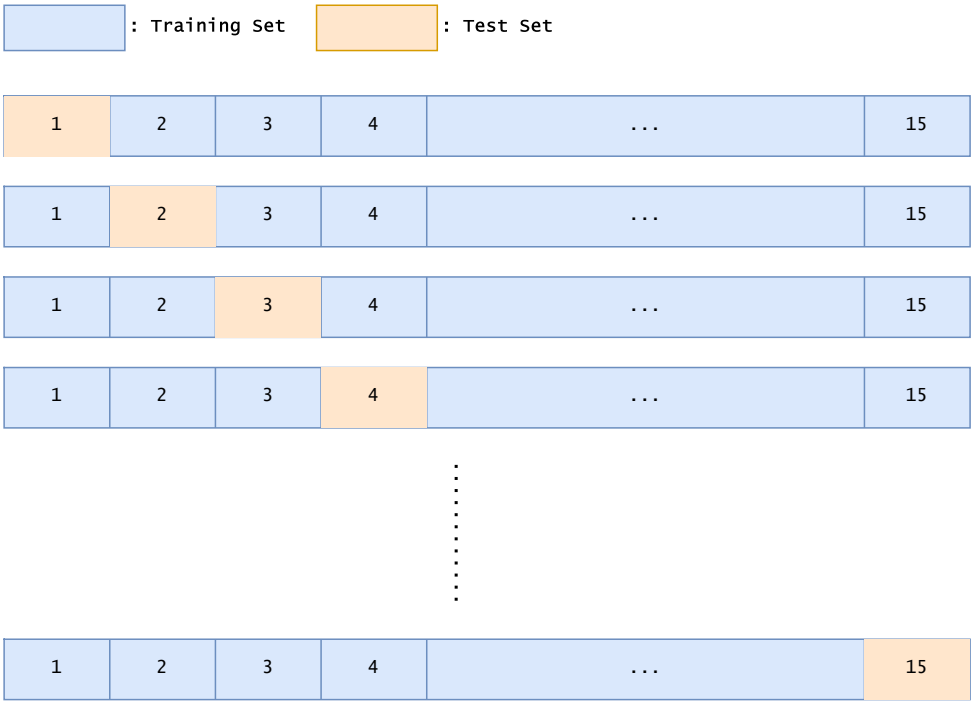


Figure 3.7 - Leave-one-person-out cross-validation across 15 people

```
test configuration equal gpu_id=cuda:0, batch_size=100, model_arch=ResNet50
Start testing dataset=mpiigaze, fold=2--------------------------------------

[_epoch_1.pkl---mpiigaze] Total Num:3000,MAE:6.100077058549096
```

Figure 3.8 - Sample output of testing process

In assessing the precision of gaze estimation models, the Mean Angular Error (MAE) serves as a pivotal metric. MAE measures the average angular deviation between the actual ground-truth vectors and the gaze vectors predicted by the model. Concretely, for a set of n predictions, the MAE is the mean of the arccosine of the cosine similarities between the predicted and true gaze vectors, expressed in degrees. Formally, the MAE can be described as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left(\cos^{-1}\left(\frac{\mathbf{g}_i \cdot \hat{\mathbf{g}}_i}{|\mathbf{g}_i||\hat{\mathbf{g}}_i|}\right)\right) \times \frac{180}{\pi} \tag{3.1}$$

where $\mathbf{g}_i$ and $\hat{\mathbf{g}}_i$ represent the ground-truth and predicted gaze vectors for the i-th observation, respectively, $\cdot$ denotes the dot product, $\|\cdot\|$ denotes

the Euclidean norm, $\cos^{-1}$ is the arccosine function, and n is the number of observations.

- **Ensemble Strategy:** We employed a simple yet effective ensemble strategy—averaging the predictions. Each of the 15 models, trained via a leave-one-out strategy on our dataset folds, predicts gaze angles (pitch and yaw). The final gaze estimation for each input is obtained by averaging these predictions. This method is chosen for its simplicity and effectiveness in reducing errors that might be present in individual model predictions.
  Given a set of M trained models, the ensemble prediction for a given input can be computed by averaging the predictions from all models. For a specific gaze estimation, let $\theta_{i,j}$ be the predicted gaze angle from the i-th model for the j-th observation. The ensemble prediction $\theta_{\text{ensemble},j}$ is then given by:

$$\theta_{\text{ensemble},j} = \frac{1}{M} \sum_{i=1}^{M} \theta_{i,j} \tag{3.2}$$

  where $\theta_{i,j}$ is the gaze angle estimate from the i-th model for the j-th observation, and M is the total number of models in the ensemble.
- **Model Saving and Deployment:** Post-training, the models were saved locally. Snapshotting capabilities ensured that the best-performing models were preserved at each training epoch.

3.1.4 Web-Based Interface Development

3.1.4.1 Design Considerations

This section details the development of a comprehensive web-based application that is hosted locally. The application is designed to interact with a ML model to perform real-time gaze estimation, with a frontend for capturing video and displaying results.

3.1.4.2 Implementation Technicalities

- **User Interface Design:** The web application provides a user-friendly interface that includes a live video feed, control buttons to start and stop the camera, and visual indicators that display real-time gaze direction. The design is focused on simplicity and ease of use, ensuring accessibility for all users.
- **Video Capture and Processing:** Utilizing HTML5 and the WebRTC API, the application captures video streams directly from the user's webcam. JavaScript is employed to handle the video data, capturing frames at a defined interval, resizing them, and preparing them for upload to the backend hosted on local Flask application.
- **Image Resizing:** Resizing is performed to adjust the dimensions of the captured image to a standard size (224x224). This is accomplished using the HTML canvas element. Initially, a canvas is created, and its dimensions are set to the desired width and height. The captured image from the video feed

is then drawn onto this canvas, effectively scaling it to the new dimensions. This step is vital as it ensures that all images fed into the model are of consistent size, thereby maintaining uniformity and reducing computational complexity.

- **Image Normalization:** Normalization adjusts the pixel values of the image to a specific range, typically between 0 and 1. This process involves extracting the image data from the canvas and iterating through the pixel values. Each RGB value is divided by 255, converting the pixel intensity from a 0-255 range to a 0-1 range. Normalizing the image data is essential because it ensures that the pre-trained model can handle variations in lighting conditions and image quality effectively, leading to more accurate predictions.
- **Dynamic Updates and Real-Time Interaction:** The frontend employs AJAX to send the video frames asynchronously to the backend and to receive gaze estimation results. These results are then dynamically displayed on the UI as overlays on the video feed, providing immediate feedback on the user's gaze direction.
- **Backend Hosting Locally:** The backend of the web application is developed using Flask, a lightweight Python web framework. The application is hosted locally, which allows for direct control over the deployment, management, and scaling processes. By developing and hosting the backend locally, we can easily test and refine the application logic in a controlled environment before considering any potential future deployment to a cloud platform. The backend is responsible for processing incoming video frames. This involves loading the trained model, calling the prediction pipeline, and calculating the gaze angle. The Flask application efficiently handles the entire workflow, from receiving video frames to returning the gaze estimation results, ensuring real-time interaction and accurate focus tracking. The backend expects the base64 encoded image with POST request. The image is then decoded from base64 and converted to required format for model prediction. Within the \predict function, the model receives the image for inference.
- **L2CS Model Integration:** Upon initialization, the backend class accepts parameters such as the path to the model weights, the type of model architecture, the device for computation (CPU or GPU), whether to include face detection, and the confidence threshold for face detection. The L2CS model is constructed according to the specified architecture and loaded with the provided weights. If face detection is enabled, a RetinaFace detector is instantiated. Additionally, a softmax layer and an index tensor are initialized for processing the model's output. Faces detected with confidence scores above the specified threshold are processed further. If no faces are detected, the function proceeds directly to predict the gaze direction from the entire frame.The model returns results that include the pitch and yaw angles for gaze estimation. These angles are converted from radians to degrees and formatted into a JSON response, which is then sent back to the client. Also, the response contains boudning boxes for each of the face identified, with coordinates of each of the 4 points of the bounding rectangle.

■ **Calculation of Gaze Point on Screen:**

We first need to understand the yaw and pitch angles that the gaze estimation model provides in order to identify the gaze point on the screen. We can tell which way the person is looking depending on the yaw and pitch angles. Using these angles along with the distance from the camera to the screen, we can calculate the coordinates of the gaze point on the screen.

The angular measurements Yaw (θ) and pitch (ϕ) correspond to the gaze's horizontal and vertical orientations, respectively. These values are provided by the gaze estimation model in radians.

To convert these angles into a point on the screen, we use the following formulas:

$$dx = -D \cdot \tan(\theta) \tag{3.3}$$

$$dy = -D \cdot \cos(\theta) \cdot \tan(\phi) \tag{3.4}$$

where:

- D is the distance from the camera to the screen.
- θ is the yaw angle.
- ϕ is the pitch angle.

These calculations yield dx and dy, which are the offsets from the center of the screen where the person is looking. We then adjust these offsets based on the actual resolution of the video frame to determine the exact coordinates on the screen.

Assuming the center of the screen corresponds to the midpoint of the video frame, we calculate the gaze point coordinates as follows:

$$gazePointX = \frac{imageWidth}{2} + dx \tag{3.5}$$

$$gazePointY = \frac{imageHeight}{2} + dy \tag{3.6}$$

where:

- $imageWidth$ is the width of the video frame.
- $imageHeight$ is the height of the video frame.

By combining these steps, we can accurately plot the gaze point on the screen. Below is a sample JavaScript function illustrating these calculations:

```
function calculateGazePoint(yaw, pitch, imageWidth, imageHeight,
    distance) {
    const dx = -distance * Math.tan(yaw);
    const dy = -distance * Math.cos(yaw) * Math.tan(pitch);
    const gazePointX = (imageWidth / 2) + dx;
    const gazePointY = (imageHeight / 2) + dy;

    return { x: gazePointX, y: gazePointY };
}
```

In this implementation, the `calculateGazePoint` function takes the yaw and pitch angles, the dimensions of the video frame, and the distance to the screen as inputs, and returns the calculated gaze point coordinates.
This approach ensures that the gaze point is accurately determined and can be visualized on the screen, providing real-time feedback on where the user is looking.

3.1.5 Alert System Design and Threshold Determination

3.1.5.1 Alert Mechanism

The alert mechanism in our gaze estimation system is used to notify users of significant deviations from expected gaze directions based on locally determined thresholds. These thresholds were calculated by measuring the yaw and pitch angles necessary to look at all four edges of a 14-inch laptop screen. In the scope of this research, it was set to 15 degrees for both of the angles.

When the gaze estimation system detects angles exceeding these calculated thresholds, it triggers an alert. This is crucial in applications like attentive user interfaces, where maintaining focus on the screen is essential. The alerts are visually indicated on the user interface by overlaying a red border around the video frame and displaying alert text in red, which notifies users that their gaze has moved outside the acceptable range.

3.1.5.2 User Interaction and Feedback

The system includes a simple feedback mechanism to refine its performance based on user interaction. Users can provide feedback directly through the interface if they believe an alert was incorrectly triggered. For example, if a user was looking at areas within the scope of a laptop monitor but the system flagged this as an alert, the user can dismiss the alert through a feedback button. This input is valuable for continuously adjusting the sensitivity and accuracy of the alert system in future works.

Chapter 4

Discussion and Results

4.1 Results

4.1.1 Accuracy of Gaze Estimation Model

The L2CS-Net model demonstrated varying levels of precision across the 15 different folds, as evidenced by the mean angular error rates. These metrics provide a detailed measure of the model's accuracy in estimating gaze direction. For instance, the lowest mean angular error recorded was 2.887302354 in fold 5, indicating high precision, while the highest error was 6.100077059 in fold 3, suggesting areas for potential improvement. The mean angular errors from each fold serve as a practical benchmark for comparing the model's performance against theoretical expectations and underline the importance of robust validation across diverse subsets of data.

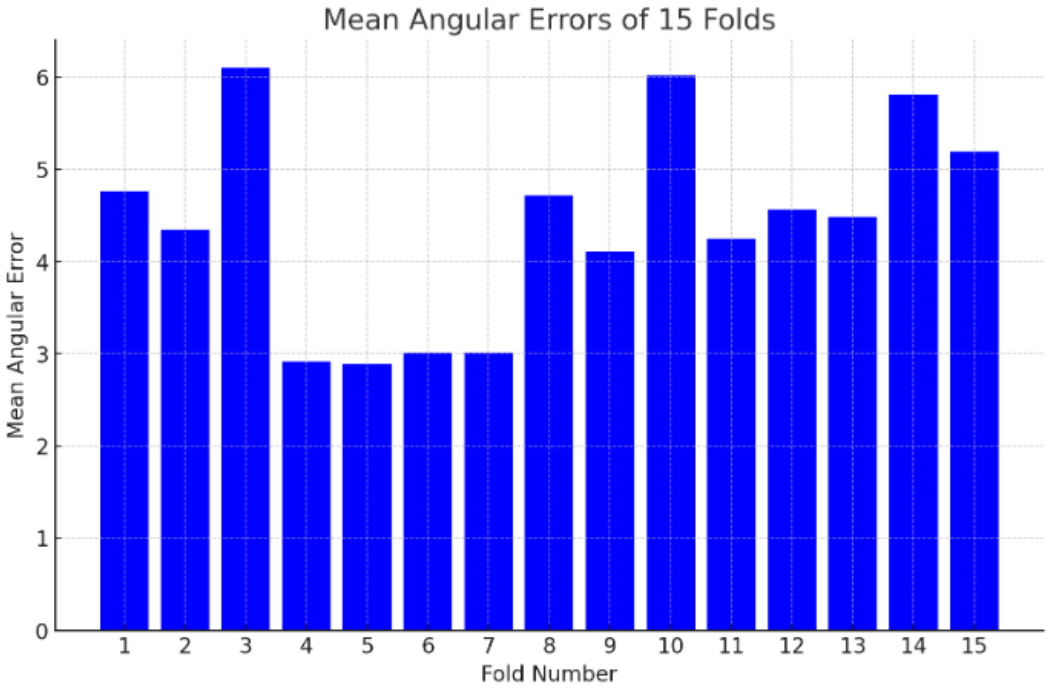
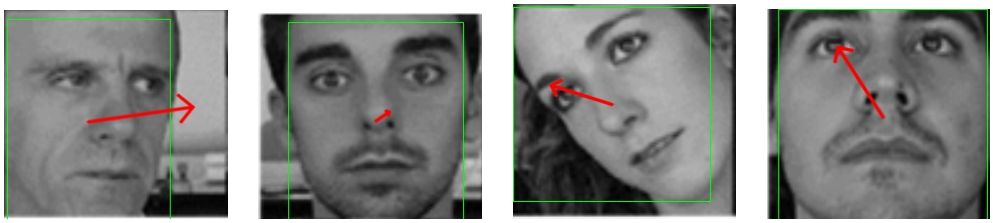


Figure 4.1 - Results of testing across 15 folds

The bar chart in Figure 4.1 above demonstrates the MAE across different folds. The Figure 4.2 below demonsrates the face identified and gaze vector shown by red arrow.



Yaw: 5.27° Pitch: -40.05° | Yaw: 3.06° Pitch: -4.22° | Yaw: 5.44° Pitch: 18.15° | Yaw: 19.61° Pitch: 13.28°

Figure 4.2 - Output of L2CS on sample dataset images.

4.1.2 System Responsiveness and Efficiency

Processing Speed: We computed the average processing time per frame at three different input resolutions in order to assess the effectiveness of our gaze estimating method. The results provide insight into the system's ability to meet real-time processing demands and how input resolution affects processing speed.

- 640x480: This resolution had the highest processing time at approximately 0.186 seconds per frame.
- 800x600: Processing at this resolution was slightly slower, averaging about 0.223 seconds per frame.
- 1024x768: The slowest average processing time was observed at this resolution, around 0.260 seconds per frame.

These findings suggest that as resolution increases, so does the complexity of processing tasks, leading to longer computational times. This is likely due to the larger data volume per image, which demands more from the computational resources, including memory bandwidth and processing power. Such trends underscore the importance of optimizing algorithms and hardware configurations when dealing with high-resolution inputs to ensure efficiency, especially in applications where processing speed is critical.

Table 4.1 - Average Processing Times for Different Resolutions

Resolution	**Average Processing Time (seconds)**
640x480	0.186
800x600	0.223
1024x768	0.260

4.2 Resource Utilization

The processing of images at varying resolutions has presented notable differences in CPU and memory utilization, which are essential for understanding the com-

putational demands of the system under different operational conditions.

4.2.1 CPU Utilization

The CPU usage exhibits variability across different resolutions:

- At a resolution of **640x480**, there is an average increase of **9.0%** in CPU usage, which indicates a moderate load on the system.
- At At a resolution of **800x600** it shows an average increase in CPU usage by 13.0%.
- A significant increase in CPU load is observed at the highest tested resolution of **1024x768**, with an increase of **67.0%**. This suggests that higher resolutions significantly tax the system's processing capabilities, reflecting the expected trend where higher data volumes require more computational power.

Table 4.2 - Average CPU Usage Increases for Different Resolutions

Resolution	**Average CPU Increase (%)**
640x480	9.0%
800x600	13.0%
1024x768	67.0%

4.2.2 Memory Utilization

Memory increases were more consistent across different resolutions:

- Both **640x480** and **800x600** resolutions saw a minimal increase in memory usage by approximately **1.0%**, indicating that memory requirements are not heavily impacted by resolution changes within this range.
- Interestingly, no increase in memory usage was observed at **1024x768**, which might suggest efficient memory handling or that the memory usage is capped by system constraints.

4.2.3 Load testing of the application

To evaluate the performance and scalability of the Flask API for gaze estimation, a parallel testing experiment was conducted. The test aimed to assess the API's ability to handle multiple requests efficiently and provide real-time gaze estimation. A Python script utilizing several libraries, including requests, base64, time, PIL, io, and concurrent.futures, was developed to send multiple requests to the Flask API in parallel.

The experiment revealed that the total time to process 100 concurrent requests was approximately 50.83 seconds. The average response time per request was 0.58 seconds, with the maximum response time observed at 0.82 seconds and the minimum response time at 0.34 seconds. These results indicate that the Flask API can handle multiple concurrent requests efficiently, with a consistent average response time.

The average response time of 0.58 seconds suggests that the API is suitable for real-time gaze estimation applications. However, further optimization and scaling strategies may be needed to handle a larger number of concurrent users in a production environment.

4.2.4 System Specifications

The system operates with **8 physical cores** and a **current CPU frequency of 2200.23 MHz**. During testing, the total CPU usage peaked at **15.6%**, with individual cores showing varied levels of engagement. The highest single-core usage was **50.0%**, which indicates that certain processes may be more CPU-intensive, likely related to image processing tasks.

4.2.5 Web Application Interface and Alert System

The web application for gaze estimation offers an intuitive and user-friendly interface, enabling seamless interaction and real-time monitoring. Upon accessing the application, users are greeted with a clean layout that features a live video feed from the user's webcam, allowing continuous monitoring of gaze behavior. The video feed is captured and processed at regular intervals, in our case in every three seconds. A canvas element captures frames from the video feed, overlaid with visual indicators to highlight areas of interest or concern. The application also includes a section displaying real-time data related to the user's gaze, such as pitch and yaw angles, providing insights into the user's focus and engagement. The Figure 4.3 demonstrates two scenarios. In first case, gaze of the user is within the image frame and the estimated gaze point is marked by green dot. In second case, gaze is directed out of the image frame and alert message is shown.

Figure 4.3 - Screenshots from web application interface

4.3 Discussion

The development and evaluation of the gaze estimation system incorporated several crucial components, each contributing to the system's overall effectiveness and efficiency. This section discusses the interconnected aspects of model architecture, training, testing, deployment, and real-world application.

The gaze estimation model was built on the ResNet-50 architecture, known for its deep network capabilities and efficiency in handling complex image data. This choice was pivotal in achieving high accuracy due to ResNet-50's ability to extract nuanced features from eye region images, which are critical for accurate gaze prediction.

A large set of over 200,000 photos that varied greatly in terms of demographic and environmental characteristics was used to train the algorithm. Some parameters, like the learning rate—which was initially set at 0.001 then was refined to 0.0001. The batch size was experimented with and settled at 16 to balance computational efficiency and model performance effectively. These parameters were crucial for stabilizing the model training and enhancing performance. The training was conducted over 5 epochs, which proved sufficient to achieve robust performance without excessive computation.

The testing phase employed a separate set of 9,000 (for each person) images to evaluate the model's generalization capabilities. The mean angular error (MAE) observed was 4.4 degrees, indicating a relatively high level of precision in gaze direction estimation. However, variations in results across different participants highlighted the need for further model refinement.

Deploying the model involved setting it up locally using Flask API to handle real-time gaze estimation. The model was integrated into a web application through a REST API, which allowed for real-time interaction and feedback. Users could interact with the system through a web interface, which captured images from the user's webcam at intervals and displayed gaze predictions directly on the screen.

The user interface was designed to be intuitive and user-friendly. It included visual cues such as a red alert text inside image frame when the gaze deviated beyond predefined thresholds. This immediate feedback was crucial for applications requiring user attention, such as in accessibility technologies and driver monitoring systems.

The development of the gaze estimation system was met with several significant challenges. One of the primary difficulties was managing the extensive dataset of 200,000 images, which posed issues related to data storage, retrieval, and preprocessing efficiency. The large volume of data significantly slowed down the training process, necessitating optimizations in data handling and model training techniques to improve efficiency.

Hardware limitations were another critical challenge. These restrictions impacted the real-time processing capabilities of the system, particularly during the model training phase where the need for substantial computational resources was critical. The limited GPU availability occasionally resulted in prolonged training times and affected the overall development timeline.

Chapter 5

Conclusions and future work

5.1 Conclusions

Using the L2CS-Net model, this thesis has effectively proven the integration and real-world implementation of a real-time focus tracking system. The model was trained with increased batch size (20) and learning rate 0.00001 on massive dataset of around 200.000 images. Although number of epochs were limited to 5 due to resources constraint, resulting models for each fold demonstrated average MAE of 4.4 degrees. Moreover, the performance of the model as low as 0.26 seconds for high resolution image and almost zero memory increase proves that the model can act as backing model in gaze estimation platform.

For the deployment and user interaction aspects, the research introduced a web-based interface to facilitate real-time processing and display of gaze data. This interface, developed as part of the project, allowed users to interact seamlessly with the system, providing immediate visual feedback and alerts. The alert mechanism, designed to notify users of significant gaze deviations, plays an important role in maintaining user engagement and focus during tasks, thereby addressing the second research question regarding the efficient use of a web interface in real-time gaze tracking.

The novelty of this research lies in the real-time implementation of the L2CS-Net model for gaze estimation, optimized for diverse real-world settings and integrated into a web-based application. This approach enhances user engagement in e-learning environments and ensures the integrity of virtual examinations through effective gaze tracking and alert mechanisms.

Furthermore, the project utilized model deployment technologies, including Flask API, to deploy the gaze estimation system. Successful deployment of the model and connection of different parts showcased that the setup can ensure scalable, robust, and accessible gaze estimation platform, available in different platforms.

5.2 Future works

Future work could expand the dataset to include more varied environmental and demographic conditions, and extend the training to more epochs to potentially

enhance the model's accuracy and robustness. Investigating the effects of different preprocessing techniques and augmentation methods could further improve the model's performance across diverse settings.

Developing more advanced algorithms or optimizing existing ones to reduce the latency in real-time processing could enhance the user experience. Exploring the integration of lightweight models or edge computing could help achieve faster response times necessary for real-time applications.

Improving the alert mechanism to include more nuanced feedback based on the severity and frequency of gaze deviation could make the system more adaptive and sensitive to user behavior. Integrating machine learning to predict potential distractions before they occur could proactively manage user focus.

Extending the application areas of the gaze tracking system to other fields such as virtual reality, augmented reality, and driver monitoring systems could open new avenues for research. Each application would bring unique challenges and opportunities for utilizing gaze tracking technology to enhance safety and user interaction.

Bibliography

[1] Phil Newton and Keioni Essex. How common is cheating in online exams and did it increase during the covid-19 pandemic? a systematic review. 10 2022. doi: 10.21203/rs.3.rs-2187710/v1.

[2] Andrew E. Fluck. An international review of eexam technologies and impact. *Computers Education*, 132:1–15, 2019. ISSN 0360-1315. doi: https://doi.org/10.1016/j.compedu.2018.12.008. URL https://www.sciencedirect.com/science/article/pii/S0360131518303270.

[3] Anil Kumar, Poonam Kumar, Shailendra Palvia, and Sanjay Verma. Online education worldwide: Current status and emerging trends. *Journal of Information Technology Case and Application Research*, 19:1–7, 03 2017. doi: 10.1080/15228053.2017.1294867.

[4] Hao Lei, Yunhuo Cui, and Wenye Zhou. Relationships between student engagement and academic achievement: A meta-analysis. *Social Behavior and Personality: an international journal*, 46:517–528, 03 2018. doi: 10.2224/sbp.7054.

[5] Fatima Ahmed, Thowiba Ahmed, Rashid Saeed, Hesham Alhumyani, S. Abdel-Khalek, and Hanaa Abu-Zinadah. Analysis and challenges of robust e-exams performance under covid-19. *Results in Physics*, 23:103987, 02 2021. doi: 10.1016/j.rinp.2021.103987.

[6] Hammad Naeem, Jawad Ahmad, and Muhammad Tayyab. Real-time object detection and tracking. pages 148–153, 12 2013. ISBN 978-1-4799-3043-2. doi: 10.1109/INMIC.2013.6731341.

[7] Sarkar Soumik. A brief history of online education. https://adamasuniversity.ac.in/a-brief-history-of-online-education/, May 2020. https://adamasuniversity.ac.in/a-brief-history-of-online-education/.

[8] Abid Haleem, Mohd Javaid, Mohd Qadri, and Rajiv Suman. Understanding the role of digital technologies in education: A review. *Sustainable Operations and Computers*, 3, 05 2022. doi: 10.1016/j.susoc.2022.05.004.

[9] Cereneo Santiago Jr, Joseph Callanta, Ma. Leah Ulanday, Zarah Centeno, and Ma.Cristina Bayla. Flexible learning adaptabilities in the new normal: E-learning resources, digital meeting platforms, online learning systems and learning engagement. 16:38–56, 01 2022. doi: 10.5281/zenodo.5762474.

[10] Muhammad Suleiman and Bilkisu Danmuchikwali. Digital education: Opportunities, threats, and challenges. 10 2020.

[11] Robert Carini, George Kuh, and Stephen Klein. Student engagement and student learning: Testing the linkages*. *Research in Higher Education*, 47: 1–32, 02 2006. doi: 10.1007/s11162-005-8150-9.

[12] Jorge Batista. A real-time driver visual attention monitoring system. volume 3522, pages 200–208, 06 2005. ISBN 978-3-540-26153-7. doi: 10.1007/11492429_25.

[13] Kapotaksha Das, Michalis Papakostas, Kais Riani, Andrew Gasiorowski, Mohamed Abouelenien, Mihai Burzo, and Rada Mihalcea. Detection and recognition of driver distraction using multimodal signals. *ACM Transactions on Interactive Intelligent Systems*, 12:1–28, 12 2022. doi: 10.1145/3519267.

[14] Santhoshikka R, Laranya R, and Harshavarthini C. Eye tracking and its applications. *IARJSET*, 8, 08 2021. doi: 10.17148/IARJSET.2021.8824.

[15] Jacob Whitehill, Zewe Serpell, Yi-Ching Lin, Aysha Foster, and Javier Movellan. The faces of engagement: Automatic recognition of student engagement-from facial expressions. *Affective Computing, IEEE Transactions on*, 5:86–98, 04 2014. doi: 10.1109/TAFFC.2014.2316163.

[16] Q. Liu, X. Yang, Z. Chen, et al. Using synchronized eye movements to assess attentional engagement. *Psychological Research*, 87:2039–2047, 2023. doi: 10.1007/s00426-023-01791-2. URL https://doi.org/10.1007/s00426-023-01791-2.

[17] Raquel Martins and Jorge Carvalho. Eye blinking as an indicator of fatigue and mental load – a systematic review. 02 2015. ISBN 9780429226526. doi: 10.1201/b18042-48.

[18] P. Ekman, W.V. Friesen, and J.C. Hager. Facial action coding system (facs): A technique for the measurement of facial action, 1978. 22.

[19] M. Dewan, Mahbub Murshed, and Fuhua Lin. Engagement detection in online learning: a review. *Smart Learning Environments*, 6, 01 2019. doi: 10.1186/s40561-018-0080-z.

[20] Mar Saneiro, Olga C. Santos, Sergio Salmeron-Majadas, and Jesus G. Boticario. Towards emotion detection in educational scenarios from facial expressions and body movements through multimodal approaches. *The Scientific World Journal*, 2014, 04 2014. doi: 10.1155/2014/484873.

[21] M. Elbawab and R. Henriques. Machine learning applied to student attentiveness detection: Using emotional and non-emotional measures. *Educational Information Technology*, 28:15717–15737, 2023. doi: 10.1007/s10639-023-11814-5. URL https://doi.org/10.1007/s10639-023-11814-5.

[22] Mustafa Uçar and Ersin Özdemir. Recognizing students and detecting student

engagement with real-time image processing. *Electronics*, 11, 05 2022. doi: 10.3390/electronics11091500.

[23] S. Kaddoura, D.E. Popescu, and J.D. Hemanth. A systematic review on machine learning models for online learning and examination systems. *PeerJ Computer Science*, 8:e986, 5 2022. doi: 10.7717/peerj-cs.986. URL https://doi.org/10.7717/peerj-cs.986.

[24] Janez Zaletelj and Andrej Kosir. Predicting students' attention in the classroom from kinect facial and body features. *EURASIP Journal on Image and Video Processing*, 2017, 12 2017. doi: 10.1186/s13640-017-0228-8.

[25] Prabin Sharma, Shubham Joshi, Subash Gautam, Vítor Filipe, and Manuel Reis. Student engagement detection using emotion analysis, eye tracking and head movement with machine learning, 09 2019.

[26] Sinem Aslan, Zehra Cataltepe, Itai Diner, Onur Dundar, Asli Esme, Ron Ferens, Gila Kamhi, Ece Oktay, and Canan Soysal. Learner engagement measurement and classification in 1:1 learning. *Proceedings - 2014 13th International Conference on Machine Learning and Applications, ICMLA 2014*, pages 545–552, 02 2015. doi: 10.1109/ICMLA.2014.111.

[27] Mahbub Murshed, M. Dewan, Fuhua Lin, and Dunwei Wen. Engagement detection in e-learning environments using convolutional neural networks. pages 80–86, 08 2019. doi: 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00028.

[28] Michael Argyle. Non-verbal communication and language. *Royal Institute of Philosophy Supplement*, 10:63–78, 1976. doi: 10.1017/s0080443600011079.

[29] Hr Chennamma and Xiaohui Yuan. A survey on eye-gaze tracking techniques. *Indian Journal of Computer Science and Engineering*, 4, 12 2013.

[30] Andronicus A. Akinyelu and Pieter Blignaut. Convolutional neural network-based methods for eye gaze estimation: A survey. *IEEE Access*, 8:142581–142605, 2020. doi: 10.1109/ACCESS.2020.3013540.

[31] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. pages 4511–4520, 06 2015. doi: 10.1109/CVPR.2015.7299081.

[32] Seonwook Park, Adrian Spurr, and Otmar Hilliges. *Deep Pictorial Gaze Estimation*, pages 741–757. 09 2018. ISBN 978-3-030-01260-1. doi: 10.1007/978-3-030-01261-8_44.

[33] Tobias Fischer, Hyung Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. 09 2018. ISBN 978-3-030-01248-9. doi: 10.1007/978-3-030-01249-6_21.

[34] Yihua Cheng, Feng Lu, and Xucong Zhang. *Appearance-Based Gaze Estimation via Evaluation-Guided Asymmetric Regression: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XIV*, pages

105–121. 10 2018. ISBN 978-3-030-01263-2. doi: 10.1007/978-3-030-01264-9_7.

[35] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:10623–10630, 04 2020. doi: 10.1609/aaai.v34i07.6636.

[36] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. 01 2019.

[37] Swati Jindal and Roberto Manduchi. Contrastive representation learning for gaze estimation, 10 2022.

[38] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. 11 2016.

[39] Eunji Chong, Nataniel Ruiz, Yongxin Wang, Yun Zhang, Agata Rozga, and James Rehg. *Connecting Gaze, Scene, and Attention: Generalized Attention Estimation via Joint Modeling of Gaze and Scene Saliency: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part V*, pages 397–412. 09 2018. ISBN 978-3-030-01227-4. doi: 10.1007/978-3-030-01228-1_24.

[40] Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. Following gaze in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1435–1443, 2017.

[41] Bhanuka Mahanama, Yasith Jayawardana, and Sampath Jayarathna. Gaze-net: appearance-based gaze estimation using capsule networks. pages 1–4, 05 2020. doi: 10.1145/3396339.3396393.

[42] Gang Liu, Yu Yu, Kenneth Funes Mora, and Jean-Marc Odobez. A differential approach for gaze estimation, 04 2019.

[43] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. 06 2016.

[44] Zhaokang Chen and Bertram E. Shi. Towards high performance low complexity calibration in appearance based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1174–1188, 2023. doi: 10.1109/TPAMI.2022.3148386.

[45] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9367–9376, 2019. doi: 10.1109/ICCV.2019.00946.

[46] Yiran Guan, Zhuoguang Chen, Wenzheng Zeng, Zhiguo Cao, and Yang Xiao. End-to-end video gaze estimation via capturing head-face-eye spatial-

temporal interaction context. *IEEE Signal Processing Letters*, 30:1687–1691, 2023. doi: 10.1109/LSP.2023.3332569.

[47] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation, 07 2020.

[48] Joohwan Kim, Michael Stengel, Alexander Majercik, Shalini Mello, David Dunn, Samuli Laine, Morgan McGuire, and David Luebke. Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation. pages 1–12, 05 2019. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300780.

[49] Brian Smith, Qi Yin, Steven Feiner, and Shree Nayar. Gaze locking: Passive eye contact detection for human-object interaction. pages 271–280, 10 2013. doi: 10.1145/2501988.2501994.

[50] Henri Tomas, Marcus Reyes, Raimarc Dionido, Mark Vincent Ty, Jonric Mirando, Joel Casimiro, Rowel Atienza, and Richard Guinto. Goo: A dataset for gaze object prediction in retail environments. pages 3119–3127, 06 2021. doi: 10.1109/CVPRW53098.2021.00349.

[51] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. pages 1821–1828, 06 2014. doi: 10.1109/CVPR.2014.235.

[52] Kenneth Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: a database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. pages 255–258, 03 2014. doi: 10.1145/2578153.2578190.

[53] Feng Lu, Takahiro Okabe, Yusuke Sugano, and Yoichi Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 32, 03 2014. doi: 10.1016/j.imavis.2014.01.005.

[54] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, 28, 08 2017. doi: 10.1007/s00138-017-0852-4.

[55] Denise R. S. Almeida, Konstantin Shmarko, and Elizabeth Lomas. The ethics of facial recognition technologies, surveillance, and accountability in an age of artificial intelligence: a comparative analysis of us, eu, and uk regulatory frameworks. *AI and Ethics*, 2, 08 2022. doi: 10.1007/s43681-021-00077-w.

[56] D. Woldeab and T. Brothen. 21st century assessment: Online proctoring, test anxiety, and student performance. *International Journal of E-Learning & Distance Education / Revue Internationale Du E-Learning Et La Formation à Distance*, 34(1), 2019. URL https://www.ijede.ca/index.php/jde/article/view/1106.

[57] George Raptis, Christina Katsini, Marios Belk, Christos Fidas, George Samaras, and Nikolaos Avouris. Using eye gaze data and visual activities to infer

human cognitive styles: Method and feasibility studies. pages 164–173, 07 2017. doi: 10.1145/3079628.3079690.

[58] Senuri De Silva, Sanuwani Dayarathna, Gangani Ariyarathne, Dulani Meedeniya, Sampath Jayarathna, Anne Michalek, and Gavindya Jayawardena. A rule-based system for adhd identification using eye movement data. pages 538–543, 07 2019. doi: 10.1109/MERCon.2019.8818865.

[59] Haofei Wang, Jimin Pi, Tong Qin, Shaojie Shen, and Bertram E. Shi. Slam-based localization of 3d gaze using a mobile eye tracker. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357067. doi: 10.1145/3204493.3204584. URL https://doi.org/10.1145/3204493.3204584.

[60] Rencheng Zheng, Kimihiko Nakano, Hiromitsu Ishiko, Kenji Hagita, Makoto Kihira, and Toshiya Yokozeki. Eye-gaze tracking analysis of driver behavior while interacting with navigation systems in an urban area. *IEEE Transactions on Human-Machine Systems*, 46, 12 2015. doi: 10.1109/THMS.2015.2504083.

[61] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Facevr: Real-time gaze-aware facial reenactment in virtual reality. *ACM Trans. Graph.*, 37(2), jun 2018. ISSN 0730-0301. doi: 10.1145/3182644. URL https://doi.org/10.1145/3182644.

[62] Boris Velichkovsky, Mikhail Rumyantsev, and Mikhail Morozov. New solution to the midas touch problem: Identification of visual commands via extraction of focal fixations. *Procedia Computer Science*, 39, 12 2014. doi: 10.1016/j.procs.2014.11.012.

[63] Maria Mikhailenko, Nadezhda Maksimenko, and Mikhail Kurushkin. Eye-tracking in immersive virtual reality for education: A review of the current progress and applications. *Frontiers in Education*, 7:697032, 03 2022. doi: 10.3389/feduc.2022.697032.

[64] Anna Belardinelli. Gaze-based intention estimation: principles, methodologies, and applications in hri, 02 2023.

[65] Pavan Sharma and Pranamesh Chakraborty. A review of driver gaze estimation and application in gaze behavior understanding, 07 2023.

[66] Ahmed Abdelrahman, Thorsten Hempel, Aly Khalifa, and Ayoub Al-Hamadi. L2cs-net: Fine-grained gaze estimation in unconstrained environments, 03 2022.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016. doi: 10.1109/CVPR.2016.90.