



Evaluating the Feasibility of Integrating Webcam-Based Eye Tracking into Hazard Perception Assessments

by

Ryan Evans

Supervisor: Nathan Morgan

Second Marker: Mark Huntly

Project submitted as part of the requirements for the award of

MSc Software Engineering (Web Development)

April 2026

Declaration

I, Ryan Evans, declare that I am the sole author of this project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: R. Evans

Date: 22nd April 2026

Form of Consent

I, Ryan Evans, hereby consent that my project, submitted in candidature for the MSc Software Engineering (Web Development) degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed: R. Evans

Date: 22nd April 2026

Abstract

Hazard perception is fundamental to driver safety, yet current assessment methods, including those employed by the UK Driver and Vehicle Standards Agency (DVSA), rely primarily on reaction-time-based scoring and offer limited insight into the attentional processes underlying hazard detection. Although eye-tracking research has demonstrated the importance of visual attention in hazard perception, traditional approaches depend on specialised hardware, restricting their scalability and integration into real-world assessment environments.

This study presents a proof-of-concept investigation into the integration of webcam-based gaze estimation within a hazard perception testing framework. A prototype system was developed using MediaPipe to estimate gaze direction during video-based hazard perception tasks, enabling the capture of low-resolution attentional signals using standard consumer hardware. Ten domain-informed participants, all software engineers within the UK Department for Transport, completed the task and provided structured feedback through a mixed-methods evaluation.

Findings demonstrate strong support for the integration of gaze-based feedback, with all participants indicating that such information would enhance hazard perception training or testing. Despite recognised limitations in accuracy, calibration stability, and sensitivity to environmental conditions, participants considered coarse gaze data sufficient to provide meaningful insights into attentional behaviour. The results highlight a clear distinction between the requirements of high-stakes assessment and training applications, with gaze estimation showing particular promise as a diagnostic and feedback tool.

This research contributes an initial validation of the feasibility of incorporating low-resolution gaze estimation into hazard perception systems in a manner that is interpretable, accessible, and aligned with existing assessment constraints. The findings establish a foundation for future work exploring gaze-informed feedback, improved calibration techniques, and the development of scalable, attention-aware driver training systems.

Contents

List of Abbreviations	7
1 Introduction	8
1.1 Problem Definition	9
2 Literature Review	10
2.1 Introduction to the Subject Area	10
2.2 Critical Literature Review	10
2.2.1 Theoretical Foundations	10
2.2.2 Current State of Research	12
2.2.3 Industrial Applications and Practice	15
2.2.4 Gaps and Opportunities	16
2.3 Research Rationale and Justification	18
2.3.1 Rationale for the Study	18
2.3.2 Justification of Research Focus	19
2.3.3 Academic Contribution	19
2.3.4 Practical Significance	19
2.3.5 Methodological Justification	20
2.3.6 Research Gap Analysis	20
3 System Design	23
3.1 Design Objectives	23
3.2 System Architecture	23
3.3 Gaze Estimation Pipeline	24
3.4 Hazard Mapping and Alignment	25
3.5 Implementation Details	25
4 Research Philosophy and Design Strategy	27
4.1 Research Approach	27
4.2 System Implementation	28
4.3 System Design Methodology	28
4.4 Participants	29
4.5 Survey Design	29
4.6 Data Collection Procedure	30
4.7 Data Analysis	31
4.8 Methodological Considerations	31
4.8.1 Participant Awareness of System Operation	31
4.8.2 Technical Constraints of Webcam-Based Gaze Estimation	31
4.8.3 Exploratory Study Scope	32
4.8.4 Ethical Considerations	32
4.9 Summary	32
5 Results	33
5.1 Overview of Responses	33
5.2 Calibration and Setup Experience	33
5.3 System Usability	34
5.4 Physical Comfort and Webcam Awareness	35

5.5	Hazard Perception Task Experience	36
5.6	Attitudes Toward Gaze Tracking	36
5.7	Qualitative Feedback	37
6	Discussion	39
6.1	Overview of Findings	39
6.2	Interpretation of Key Results	39
6.3	Comparison with Existing Literature	41
6.4	Implications for Hazard Perception Systems	42
6.5	Limitations	43
6.6	Future Work	44
6.7	Reflections on the Project	45
7	Conclusions and Recommendations	46
A	System Screenshots	53
A.1	Welcome Screen	53
A.2	Instructions Screen	53
A.3	Gaze Calibration	54
A.4	Calibration Validation	55
A.5	Test Phase	56
A.6	Debug Mode	56
A.7	Results Screen	57
A.8	AoI Marker Tool	57
B	Source Code	58
B.1	Application Entry Point	58
B.2	Type Definitions	58
B.3	Main Application Component	58
B.4	MediaPipe Gaze Tracker	70
B.5	Affine Calibration Transform	72
B.6	Gaze Utilities	73
B.7	Calibration Component	73
B.8	Video Player Component	76
B.9	Card Component	77
B.10	Status Badge Component	78
B.11	Data Capture Manager	78
B.12	AoI Marker Tool	80

List of Figures

1	System architecture of the prototype hazard perception platform	24
2	The initial setup screen prompting the user to start the test.	53
3	Instructions presented to the user before calibration.	53
4	The 9-point gaze calibration interface.	54
5	The validation screen showing the gaze dot tracking the user’s eye position against reference points.	55
6	A hazard perception clip during the test phase.	56
7	Debug mode showing the gaze overlay, hazard AOI bounding boxes, and camera preview.	56
8	The results screen displaying the user’s score and data export options. . .	57
9	The standalone AoI Marker Tool used to define hazard regions on video clips.	57

List of Tables

1	Participant perceptions of the calibration process	33
2	Participant perceptions of system usability	34
3	Participant perceptions of physical comfort and webcam awareness	35
4	Participant perceptions of the hazard perception task	36
5	Participant attitudes toward gaze-based hazard perception systems . . .	37

List of Abbreviations

DfT	Department for Transport
DVSA	Driver and Vehicle Standards Agency
DMS	Driver Monitoring System
MSc	Master of Science
PC	Personal Computer
SA	Situational Awareness
UI	User Interface
WBET	Webcam-Based Eye Tracking

1 Introduction

Road safety remains a significant public concern, with human factors continuing to play a central role in traffic collisions [1, 2]. Among these factors, hazard perception is widely recognised as a critical component of driver competence [3, 4, 5]. It refers to the ability to identify, interpret, and respond to potentially dangerous situations as they develop within the driving environment. Deficiencies in hazard perception are strongly associated with increased crash risk, particularly among novice drivers, who often exhibit delayed hazard recognition and less effective visual engagement with the road scene [3, 6, 7].

In the United Kingdom, hazard perception is formally assessed through video-based tests administered as part of the driving theory test. These assessments require candidates to observe pre-recorded driving scenarios and indicate when a developing hazard is detected. While this approach offers scalability, standardisation, and ease of deployment, it relies heavily on response timing as the primary measure of performance [4, 8]. As a result, it provides limited insight into the perceptual and cognitive processes that underpin hazard detection. In particular, existing assessment methods do not capture how drivers allocate visual attention, despite substantial evidence that attentional behaviour plays a central role in hazard perception [7, 9, 10].

Eye-tracking research has provided important insights into the relationship between visual attention and hazard perception. Studies consistently show that experienced drivers tend to exhibit broader scanning patterns, earlier fixation on potential hazards, and more anticipatory visual behaviour compared to novice drivers [7, 4, 9]. However, much of this research relies on laboratory-based infrared eye-tracking systems, which require specialised equipment and controlled environments [11, 12]. These constraints limit their applicability in large-scale or assessment-oriented contexts, where systems must be accessible, robust, and capable of operating under varying conditions [12, 13].

Recent advances in computer vision have led to the development of webcam-based eye tracking (WBET), which enables gaze estimation using standard consumer hardware [14, 15, 16]. Although WBET operates at lower spatial and temporal resolution than traditional eye-tracking systems, it offers a more accessible and scalable alternative [12, 17]. This raises an important question: whether a lower-resolution gaze signal may still provide meaningful insight into attentional behaviour within hazard perception tasks, particularly when aligned with the constraints of existing assessment methods [18, 10].

This dissertation investigates the feasibility of integrating webcam-based gaze estimation into a hazard perception testing framework. Rather than attempting to model detailed visual search strategies, the study adopts a proof-of-concept approach focused on whether coarse gaze information can be used to validate attentional alignment at the point of response. This reflects the practical limitations of WBET and the structure of existing hazard perception assessments, which rely on discrete user responses evaluated within defined time windows [12, 17].

The research is guided by two key questions:

- Can webcam-based gaze estimation be feasibly integrated into a hazard perception system using standard hardware?
- Do domain-informed users perceive value in incorporating gaze information into hazard perception training or assessment?

To address these questions, a prototype hazard perception system incorporating webcam-based gaze estimation was developed and evaluated through interaction with

participants. Feedback was collected using a structured survey and supported by qualitative observations from post-task discussions. The study therefore considers both the technical feasibility of the approach and its perceived usefulness within a transport-related context [19, 20].

The contribution of this work is intentionally scoped as exploratory. It does not seek to replace existing hazard perception assessment methods or to make claims about detailed attentional processes. Instead, it evaluates whether a minimal and interpretable gaze-based signal can be captured and meaningfully integrated into a hazard perception task. In doing so, the research provides a foundation for future work exploring the use of gaze-based feedback in driver training and the potential evolution of hazard perception assessment practices [21, 22, 23].

1.1 Problem Definition

Existing hazard perception assessments, including those used by the DVSA, rely primarily on reaction-time-based scoring and provide limited insight into the attentional processes that underpin hazard detection. As a result, they cannot distinguish between timely responses that reflect effective visual engagement with a developing hazard and those that arise from coincidental timing. At the same time, traditional infrared eye-tracking systems are too costly and specialised for widespread deployment, creating a gap for more accessible, webcam-based approaches that can provide a coarse but interpretable attentional signal suitable for integration into hazard perception tasks.

2 Literature Review

2.1 Introduction to the Subject Area

Hazard perception is widely regarded as a fundamental component of road safety and driver competence. It is commonly defined as the ability to detect, interpret, and respond to hazards as they emerge within the driving environment. Crucially, hazard perception extends beyond the mere detection of objects in the visual field, involving judgement under time pressure and the recognition of situations that may become dangerous before they fully materialise. Impairments in hazard perception are strongly associated with increased crash risk, a relationship that is particularly well documented among novice drivers, who typically recognise hazards later and exhibit less effective visual engagement than experienced drivers [3, 4, 5].

Despite this conceptual complexity, hazard perception is still most commonly assessed using outcome-based measures such as reaction time. In the United Kingdom, for example, the DVSA’s video-based hazard perception tests require candidates to indicate developing hazards within predefined scoring windows, an approach that is scalable and standardised but provides limited insight into the attentional processes underlying those responses [8, 1].

Eye-tracking research has demonstrated that visual attention plays a central role in hazard perception, with experienced drivers typically exhibiting earlier and more consistent visual engagement with potential hazards than novice drivers [7, 4]. Much of this evidence, however, is derived from laboratory-based infrared eye-tracking systems, which offer high precision but require specialised equipment and controlled environments. These constraints limit their suitability for large-scale or assessment-integrated deployment and raise questions regarding ecological validity [12, 13].

WBET has emerged as a more accessible alternative, enabling gaze estimation using standard consumer webcams and computer vision techniques [12, 14]. Although such systems operate at lower spatial and temporal resolution than infrared trackers, they are capable of capturing broad attentional information that may be sufficient for assessment contexts. Their low cost and ease of deployment make them viable candidates for integration into existing hazard perception testing frameworks [17, 19].

Accordingly, this study is positioned as a proof-of-concept (POC). Rather than attempting to model full scan-path behaviour or anticipatory visual strategies, it examines whether coarse, assessment-compatible gaze information obtained via WBET can be feasibly integrated into a hazard perception test. Specifically, the study explores whether validating attentional alignment at the point of response can provide additional information beyond reaction time alone. This framing prioritises feasibility, robustness, and alignment with current assessment practices, while explicitly acknowledging the need for further research to address finer-grained attentional processes [18, 22].

2.2 Critical Literature Review

2.2.1 Theoretical Foundations

Hazard perception is fundamentally grounded in cognitive psychology, drawing on theoretical models that explain how individuals perceive, interpret, and act upon information in dynamic and time-critical environments. Driving represents a particularly demanding cognitive task, requiring continuous visual monitoring, rapid interpretation of unfold-

ing events, and anticipatory decision-making under conditions of uncertainty. As such, theoretical frameworks concerned with situational awareness (SA), attentional resource allocation, and cognitive processing provide essential foundations for understanding hazard perception and its failure modes [24, 6].

One of the most influential theoretical models in this domain is Endsley’s theory of SA, which conceptualises awareness as a three-level construct consisting of the perception of environmental elements, comprehension of their meaning, and projection of their future status [24]. Within the driving context, this model offers a structured account of how drivers move from detecting visual cues, such as vehicles or pedestrians, to interpreting their relevance and anticipating potential hazards. Failures in hazard perception may occur at any stage of this process, for example when a driver visually registers an object but fails to recognise its potential threat, or when they are unable to predict how a situation is likely to develop.

Although widely adopted, the SA model has been subject to critique, particularly regarding its apparent linearity and the difficulty of empirically separating its constituent levels. Endsley later clarified that SA should not be interpreted as a rigid, sequential pipeline, but rather as a dynamic and continuously updated cognitive state shaped by both bottom-up sensory input and top-down goal-driven attention [25]. This clarification is particularly relevant in driving contexts, where hazard perception depends on continuous updating of expectations as situations evolve. Delayed hazard recognition may therefore arise from difficulties in attentional prioritisation or expectation management, rather than from failures of visual perception alone.

Closely related to SA is Wickens’ Multiple Resource Theory, which proposes that human cognitive capacity is limited and distributed across multiple resource pools, including visual, auditory, cognitive, and motor modalities [26]. According to this theory, task interference occurs when concurrent demands compete for the same cognitive resources, resulting in degraded performance. Driving frequently involves such competition, particularly in complex traffic environments where visual attention must be shared between monitoring hazards, reading signage, and maintaining vehicle control. Under high cognitive load, the allocation of attentional resources to hazard detection may therefore be compromised.

Multiple Resource Theory provides a useful explanatory framework for interpreting gaze behaviour in driving contexts. Narrow scanning patterns or prolonged fixations may indicate cognitive overload rather than deliberate attentional focus, while reduced visual exploration may reflect an adaptive but ultimately risky coping strategy. When examined qualitatively, gaze behaviour can provide insight into the cognitive demands experienced by drivers during hazard perception tasks. This perspective aligns well with WBET, which is more suited to identifying broad attentional patterns than to capturing fine-grained oculomotor detail [12, 17].

Selective attention theory further contributes to hazard perception research by explaining how individuals prioritise behaviourally relevant stimuli while filtering out irrelevant or distracting information. In real-world driving environments, drivers are exposed to a large volume of visual input, only a subset of which is directly relevant to safety. Effective hazard perception depends on the ability to selectively attend to cues that signal potential danger, such as changes in vehicle trajectory or pedestrian movement, while suppressing non-critical stimuli [6]. Novice drivers often struggle with this prioritisation, leading to attentional capture by visually salient but low-risk elements of the scene.

Dual-process theories of cognition also provide valuable insight into hazard percep-

tion by distinguishing between fast, automatic processing and slower, more deliberative reasoning [27]. Experienced drivers are thought to rely more heavily on rapid, intuitive processes developed through repeated exposure, enabling them to anticipate hazards with minimal conscious effort. In contrast, novice drivers may depend more on slower analytical processing, which can be overwhelmed in time-critical situations. This distinction helps explain why novice drivers may recognise hazards too late, even when their gaze is directed towards relevant areas of the scene [7, 4].

Empirical eye-tracking research supports these theoretical accounts, demonstrating that experienced drivers exhibit more anticipatory gaze behaviour, broader scanning patterns, and earlier fixation on potential hazards compared to novice drivers [7, 4]. These findings suggest that expertise in hazard perception is characterised less by faster reaction times and more by predictive attentional strategies. Such evidence reinforces the view that hazard perception is fundamentally an anticipatory cognitive skill rather than a simple reflexive response [10, 28].

Despite the explanatory value of these theoretical models, their empirical operationalisation remains challenging. Constructs such as situational awareness and attentional allocation are inherently abstract and cannot be directly observed. Eye tracking provides a useful window into visual attention, but gaze data alone cannot reliably represent comprehension or projection. This limitation has prompted debate regarding the extent to which gaze metrics can meaningfully support claims about higher-level cognitive processes [29].

For this reason, the present research adopts a deliberately constrained interpretation of gaze data. Rather than treating webcam-based gaze estimation as a proxy for anticipatory attention or situational awareness, the study evaluates whether coarse gaze information can validate attentional alignment at the point of response within a hazard perception test. Theoretical frameworks remain useful for motivating why attentional engagement matters, but the empirical claims of this study are limited to what can be supported by a low-resolution gaze signal [18, 17].

2.2.2 Current State of Research

Research into hazard perception has evolved substantially over the past two decades, driven by advances in sensing technologies and a growing recognition of the role of visual attention in driving safety. Early studies in this area focused primarily on behavioural outcomes such as reaction time and error rates, often using video-based hazard perception tasks to assess driver performance. While these approaches provided valuable large-scale insights, they offered limited visibility into the perceptual and cognitive processes underlying hazard recognition [4, 8].

Eye-tracking technology has since become a central tool in hazard perception research, enabling direct observation of how drivers allocate visual attention in dynamic traffic environments. A substantial body of work has demonstrated systematic differences between novice and experienced drivers, with experienced drivers exhibiting broader scanning patterns, earlier fixation on potential hazards, and greater anticipatory attention to areas of latent risk [7, 4]. These findings have shifted the focus of hazard perception research away from simple response timing and towards the study of attentional strategies and visual search behaviour [10, 30].

Traditionally, eye-tracking studies in driving research have relied on infrared (IR) systems, which provide high spatial accuracy and temporal resolution. Such systems are well

suiting to controlled experimental environments and have enabled detailed measurement of fixations, saccades, and gaze dispersion. However, IR-based eye trackers are costly, require specialised setup, and often constrain head movement, which can limit ecological validity. Participants may behave differently under laboratory conditions than they would in naturalistic or assessment-based settings, raising questions about the generalisability of findings to real-world driving behaviour [8, 13].

In response to these limitations, recent research has increasingly explored WBET as a more accessible alternative. Advances in computer vision, facial landmark detection, and machine learning have improved the accuracy and robustness of webcam-based gaze estimation, bringing performance closer to laboratory standards for certain constrained applications [12, 14]. However, compared to infrared systems, webcam-based solutions typically operate at lower frame rates and are more susceptible to noise arising from lighting conditions and head movement. In pilot testing, accuracy degraded noticeably under overhead lighting glare and when participants shifted posture after calibration, which reinforced the decision to adopt a deliberately broad interpretation of gaze. Under these limitations, WBET can still provide broad indicators of attentional direction that may be sufficient for assessment-oriented applications, particularly when the aim is to validate general gaze alignment rather than to model detailed scan-path behaviour [17, 12].

Webcam-Based Eye Tracking Frameworks In practice, WBET is not a single method but a family of implementation approaches that differ in how gaze is inferred from webcam imagery. Two particularly relevant frameworks in this space are WebGazer.js and MediaPipe, as they represent contrasting design philosophies for browser-based gaze estimation.

WebGazer.js is an open-source webcam eye-tracking library designed to infer a user’s gaze location on a web page in real time using a standard webcam. A distinguishing feature of WebGazer is that it self-calibrates by observing user interactions with the page and learning a mapping between eye features and screen positions. This makes it attractive for web-based applications because it can be integrated directly into browser environments and used without specialist hardware [31]. However, the same self-calibrating behaviour that makes WebGazer flexible can also introduce instability in contexts where user interaction patterns are limited, atypical, or difficult to separate from the primary task. In assessment-oriented environments, where consistency and interpretability are important, this can create practical challenges [19].

By contrast, MediaPipe does not function as a dedicated eye-tracking framework in the same sense. Instead, it provides real-time facial landmark detection and related vision tasks that can operate on continuous image or video streams. In its web implementation, MediaPipe’s Face Landmarker can detect facial landmarks from webcam input and can therefore be used as the basis for a custom gaze estimation pipeline [32]. This makes MediaPipe more flexible at the implementation level, as developers can derive coarse gaze direction from facial and eye landmarks rather than relying on an end-to-end gaze prediction library. The trade-off is that additional application logic is required to transform landmark output into usable gaze estimates [33, 34].

The distinction between these frameworks is methodologically important. WebGazer offers a more direct path to browser-based gaze estimation, but its self-training approach may be harder to control in dynamic hazard perception tasks. MediaPipe, by contrast, provides lower-level computer vision outputs that can be incorporated into a more con-

strained and transparent estimation pipeline, albeit with less direct support for screen-based gaze prediction. For proof-of-concept research such as the present study, this distinction matters because implementation stability and interpretability may be more important than attempting to maximise gaze precision [12, 17].

This comparison also reinforces a broader point in the WBET literature: feasibility depends not only on the general capabilities of webcam-based tracking, but also on the characteristics of the framework used to operationalise it. Different frameworks embed different assumptions about calibration, interaction, and inference, and these assumptions affect their suitability for hazard perception contexts [14, 35].

The growing viability of WBET has led to its adoption across a range of domains, including human-computer interaction, online learning, and driver monitoring systems (DMS) [14, 19]. In driving research, webcam-based approaches are particularly attractive for large-scale and remote studies, where traditional eye-tracking hardware may be impractical. These systems enable researchers to study driver attention in environments that more closely resemble real-world testing contexts, such as desktop-based simulations or web-delivered hazard perception assessments [12, 36].

Despite these advantages, WBET remains underutilised in hazard perception research, particularly in studies that seek to interpret gaze behaviour qualitatively. Much of the existing literature continues to prioritise quantitative performance metrics, such as mean fixation duration or reaction time, even when using eye-tracking data. While these metrics provide useful summary indicators, they often fail to capture the dynamic and context-dependent nature of visual attention during hazard encounters. As a result, important aspects of driver behaviour, such as ineffective scanning strategies or delayed situational interpretation, may be overlooked [18, 10].

In parallel with technological developments, hazard perception assessment methods have remained relatively static. Video-based tests, including those used by licensing authorities such as the UK DVSA, continue to dominate formal assessment practices. These tests offer standardisation and scalability but are inherently passive, requiring drivers to respond to pre-recorded scenarios without interacting with the driving environment. Research has raised concerns regarding the ecological validity of such assessments, as they do not fully reflect the perceptual and cognitive demands of real-world driving [8, 2]. Moreover, the scoring mechanisms employed by these tests provide limited insight into why a driver fails to identify a hazard.

Simulated driving environments offer greater interactivity and immersion and have been widely used in experimental research. Studies using driving simulators combined with eye tracking have demonstrated their value in examining attentional behaviour under controlled yet dynamic conditions [37]. However, simulator-based studies often require specialised equipment and are resource-intensive, which restricts their scalability and applicability in training or assessment contexts. As a result, there is growing interest in hybrid and lightweight approaches that balance experimental control with accessibility [36, 23].

This matters because WBET occupies a practical compromise. While it cannot replicate the precision of IR-based systems, it may provide sufficiently reliable low-resolution indicators of attentional direction for assessment-oriented applications. Importantly, its accessibility enables investigation of whether a minimal gaze signal can be used within hazard perception tasks without specialist hardware. In the context of this study, the relevant question is not whether WBET can support rich modelling of gaze behaviour, but whether it can support a defensible, low-resolution validation of attentional alignment at

the point of response [12, 17].

The trade-off is clear: high-precision eye tracking supports detailed analysis, but it does not scale well into assessment-style settings. While eye tracking has deepened understanding of visual attention in driving, its adoption within assessment and training contexts remains limited by cost, setup complexity, and limitations on deployment at scale. WBET presents a plausible route towards more accessible integration, but its application to hazard perception assessment remains underdeveloped and requires careful scoping of what can be validly inferred from lower-resolution gaze estimates. This gap motivates research that evaluates whether WBET can contribute an assessment-compatible attentional signal, rather than assuming it can replicate laboratory-grade analyses [18, 22].

2.2.3 Industrial Applications and Practice

Advances in eye-tracking technology have increasingly influenced industrial applications in driver training, vehicle safety systems, and human-machine interaction. Within these contexts, hazard perception is not only a subject of academic investigation but also a practical concern linked to accident prevention, driver monitoring, and regulatory compliance. As a result, many contemporary systems seek to assess or influence driver attention in real time, often using eye-tracking data as a proxy for engagement and situational awareness [9, 38].

Driver training and education represent one of the earliest and most established application areas. Eye tracking has been incorporated into driving simulators to provide feedback on visual scanning behaviour and hazard anticipation skills. Research has shown that presenting drivers with feedback on their gaze behaviour can improve awareness of ineffective scanning strategies and encourage more anticipatory visual search patterns [9]. Such systems are particularly relevant for novice drivers, who often struggle to identify latent hazards despite visually fixating on the driving scene.

Beyond training environments, eye tracking has become a key component of DMS in modern vehicles. These systems are designed to assess driver attentiveness and detect states such as distraction, drowsiness, or inattention. Commercial solutions developed by companies such as Tobii and Smart Eye are increasingly integrated into production vehicles and simulators, using gaze direction, eyelid closure, and head pose to infer driver state. In advanced driver assistance and partially automated driving systems, such monitoring is essential to ensure that drivers remain sufficiently engaged to resume control when required [38, 29].

Automotive manufacturers have also begun embedding eye-tracking-based monitoring into consumer vehicles. Systems such as Tesla’s Autopilot and General Motors’ Super Cruise incorporate inward-facing cameras to track driver attention and issue warnings when gaze is diverted from the road. While these systems operate in fundamentally different contexts to desktop-based hazard perception testing, they demonstrate broader industrial confidence in camera-based attention monitoring, albeit using proprietary and non-transparent methods. However, they are primarily designed for safety enforcement rather than behavioural insight. Their internal metrics and decision thresholds are typically proprietary, limiting transparency and making them unsuitable for research or pedagogical applications [29].

Despite their increasing prevalence, industrial eye-tracking systems raise important ethical and practical concerns. The collection of biometric data, including facial features

and gaze behaviour, introduces risks related to privacy, consent, and data misuse. Smith and Miller [29] highlight that biometric technologies, when deployed without adequate safeguards, may erode user trust and raise concerns about surveillance. These issues are particularly salient in automotive contexts, where data may be collected continuously and without explicit user interaction. As such, ethical considerations must be central to the design and deployment of eye-tracking-based systems.

From a technical perspective, the reliability of eye-tracking systems in real-world conditions remains a challenge. Variations in lighting, user appearance, and environmental conditions can degrade tracking accuracy, particularly for camera-based systems deployed outside controlled laboratory settings. Studies examining driver monitoring and drowsiness detection have noted that performance can be inconsistent across individuals and contexts, highlighting the need for robust validation and cautious interpretation of results [38]. These limitations are especially relevant when eye tracking is used to support high-stakes decisions, such as driver warnings or safety interventions.

Importantly, industrial applications tend to prioritise binary or threshold-based outcomes, such as whether a driver is attentive or distracted. While appropriate for safety systems, this approach offers limited insight into the nuanced perceptual and cognitive processes underlying hazard perception. For example, a driver may be visually attentive yet fail to interpret a developing hazard correctly. This distinction is critical for assessment and training contexts, where understanding why a hazard was missed is often more valuable than simply identifying that it was missed [18, 23].

WBET occupies a distinct position within this industrial landscape. Although it lacks the precision and robustness required for safety-critical in-vehicle systems, it offers sufficient fidelity for training, assessment, and research applications. Its low cost and ease of deployment make it particularly suitable for large-scale or remote hazard perception studies, where traditional hardware-based solutions are impractical. Furthermore, webcam-based approaches can offer greater transparency and flexibility in analysis, supporting interpretable use of gaze-related signals rather than reliance on opaque proprietary metrics [12, 17].

Overall, industrial applications of eye tracking illustrate both the strengths and limitations of gaze-based monitoring in driving contexts. While commercial systems have successfully incorporated eye tracking into safety and training applications, their emphasis on enforcement and automation limits their usefulness for understanding hazard perception as a cognitive skill. This reinforces the value of research-focused approaches that prioritise transparency and behavioural interpretation [29, 18].

2.2.4 Gaps and Opportunities

Despite substantial progress in hazard perception research, several important gaps remain, particularly in relation to methodological accessibility, interpretability, and the integration of attentional data into assessment contexts. While eye tracking has significantly enhanced understanding of how drivers allocate visual attention, its application has been constrained by reliance on specialist hardware, quantitative performance metrics, and laboratory-based methodologies. These constraints have limited both the scalability of research and the extent to which findings can inform real-world hazard perception assessment and training [8, 13].

A key gap lies in the underutilisation of WBET within hazard perception research. Although recent studies have demonstrated that modern webcam-based systems can achieve

performance approaching laboratory standards for certain gaze metrics [12, 17], their adoption in time-sensitive, dynamic driving tasks remains limited. Existing work has predominantly focused on feasibility or technical validation, rather than on applying WBET to investigate how drivers visually engage with hazards in realistic assessment scenarios. This remains an underexplored area, particularly given the widespread availability of webcam-equipped devices and the increasing demand for scalable research methods [19, 35].

Closely related to this is a methodological gap in how gaze data are analysed and interpreted. Much of the hazard perception literature continues to prioritise quantitative outcome measures, such as reaction time, fixation counts, or aggregate gaze statistics. While these measures are useful for summarising performance, they provide limited insight into the strategies and perceptual processes that underpin hazard recognition. As a result, they often fail to explain why drivers miss hazards or respond inappropriately. Despite their potential to reveal how attentional strategies unfold over time, qualitative approaches to analysing gaze behaviour remain relatively uncommon within hazard perception research [18, 10].

Another significant gap concerns the alignment between research methodologies and formal hazard perception assessment practices. Video-based tests used in licensing contexts, such as those employed by the UK DVSA, continue to rely on passive viewing and binary scoring systems. These assessments offer little transparency regarding the perceptual behaviours associated with success or failure, limiting their diagnostic and pedagogical value. Although research has highlighted differences in gaze behaviour between novice and experienced drivers, these insights have not been systematically integrated into assessment design or feedback mechanisms [4, 8, 21].

There is also a theoretical opportunity to more explicitly connect cognitive models of attention and SA with observable gaze behaviour in hazard perception tasks. While frameworks such as SA and Multiple Resource Theory are frequently cited, their empirical application often remains indirect. Eye tracking is commonly treated as a proxy for attention without sufficient consideration of how gaze behaviour reflects higher-level cognitive processes such as comprehension and anticipation. Qualitative interpretation of gaze data offers a pathway for bridging this gap, enabling researchers to examine how theoretical constructs manifest in practice [25, 18].

From a practical perspective, the emphasis on high-precision eye tracking has inadvertently limited participation and diversity in hazard perception research. Studies conducted using laboratory-grade equipment often involve small, homogenous samples and artificial testing environments. Webcam-based approaches offer an opportunity to address this limitation by enabling data collection in more naturalistic settings and from broader participant populations. This is particularly relevant for novice drivers, who are both at greatest risk and most likely to benefit from improved assessment and training tools [1, 22].

Finally, ethical and transparency considerations present both a challenge and an opportunity. Commercial DMS increasingly collect gaze and facial data, yet their internal decision-making processes are often opaque. Research-oriented webcam-based systems, by contrast, allow for greater methodological transparency and participant agency. When combined with qualitative analysis, these systems support more interpretable and ethically grounded investigations of driver behaviour, which is especially important in educational and assessment contexts [29, 23].

In summary, the literature reveals clear gaps in the assessment-oriented application of

WBET to hazard perception, particularly in relation to feasibility, interpretability, and integration with existing scoring models. While richer gaze analysis remains a longer-term research direction, a more immediate gap concerns whether WBET can provide an assessment-compatible signal that helps distinguish between timely responses that reflect attentional engagement with the hazard and those that do not. By focusing on low-resolution gaze alignment at the point of response, this research addresses a foundational feasibility question and establishes a baseline for future work that may extend towards higher-resolution gaze representations and more detailed behavioural interpretation [12, 17, 18].

2.3 Research Rationale and Justification

This chapter outlines the rationale for the proposed study and justifies its focus, scope, and methodological approach. Building on the gaps identified in the literature, the chapter explains why a POC investigation into the feasibility of integrating WBET within hazard perception tasks is timely and necessary. The justification is presented in terms of topical relevance, practical significance, academic contribution, and methodological suitability, with claims deliberately constrained to what can be supported by a low-resolution gaze signal suitable for assessment contexts [12, 17].

2.3.1 Rationale for the Study

Hazard perception remains a central concern in road safety research and driver assessment, particularly among novice drivers, who are consistently associated with higher collision risk in the road safety literature [1, 3].

At the same time, the literature consistently suggests that hazard perception performance is strongly linked to visual attention and the ability to allocate attention to relevant cues as situations unfold. Eye-tracking studies have demonstrated differences between novice and experienced drivers in how they visually engage with hazards, yet these insights have not been systematically integrated into formal assessment practices. As a result, a persistent disconnect remains between research evidence on visual attention and the practical reality of hazard perception testing, where outcomes are scored without reference to where drivers are looking [7, 18].

Recent advances in WBET provide an opportunity to address this disconnect in a pragmatic way. Unlike laboratory-based infrared eye tracking, webcam-based approaches can be deployed using widely available consumer hardware, lowering cost and reducing barriers to integration within existing testing environments [12, 14]. However, WBET is inherently limited in spatial and temporal precision and is sensitive to calibration drift, lighting variation, and head movement. These constraints mean that the most defensible near-term application is not detailed modelling of visual search strategies, but the extraction of coarse, interpretable attentional signals aligned with assessment limitations [17, 13].

Against this background, the study evaluates whether broad gaze information captured via WBET can be feasibly incorporated into a hazard perception test and whether it can add interpretive value beyond reaction time alone. Specifically, the study examines attentional alignment at the point of response as a minimal signal of whether a candidate was visually engaged with the hazard when they indicated detection [18, 10].

2.3.2 Justification of Research Focus

The decision to focus on WBET is justified on both practical and methodological grounds. From a practical perspective, infrared eye trackers are costly, require specialist setup, and are typically confined to controlled laboratory environments. These practicalities limit scalability and reduce the feasibility of embedding eye tracking into assessment contexts that must be standardised, repeatable, and robust across large candidate populations. Webcam-based systems, by contrast, can be deployed with standard computing equipment, enabling broader participation and supporting evaluation in contexts closer to real-world hazard perception testing [12, 20].

Methodologically, the study adopts a deliberately constrained approach that matches the capabilities and limitations of WBET. Rather than attempting to infer detailed scan-path behaviour, fixation dynamics, or anticipatory gaze patterns, the research focuses on whether a low-resolution attentional signal can be derived reliably enough to support assessment augmentation. This is operationalised through evaluating whether gaze is broadly aligned with the hazard, operationalised at the quadrant level, at the moment of response. This focus avoids false precision and aligns with the structure of existing hazard perception assessments, which rely on discrete responses evaluated within predefined temporal windows [4, 21].

This POC framing is also justified by the current state of the literature. While WBET has been examined in other domains and has improved substantially in robustness, there remains limited work assessing how it can be integrated into hazard perception testing in a way that prioritises interpretability and feasibility. Establishing whether a minimal, coarse gaze-response congruence signal is viable is a necessary first step before more complex gaze-based approaches can be credibly proposed for assessment settings [12, 17].

2.3.3 Academic Contribution

From an academic standpoint, this research contributes to hazard perception and applied attention research by addressing a practical methodological gap: whether webcam-based gaze estimation can support an attentional signal within hazard perception tasks. The contribution is intentionally foundational. Instead of advancing claims about anticipatory attention or situational awareness inference from gaze, the study evaluates the feasibility and interpretive usefulness of a minimal signal that can be transparently operationalised and realistically deployed [18, 17].

The study also contributes to ongoing discussion around the operationalisation of attention-related constructs in applied settings. The literature frequently motivates hazard perception in terms of attentional allocation and situational awareness, but empirical work often either relies on laboratory-grade instrumentation or treats attention as equivalent to reaction time. By examining broad gaze alignment at the point of response, this research offers a pragmatic bridge between theory and assessment practice, while explicitly limiting inference to what the measurement can support [25, 39].

2.3.4 Practical Significance

The findings of this study have potential implications for hazard perception assessment and, indirectly, for training. If low-resolution gaze alignment can be captured reliably enough to contextualise a response, hazard perception testing could move beyond purely outcome-based measures towards a more interpretable account of performance. In prac-

tical terms, a response that falls within the scoring window but is accompanied by gaze directed away from the hazard may warrant a different interpretation than one where gaze falls within the same screen quadrant. This could support the development of more diagnostic assessment feedback without requiring fundamental changes to existing test formats [22, 40].

In addition, the accessibility of WBET aligns with broader trends towards digital and remote assessment. Demonstrating feasibility in controlled, assessment-like conditions is a prerequisite for considering wider deployment. While this study does not claim to deliver a complete assessment system, it provides evidence regarding whether WBET can contribute a usable attentional signal under realistic limitations [1, 23].

2.3.5 Methodological Justification

The methodological choices underpinning this study follow directly from the research aim. The goal is not to predict crash risk or to automate hazard detection, but to assess whether gaze information from WBET can feasibly augment hazard perception testing in an interpretable manner. A coarse spatial representation of gaze is therefore appropriate, as it is more robust to the noise and variability inherent in webcam-based estimation and avoids implying precision that cannot be supported [12, 17].

By constraining analysis to attentional alignment at the point of response, the methodology avoids overextending claims about higher-level cognition. This positioning supports a defensible POC contribution: establishing whether a minimal gaze–response congruence signal can be obtained reliably enough to justify subsequent research into richer gaze representations, longer temporal analysis windows, or more detailed behavioural interpretation under more controlled conditions [18, 13].

Overall, the study is motivated by limitations in current hazard perception assessment practices, practical barriers to traditional eye tracking, and the emerging potential of WBET. By adopting a POC approach focused on broad gaze alignment at response, the research aims to provide a realistic and defensible foundation for future work on integrating gaze-based signals into hazard perception assessment and training [17, 22].

2.3.6 Research Gap Analysis

This section synthesises the key gaps identified in the literature and clarifies how the proposed study addresses them. While hazard perception has been extensively studied within road safety research, significant limitations remain in how driver attention is measured, interpreted, and integrated into assessment contexts [4, 8]. These gaps span methodological, theoretical, and practical dimensions, and collectively justify the need for the present research [12, 17].

A primary gap concerns the continued reliance on outcome-based measures in hazard perception assessment. Much of the existing research, as well as formal testing practices, focuses on reaction time, hazard detection accuracy, or binary scoring outcomes [4, 21]. While these measures offer scalable and standardised evaluation, they provide limited insight into the perceptual and cognitive processes that underpin performance. As a result, they often fail to explain why hazards are missed or detected late, particularly among novice drivers. This limits the diagnostic value of hazard perception testing and constrains its potential as a training tool [22, 10].

Although eye tracking has been widely used to study visual attention in driving, its application within hazard perception research has been shaped by methodological

practicalities. The majority of studies employ IR eye-tracking systems, which offer high precision but require specialist equipment, controlled environments, and small participant samples [7, 37]. These constraints reduce ecological validity and limit the feasibility of deploying eye tracking in assessment-like or remote contexts. Consequently, insights gained from eye-tracking research have had limited impact on real-world hazard perception assessment practices [8, 13].

A further gap exists in the underrepresentation of WBET within hazard perception research. While recent studies demonstrate that modern webcam-based systems can capture gaze behaviour with sufficient accuracy for behavioural analysis, their use in dynamic, time-sensitive hazard perception tasks remains limited [12, 17]. Existing work has largely focused on technical validation or feasibility rather than on applying WBET to examine how drivers visually engage with hazards during assessment scenarios. This represents a missed opportunity, particularly given the widespread availability of webcam-equipped devices and the growing demand for scalable and accessible research methods [19, 35].

The literature also reveals a conceptual gap in how gaze data are analysed. Many studies treat eye-tracking outputs as quantitative proxies for attention, relying on aggregate metrics such as fixation duration or gaze frequency. While useful, these metrics often fail to capture the temporal and contextual dynamics of visual attention during hazard encounters. Qualitative approaches to gaze analysis, which emphasise scanning strategies, attentional shifts, and the sequencing of visual behaviour, are comparatively rare. This limits the field’s ability to link observable gaze behaviour to higher-level cognitive constructs such as SA and anticipatory processing [18, 10].

Additionally, there is a gap between theoretical models of hazard perception and their empirical operationalisation. Frameworks such as SA, Multiple Resource Theory, selective attention, and dual-process cognition are frequently cited, yet their application often remains abstract [24, 26, 6, 27]. Eye tracking is commonly used as an indirect indicator of attention without sufficient consideration of how gaze behaviour reflects comprehension, expectation, or projection. More interpretative approaches are required to bridge this gap and to examine how theoretical constructs manifest in real hazard perception tasks [25, 18].

From a practical perspective, existing hazard perception assessments offer limited feedback to learners and instructors. Licensing tests typically indicate whether a hazard was detected within an acceptable time window but provide no information about visual search behaviour or attentional strategy [4, 8]. This constrains opportunities for targeted training and behavioural improvement. Research methods that prioritise interpretability and behavioural insight are therefore needed to support more informative assessment and feedback mechanisms [21, 23].

In response to these gaps, the present study adopts a POC approach to investigating hazard perception using WBET. Rather than attempting to characterise full visual search strategies, the study evaluates whether coarse gaze alignment at the point of response can be captured robustly enough to augment existing outcome-based assessment [18, 17]. WBET offers an accessible means of capturing this minimal attentional signal in assessment-like contexts, while keeping empirical claims aligned with the limitations of low-resolution gaze estimation [12, 13].

In summary, the research gap addressed by this study lies at the intersection of accessibility, interpretability, and theoretical integration. Existing hazard perception research has yet to establish whether WBET can provide a robust and interpretable attentional

signal that is feasible to integrate alongside existing outcome-based measures [17, 22].

3 System Design

This chapter describes the design and implementation of the prototype hazard perception system developed in this study. The system was designed to demonstrate how webcam-based gaze estimation could be integrated into a hazard perception testing environment using standard consumer hardware, following recent work on browser-based eye tracking using MediaPipe-style pipelines [12, 17, 35]. The design prioritises simplicity, robustness, and compatibility with existing hazard perception interaction models.

3.1 Design Objectives

The primary objective of the system was to evaluate whether gaze estimation could be incorporated into a hazard perception task without negatively affecting usability or user experience. To achieve this, the system was designed around the following goals:

- Integrate gaze estimation into a hazard perception task using standard webcam hardware
- Maintain a user experience consistent with conventional hazard perception tests
- Ensure the system operates unobtrusively without distracting the user
- Provide a foundation for visualising gaze behaviour for training purposes

Given the exploratory nature of the study, the system prioritises feasibility and robustness over high-precision gaze tracking, consistent with recommendations for low-resolution, browser-based gaze estimation in applied settings [14, 17].

3.2 System Architecture

The prototype system was implemented as a web-based application running locally in a browser environment, in line with recent scalable eye-tracking frameworks that emphasise commodity hardware and on-device processing [35, 34]. The system consists of three main components: hazard perception video playback, user interaction handling, and gaze estimation.

Figure 1 illustrates the high-level architecture of the prototype system, showing how webcam input, gaze estimation, and hazard mapping interact within the application.

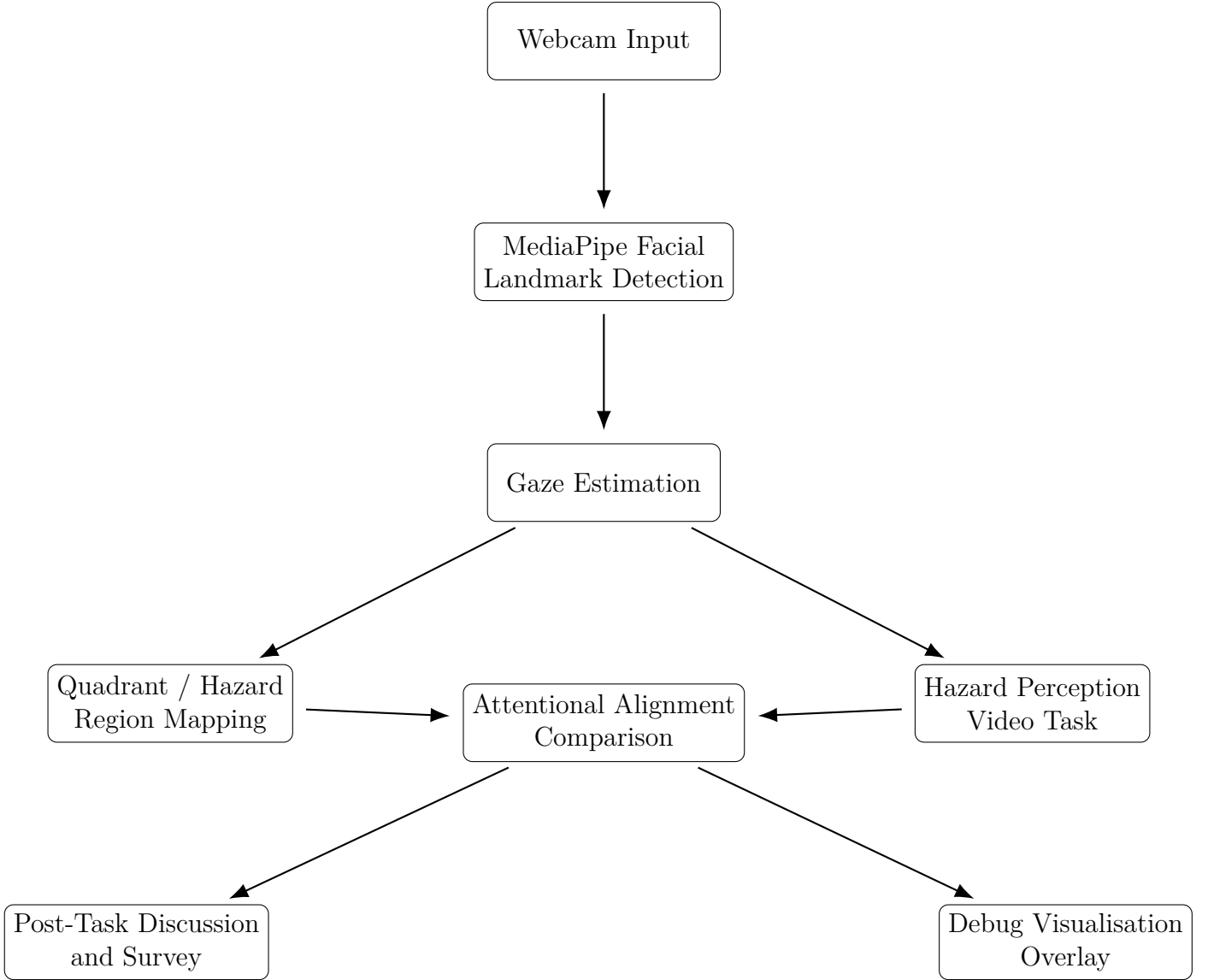


Figure 1: System architecture of the prototype hazard perception platform

As shown in Figure 1, the system processes webcam input using MediaPipe to estimate gaze direction, which is then mapped to predefined hazard regions and compared with user interaction events [32, 33]. Additional components support debugging and post-task evaluation.

3.3 Gaze Estimation Pipeline

The gaze estimation pipeline is based on Google’s MediaPipe framework, which provides real-time facial landmark detection using standard webcam input [32]. The system detects key facial and eye landmarks and uses these to approximate gaze direction relative to the screen, following common practice in recent MediaPipe-based gaze tracking work [33, 34].

The pipeline consists of the following stages:

1. Webcam input capture and frame processing

2. Facial landmark detection using MediaPipe
3. Extraction of eye and head pose features
4. Mapping of landmark positions to screen coordinates

Because MediaPipe does not provide dedicated eye-tracking functionality, gaze estimation is approximate and operates at a coarse spatial level. This approach was considered sufficient for the purposes of this study, which focuses on feasibility rather than precise oculomotor analysis, and is consistent with guidance that webcam-based systems are best used to capture broad attentional direction under realistic conditions [12, 17].

3.4 Hazard Mapping and Alignment

The hazard perception clips used in this study were sourced from a publicly available online practice resource provided by TheoryPass, which offers DVSA-style video scenarios for learner drivers [41].

To enable comparison between gaze direction and hazard location, the video display area was divided into a set of predefined spatial regions. In this study, a quadrant-based approach was used to represent coarse areas of the screen, providing robustness to noise and head-movement-induced drift that commonly affect webcam-based trackers [12, 13].

Hazard locations within each video clip were manually annotated and mapped to these regions. During playback, the system determines whether the estimated gaze position falls within the same region as the hazard at the moment a user indicates detection.

This approach allows for a simple form of attentional alignment, where gaze and hazard location are compared at a broad spatial level. The use of quadrants provides tolerance to noise and drift in gaze estimation while still allowing meaningful distinctions between attention directed toward and away from hazards, aligning with recent recommendations to avoid over-interpreting fine-grained webcam gaze data in dynamic driving tasks [18, 17].

3.5 Implementation Details

The prototype was implemented using a web-based technology stack, enabling it to run on standard desktop hardware without the need for specialised equipment. The system was developed using TypeScript to provide type safety and improve code maintainability during development.

MediaPipe was integrated into the application to handle real-time facial landmark detection, mirroring current browser-based eye tracking examples that use MediaPipe or similar facial landmark models as a backbone [32, 34]. The gaze estimation logic was implemented within the client-side application, allowing all processing to occur locally without reliance on external services, in line with privacy- and latency-motivated designs in recent scalable browser trackers [35].

The user interface was designed to be minimal and consistent with conventional hazard perception systems. Video playback, user input, and gaze estimation operate concurrently within the same environment, ensuring a seamless user experience that closely resembles existing PC-based hazard perception platforms [21, 23].

Overall, the system design demonstrates that webcam-based gaze estimation can be integrated into hazard perception tasks using accessible technologies, providing a foundation for further development of gaze-based training tools and diagnostic feedback mechanisms [42, 43].

4 Research Philosophy and Design Strategy

This study is informed by a pragmatic research philosophy, which focuses on selecting methods that best address the research problem rather than adhering to a single methodological paradigm. The aim of the research is to evaluate the feasibility and perceived usefulness of integrating webcam-based gaze estimation into hazard perception systems, building on recent work that emphasises feasibility and interpretability in applied gaze estimation [12, 17]. As such, both technical implementation and participant perception are considered equally important.

In terms of research approach, the study adopts an exploratory, proof-of-concept strategy. Rather than testing predefined hypotheses, the research seeks to investigate whether a novel combination of technologies can be applied within a specific domain, consistent with design-oriented feasibility studies in driving and training contexts [42, 13]. This aligns with an inductive approach, where insights are derived from observed system behaviour and participant feedback.

The methodological choice can be described as a mixed-methods approach. Quantitative data is collected through Likert-scale survey responses, providing measurable indicators of participant perception. This is complemented by qualitative data in the form of open-ended responses and informal discussions, which provide deeper insight into user attitudes and system limitations, in line with recommendations to interpret gaze-based systems through both quantitative and qualitative lenses [18, 19].

From a research strategy perspective, the study combines elements of design science and user evaluation. A prototype system is developed to demonstrate the technical feasibility of the concept, and this artefact is then evaluated through interaction with participants. This aligns with design-oriented research approaches, where artefact creation and evaluation are central to knowledge generation [44, 42].

The time horizon of the study is cross-sectional, as data is collected from participants during a single session rather than over an extended period. This reflects the exploratory nature of the research and the focus on initial feasibility rather than long-term behavioural change.

Overall, the research design reflects the layered structure of the research onion, progressing from a pragmatic philosophical stance through an exploratory approach, mixed-methods data collection, and a design-oriented evaluation strategy [12, 17].

4.1 Research Approach

The research adopts an exploratory proof-of-concept approach. The aim of the study is not to produce statistically generalisable behavioural findings but to evaluate whether webcam-based gaze estimation could realistically be integrated into hazard perception testing systems [12, 17].

Hazard perception assessments used in driver licensing contexts must operate at large scale and under a wide range of technical conditions. Consequently, any gaze-based augmentation must be compatible with commodity hardware and robust to variation in lighting, camera quality, and participant behaviour [12, 13]. These constraints informed the design of the prototype system developed in this study.

The research therefore focuses on two key questions:

- Whether webcam-based gaze estimation can be integrated into a hazard perception environment using widely available technology.

- Whether professionals working within the transport domain consider such a system to have potential value for hazard perception training or assessment.

By combining a prototype demonstration with structured participant feedback, the study provides an initial evaluation of the feasibility and perceived usefulness of the concept, consistent with early-stage applications of WBET in other applied domains [20, 19].

4.2 System Implementation

To explore the feasibility of the concept, a prototype hazard perception platform was developed. The prototype demonstrates how webcam-based gaze estimation could be integrated into a hazard perception testing environment using standard consumer hardware [35, 34].

Initial development attempted to integrate the WebGazer.js library, a browser-based eye-tracking framework commonly used in webcam gaze estimation research. However, during early testing the library proved difficult to stabilise within the hazard perception environment due to calibration inconsistencies associated with its continuous training mechanism, echoing previously reported challenges with interaction-dependent calibration [31, 19].

As a result, the implementation was revised to use Google’s MediaPipe framework. MediaPipe provides real-time facial landmark detection and head pose estimation using standard webcams [32]. Within the prototype system, MediaPipe was used to detect facial and eye landmarks and estimate approximate gaze direction relative to the screen, following recent work that derives gaze from MediaPipe-based landmark pipelines [33, 34].

While MediaPipe does not provide dedicated eye tracking functionality comparable to laboratory-grade systems, its facial landmark model provides sufficient spatial information to approximate gaze direction at a coarse level. For the purposes of this study, this level of accuracy is sufficient to demonstrate the potential integration of gaze-based signals within a hazard perception system, consistent with recommendations to use WBET for broad attentional indicators rather than fine-grained oculomotor analysis [12, 17].

The prototype system demonstrates how gaze estimation could operate alongside hazard perception video playback. During a hazard perception clip, the system estimates gaze direction relative to the video frame. This information can be visualised using debugging tools to illustrate how participant gaze might align with predefined hazard regions. The system therefore illustrates how gaze information could be used to augment hazard perception feedback in training or assessment contexts [42, 18].

4.3 System Design Methodology

The development of the prototype system followed an iterative and experimental design process. Given the exploratory nature of the study, the system was not developed using a rigid software development lifecycle, but instead evolved through successive refinement based on technical feasibility and observed behaviour [44, 35].

Initial implementation focused on integrating existing webcam-based gaze estimation libraries into a hazard perception environment. Early attempts using WebGazer.js highlighted challenges relating to calibration stability and interaction with dynamic video

content. These limitations informed a design pivot toward the MediaPipe framework, which provided more stable facial landmark detection and allowed for a more robust implementation [31, 32].

The system was developed incrementally, with individual components such as gaze estimation, video playback, and hazard mapping implemented and tested independently before being integrated. This modular approach allowed specific challenges, such as coordinate mapping and gaze alignment, to be addressed in isolation, reflecting patterns in other WBET feasibility studies [20, 19].

Design decisions were guided by practical constraints, including the need to operate on standard hardware and to maintain a user experience consistent with conventional hazard perception systems. As a result, the system adopts a coarse-grained approach to gaze estimation, prioritising robustness and usability over fine-grained accuracy [12, 13].

Overall, the system design process reflects an iterative prototyping methodology, where implementation decisions are continuously informed by technical limitations, user interaction requirements, and the overarching goal of evaluating feasibility.

4.4 Participants

Ten participants took part in the study. All participants were software engineers working within the UK Department for Transport (DfT). Participants were therefore familiar with digital systems and technology used within transport-related environments.

Participants were recruited through voluntary participation within the organisation. All participants were over 18 years of age and reported normal or corrected-to-normal vision.

The participant group was intentionally limited in size because the study focuses on evaluating the feasibility of the proposed concept rather than producing statistically generalisable behavioural findings. As professionals working within the transport domain, participants were well positioned to provide informed feedback regarding the potential practical value, implementation challenges, and relevance of the proposed system within real-world transport environments [1, 21, 23].

4.5 Survey Design

Following completion of the hazard perception task and the subsequent system walk-through, participants were asked to complete a structured survey designed to collect feedback on the perceived usefulness and feasibility of the approach.

The survey was implemented using Google Forms, which provided a straightforward and reliable mechanism for collecting responses. The survey included a mixture of Likert-scale questions and short-response questions.

The Likert-scale questions were designed to measure participant agreement with statements relating to:

- the perceived usefulness of gaze feedback in hazard perception training
- the potential value of gaze information within hazard perception testing
- the practicality of implementing webcam-based gaze estimation in real-world systems

Short-response questions allowed participants to provide additional comments regarding potential benefits, limitations, or concerns associated with the approach. These responses were used to capture qualitative insights that may not be reflected in structured survey responses.

This combination of quantitative and qualitative questions allowed the study to capture both measurable perceptions of usefulness and more detailed participant observations, mirroring approaches used in other applied webcam eye-tracking evaluations [20, 19, 43].

4.6 Data Collection Procedure

Participants completed the study individually using the prototype hazard perception system. Each session lasted approximately 20–30 minutes and consisted of three stages: completion of the hazard perception task, a post-task system walkthrough and discussion, and completion of a structured feedback survey.

During the first stage, participants completed a hazard perception task while the webcam-based gaze estimation functionality, implemented using MediaPipe, operated in the background. This allowed participants to gain direct experience of how the system estimated gaze direction during hazard perception video playback.

Participants interacted with the system in a manner similar to a conventional hazard perception test, observing driving scenarios and identifying developing hazards. During this stage, the gaze estimation component operated in the background, estimating gaze direction using webcam-based facial landmark tracking. No visual feedback or overlays were presented to participants during the test itself, allowing the task to more closely resemble a conventional hazard perception assessment [4, 21].

After completing the hazard perception task, a short informal interview and system walkthrough was conducted with each participant. During this stage, the technical components of the prototype were explained in more detail. Using the system’s built-in debugging tools, participants were shown how gaze estimation operated during the test. This included visualising the estimated gaze position overlay, the screen quadrants used for attentional alignment, and the mapping of hazard locations within the video frame.

This walkthrough allowed participants to better understand how the system interpreted gaze behaviour and how gaze data could potentially be integrated into hazard perception training or assessment environments. Participants were able to ask questions and provide immediate feedback during this stage, similar to debrief procedures used in prior gaze-based hazard perception studies [42, 18].

Following the discussion and system demonstration, participants were asked to complete a structured survey implemented using Google Forms. The survey collected participant feedback on the perceived usefulness, practicality, and potential applications of integrating webcam-based gaze estimation into hazard perception testing or training systems.

Participants were asked to reflect on their experience using the prototype system and the explanations provided during the walkthrough. The survey consisted primarily of Likert-scale questions along with optional short-response questions to capture additional observations.

All survey responses were automatically recorded by the Google Forms platform and exported for analysis.

Including the hazard perception task ensured that participant feedback was informed by direct interaction with the system rather than by a purely conceptual demonstration.

4.7 Data Analysis

Survey responses were analysed using a combination of descriptive statistics and qualitative analysis. Given the exploratory nature of the study and the small participant group, the analysis focused on identifying trends in participant opinion rather than performing inferential statistical analysis.

Responses to Likert-scale questions were summarised using frequency counts and percentage distributions to identify general trends in participant opinion. These results provide an overview of how participants perceived the usefulness and feasibility of the proposed approach.

Responses to open-ended questions were analysed qualitatively by reviewing participant comments and identifying recurring themes. This process allowed common observations, concerns, or suggestions to be identified across participants, consistent with qualitative analysis practices in exploratory eye-tracking research [18, 19].

Because the study is exploratory and the sample size is small, the analysis focuses on identifying patterns and insights rather than performing statistical hypothesis testing.

4.8 Methodological Considerations

Several methodological considerations influenced the design of the study, including the limitations of webcam-based gaze estimation, the exploratory nature of the research, and the use of domain-informed participants.

4.8.1 Participant Awareness of System Operation

Participants were informed about the technical operation of the prototype during the post-task walkthrough. While this explanation may influence participant perceptions of the system, it was intentionally included so that survey responses reflected an informed understanding of the system’s capabilities and limitations, which is particularly important when evaluating novel biometric technologies [29].

4.8.2 Technical Constraints of Webcam-Based Gaze Estimation

Webcam-based gaze estimation introduces several technical limitations compared to specialised infrared eye-tracking systems. These include lower frame rates, reduced spatial accuracy, and sensitivity to lighting conditions and participant movement [12]. As a result, webcam-based approaches cannot provide the same level of precision as laboratory-grade eye trackers.

In the context of this study, the aim was not to obtain precise eye movement measurements but rather to demonstrate that approximate gaze direction signals could be captured using widely available hardware. This level of accuracy is sufficient for evaluating whether gaze estimation could plausibly be incorporated into hazard perception training or testing tools [12, 17].

4.8.3 Exploratory Study Scope

The study is exploratory in nature and focuses on evaluating the feasibility of the concept rather than producing behavioural conclusions about hazard perception performance. The prototype system and survey feedback therefore provide indicative insights into the technical and practical potential of the approach.

Future work could extend this research by conducting larger studies with more diverse participant groups and by integrating gaze estimation into fully interactive hazard perception testing environments [22, 40].

4.8.4 Ethical Considerations

Ethical considerations were taken into account throughout the study. Participants were informed about the purpose of the research and the nature of the data being collected. Participation was voluntary and participants were free to withdraw from the study at any time.

Survey responses were collected anonymously using Google Forms and no personally identifiable data was stored. All data were handled in accordance with institutional data protection policies and broader guidance on the ethical use of biometric and gaze-related data [29, 43].

4.9 Summary

This chapter has described the research design used to evaluate the feasibility of integrating webcam-based gaze estimation into hazard perception testing systems. A prototype implementation was developed using MediaPipe to demonstrate how gaze estimation could operate within a hazard perception environment [32, 33]. Feedback from domain-informed participants working within the Department for Transport was collected using a structured survey. The methodology reflects the exploratory nature of the research while providing insight into both the technical feasibility and perceived usefulness of the proposed approach [12, 17].

5 Results

5.1 Overview of Responses

A total of ten participants completed the post-session survey following their interaction with the prototype hazard perception system. The survey was designed to collect participant perceptions regarding the usability of the prototype, the practicality of webcam-based gaze estimation, and the potential value of integrating gaze information into hazard perception testing or training systems [12, 17].

Responses consisted primarily of Likert-scale ratings supported by optional short-response comments. Because the study involved a relatively small number of participants and was exploratory in nature, the results are presented using descriptive statistics rather than inferential statistical testing, in line with common practice in early-stage feasibility work [20, 19]. Mean scores are used to provide a general indication of participant attitudes across different areas of the system.

The results are organised into several categories corresponding to different aspects of the system: calibration experience, usability of the interface, physical comfort and awareness of the webcam, interaction with the hazard perception task, attitudes toward gaze tracking, and overall acceptance of the technology.

5.2 Calibration and Setup Experience

Before interacting with the hazard perception system, participants completed a calibration procedure that allowed the gaze estimation model to approximate the participant’s gaze direction relative to the display. Calibration is a critical step in webcam-based gaze estimation because the system must establish a mapping between facial landmark positions and screen coordinates [14, 12].

Participants were therefore asked several questions relating to the clarity and usability of the calibration process. These questions aimed to evaluate whether the calibration stage presented any usability challenges that might affect the practicality of deploying the system in real-world environments, reflecting concerns raised in prior webcam-based tracking research [15, 17].

Question	Mean Score (1–5)
Calibration instructions were clear	4.6
Calibration was quick and easy	4.7
Confidence calibration worked correctly	3.4

Table 1: Participant perceptions of the calibration process

The results indicate that participants generally found the calibration process easy to complete. The question relating to the clarity of the calibration instructions received a mean score of 4.6, suggesting that most participants felt that the instructions provided sufficient guidance to complete the process successfully.

Similarly, participants reported that the calibration procedure itself was quick and straightforward, with a mean score of 4.7. This suggests that the calibration stage did not introduce significant delays or complexity before participants could begin the hazard

perception task. In the context of real-world deployment, a calibration process that is perceived as simple and efficient is important because lengthy or confusing setup procedures can discourage user engagement [20, 19].

However, the question relating to participant confidence in the calibration accuracy received a lower mean score of 3.4. This indicates that while participants were comfortable completing the calibration process, they were somewhat less confident that the resulting gaze estimation would be fully accurate.

This result reflects a broader theme that emerged across several survey responses. Participants generally appeared to support the concept of gaze estimation within hazard perception systems but expressed some uncertainty regarding the reliability of webcam-based tracking compared with dedicated eye-tracking hardware, a concern echoed in comparative evaluations of WBET and infrared systems [12, 13].

The difference between perceived ease of use and perceived reliability suggests that participants viewed the system as technically promising but recognised that further refinement may be required before such technology could be relied upon in formal assessment contexts [18, 43].

5.3 System Usability

Participants were asked several questions relating to the usability of the prototype hazard perception system. These questions aimed to evaluate how easily participants could interact with the interface and whether the system allowed them to complete the hazard perception task without difficulty. Usability is an important consideration when evaluating prototype systems, as complex or unintuitive interfaces may influence participant behaviour and reduce the practicality of real-world deployment [42, 21].

The usability questions focused on three key aspects of interaction with the system: the overall ease of use of the interface, the participant’s confidence while using the system, and whether the interface allowed hazards to be identified and responded to quickly.

Question	Mean Score (1–5)
The system was easy to use	4.9
I felt confident using the system	4.3
The interface allowed quick responses	4.6

Table 2: Participant perceptions of system usability

The results indicate that participants generally found the prototype system highly usable. The question relating to overall ease of use received the highest mean score of 4.9, suggesting that participants were able to understand and interact with the interface with minimal difficulty. This indicates that the prototype successfully replicated the basic interaction model used in conventional hazard perception tests [4, 21].

Participants also reported relatively high confidence when using the system, with a mean score of 4.3. This suggests that participants felt comfortable navigating the interface and interacting with the hazard perception task. Although slightly lower than the ease-of-use score, this value still indicates a positive overall perception of the system.

The ability to respond quickly to hazards is an important aspect of hazard perception testing, as the timing of hazard identification forms a key component of such assess-

ments. The mean score of 4.6 for the question relating to response speed suggests that participants did not feel constrained by the interface when indicating hazards.

Overall, these results suggest that the prototype interface did not introduce usability barriers that could interfere with participant interaction during the hazard perception task. Participants were able to focus primarily on identifying hazards rather than navigating the interface itself.

5.4 Physical Comfort and Webcam Awareness

Because the prototype system relied on a webcam to estimate gaze direction, participants were also asked about their physical comfort during the task and whether the presence of the webcam influenced their behaviour. Webcam-based gaze estimation systems must operate unobtrusively if they are to be used effectively in real-world environments. If users feel uncomfortable or overly aware of the webcam, this may alter their natural behaviour during hazard perception tasks [12, 17].

Participants were therefore asked to rate their level of physical comfort while using the system, whether they were able to sit naturally during the task, and whether the webcam itself caused any distraction.

Question	Mean Score (1–5)
I felt physically comfortable during the task	4.6
I was able to sit naturally while using the system	4.0
The webcam distracted me during the task	1.1

Table 3: Participant perceptions of physical comfort and webcam awareness

The results indicate that participants generally felt comfortable while completing the hazard perception task. The question relating to physical comfort received a mean score of 4.6, suggesting that the system did not require participants to adopt unusual or restrictive positions while interacting with the interface. This is an important consideration when evaluating systems intended for widespread use, as uncomfortable setups may discourage user engagement.

Participants also reported that they were generally able to sit naturally while using the system, with a mean score of 4.0. Although slightly lower than the overall comfort score, this still indicates that participants did not feel significantly constrained by the webcam-based gaze estimation component.

Notably, the question relating to webcam distraction produced a very low mean score of 1.1. This suggests that participants were largely unaware of the webcam during the task and did not feel that it interfered with their ability to concentrate on identifying hazards. From a usability perspective, this is a desirable outcome, as gaze estimation systems intended for training or assessment should ideally operate without drawing attention away from the primary task [42, 43].

Overall, these results suggest that the presence of the webcam did not negatively affect participant comfort or behaviour during the hazard perception task.

5.5 Hazard Perception Task Experience

Participants were asked several questions relating to their experience completing the hazard perception task using the prototype system. These questions aimed to evaluate whether the task was clear and understandable, and whether the interface supported timely responses when participants identified developing hazards.

Ensuring that participants clearly understand the hazard perception task is important when evaluating prototype systems. If participants find the task confusing or difficult to follow, this may affect their responses and reduce the reliability of the feedback provided during the survey [4, 21].

Question	Mean Score (1–5)
The hazards were clear to identify	4.3
I understood when to click to indicate a hazard	4.3
The interface allowed me to respond quickly	4.6

Table 4: Participant perceptions of the hazard perception task

The results suggest that participants generally found the hazard perception scenarios clear and understandable. The question relating to hazard clarity received a mean score of 4.3, indicating that participants were able to recognise developing hazards within the video scenarios without significant confusion.

Participants also reported that they understood when to click to indicate a hazard, which also received a mean score of 4.3. This suggests that the interaction model used in the prototype system successfully replicated the basic behaviour expected in conventional hazard perception tests, where users indicate the moment they believe a hazard is developing [4, 6].

The ability to respond quickly to hazards is an important aspect of hazard perception testing, as reaction timing is typically used as a measure of hazard awareness. The relatively high score of 4.6 for the question relating to response speed indicates that participants felt the interface allowed them to react promptly when identifying hazards.

Overall, these results suggest that the prototype system successfully reproduced the basic interaction model used in traditional hazard perception tests. Participants were able to focus on identifying hazards within the driving scenarios without being hindered by the interface or task design.

5.6 Attitudes Toward Gaze Tracking

Participants were also asked several questions relating to their attitudes toward the use of gaze estimation within hazard perception systems. These questions aimed to evaluate whether participants believed that gaze information could provide useful insight into driver behaviour and whether they felt comfortable with such technology being incorporated into hazard perception training or assessment environments.

Question	Mean Score (1–5)
It is reasonable to consider where someone is looking	4.4
I trust gaze tracking at a broad level	3.3
I would be comfortable using this for training	4.9
I would be comfortable using this in formal assessment	4.8

Table 5: Participant attitudes toward gaze-based hazard perception systems

The results suggest that participants were generally supportive of the concept of incorporating gaze information into hazard perception systems. The question relating to whether it is reasonable to consider where someone is looking received a mean score of 4.4, indicating that participants broadly agreed that gaze direction could provide meaningful information about how individuals perceive hazards within driving scenarios, consistent with prior work linking gaze patterns to hazard perception performance [7, 10].

However, participants expressed somewhat lower confidence in the reliability of gaze estimation itself. The question relating to trust in gaze tracking received a mean score of 3.3, which suggests that participants were more cautious when considering the accuracy of webcam-based gaze estimation technologies. While participants appeared receptive to the concept, this result indicates that some uncertainty remains regarding the precision of the technology, mirroring concerns raised in earlier webcam-based tracking and driver monitoring studies [12, 17, 13].

Participants reported particularly strong acceptance of the technology when considered in a training context. The question relating to comfort using the system for training received a mean score of 4.9, indicating that participants viewed gaze-based feedback as potentially useful for improving hazard perception skills, in line with prior findings that gaze feedback can support more effective scanning strategies [9, 43].

Similarly, participants reported high levels of acceptance when asked whether they would be comfortable using such technology within formal assessment environments. The mean score of 4.8 suggests that participants were generally open to the idea of gaze-based systems being integrated into hazard perception testing, provided that the technology operates reliably.

Overall, these results indicate that participants viewed gaze estimation as a promising addition to hazard perception systems, particularly for training purposes. However, the lower trust score highlights that perceptions of technological reliability may influence how such systems are adopted in practice.

5.7 Qualitative Feedback

In addition to Likert-scale questions, participants were invited to provide open-ended feedback about their experience using the prototype system. These responses provided qualitative insight into how participants perceived the system, the limitations they observed, and how the concept might be improved in future implementations.

Several common themes emerged from participant responses.

Calibration and Tracking Limitations. Participants were asked whether they noticed any factors that may have affected calibration accuracy. While three participants reported no issues, others identified factors such as head movement, webcam positioning, lighting conditions, and the presence of glasses as potential influences on tracking performance. These responses are consistent with known limitations of webcam-based gaze

estimation systems, which can be sensitive to lighting variation, head pose changes, and reflective surfaces such as eyeglass lenses [12, 14, 45].

Perceived Realism of the Hazard Perception Task. Participants were also asked how the prototype hazard perception task compared with existing DVSA hazard perception tests. Four participants reported that the task felt approximately the same as the official test format, while three participants considered the prototype to be slightly or significantly more realistic. Only one participant indicated that it felt slightly less realistic, and two participants reported that they were unsure. Overall, the responses suggest that participants perceived the prototype system as broadly comparable to existing hazard perception testing environments, aligning with prior work showing that desktop and simulator-based tasks can approximate formal hazard perception assessments [4, 8, 42].

Interpretation of Gaze and Click Mismatch. Participants were asked to explain potential reasons why the system’s estimated gaze location might not align with the moment a hazard was identified. The most common explanation provided was that the gaze tracking itself may be inaccurate, which was mentioned by four participants. Three participants suggested that peripheral vision could allow hazards to be detected without directly looking at them, while others suggested timing differences between when the hazard was perceived and when the click response occurred. These responses indicate that participants were aware of both technological and behavioural factors that could influence gaze alignment, reflecting similar discussions in eye-tracking studies of driving and hazard perception [18, 10, 30].

Value of Gaze-Based Feedback. Participants were also asked whether they would find it useful to receive feedback about where the system estimated they were looking during hazard perception scenarios. All ten participants responded positively to this question, indicating that gaze-based feedback could provide useful insight into their hazard perception behaviour. Several participants suggested that such feedback could help drivers understand whether they were scanning the road environment effectively, echoing prior findings that gaze feedback can support training in simulated driving tasks [9, 42, 21].

Participants also proposed several types of feedback that could be incorporated into such a system. These included visual heatmaps of gaze distribution, screenshots highlighting gaze location when hazards appear, quadrant-based indicators, and timeline views showing gaze position relative to hazard emergence, all of which are consistent with visualisation techniques used in existing gaze-analysis tools [18, 19].

Suggested Improvements. Participants also suggested several improvements to the prototype system. The most common suggestions related to improving gaze tracking accuracy, making the calibration process more robust, allowing greater head movement during tracking, and improving support for users wearing glasses. Some participants also suggested minor interface improvements such as full-screen video playback and clearer scoring feedback. These suggestions align with technical and usability challenges identified in other WBET implementations [12, 31, 34].

Overall, the qualitative responses were strongly positive toward the concept of integrating gaze estimation into hazard perception systems. Participants generally viewed the system as easy to use and believed that gaze-based feedback could provide valuable insight for training purposes, supporting the view that webcam-based gaze estimation can play a useful role in applied training and assessment contexts despite its limitations [17, 23].

6 Discussion

6.1 Overview of Findings

This study set out to evaluate the feasibility of integrating webcam-based gaze estimation into a hazard perception testing framework. The results presented in the previous chapter provide insight into both the usability of the prototype system and participant attitudes toward the use of gaze information within hazard perception environments [12, 17].

Overall, participants reported a high level of usability when interacting with the prototype system. The interface was perceived as easy to use, and participants were able to complete the hazard perception task without difficulty. In addition, participants reported minimal distraction from the webcam component, suggesting that webcam-based gaze estimation can operate unobtrusively within a hazard perception context, in line with findings from other desktop and simulator-based eye-tracking studies [42, 43]. One participant did recommend removing the calibration check screen to further obfuscate the tracking system; however, this would require a very high level of confidence in the calibration and is not feasible with the current prototype, where many participants recalibrated after their first attempt due to inaccuracy on the check screen.

Participants also expressed strong support for the concept of incorporating gaze information into hazard perception systems. All participants indicated that they would find gaze-based feedback useful, particularly in a training context. This suggests that there is clear potential for gaze estimation to enhance hazard perception training by providing additional insight into where attention is directed during driving scenarios, echoing prior work showing that gaze feedback can support more effective scanning behaviour [9, 21].

However, despite this strong conceptual acceptance, participants expressed lower confidence in the accuracy of the gaze estimation itself. This indicates a distinction between acceptance of the idea and trust in the underlying technology. Participants appeared willing to engage with gaze-based systems but remained cautious about the reliability of webcam-based tracking, particularly in scenarios where such data might influence formal assessment outcomes, a concern consistent with comparisons between webcam-based and infrared systems [12, 13].

Qualitative feedback further supported these findings, with participants identifying factors such as lighting conditions, head movement, and the use of glasses as potential sources of inaccuracy. These limitations align with those identified in existing research on WBET, reinforcing the importance of considering technical constraints when evaluating the feasibility of such systems [14, 45].

Taken together, these findings suggest that webcam-based gaze estimation is a promising addition to hazard perception systems, particularly for training applications. However, concerns regarding accuracy and robustness highlight the need for further development before such systems could be confidently deployed in high-stakes assessment contexts [17, 18].

6.2 Interpretation of Key Results

The results of this study highlight several important themes relating to the feasibility and potential application of webcam-based gaze estimation within hazard perception systems. These themes centre around system usability, participant acceptance of gaze-based feedback, and concerns regarding the reliability of the underlying technology.

Usability and System Integration. Participants consistently reported high usability scores, indicating that the prototype system was easy to interact with and did not introduce barriers to completing the hazard perception task. This is a significant finding in the context of hazard perception testing, where the interface must not interfere with the user’s ability to detect and respond to hazards [4, 21]. The high usability scores suggest that gaze estimation can be integrated into hazard perception environments without negatively affecting the user experience.

The finding that participants reported minimal distraction from the webcam component is particularly important. For gaze-based systems to be viable in real-world contexts, they must operate unobtrusively. The results suggest that webcam-based tracking can function in the background without altering user behaviour, which supports its potential use in both training and assessment settings [42, 12].

Acceptance of Gaze-Based Feedback. A key finding of this study is the strong participant support for gaze-based feedback. All participants indicated that they would find it useful to receive feedback about where the system estimated they were looking during hazard perception tasks. This suggests that gaze estimation has clear potential as a tool for enhancing driver training, especially when combined with rich visualisations such as heatmaps and timelines [18, 19].

From a theoretical perspective, this aligns with models of situational awareness, which emphasise the importance of attention allocation in perceiving and responding to hazards [24, 25]. Providing feedback on gaze behaviour may help individuals better understand how they attend to relevant elements within the driving environment, potentially improving hazard perception performance over time [7, 10].

Acceptance Versus Trust in Technology. Despite strong support for the concept, participants expressed lower confidence in the accuracy of the gaze estimation itself. This distinction between acceptance of the concept and trust in the technology is a central finding of the study and mirrors wider concerns about user trust in imperfect sensing technologies [12, 17].

Participants appeared willing to engage with gaze-based systems, particularly in training contexts, but were more cautious when considering their use in formal assessment. This suggests that perceived reliability plays a key role in determining whether such systems are considered appropriate for high-stakes applications, much as it does in driver monitoring and safety-critical systems [38, 43].

Perceived Causes of Gaze Mismatch. Participants frequently attributed discrepancies between gaze estimation and hazard identification to limitations in the tracking system rather than to their own behaviour. This suggests that participants were aware of the limitations of webcam-based tracking and were cautious in interpreting its outputs.

Interestingly, some participants also identified behavioural factors such as peripheral vision and timing differences between perception and response. This highlights the complexity of interpreting gaze data in hazard perception contexts, where attention and action do not always align directly [18, 10, 30].

Implications for Training Versus Assessment. Taken together, the results suggest that webcam-based gaze estimation is particularly well suited to training applications. Participants were highly receptive to the idea of receiving gaze-based feedback as a means of improving their hazard perception skills, which aligns with prior evidence that targeted hazard perception training can improve performance [21, 22].

However, the lower levels of trust in tracking accuracy indicate that caution would be required if such systems were to be used in formal assessment contexts. In such scenarios,

inaccuracies in gaze estimation could lead to unfair or misleading evaluations of driver performance, especially when hazards may be detected through peripheral vision or when gaze does not directly coincide with the response location [18, 12].

Overall, these findings suggest that while webcam-based gaze estimation shows strong potential as a training tool, further improvements in accuracy and robustness would be required before it could be reliably used within high-stakes testing environments.

6.3 Comparison with Existing Literature

The findings of this study align with several key themes identified within existing hazard perception and eye-tracking research. In particular, the importance of visual attention in hazard perception, the differences between novice and experienced drivers, and the limitations of WBET are all reflected in the results.

Previous research has consistently demonstrated that effective hazard perception is closely linked to how drivers allocate visual attention within the driving environment [7, 4]. Studies have shown that experienced drivers exhibit more efficient scanning patterns and are more likely to fixate on potential hazards earlier than novice drivers. The strong participant support for gaze-based feedback observed in this study is consistent with these findings, as it suggests that individuals recognise the importance of where attention is directed during hazard perception tasks [10, 28].

The results also align with theoretical models of situational awareness. Endsley’s model [24] emphasises the role of perception as the first stage in understanding dynamic environments. In the context of hazard perception, this involves identifying relevant elements within the visual field. The positive response to gaze-based feedback suggests that participants intuitively understand the value of monitoring attentional focus as part of hazard detection and decision-making processes.

Similarly, Wickens’ Multiple Resource Theory [26] highlights the limitations of cognitive resources and the need to allocate attention effectively across competing stimuli. The interest in feedback relating to gaze behaviour may reflect an awareness that attention is a limited resource and that improving how it is distributed could enhance hazard perception performance [6, 39].

The distinction observed in this study between acceptance of gaze tracking as a concept and lower confidence in its accuracy is also consistent with existing research on WBET. While recent studies have shown that webcam-based systems can approach the performance of laboratory-based systems under certain conditions [12], they remain sensitive to environmental factors such as lighting, head movement, and occlusion [14, 45]. The qualitative responses in this study, which identified glasses, lighting, and posture as sources of inaccuracy, directly reflect these known limitations.

Furthermore, the finding that participants often attributed mismatches between gaze estimation and hazard identification to technological limitations rather than their own behaviour is notable. This suggests that users may be cautious in trusting automated interpretations of attentional data, particularly when the system lacks the precision of dedicated eye-tracking hardware, echoing broader concerns around the interpretation of gaze metrics in applied contexts [18, 29].

Finally, the strong preference for using gaze tracking as a training tool rather than a strict assessment mechanism is consistent with prior critiques of hazard perception testing. Traditional DVSA-style hazard perception tests have been criticised for their limited ecological validity and reliance on simplified response measures [2, 8]. The results of

this study suggest that gaze-based systems may offer a way to enrich training by providing additional insight into attentional behaviour, without necessarily replacing existing assessment methods.

Overall, the findings of this study broadly support prior work showing strong links between gaze behaviour and hazard perception but do so under less controlled and more assessment-like conditions than many infrared-based studies, which strengthens the case for considering webcam-based methods as a pragmatic, lower-resolution alternative in training-focused environments [17, 13].

6.4 Implications for Hazard Perception Systems

The findings of this study have several implications for the design and application of hazard perception systems, particularly within the context of driver training and licensing. The results suggest that webcam-based gaze estimation has clear potential to enhance existing hazard perception approaches, although its role may differ depending on whether it is used for training or formal assessment.

One of the most significant implications is the potential use of gaze estimation as a training tool. Participants expressed strong interest in receiving feedback about where they were looking during hazard perception tasks, with all participants indicating that such feedback would be useful. This suggests that gaze-based systems could be used to provide learners with additional insight into their attentional behaviour, helping them to understand whether they are effectively scanning the driving environment [9, 42, 21]. Such feedback could complement existing hazard perception training methods, which typically rely on response timing alone [4, 22].

In contrast, the findings indicate that the use of gaze estimation within formal assessment contexts would require more careful consideration. Although participants were generally open to the idea of gaze-based systems being incorporated into hazard perception testing, lower levels of trust in tracking accuracy suggest that such systems may not yet be suitable for high-stakes evaluation. Inaccuracies in gaze estimation could lead to incorrect interpretations of driver behaviour, particularly in cases where hazards are detected using peripheral vision or where gaze does not directly align with the point of response [18, 12].

These findings support a distinction between formative and summative use of gaze-based systems. In a formative context, such as training, approximate gaze information may be sufficient to provide meaningful feedback without the risk of penalising users unfairly. In a summative context, such as driver licensing assessments, higher levels of accuracy and reliability would be required before gaze-based measures could be incorporated into scoring frameworks [40, 43].

The results also highlight the importance of system design considerations when integrating gaze estimation into hazard perception environments. Participants suggested a range of potential feedback mechanisms, including heatmaps, gaze overlays, and timeline visualisations. These features could be used to present gaze information in an accessible and interpretable manner, allowing users to reflect on their performance without requiring detailed technical knowledge of eye-tracking systems [18, 19].

From a broader perspective, the findings suggest that gaze-based hazard perception systems could contribute to more comprehensive models of driver behaviour. By combining traditional response-based measures with information about attentional focus, it may be possible to develop richer assessments of hazard perception that account for both

what drivers respond to and how they visually engage with the driving environment [7, 28].

Overall, the results indicate that webcam-based gaze estimation has strong potential as a tool for enhancing hazard perception training. However, its application within formal assessment contexts remains dependent on improvements in accuracy, robustness, and user trust [12, 17].

6.5 Limitations

While this study provides useful insight into the feasibility of integrating webcam-based gaze estimation into hazard perception systems, several limitations must be acknowledged.

Firstly, the study involved a relatively small sample size of ten participants. As a result, the findings are not intended to be statistically generalisable to a wider population. Instead, the results provide an exploratory indication of how such a system may be perceived by users, consistent with other early-stage WBET feasibility studies [20, 19]. A larger and more diverse participant group would be required to draw broader conclusions about user acceptance and system effectiveness.

Secondly, all participants were software engineers working within the Department for Transport. While this group was well positioned to provide informed and technically aware feedback, it does introduce a potential bias. Participants may have been more familiar with digital systems and more receptive to emerging technologies than the general population. As such, their responses may not fully reflect the perceptions of typical learner drivers or members of the public [1, 23].

Another limitation relates to the use of webcam-based gaze estimation itself. As identified in both the quantitative and qualitative results, the system is sensitive to environmental and user-related factors such as lighting conditions, head movement, and the presence of glasses. These factors can reduce tracking accuracy and introduce uncertainty into the interpretation of gaze data [12, 14, 45]. While the study deliberately adopted a coarse-grained approach to gaze estimation, these limitations still affect the reliability of the system.

The study is also limited by its focus on participant perception rather than objective performance measures. No quantitative data was collected relating to hazard detection accuracy or reaction time. As a result, the study does not assess whether gaze-based feedback improves hazard perception performance, but instead evaluates whether such an approach is perceived as useful and feasible. Future work will therefore need to incorporate performance metrics to determine the impact of gaze-based feedback on driving-related outcomes [22, 40].

Finally, the prototype system was evaluated within a controlled environment using the researcher’s machine. This limits the ability to assess how the system would perform across a wider range of hardware configurations, camera qualities, and environmental conditions. Real-world deployment may introduce additional variability that was not captured within this study [13, 36].

These limitations highlight that while the study provides valuable insight into the feasibility of the approach, further research would be required to evaluate system performance, robustness, and user acceptance at scale.

6.6 Future Work

The findings of this study highlight several areas for further development, both in terms of improving the technical implementation of webcam-based gaze estimation and expanding its application within hazard perception training systems.

A key area for future work is the exploration of alternative gaze estimation approaches beyond the MediaPipe framework used in this study. While MediaPipe provided a stable and accessible baseline for estimating gaze direction, it does not offer dedicated eye-tracking capabilities and is limited in terms of precision [32, 33]. Future work could investigate specialised webcam-based gaze estimation models or hybrid approaches that combine facial landmark detection with machine learning-based gaze prediction. Such approaches may improve spatial accuracy and robustness, particularly under varying lighting conditions and head movements [35, 34].

Improving the calibration process represents another important area for development. Although participants found the calibration procedure easy to complete, confidence in its accuracy was lower. Future systems could incorporate adaptive or continuous calibration techniques that update the gaze model dynamically during use, as explored in other webcam-based tracking work [31, 20]. Additionally, calibration procedures could be refined to better account for variations in head position, camera angle, and the presence of glasses. Providing real-time feedback on calibration quality may also help users achieve more reliable tracking performance [12, 45].

The results of this study strongly support the use of gaze estimation as a training tool rather than as a strict assessment mechanism. Future work should therefore focus on the development of a dedicated training platform that leverages gaze data to provide meaningful and actionable feedback to users [21, 22].

One potential direction is the implementation of a post-task feedback dashboard. Following each hazard perception clip, users could be presented with a summary of their performance, including where they were looking during key moments in the scenario. This could be supported by visualisations such as heatmaps that highlight areas of visual attention across the duration of the clip [18, 19].

To further enhance usability, users could be given the ability to review individual clips interactively. By scrubbing through the video timeline, users could observe how their gaze behaviour evolved over time and how it aligned with the emergence of hazards. This would allow users to identify missed hazards or delayed responses and understand how their visual attention contributed to these outcomes.

In addition to individual feedback, future systems could incorporate comparative metrics. For example, aggregated gaze data from multiple users could be used to generate baseline or reference gaze patterns. Individual users could then compare their gaze behaviour against these reference patterns to identify areas where their visual scanning differs from expected or optimal behaviour. Such comparisons may provide valuable insight into how attention is distributed during hazard perception tasks [10, 28].

Finally, future work could evaluate the impact of gaze-based feedback on hazard perception performance. This would involve conducting larger-scale studies that measure whether providing users with feedback on their visual attention leads to measurable improvements in hazard detection accuracy or response timing. Such studies would help determine whether gaze-based training systems can contribute to improved driver safety outcomes, addressing calls in the literature to link attentional measures more directly to performance and crash risk [22, 40].

Overall, future research should focus on improving the accuracy and robustness of webcam-based gaze estimation while developing user-centred feedback mechanisms that translate gaze data into meaningful training insights [17, 23].

6.7 Reflections on the Project

From a personal and professional perspective, the project underscored the importance of balancing research ambition with technical feasibility when working with emerging tools such as webcam-based gaze estimation. A substantial amount of time was ultimately devoted to stabilising tracking and adapting the study design around practical constraints, which highlighted the value of iterating on small, testable components rather than assuming that existing frameworks would transfer directly into an assessment-style environment. The experience also reinforced the benefits of combining technical prototyping with early, domain-informed user feedback, as participants' perceptions of usefulness, trust, and realism significantly shaped how the concept should evolve beyond this initial proof of concept.

7 Conclusions and Recommendations

This dissertation investigated the feasibility of integrating webcam-based gaze estimation into a hazard perception testing framework. The research was motivated by the limitations of existing hazard perception assessments, which rely primarily on response timing and provide limited insight into how users allocate visual attention during driving scenarios [4, 8]. By combining a prototype implementation with participant feedback, the study explored both the technical viability of webcam-based gaze estimation and its perceived usefulness within hazard perception contexts [12, 17].

The results demonstrate that webcam-based gaze estimation can be successfully integrated into a hazard perception system using standard hardware. Participants were able to complete the task without difficulty, and the system was perceived as easy to use and non-intrusive [42, 20]. Importantly, the presence of the webcam did not significantly affect participant comfort or behaviour, suggesting that such systems can operate unobtrusively within realistic testing environments [12, 43].

A key finding of the study is the strong support for gaze-based feedback. All participants indicated that they would find it useful to receive information about where they were looking during hazard perception tasks. This highlights a clear opportunity to enhance hazard perception training by providing users with additional insight into their attentional behaviour, consistent with evidence that gaze feedback can improve scanning and hazard anticipation [9, 21]. Participants recognised that understanding where attention is directed could help improve hazard detection and overall driving awareness [7, 10].

However, the study also identified a critical distinction between acceptance of the concept and trust in the technology. While participants were highly receptive to the idea of gaze-based systems, confidence in the accuracy of webcam-based gaze estimation was more limited. Factors such as lighting conditions, head movement, and the use of glasses were identified as potential sources of inaccuracy, reflecting known limitations of WBET [12, 14, 45]. This suggests that while the concept is well received, the reliability of the underlying technology remains a key barrier to adoption, particularly in high-stakes assessment contexts [13, 18].

These findings support the conclusion that webcam-based gaze estimation is currently best suited to training applications rather than formal assessment. In training environments, approximate gaze information can provide meaningful feedback without the risk of penalising users unfairly [21, 22]. In contrast, the use of gaze data within licensing or assessment frameworks would require significantly higher levels of accuracy and robustness to ensure fairness and reliability, comparable to expectations for in-vehicle monitoring and assessment tools [40, 43].

From a broader perspective, this study contributes to the understanding of how gaze estimation technologies can be applied within hazard perception systems. It demonstrates that even coarse-grained gaze information can be perceived as valuable by users and can complement existing response-based measures [17, 18]. By combining behavioural responses with attentional data, future systems may be able to provide a more comprehensive view of driver awareness and decision-making processes [7, 28].

The study is exploratory in nature and has several limitations, including a small and domain-specific participant group and a focus on perceived usefulness rather than objective performance outcomes. However, these limitations are consistent with the proof-of-concept approach adopted in this research and with other early-stage evaluations of

webcam-based gaze systems [20, 19]. The findings provide a foundation for future work that can build on both the technical implementation and the evaluation of gaze-based feedback mechanisms.

Future development should focus on improving the accuracy and robustness of webcam-based gaze estimation, as well as designing user-centred feedback systems that translate gaze data into actionable insights. In particular, the development of interactive training tools incorporating features such as gaze heatmaps, timeline-based analysis, and reference gaze patterns offers a promising direction for enhancing hazard perception training [18, 19, 23].

In conclusion, this study demonstrates that webcam-based gaze estimation is a feasible and well-received addition to hazard perception systems [12, 17]. While current limitations restrict its use in formal assessment, the strong participant support for gaze-based feedback highlights its potential as a valuable training tool. With further development, gaze-based systems may play an important role in improving driver awareness and contributing to road safety outcomes [22, 40].

References

- [1] Department for Transport. *Reported road casualties in Great Britain: younger driver factsheet, 2024*. <https://www.gov.uk/government/statistics/reported-road-casualties-great-britain-older-and-younger-driver-factsheets-2024/reported-road-casualties-in-great-britain-younger-driver-factsheet-2024>. 2024.
- [2] E. Papazikou et al. “Unpacking the relationship between task complexity and driving risk: Insights from a UK on-road trial”. In: *IATSS Research* 49.2 (July 2025), pp. 127–136. DOI: 10.1016/j.iatssr.2025.03.003.
- [3] S. Payani, H. Hamid, and T. Hua Law. “A review on impact of human factors on road safety with special focus on hazard perception and risk-taking among young drivers”. In: *IOP Conference Series: Earth and Environmental Science*. Institute of Physics Publishing, Nov. 2019. DOI: 10.1088/1755-1315/357/1/012041.
- [4] D. L. Fisher, A. P. Pollatsek, and A. Pradhan. “Can novice drivers be trained to scan for information that will reduce their likelihood of a crash?” In: *Injury Prevention* (2006). DOI: 10.1136/ip.2006.012021.
- [5] Y. B. Wang, W. Zhang, and G. Salvendy. “A comparative study of two hazard handling training methods for novice drivers”. In: *Traffic Injury Prevention* 11.5 (2010), pp. 483–491. DOI: 10.1080/15389588.2010.489242.
- [6] S. E. Lee et al. “Detection of road hazards by novice teen and experienced adult drivers”. In: *Transportation Research Record* 2078 (2008), pp. 26–32. DOI: 10.3141/2078-04.
- [7] G. Underwood et al. “Visual attention while driving: Sequences of eye fixations made by experienced and novice drivers”. In: *Ergonomics* 46.6 (May 2003), pp. 629–646. DOI: 10.1080/0014013031000090116.
- [8] M. Meocci et al. “Driving Simulator for Road Safety Design: A Comparison between Virtual Reality Tests and On-Field Tests”. In: *Journal of Information Technology in Construction* 29 (2024), pp. 1005–1025. DOI: 10.36680/j.itcon.2024.044.
- [9] A. K. Mackenzie and J. M. Harris. “Eye movements and hazard perception in active and passive driving”. In: *Visual Cognition* 23.6 (July 2015), pp. 736–757. DOI: 10.1080/13506285.2015.1079583.
- [10] X. Li, Y. Zhang, and J. Chen. “Exploration of the Relationships between Hazard Perception and Eye-Movement Indicators in Young Drivers”. In: *Journal of Advanced Transportation* 2021 (2021), p. 6642999. DOI: 10.1155/2021/6642999. URL: <https://onlinelibrary.wiley.com/doi/10.1155/2021/6642999>.
- [11] B. Hosp et al. “RemoteEye: An open-source high-speed remote eye tracker: Implementation insights of a pupil- and glint-detection algorithm for high-speed remote eye tracking”. In: *Behavior Research Methods* 52.3 (June 2020), pp. 1387–1401. DOI: 10.3758/s13428-019-01305-2.
- [12] T. Kaduk et al. “Webcam eye tracking close to laboratory standards: Comparing a new webcam-based system and the EyeLink 1000”. In: *Behavior Research Methods* 56.5 (Aug. 2024), pp. 5002–5022. DOI: 10.3758/s13428-023-02237-8.

- [13] P. Alves, A. Sousa, and M. Pereira. “Comparing Eye-Tracking System Effectiveness in Field and Driving Simulator Studies”. In: *The Open Transportation Journal* 17 (2023), e187444782301191. DOI: 10.2174/1874447802317010191. URL: <https://www.opentransportationjournal.com/VOLUME/17/ELOCATOR/e187444782301191/FULLTEXT/>.
- [14] C. Zheng and T. Usagawa. “A Rapid Webcam-Based Eye Tracking Method for Human Computer Interaction”. In: *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, Oct. 2018, pp. 133–136. DOI: 10.1109/ICCAIS.2018.8570532.
- [15] H. Skovsgaard et al. “Evaluation of a remote webcam-based eye tracker”. In: *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*. New York, NY, USA: ACM, May 2011, pp. 1–4. DOI: 10.1145/1983302.1983309.
- [16] L. Falch and K. S. Lohan. “Webcam-based gaze estimation for computer screen interaction”. In: *Frontiers in Robotics and AI* 11 (2024). DOI: 10.3389/frobt.2024.1369566.
- [17] R. Sharma and A. Chakraborty. “A Review of Driver Gaze Estimation and Application in Gaze-Based Driver Assistance Systems”. In: *arXiv preprint arXiv:2307.01470* (2023). URL: <https://arxiv.org/abs/2307.01470>.
- [18] G. Kountouriotis, P. Spyridakos, and N. Merat. “Measuring Driver Perception: Combining Eye-Tracking and Road-Scene Understanding”. In: *Cognitive Research: Principles and Implications* 5.1 (2020), p. 32. DOI: 10.1186/s41235-020-00246-8. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC9136390/>.
- [19] N. Dostálová and L. Plch. “A Scoping Review of Webcam Eye Tracking in Learning and Education”. In: *Studia Paedagogica* (2023). DOI: 10.5817/SP2023-3-5.
- [20] T. Robal et al. “Webcam-based attention tracking in online learning: A feasibility study”. In: *Proceedings of the 23rd International Conference on Intelligent User Interfaces*. Association for Computing Machinery, Mar. 2018, pp. 189–197. DOI: 10.1145/3172944.3172987.
- [21] K. Jones, T. Miller, and P. Wilson. “A Brief PC-Based Hazard Prediction Training Program Improves Young Drivers’ Skills”. In: *Accident Analysis & Prevention* (2024). Online ahead of print. URL: <https://www.sciencedirect.com/science/article/pii/S1369847824000287>.
- [22] S. Taylor, M. Horswill, and F. McKenna. “The Efficacy of Hazard Perception Training and Education: A Systematic Review”. In: *Accident Analysis & Prevention* (2024). Online ahead of print. URL: <https://www.sciencedirect.com/science/article/pii/S000145752400099X>.
- [23] Esitu Solutions. *How Hazard Perception Can Identify Risk in Professional Drivers*. <https://esitusolutions.com/how-hazard-perception-can-identify-risk-in-professional-drivers/>. Accessed: 2026-04-07. 2025.
- [24] M. R. Endsley. “Toward a Theory of Situation Awareness in Dynamic Systems”. In: *Human Factors* 37.1 (Mar. 1995), pp. 32–64. DOI: 10.1518/001872095779049543.
- [25] M. R. Endsley. “Situation awareness misconceptions and misunderstandings”. In: *Journal of Cognitive Engineering and Decision Making* 9.1 (Mar. 2015), pp. 4–32. DOI: 10.1177/1555343415572631.

- [26] C. D. Wickens. “The Effects of Divided Attention on Information Processing in Manual Tracking”. In: *Journal of Experimental Psychology: Human Perception and Performance* (1976).
- [27] P. C. Wason and J. St. B. T. Evans. “Dual processes in reasoning?” In: *Cognition* 3.2 (Jan. 1974), pp. 141–154. DOI: 10.1016/0010-0277(74)90017-1.
- [28] H. Wang, Z. Liu, and P. Zhao. “A Study on Traffic Hazard Perception of Drivers on Urban Expressway Based on Eye Movement Behavior Analysis”. In: *SSRN Electronic Journal* (2025). Preprint. DOI: 10.2139/ssrn.4962297. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4962297.
- [29] M. Smith and S. Miller. “The ethical application of biometric facial recognition technology”. In: *AI & Society* 37.1 (Mar. 2022), pp. 167–175. DOI: 10.1007/s00146-021-01199-9.
- [30] J. Park, K. Choi, and S. Lee. “Understanding Hazard Recognition Behavior Using Eye-Tracking in Simulated Urban Driving”. In: *Journal of Safety Research* (2025). Online ahead of print. URL: <https://www.sciencedirect.com/science/article/pii/S1226798825002880>.
- [31] Brown University. *WebGazer.js*. <https://webgazer.cs.brown.edu/>. Accessed: 2026-03-23. 2026.
- [32] Google AI Edge. *Face Landmarker for Web*. https://ai.google.dev/edge/media-pipe/solutions/vision/face_landmarker/web_js. Accessed: 2026-03-23. 2026.
- [33] D. Putra, M. Pratama, and A. Santoso. “MediaPipe Iris and Kalman Filter for Robust Eye Gaze Tracking”. In: *Proceedings of the International Conference on Science and Information Engineering*. Atlantis Press, 2025. DOI: 10.2991/978-94-6463-410-1_16. URL: <https://www.atlantis-press.com/proceedings/icsice-24/126011300>.
- [34] N. Johnson. *How to Build Real-Time Eye Tracking in the Browser*. <https://blog.roboflow.com/build-eye-tracking-in-browser/>. Accessed: 2026-04-07. 2025.
- [35] J. Doe, A. Smith, and L. Zhang. “\webeyetrack: Scalable Eye-Tracking for the Browser via On-Device Models”. In: *arXiv preprint arXiv:2508.19544* (2025). URL: <https://arxiv.org/abs/2508.19544>.
- [36] M. Carvalho and H. Hangan. “Modelling Weather Precipitation Intensity on Surfaces in Motion with Application to Autonomous Vehicles”. In: *Sensors* 23.19 (Oct. 2023). DOI: 10.3390/s23198034.
- [37] S. El Hamdani et al. “The Impact of Physical Motion Cues on Driver Braking Performance: A Clinical Study Using Driving Simulator and Eye Tracker”. In: *Sensors* 23.1 (Jan. 2023). DOI: 10.3390/s23010042.
- [38] L. Nahar et al. “Real Time Driver Drowsiness Monitoring System by Eye Tracking Using Mean Shift Algorithm”. In: *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE, Sept. 2019, pp. 27–31. DOI: 10.1109/ICAEE48663.2019.8975593.
- [39] E. J. Nilsson et al. “Let Complexity Bring Clarity: A Multidimensional Assessment of Cognitive Load Using Physiological Measures”. In: *Frontiers in Neuroergonomics* 3 (2022), p. 787295. DOI: 10.3389/fnrgo.2022.787295.

- [40] M. Garcia, M. Rizzo, and A. Allen. “Eye-Tracking and Visual Processing Tests for Assessing Driving Ability and Hazard Perception”. In: *Accident Analysis & Prevention* (2026). Online ahead of print. URL: <https://www.sciencedirect.com/science/article/pii/S2214140526000496>.
- [41] TheoryPass. *Hazard perception practice clips*. DVSA-style online hazard perception practice resource. 2026. URL: <https://theorypass.co.uk/practice/hazard-perception> (visited on 04/07/2026).
- [42] C. Mark, A. McGowan, and L. Maguire. “Virtual Reality Simulation and Eye Tracking to Assess Hazard Perception”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* (2018). Preprint available from Ulster University Research Portal. URL: <https://pure.ulster.ac.uk/en/publications/virtual-reality-simulation-and-eye-tracking-to-assess-hazard-perc>.
- [43] Y. Liu, H. Chen, and S. Park. “A Gaze-Based Driver Distraction Countermeasure: Comparing Effects of Different Feedback Strategies”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* (2024). Online ahead of print. URL: <https://www.sciencedirect.com/science/article/pii/S1071581924001496>.
- [44] B. Anderson, C. Brown, and D. Lee. *Eye Tracking Methodology for a Static Driving Hazard Perception Task*. Tech. rep. Class project report. Clemson University, 2019. URL: <http://andrewd.ces.clemson.edu/courses/cpsc412/fall23/teams/reports/group04.pdf>.
- [45] R. Khan, P. Gupta, and S. Verma. “Webcam Based Eye Tracking for Hands-Free Navigation”. In: *International Journal for Research in Applied Science and Engineering Technology* 13.8 (2025), pp. 1–7. URL: <https://www.ijraset.com/research-paper/webcam-based-eye-tracking-for-hands-free-navigation>.

Bibliography

- [46] Long Sun and Lingsen Hua. “Effects of hazard types on drivers’ risk rating and hazard response in a video-based hazard perception task”. In: *PLoS ONE* 14 (3 Mar. 2019). ISSN: 19326203. DOI: 10.1371/journal.pone.0214226.
- [47] Seonwook Park, Adrian Spurr, and Otmar Hilliges. “Deep Pictorial Gaze Estimation”. In: 2018, pp. 741–757. DOI: 10.1007/978-3-030-01261-8_44.
- [48] David Crundall, Geoffrey Underwood, and Peter Chapman. “Driving experience and the functional field of view”. In: *Perception* 28 (9 1999), pp. 1075–1087. ISSN: 0301-0066. DOI: 10.1068/p2894.
- [49] Avinoam Borowsky, David Shinar, and Tal Oron-Gilad. “Age, skill, and hazard perception in driving”. In: *Accident Analysis & Prevention* 42 (4 July 2010), pp. 1240–1249. ISSN: 00014575. DOI: 10.1016/j.aap.2010.02.001.
- [50] Tadas Baltrusaitis et al. “OpenFace 2.0: Facial Behavior Analysis Toolkit”. In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, May 2018, pp. 59–66. ISBN: 978-1-5386-2335-0. DOI: 10.1109/FG.2018.00019.
- [51] T. K. Akshay, Nishant Mukund Pawar, and Kirti Mahajan. “Modeling response time and hazard identification in a video-based hazard perception test for car drivers”. In: *International Journal of Transportation Science and Technology* (2025). ISSN: 20460449. DOI: 10.1016/j.ijtst.2025.02.012.
- [52] Yu Zhang et al. “The effects of visual and cognitive distractions on operational and tactical driving behaviors”. In: *Human Factors* 56 (3 2014), pp. 592–604. ISSN: 15478181. DOI: 10.1177/0018720813495279.
- [53] Tanmay Jain et al. “Real-Time Webcam-Based Eye Tracking for Gaze Estimation: Applications and Innovations”. In: *2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024*. Institute of Electrical and Electronics Engineers Inc., 2024. ISBN: 9798350370249. DOI: 10.1109/ICCCNT61001.2024.10724037.
- [54] Alessandro Bruno et al. “Detecting colour vision deficiencies via Webcam-based Eye-tracking: A case study”. In: *Eye Tracking Research and Applications Symposium (ETRA)*. Association for Computing Machinery, May 2023. ISBN: 9798400701504. DOI: 10.1145/3588015.3590133.
- [55] August Romeo et al. “Cognitive Vergence Recorded with a Webcam-Based Eye-Tracker during an Oddball Task in an Elderly Population”. In: *Sensors* 24 (3 Feb. 2024). ISSN: 14248220. DOI: 10.3390/s24030888.
- [56] Liz Burton, William Albert, and Mark Flynn. “A comparison of the performance of webcam vs. infrared eye tracking technology”. In: *Proceedings of the Human Factors and Ergonomics Society*. Vol. 2014-January. Human Factors and Ergonomics Society Inc., 2014, pp. 1437–1441. ISBN: 9780945289456. DOI: 10.1177/1541931214581300.

A System Screenshots

The following screenshots illustrate the key screens and states of the hazard perception test prototype.

A.1 Welcome Screen

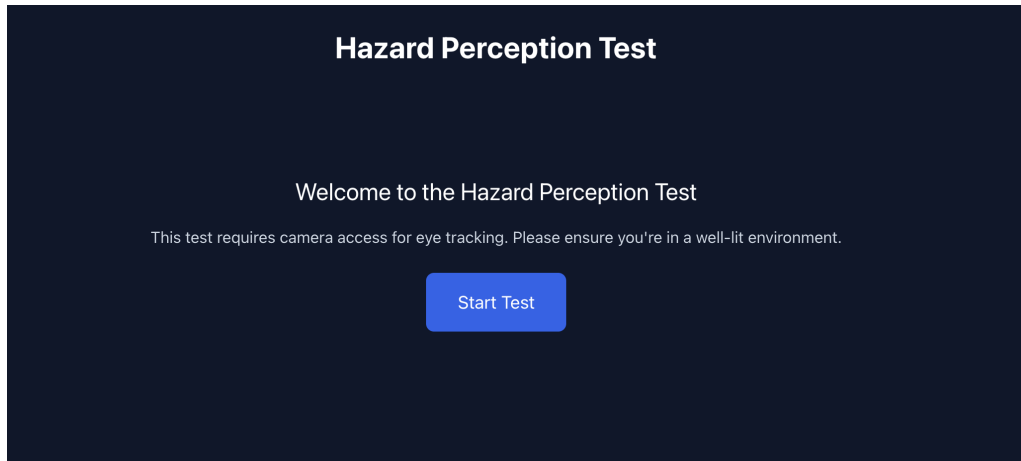


Figure 2: The initial setup screen prompting the user to start the test.

A.2 Instructions Screen

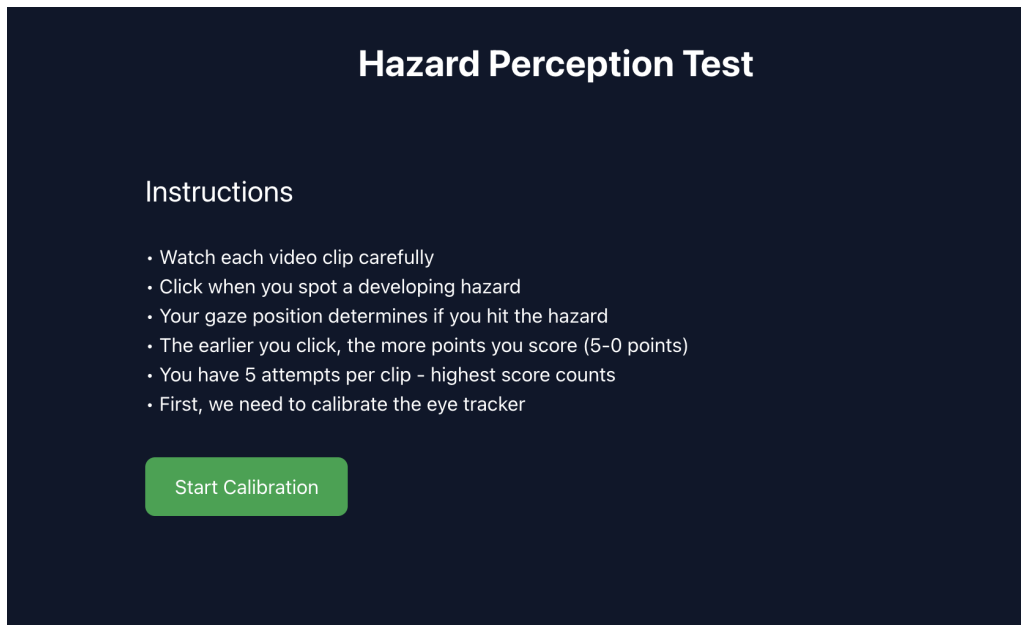


Figure 3: Instructions presented to the user before calibration.

A.3 Gaze Calibration

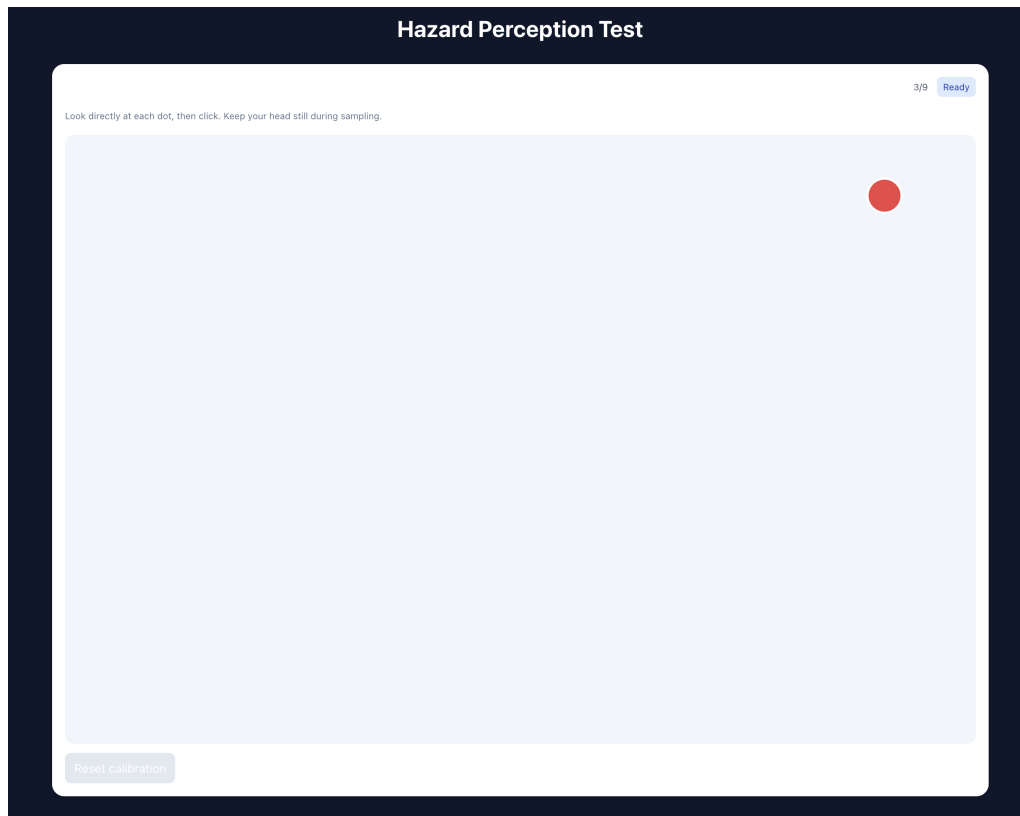


Figure 4: The 9-point gaze calibration interface.

A.4 Calibration Validation

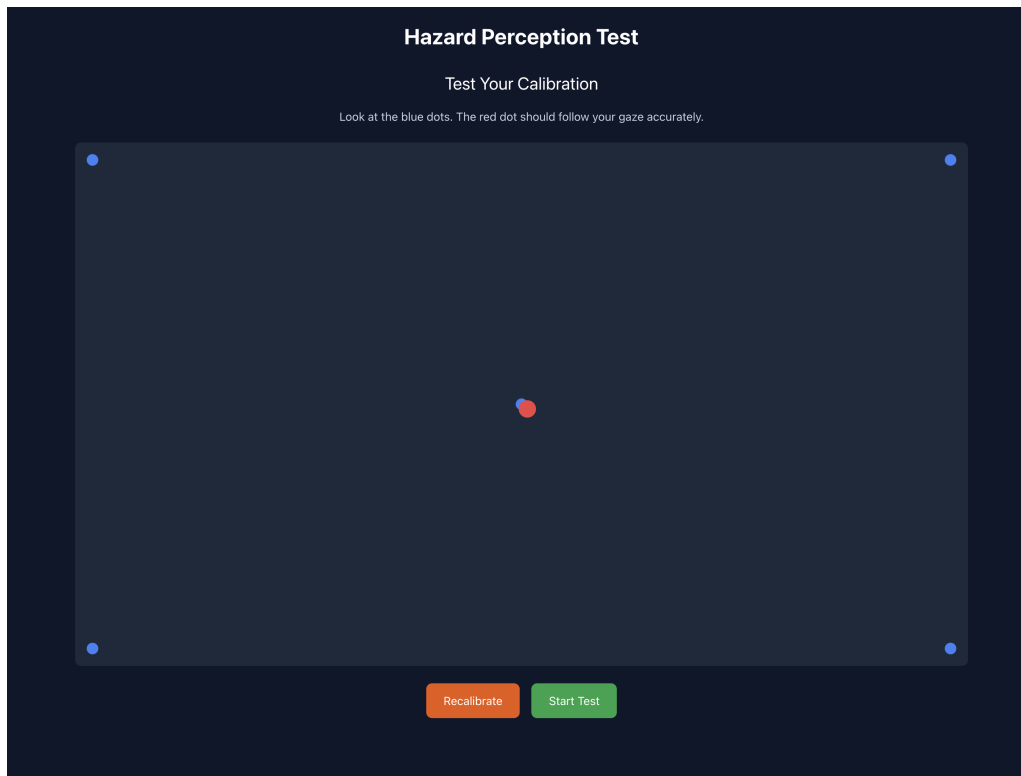


Figure 5: The validation screen showing the gaze dot tracking the user's eye position against reference points.

A.5 Test Phase

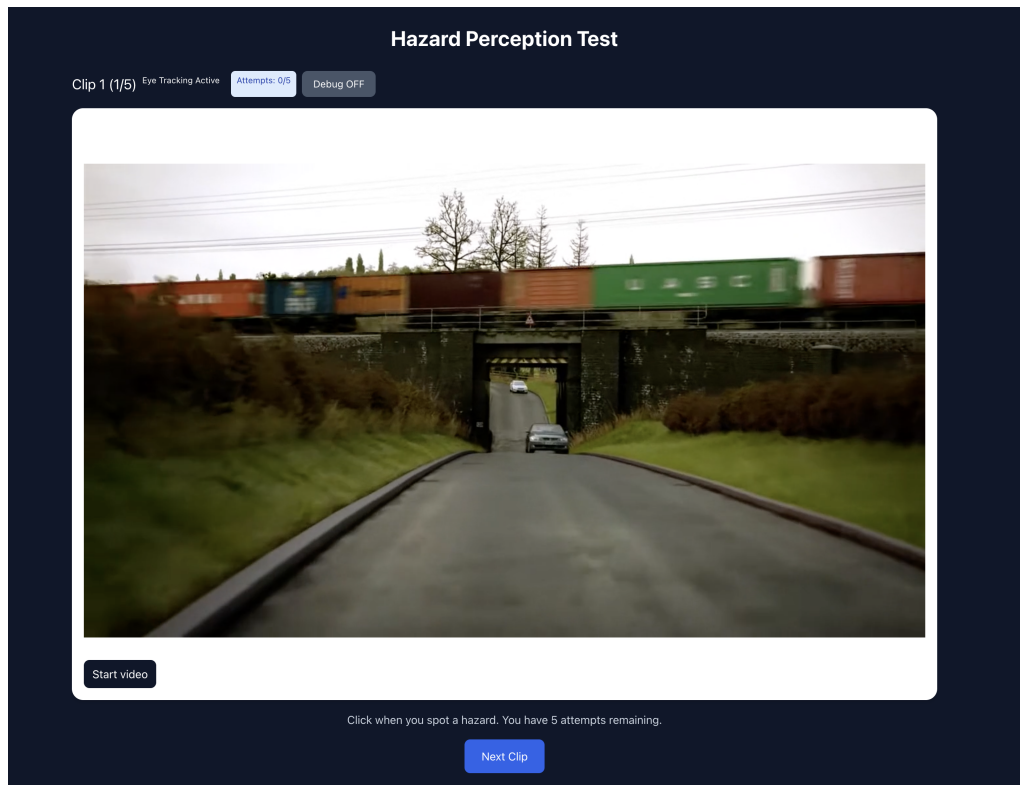


Figure 6: A hazard perception clip during the test phase.

A.6 Debug Mode

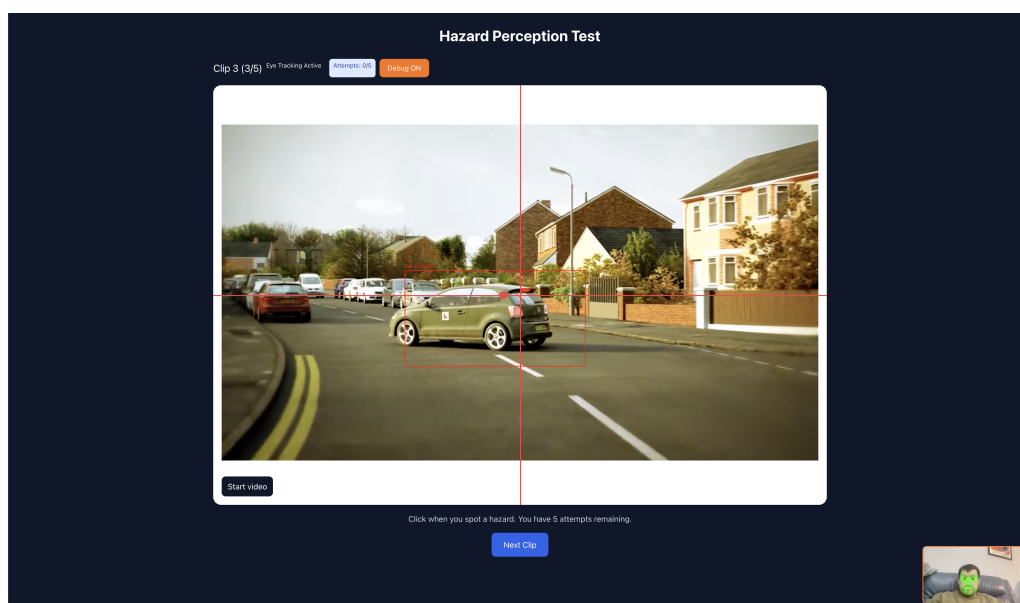


Figure 7: Debug mode showing the gaze overlay, hazard AOI bounding boxes, and camera preview.

A.7 Results Screen

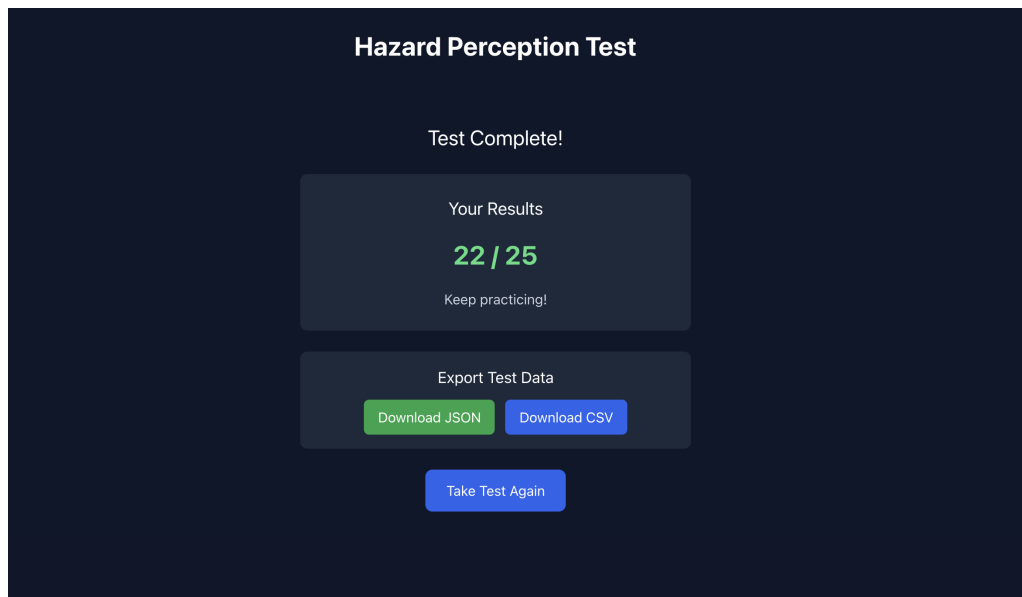


Figure 8: The results screen displaying the user's score and data export options.

A.8 AoI Marker Tool

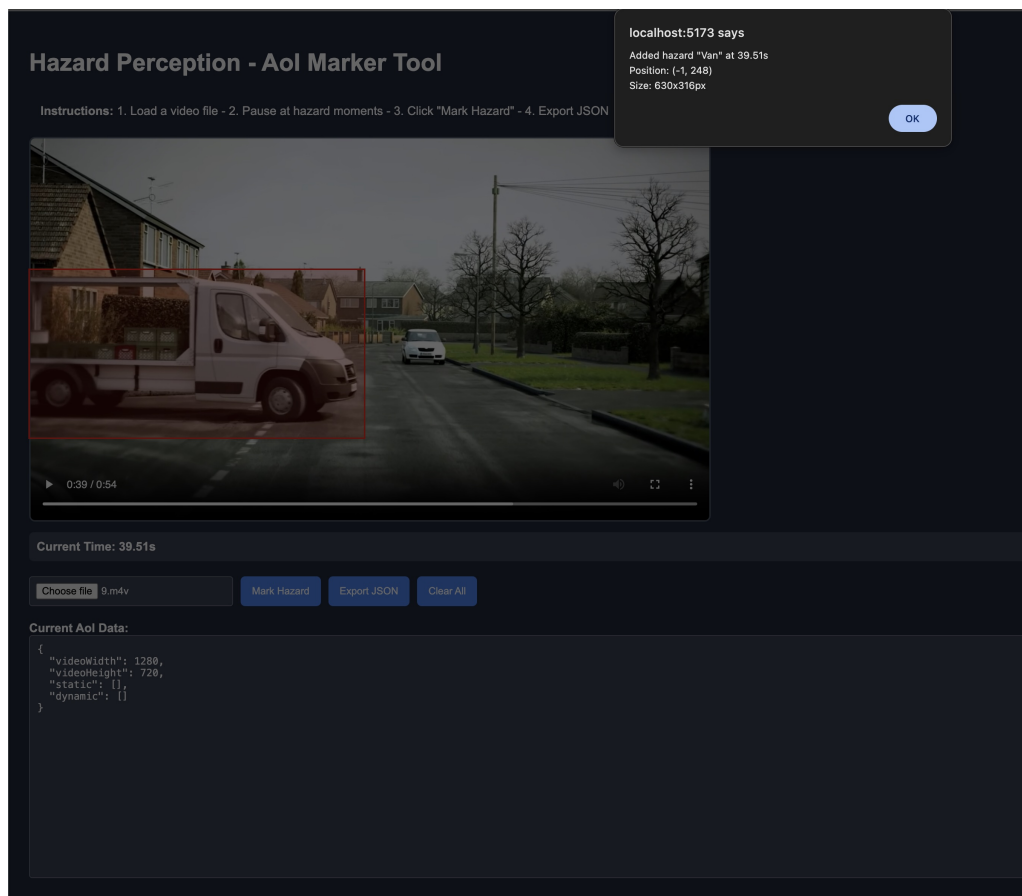


Figure 9: The standalone AoI Marker Tool used to define hazard regions on video clips.

B Source Code

B.1 Application Entry Point

```
1 import React from 'react'
2 import ReactDOM from 'react-dom/client'
3 import App from './App'
4 import '../styled-system/styles.css'
5
6 ReactDOM.createRoot(document.getElementById('root')).render(
7   <React.StrictMode>
8     <App />
9   </React.StrictMode>
10 )
```

Listing 1: main.tsx – Application bootstrap

B.2 Type Definitions

```
1
2 export type GazePoint = {
3   x: number
4   y: number
5   t: number
6   conf?: number
7 }
```

Listing 2: types.ts – Shared type definitions

B.3 Main Application Component

```
1 import React from 'react'
2 import { css } from '../styled-system/css'
3 import { initMP, getLastLandmarks, getCameraStream } from '@gaze/mediapipe'
4 import { getCalibratedGaze } from '@utils/gazeUtils'
5 import StatusBadge from './components/StatusBadge'
6 import Calibration from './components/Calibration'
7 import VideoPlayer from './components/VideoPlayer'
8 import { dataCapture, TestSession } from '@utils/dataCapture'
9
10 type TestPhase = 'setup' | 'instructions' | 'calibration' | 'validation' | 'test' | 'results'
11 type TestClip = { id: number; path: string; name: string }
12
13 const TEST_CLIPS: TestClip[] = [
14   { id: 1, path: '/clips/done/1.m4v', name: 'Clip 1' },
15   { id: 2, path: '/clips/done/2.m4v', name: 'Clip 2' },
16   { id: 3, path: '/clips/done/3.m4v', name: 'Clip 3' },
17   { id: 4, path: '/clips/done/4.m4v', name: 'Clip 4' },
18   { id: 5, path: '/clips/done/5.m4v', name: 'Clip 5' }
19 ]
20
21 function getVideoDisplayBounds(video: HTMLVideoElement) {
22   const rect = video.getBoundingClientRect()
23   const videoAspect = video.videoWidth / video.videoHeight
24   const containerAspect = rect.width / rect.height
25
26   let displayWidth, displayHeight, offsetX, offsetY
27
28   if (containerAspect > videoAspect) {
29     displayHeight = rect.height
30     displayWidth = displayHeight * videoAspect
31     offsetX = (rect.width - displayWidth) / 2
32     offsetY = 0
33   } else {
```

```

34     displayWidth = rect.width
35     displayHeight = displayWidth / videoAspect
36     offsetX = 0
37     offsetY = (rect.height - displayHeight) / 2
38 }
39
40     return { displayWidth, displayHeight, offsetX, offsetY }
41 }
42
43 export default function App() {
44     const [phase, setPhase] = React.useState<TestPhase>('setup')
45     const [camReady, setCamReady] = React.useState(false)
46     const [calibrated, setCalibrated] = React.useState(false)
47     const [currentClip, setCurrentClip] = React.useState(0)
48     const [clipScores, setClipScores] = React.useState<{ clipId: number; score:
49         number; maxScore: number }[]>([])
50     const [clicks, setClicks] = React.useState<{ time: number; score: number }[]>([])
51     const [debugMode, setDebugMode] = React.useState(false)
52     const [currentGaze, setCurrentGaze] = React.useState<{ x: number; y: number } |
53         null>(null)
54     const [sessionData, setSessionData] = React.useState<TestSession | null>(null)
55     const [aoiData, setAoiData] = React.useState<any>(null)
56
57     const previewRef = React.useRef<HTMLVideoElement>(null)
58     const overlayCanvasRef = React.useRef<HTMLCanvasElement>(null)
59     const videoRef = React.useRef<HTMLVideoElement>(null)
60     const hazardOverlayRef = React.useRef<HTMLCanvasElement>(null)
61     const validationRef = React.useRef<HTMLDivElement>(null)
62
63     async function startCamera() {
64         try {
65             const originalLog = console.log
66             console.log = (...args) => {
67                 if (typeof args[0] === 'string' && args[0].includes('TensorFlow Lite'))
68                     return
69                 originalLog(...args)
70             }
71
72             await new Promise(resolve => setTimeout(resolve, 100))
73             await initMP()
74
75             console.log = originalLog
76             await new Promise(r => setTimeout(r, 1000))
77             setCamReady(true)
78             const session = dataCapture.startSession()
79             setSessionData(session)
80             setPhase('instructions')
81         } catch (error) {
82             console.error('Camera initialization failed:', error)
83             alert('Camera initialization failed: ${error instanceof Error ? error.
84                 message : 'Unknown error'}. Please check camera permissions and try
85                 again.')
86             setCamReady(false)
87             setPhase('setup')
88         }
89     }
90
91     function handleCalibrated() {
92         setCalibrated(true)
93         setPhase('validation')
94     }
95
96     async function handleVideoClick() {
97         const video = videoRef.current
98         if (!video || !calibrated) return
99
100         const currentTime = video.currentTime
101
102         // Check if already have 5 attempts
103         if (clicks.length >= 5) return
104
105         let score = 0
106         let hitHazard = false

```

```

102     let hazardId: string | undefined
103
104     try {
105         const gaze = await getCalibratedGaze()
106         if (!gaze || !aoiData) return
107
108         const rect = video.getBoundingClientRect()
109         const { displayWidth, displayHeight, offsetX, offsetY } =
            getVideoDisplayBounds(video)
110
111         const relativeX = gaze.x - rect.left - offsetX
112         const relativeY = gaze.y - rect.top - offsetY
113
114         const videoX = (relativeX / displayWidth) * aoiData.videoWidth
115         const videoY = (relativeY / displayHeight) * aoiData.videoHeight
116
117         const midX = aoiData.videoWidth / 2
118         const midY = aoiData.videoHeight / 2
119
120         let gazeQuadrant = ''
121         if (videoX < midX && videoY < midY) gazeQuadrant = 'top-left'
122         else if (videoX >= midX && videoY < midY) gazeQuadrant = 'top-right'
123         else if (videoX < midX && videoY >= midY) gazeQuadrant = 'bottom-left'
124         else gazeQuadrant = 'bottom-right'
125
126         for (const hazard of aoiData.dynamic || []) {
127             const frames = hazard.frames.sort((a, b) => a.t - b.t)
128
129             let activeFrame = null
130
131             // Check if between two frames for interpolation
132             for (let i = 0; i < frames.length - 1; i++) {
133                 const frame1 = frames[i]
134                 const frame2 = frames[i + 1]
135
136                 if (currentTime >= frame1.t && currentTime <= frame2.t) {
137                     const ratio = (currentTime - frame1.t) / (frame2.t - frame1.t)
138                     activeFrame = {
139                         x: frame1.x + (frame2.x - frame1.x) * ratio,
140                         y: frame1.y + (frame2.y - frame1.y) * ratio,
141                         w: frame1.w + (frame2.w - frame1.w) * ratio,
142                         h: frame1.h + (frame2.h - frame1.h) * ratio,
143                         t: currentTime
144                     }
145                     break
146                 }
147             }
148
149             // If not between frames, check tolerance around each frame
150             if (!activeFrame) {
151                 for (const frame of frames) {
152                     if (Math.abs(currentTime - frame.t) < 2.5) {
153                         activeFrame = frame
154                         break
155                     }
156                 }
157             }
158
159             if (activeFrame) {
160                 // Check which quadrants the hazard overlaps
161                 const hazardQuadrants = []
162                 const left = activeFrame.x < midX
163                 const right = activeFrame.x + activeFrame.w > midX
164                 const top = activeFrame.y < midY
165                 const bottom = activeFrame.y + activeFrame.h > midY
166
167                 if (left && top) hazardQuadrants.push('top-left')
168                 if (right && top) hazardQuadrants.push('top-right')
169                 if (left && bottom) hazardQuadrants.push('bottom-left')
170                 if (right && bottom) hazardQuadrants.push('bottom-right')
171
172                 if (hazardQuadrants.includes(gazeQuadrant)) {

```

```

173         hitHazard = true
174         hazardId = hazard.id
175
176         // Find the earliest hazard frame time for this hazard
177         const earliestFrame = Math.min(...hazard.frames.map(f => f.t)
178         )
179         const timeDiff = currentTime - earliestFrame
180
181         if (timeDiff < 0) {
182             score = 0
183         } else if (timeDiff <= 1) {
184             score = 5
185         } else if (timeDiff <= 2) {
186             score = 4
187         } else if (timeDiff <= 3) {
188             score = 3
189         } else if (timeDiff <= 4) {
190             score = 2
191         } else if (timeDiff <= 5) {
192             score = 1
193         } else {
194             score = 0
195         }
196
197         break
198     }
199 }
200
201 setClicks(prev => [...prev, { time: currentTime, score }])
202
203 // Store click in session
204 const session = dataCapture.getCurrentSession()
205 if (session) {
206     const currentClip = session.testResults[session.testResults.length -
207     1]
208     if (currentClip) {
209         currentClip.clicks.push({
210             timestamp: Date.now(),
211             videoTime: currentTime,
212             score,
213             hitHazard,
214             hazardId
215         })
216     }
217 }
218 } catch {
219     // scoring failed
220 }
221 }
222
223 const isTransitioning = React.useRef(false)
224
225 function nextClip() {
226     if (isTransitioning.current) return
227
228     isTransitioning.current = true
229
230     const clipId = TEST_CLIPS[currentClip].id
231     const highestScore = Math.max(...clicks.map(click => click.score), 0)
232
233     dataCapture.endClip(highestScore)
234     setClipScores(prev => [...prev, { clipId, score: highestScore, maxScore: 5
235     }])
236
237     if (currentClip < TEST_CLIPS.length - 1) {
238         setCurrentClip(currentClip + 1)
239         setClicks([])
240         dataCapture.startClip(TEST_CLIPS[currentClip + 1].id, TEST_CLIPS[
241             currentClip + 1].name)
242         isTransitioning.current = false
243     } else {

```

```

242         const finalSession = dataCapture.endSession()
243         setSessionData(finalSession || sessionData)
244         setPhase('results')
245     }
246 }
247
248 function downloadJSON() {
249     const json = dataCapture.exportAsJSON()
250     if (!json) return
251
252     const blob = new Blob([json], { type: 'application/json' })
253     const url = URL.createObjectURL(blob)
254     const a = document.createElement('a')
255     a.href = url
256     a.download = `hpt_session_${sessionData?.sessionId}.json`
257     a.click()
258     URL.revokeObjectURL(url)
259 }
260
261 function downloadCSV() {
262     const csv = dataCapture.exportAsCSV()
263     if (!csv) return
264
265     const blob = new Blob([csv], { type: 'text/csv' })
266     const url = URL.createObjectURL(blob)
267     const a = document.createElement('a')
268     a.href = url
269     a.download = `hpt_session_${sessionData?.sessionId}.csv`
270     a.click()
271     URL.revokeObjectURL(url)
272 }
273
274 React.useEffect(() => {
275     if (phase === 'test' && currentClip === 0) {
276         dataCapture.startClip(TEST_CLIPS[0].id, TEST_CLIPS[0].name)
277     }
278 }, [phase])
279
280 React.useEffect(() => {
281     if (phase === 'test') {
282         const clipId = TEST_CLIPS[currentClip].id
283         fetch(`/clips/done/${clipId}.aois.json`)
284             .then(res => res.json())
285             .then(data => setAoiData(data))
286             .catch(e => console.log('Failed to load AOI data:', e))
287     }
288 }, [phase, currentClip])
289
290 React.useEffect(() => {
291     if (!debugMode || phase !== 'test' || !videoRef.current || !aoiData) return
292
293     let animationId: number
294     const drawHazards = () => {
295         const video = videoRef.current
296         const canvas = hazardOverlayRef.current
297
298         if (!video || !canvas) return
299
300         const ctx = canvas.getContext('2d')
301         if (!ctx) return
302
303         const rect = video.getBoundingClientRect()
304
305         canvas.style.left = `${rect.left}px`
306         canvas.style.top = `${rect.top}px`
307         canvas.style.width = `${rect.width}px`
308         canvas.style.height = `${rect.height}px`
309
310         const videoAspect = video.videoWidth / video.videoHeight
311         const containerAspect = rect.width / rect.height
312
313         let displayWidth, displayHeight, offsetX, offsetY

```

```

315         if (containerAspect > videoAspect) {
316             displayHeight = rect.height
317             displayWidth = displayHeight * videoAspect
318             offsetX = (rect.width - displayWidth) / 2
319             offsetY = 0
320         } else {
321             displayWidth = rect.width
322             displayHeight = displayWidth / videoAspect
323             offsetX = 0
324             offsetY = (rect.height - displayHeight) / 2
325         }
326
327         canvas.width = rect.width
328         canvas.height = rect.height
329         ctx.clearRect(0, 0, canvas.width, canvas.height)
330
331         const scaleX = displayWidth / aoiData.videoWidth
332         const scaleY = displayHeight / aoiData.videoHeight
333         const currentTime = video.currentTime
334
335         for (const hazard of aoiData.dynamic || []) {
336             const frames = hazard.frames.sort((a, b) => a.t - b.t)
337             let activeFrame = null
338
339             for (let i = 0; i < frames.length - 1; i++) {
340                 const frame1 = frames[i]
341                 const frame2 = frames[i + 1]
342
343                 if (currentTime >= frame1.t && currentTime <= frame2.t) {
344                     const ratio = (currentTime - frame1.t) / (frame2.t - frame1.t)
345
346                     activeFrame = {
347                         x: frame1.x + (frame2.x - frame1.x) * ratio,
348                         y: frame1.y + (frame2.y - frame1.y) * ratio,
349                         w: frame1.w + (frame2.w - frame1.w) * ratio,
350                         h: frame1.h + (frame2.h - frame1.h) * ratio
351                     }
352                     break
353                 }
354             }
355
356             if (!activeFrame) {
357                 for (const frame of frames) {
358                     if (Math.abs(currentTime - frame.t) < 2.5) {
359                         activeFrame = frame
360                         break
361                     }
362                 }
363             }
364
365             if (activeFrame) {
366                 ctx.strokeStyle = 'rgba(255, 0, 0, 0.8)'
367                 ctx.lineWidth = 2
368                 ctx.strokeRect(
369                     offsetX + activeFrame.x * scaleX,
370                     offsetY + activeFrame.y * scaleY,
371                     activeFrame.w * scaleX,
372                     activeFrame.h * scaleY
373                 )
374
375                 ctx.fillStyle = 'rgba(255, 0, 0, 0.9)'
376                 ctx.font = '12px Arial'
377                 ctx.fillText(
378                     hazard.id,
379                     offsetX + activeFrame.x * scaleX,
380                     offsetY + activeFrame.y * scaleY - 5
381                 )
382             }
383
384             for (const hazard of aoiData.static || []) {
385                 ctx.strokeStyle = 'rgba(255, 165, 0, 0.8)'
386                 ctx.lineWidth = 2

```

```

387         ctx.strokeRect(
388             offsetX + hazard.x * scaleX,
389             offsetY + hazard.y * scaleY,
390             hazard.w * scaleX,
391             hazard.h * scaleY
392         )
393
394         ctx.fillStyle = 'rgba(255, 165, 0, 0.9)'
395         ctx.font = '12px Arial'
396         ctx.fillText(
397             hazard.id,
398             offsetX + hazard.x * scaleX,
399             offsetY + hazard.y * scaleY - 5
400         )
401     }
402
403     animationId = requestAnimationFrame(drawHazards)
404 }
405
406 animationId = requestAnimationFrame(drawHazards)
407 return () => cancelAnimationFrame(animationId)
408 }, [debugMode, phase, aoiData])
409
410 React.useEffect(() => {
411     if (!calibrated || (phase !== 'test' && phase !== 'validation')) return
412
413     let raf = 0
414     const loop = async () => {
415         const gaze = await getCalibratedGaze()
416         if (!gaze) return
417
418         if (phase === 'validation' && validationRef.current) {
419             const rect = validationRef.current.getBoundingClientRect()
420             const boundedGaze = {
421                 x: Math.max(0, Math.min(rect.width, gaze.x - rect.left)),
422                 y: Math.max(0, Math.min(rect.height, gaze.y - rect.top))
423             }
424             setCurrentGaze(boundedGaze)
425         } else if (phase === 'test' && videoRef.current) {
426             const rect = videoRef.current.getBoundingClientRect()
427             const boundedGaze = {
428                 x: Math.max(0, Math.min(rect.width, gaze.x - rect.left)),
429                 y: Math.max(0, Math.min(rect.height, gaze.y - rect.top))
430             }
431             setCurrentGaze(boundedGaze)
432         }
433
434         raf = requestAnimationFrame(loop)
435     }
436     raf = requestAnimationFrame(loop)
437     return () => cancelAnimationFrame(raf)
438 }, [calibrated, phase])
439
440 React.useEffect(() => {
441     if (debugMode && camReady && previewRef.current) {
442         const stream = getCameraStream()
443         if (stream) {
444             previewRef.current.srcObject = stream
445             previewRef.current.play().catch(() => {})
446         }
447     }
448 }, [debugMode, camReady])
449
450 React.useEffect(() => {
451     if (!camReady || !debugMode) return
452
453     let animationId: number
454     const drawLandmarks = () => {
455         const landmarks = getLastLandmarks()
456         const canvas = overlayCanvasRef.current
457         const video = previewRef.current
458
459         if (landmarks && canvas && video) {

```

```

460     const ctx = canvas.getContext('2d')
461     if (ctx) {
462         const rect = video.getBoundingClientRect()
463         canvas.width = rect.width
464         canvas.height = rect.height
465         ctx.clearRect(0, 0, canvas.width, canvas.height)
466
467         const leftIrisCenter = {
468             x: (landmarks[33].x + landmarks[133].x) / 2,
469             y: (landmarks[33].y + landmarks[133].y) / 2
470         }
471         const rightIrisCenter = {
472             x: (landmarks[362].x + landmarks[263].x) / 2,
473             y: (landmarks[362].y + landmarks[263].y) / 2
474         }
475
476         landmarks.forEach((point) => {
477             const x = point.x * canvas.width
478             const y = point.y * canvas.height
479             ctx.fillStyle = 'rgba(0, 255, 0, 0.8)'
480             ctx.beginPath()
481             ctx.arc(x, y, 1, 0, Math.PI * 2)
482             ctx.fill()
483         })
484
485         ctx.fillStyle = 'rgba(255, 0, 0, 0.8)'
486         ctx.beginPath()
487         ctx.arc(leftIrisCenter.x * canvas.width, leftIrisCenter.y *
488             canvas.height, 2, 0, Math.PI * 2)
489         ctx.fill()
490         ctx.beginPath()
491         ctx.arc(rightIrisCenter.x * canvas.width, rightIrisCenter.y *
492             canvas.height, 2, 0, Math.PI * 2)
493         ctx.fill()
494     }
495     animationId = requestAnimationFrame(drawLandmarks)
496 }
497 animationId = requestAnimationFrame(drawLandmarks)
498 return () => cancelAnimationFrame(animationId)
499 }, [camReady, debugMode])
500
501
502 const totalScore = clipScores.reduce((sum, clip) => sum + clip.score, 0)
503 const totalMaxScore = clipScores.reduce((sum, clip) => sum + clip.maxScore, 0)
504
505 return (
506     <div className={css({ minH: '100vh', bg: 'slate.900', color: 'white' })}>
507         <div className={css({ p: 6, mx: 'auto', maxW: '7xl' })}>
508             <h1 className={css({ fontSize: '3xl', fontWeight: 'bold', mb: 6,
509                 textAlign: 'center' })}>
510                 Hazard Perception Test
511             </h1>
512
513             {phase === 'setup' && (
514                 <div className={css({ textAlign: 'center', py: 20 })}>
515                     <h2 className={css({ fontSize: '2xl', mb: 4 })}>Welcome to
516                         the Hazard Perception Test</h2>
517                     <p className={css({ mb: 6, color: 'slate.300' })}>
518                         This test requires camera access for eye tracking. Please
519                         ensure you're in a well-lit environment.
520                     </p>
521                     <button
522                         onClick={startCamera}
523                         className={css({
524                             px: 8, py: 4, bg: 'blue.600', color: 'white', rounded
525                                 : 'lg',
526                             fontSize: 'lg', _hover: { bg: 'blue.700' }
527                         })}
528                     >
529                         Start Test
530                     </button>

```

```

527     </div>
528   })
529
530   {phase === 'instructions' && (
531     <div className={css({ maxW: '2xl', mx: 'auto', py: 10 })}>
532       <h2 className={css({ fontSize: '2xl', mb: 6 })}>Instructions
533       </h2>
534       <div className={css({ space: 'y-4', mb: 8 })}>
535         <p>- Watch each video clip carefully</p>
536         <p>- Click when you spot a developing hazard</p>
537         <p>- Your gaze position determines if you hit the hazard
538         </p>
539         <p>- The earlier you click, the more points you score
540         (5-0 points)</p>
541         <p>- You have 5 attempts per clip - highest score counts
542         </p>
543         <p>- First, we need to calibrate the eye tracker</p>
544       </div>
545       <button
546         onClick={() => setPhase('calibration')}
547         className={css({
548           px: 6, py: 3, bg: 'green.600', color: 'white',
549           rounded: 'lg',
550           _hover: { bg: 'green.700' }
551         })}
552         >
553           Start Calibration
554         </button>
555     </div>
556   })
557
558   {phase === 'calibration' && (
559     <div className={css({ h: '80vh' })}>
560       <Calibration onCalibrated={handleCalibrated} />
561     </div>
562   })
563
564   {phase === 'validation' && (
565     <div>
566       <div className={css({ textAlign: 'center', mb: 6 })}>
567         <h2 className={css({ fontSize: '2xl', mb: 4 })}>Test Your
568         Calibration</h2>
569         <p className={css({ mb: 6, color: 'slate.300' })}>
570           Look at the blue dots. The red dot should follow your
571           gaze accurately.
572         </p>
573       </div>
574
575       <div ref={validationRef} className={css({ position: 'relative
576       ', h: '60vh', bg: 'slate.800', rounded: 'lg', mb: 6 })}>
577         <div className={css({ position: 'absolute', top: 4, left:
578         4, w: 4, h: 4, bg: 'blue.500', rounded: 'full' })}
579         />
580         <div className={css({ position: 'absolute', top: 4, right
581         : 4, w: 4, h: 4, bg: 'blue.500', rounded: 'full' })}
582         />
583         <div className={css({ position: 'absolute', bottom: 4,
584         left: 4, w: 4, h: 4, bg: 'blue.500', rounded: 'full'
585         })} />
586         <div className={css({ position: 'absolute', bottom: 4,
587         right: 4, w: 4, h: 4, bg: 'blue.500', rounded: 'full'
588         })} />
589         <div className={css({ position: 'absolute', top: '50%',
590         left: '50%', w: 4, h: 4, bg: 'blue.500', rounded: '
591         full', transform: 'translate(-50%, -50%)' })} />
592
593         {currentGaze && (
594           <div className={css({
595             position: 'absolute', w: 6, h: 6, bg: 'red.500',
596             rounded: 'full',
597             transform: 'translate(-50%, -50%)', zIndex: 10
598           })} style={{
599             left: currentGaze.x,

```

```

581         top: currentGaze.y
582     }} />
583     })
584 </div>
585
586 <div className={css({ textAlign: 'center', display: 'flex',
587     gap: 4, justifyContent: 'center' })}>
588     <button
589         onClick={() => setPhase('calibration')}
590         className={css({
591             px: 6, py: 3, bg: 'orange.600', color: 'white',
592             rounded: 'lg',
593             _hover: { bg: 'orange.700' }
594         })}
595     >
596         Recalibrate
597     </button>
598     <button
599         onClick={() => setPhase('test')}
600         className={css({
601             px: 6, py: 3, bg: 'green.600', color: 'white',
602             rounded: 'lg',
603             _hover: { bg: 'green.700' }
604         })}
605     >
606         Start Test
607     </button>
608 </div>
609 </div>
610 })
611
612 {phase === 'test' && (
613     <div>
614         <div className={css({ display: 'flex', justifyContent: '
615             between', alignItems: 'center', mb: 4 })}>
616             <h2 className={css({ fontSize: 'xl' })}>
617                 {TEST_CLIPS[currentClip].name} ({currentClip + 1}/{
618                     TEST_CLIPS.length})
619             </h2>
620             <div className={css({ display: 'flex', gap: 2 })}>
621                 <StatusBadge status="success">Eye Tracking Active</
622                     StatusBadge>
623                 <StatusBadge status="info">Attempts: {clicks.length
624                     }/5</StatusBadge>
625                 <button
626                     onClick={() => setDebugMode(!debugMode)}
627                     className={css({
628                         px: 4, py: 2, bg: debugMode ? 'orange.500' :
629                             'slate.600',
630                         color: 'white', rounded: 'lg', fontSize: 'sm'
631                         ,
632                         _hover: { bg: debugMode ? 'orange.600' : '
633                             slate.700' }
634                     })}
635                 >
636                     {debugMode ? 'Debug ON' : 'Debug OFF'}
637                 </button>
638                 {clicks.map((click, i) => (
639                     <StatusBadge key={i} status={click.score > 0 ? '
640                         success' : 'warning'}>
641                         {click.score} pts
642                     </StatusBadge>
643                 ))}
644             </div>
645         </div>
646
647         <div className={css({ position: 'relative', h: '70vh' })}>
648             <VideoPlayer
649                 ref={videoRef}
650                 enabled={calibrated}
651                 srcPath={TEST_CLIPS[currentClip].path}
652                 onVideoClick={handleVideoClick}
653             />

```

```

643         {debugMode && (
644             <>
645             <div className={css({
646                 position: 'absolute', top: 0, left: 0, right:
647                 0, bottom: 0,
648                 pointerEvents: 'none', zIndex: 10
649             })}>
650                 <div className={css({
651                     position: 'absolute', top: '50%', left:
652                     0, right: 0,
653                     h: '2px', bg: 'red.500'
654                 })} />
655                 <div className={css({
656                     position: 'absolute', left: '50%', top:
657                     0, bottom: 0,
658                     w: '2px', bg: 'red.500'
659                 })} />
660             </div>
661
662             <canvas
663                 ref={hazardOverlayRef}
664                 className={css({
665                     position: 'fixed',
666                     pointerEvents: 'none',
667                     zIndex: 15
668                 })}
669             />
670
671             {currentGaze && (
672                 <div className={css({
673                     position: 'absolute', zIndex: 20,
674                     pointerEvents: 'none',
675                     w: 4, h: 4, bg: 'red.500', rounded: 'full
676                     ',
677                     transform: 'translate(-50%, -50%)'
678                 })} style={{
679                     left: currentGaze.x,
680                     top: currentGaze.y
681                 }} />
682             )}
683         </>
684     )}
685
686     {debugMode && camReady && (
687         <div className={css({
688             position: 'fixed', bottom: 4, right: 4, zIndex:
689             50,
690             rounded: 'lg', overflow: 'hidden', border: '2px
691             solid', borderColor: 'orange.500'
692         })}>
693             <video
694                 ref={previewRef}
695                 autoPlay playsInline muted
696                 className={css({ w: '200px', display: 'block'
697                 , transform: 'scaleX(-1)' })}
698             />
699             <canvas
700                 ref={overlayCanvasRef}
701                 className={css({
702                     position: 'absolute', top: 0, left: 0, w:
703                     '100%', h: '100%',
704                     pointerEvents: 'none', transform: 'scaleX
705                     (-1)'
706                 })}
707             />
708         </div>
709     )}
710
711     <div className={css({ textAlign: 'center', mt: 4 })}>
712         <p className={css({ mb: 4, color: 'slate.300' })}>
713

```

```

705         Click when you spot a hazard. You have {5 - clicks.
706             length} attempts remaining.
707     </p>
708     <button
709         onClick={nextClip}
710         className={css({
711             px: 6, py: 3, bg: 'blue.600', color: 'white',
712             rounded: 'lg',
713             _hover: { bg: 'blue.700' }
714         })}
715     >
716         {currentClip < TEST_CLIPS.length - 1 ? 'Next Clip' :
717             'Finish Test'}
718     </button>
719 </div>
720 </div>
721 }}
722 {phase === 'results' && (
723     <div className={css({ textAlign: 'center', py: 10 })}>
724         <h2 className={css({ fontSize: '2xl', mb: 6 })}>Test Complete
725         !</h2>
726
727         <div className={css({ bg: 'slate.800', p: 6, rounded: 'lg',
728             maxW: 'md', mx: 'auto', mb: 6 })}>
729             <h3 className={css({ fontSize: 'xl', mb: 4 })}>Your
730             Results</h3>
731             <div className={css({ fontSize: '3xl', fontWeight: 'bold',
732                 color: 'green.400', mb: 4 })}>
733                 {totalScore} / {totalMaxScore}
734             </div>
735             <p className={css({ color: 'slate.300' })}>
736                 {totalScore >= totalMaxScore * 0.8 ? 'Excellent!' :
737                     totalScore >= totalMaxScore * 0.6 ? 'Good job!' :
738                     'Keep practicing!'}
739             </p>
740         </div>
741
742         <div className={css({ bg: 'slate.800', p: 4, rounded: 'lg',
743             maxW: 'md', mx: 'auto', mb: 6 })}>
744             <h4 className={css({ fontSize: 'lg', mb: 3 })}>Export
745             Test Data</h4>
746             <div className={css({ display: 'flex', gap: 3,
747                 justifyContent: 'center' })}>
748                 <button
749                     onClick={downloadJSON}
750                     className={css({
751                         px: 4, py: 2, bg: 'green.600', color: 'white',
752                         rounded: 'md',
753                         _hover: { bg: 'green.700' }
754                     })}
755                 >
756                     Download JSON
757                 </button>
758                 <button
759                     onClick={downloadCSV}
760                     className={css({
761                         px: 4, py: 2, bg: 'blue.600', color: 'white',
762                         rounded: 'md',
763                         _hover: { bg: 'blue.700' }
764                     })}
765                 >
766                     Download CSV
767                 </button>
768             </div>
769         </div>
770
771     <button
772         onClick={() => window.location.reload()}
773         className={css({
774             px: 6, py: 3, bg: 'blue.600', color: 'white', rounded
775             : 'lg',
776             _hover: { bg: 'blue.700' }

```

```

765         }}}}
766         >
767         Take Test Again
768     </button>
769 </div>
770   )}
771 </div>
772 </div>
773 )
774 }

```

Listing 3: App.tsx – Main application logic and test flow

B.4 MediaPipe Gaze Tracker

```

1  import type { GazePoint } from '@types'
2  import { FaceLandmarker, FilesetResolver } from '@mediapipe/tasks-vision'
3
4  let landmarker: FaceLandmarker | null = null
5  let booted = false
6
7  let cam: HTMLVideoElement | null = null
8
9  let lastXY: { x: number; y: number } | null = null
10 let lastTs = 0
11 let raf = 0
12 let lastProcessedTime = 0
13
14 let smoothK = 0.3
15 let fx: number | null = null
16 let fy: number | null = null
17
18 let lastLandmarks: Array<{x: number, y: number}> | null = null
19
20 type InitOpts = {
21   wasmRoot?: string
22   modelUrl?: string
23   smoothingK?: number
24 }
25
26 export async function initMP(opts: InitOpts = {}) {
27   if (booted) return
28   smoothK = typeof opts.smoothingK === 'number' ? opts.smoothingK : 0.3
29
30   const handleResize = () => {
31     fx = fy = null
32     lastXY = null
33   }
34
35   window.addEventListener('resize', handleResize)
36   ;(initMP as any)._resize = handleResize
37
38   const stream = await navigator.mediaDevices.getUserMedia({
39     video: {
40       facingMode: 'user',
41       width: { ideal: 1280 },
42       height: { ideal: 720 }
43     },
44     audio: false
45   })
46
47   cam = document.createElement('video')
48   cam.style.position = 'fixed'
49   cam.style.left = '-9999px'
50   document.body.appendChild(cam)
51   cam.autoplay = true
52   cam.playsInline = true
53   cam.muted = true
54   cam.srcObject = stream
55

```

```

56     await new Promise<void>((res) => {
57         const onMeta = () => { cam!.removeEventListener('loadedmetadata', onMeta);
58             res() }
59         cam!.addEventListener('loadedmetadata', onMeta)
60     })
61     await cam.play()
62
63     const wasmRoot = opts.wasmRoot ?? 'https://cdn.jsdelivr.net/npm/@mediapipe/tasks-
64         vision@latest/wasm'
65     const modelUrl = opts.modelUrl ?? 'https://storage.googleapis.com/mediapipe-
66         models/face_landmarker/face_landmarker/float16/1/face_landmarker.task'
67
68     try {
69         const fileset = await FilesetResolver.forVisionTasks(wasmRoot)
70         landmarker = await FaceLandmarker.createFromOptions(fileset, {
71             baseOptions: { modelAssetPath: modelUrl },
72             runningMode: 'VIDEO',
73             numFaces: 1
74         })
75     } catch (error) {
76         console.error('Failed to initialize MediaPipe FaceLandmarker:', error)
77         try {
78             (cam?.srcObject as MediaStream | null)?.getTracks().forEach(t => t.stop()
79                 )
80         } catch {}
81         throw new Error('MediaPipe initialization failed. Check network connection
82             and model availability.')
83     }
84
85     booted = true
86     lastXY = null
87     lastTs = 0
88     fx = fy = null
89     lastLandmarks = null
90     lastProcessedTime = 0
91
92     const vis = () => {
93         if (document.hidden) cancelAnimationFrame(raf)
94         else { cancelAnimationFrame(raf); raf = requestAnimationFrame(loop) }
95     }
96     document.addEventListener('visibilitychange', vis, { passive: true })
97     ;(initMP as any)._vis = vis
98
99     cancelAnimationFrame(raf)
100     raf = requestAnimationFrame(loop)
101 }
102
103 export async function getPrediction(maxAgeMs = 200): Promise<GazePoint | null> {
104     if (!lastXY) return null
105     const age = performance.now() - lastTs
106     if (age > maxAgeMs) return null
107     return { x: lastXY.x, y: lastXY.y, t: lastTs, conf: 0.6 }
108 }
109
110 export function resetHeadPose() {
111     // Reserved for future head-pose compensation
112 }
113
114 export function setCalibrationHeadPose() {
115     // Reserved for future calibration head-pose baseline
116 }
117
118 export function getLastLandmarks(): Array<{x: number, y: number}> | null {
119     return lastLandmarks
120 }
121
122 export function isTrackerReady(maxAgeMs = 250): boolean {
123     return !(lastTs && performance.now() - lastTs <= maxAgeMs)
124 }
125
126 export function getCameraStream(): MediaStream | null {
127     return cam?.srcObject as MediaStream | null
128 }

```

```

124
125 function loop() {
126     if (!booted || !cam || !landmarker) return
127     const t = performance.now()
128
129     if (cam.readyState >= 2) {
130         const timestamp = Math.max(t, lastProcessedTime + 1)
131         lastProcessedTime = timestamp
132
133         try {
134             const res = landmarker.detectForVideo(cam, timestamp)
135             if (res.faceLandmarks && res.faceLandmarks.length) {
136                 processLandmarks(res.faceLandmarks[0], timestamp)
137             }
138         } catch (e) {
139             console.warn('MediaPipe detection failed:', e)
140         }
141     }
142
143     raf = requestAnimationFrame(loop)
144 }
145
146 function processLandmarks(lm: Array<{x: number, y: number}>, t: number) {
147     lastLandmarks = lm
148
149     const leftIris = lm[468] || lm[33]
150     const rightIris = lm[473] || lm[362]
151
152     const screenX = ((leftIris.x + rightIris.x) / 2) * window.innerWidth
153     const screenY = ((leftIris.y + rightIris.y) / 2) * window.innerHeight
154
155     const k = smoothK
156     fx = fx == null ? screenX : (k * screenX + (1 - k) * fx)
157     fy = fy == null ? screenY : (k * screenY + (1 - k) * fy)
158
159     lastXY = { x: fx!, y: fy! }
160     lastTs = t
161 }

```

Listing 4: mediapipe.ts – Webcam-based gaze estimation via MediaPipe FaceLandmarker

B.5 Affine Calibration Transform

```

1 export type Pt = { x: number; y: number }
2 export type Affine = { a: number; b: number; c: number; d: number; e: number; f:
   number }
3
4
5 export function applyAffine(T: Affine, p: Pt): Pt {
6     return { x: T.a * p.x + T.b * p.y + T.e, y: T.c * p.x + T.d * p.y + T.f }
7 }
8
9 export function fitAffine(src: Pt[], dst: Pt[]): Affine | null {
10     const n = Math.min(src.length, dst.length)
11     if (n < 3) return null
12
13     const M = Array.from({ length: 6 }, () => Array(6).fill(0))
14     const v = Array(6).fill(0)
15
16     for (let i = 0; i < n; i++) {
17         const { x, y } = src[i]
18         const { x: u, y: w } = dst[i]
19         addOuter(M, [x, y, 0, 0, 1, 0]); addRHS(v, [x, y, 0, 0, 1, 0], u)
20         addOuter(M, [0, 0, x, y, 0, 1]); addRHS(v, [0, 0, x, y, 0, 1], w)
21     }
22
23     const theta = solve6x6(M, v)
24     if (!theta) return null
25     const [a, b, c, d, e, f] = theta
26     return { a, b, c, d, e, f }

```

```

27 }
28
29 function addOuter(M: number[][], r: number[]) {
30     for (let i = 0; i < 6; i++) for (let j = 0; j < 6; j++) M[i][j] += r[i] * r[j]
31 }
32 function addRHS(v: number[], r: number[], s: number) {
33     for (let i = 0; i < 6; i++) v[i] += r[i] * s
34 }
35
36 function solve6x6(A: number[][], b: number[]): number[] | null {
37     const n = 6
38     for (let i = 0; i < n; i++) A[i].push(b[i]) // augment
39
40     for (let col = 0; col < n; col++) {
41         let piv = col, best = Math.abs(A[col][col])
42         for (let r = col + 1; r < n; r++) {
43             const v = Math.abs(A[r][col]); if (v > best) { best = v; piv = r }
44         }
45         if (best < 1e-8) return null
46         if (piv !== col) [A[piv], A[col]] = [A[col], A[piv]]
47
48         const d = A[col][col]
49         for (let c = col; c <= n; c++) A[col][c] /= d
50
51         for (let r = 0; r < n; r++) if (r !== col) {
52             const f = A[r][col]
53             for (let c = col; c <= n; c++) A[r][c] -= f * A[col][c]
54         }
55     }
56     return A.map(row => row[n])
57 }

```

Listing 5: affine.ts – Least-squares affine transform for gaze calibration

B.6 Gaze Utilities

```

1 import { getPrediction } from '@gaze/mediapipe'
2 import { getCalibration } from '@components/Calibration'
3 import { applyAffine } from '@logic/affine'
4
5 export async function getCalibratedGaze(): Promise<{ x: number; y: number } | null> {
6     const gaze = await getPrediction()
7     if (!gaze) return null
8
9     const calib = getCalibration()
10    return calib ? applyAffine(calib, gaze) : gaze
11 }

```

Listing 6: gazeUtils.ts – Calibrated gaze helper

B.7 Calibration Component

```

1 import React from 'react'
2 import { css } from '../styled-system/css'
3 import { fitAffine, type Affine } from '@logic/affine'
4 import { getPrediction, isTrackerReady, resetHeadPose, setCalibrationHeadPose } from '@gaze/mediapipe'
5 import Card from './Card'
6 import StatusBadge from './StatusBadge'
7
8 const points = [
9     { x: 0.1, y: 0.1 }, { x: 0.5, y: 0.1 }, { x: 0.9, y: 0.1 },
10    { x: 0.1, y: 0.5 }, { x: 0.5, y: 0.5 }, { x: 0.9, y: 0.5 },
11    { x: 0.1, y: 0.9 }, { x: 0.5, y: 0.9 }, { x: 0.9, y: 0.9 }
12 ]
13
14 let _calib: Affine | null = null

```

```

15 export function getCalibration(): Affine | null { return _calib }
16 function setCalibration(T: Affine | null) { _calib = T }
17
18 async function sampleAvg(windowMs = 2000): Promise<{ x: number; y: number } | null> {
19   const tWait = performance.now()
20   while (!isTrackerReady() && performance.now() - tWait < 500) {
21     await new Promise(r => setTimeout(r, 16))
22   }
23
24   setCalibrationHeadPose()
25   await new Promise(r => setTimeout(r, 1000))
26
27   const t0 = performance.now()
28   const samples: { x: number; y: number }[] = []
29
30   while (performance.now() - t0 < windowMs) {
31     const p = await getPrediction()
32     if (p) samples.push(p)
33     await new Promise(r => setTimeout(r, 33))
34   }
35
36   if (samples.length < 30) return null
37
38   const xs = samples.map(s => s.x).sort((a, b) => a - b)
39   const ys = samples.map(s => s.y).sort((a, b) => a - b)
40
41   const q1x = xs[Math.floor(xs.length * 0.25)]
42   const q3x = xs[Math.floor(xs.length * 0.75)]
43   const iqrX = q3x - q1x
44
45   const q1y = ys[Math.floor(ys.length * 0.25)]
46   const q3y = ys[Math.floor(ys.length * 0.75)]
47   const iqrY = q3y - q1y
48
49   const filteredSamples = samples.filter(s =>
50     s.x >= q1x - 1.5 * iqrX && s.x <= q3x + 1.5 * iqrX &&
51     s.y >= q1y - 1.5 * iqrY && s.y <= q3y + 1.5 * iqrY
52   )
53
54   return {
55     x: filteredSamples.reduce((sum, s) => sum + s.x, 0) / filteredSamples.length,
56     y: filteredSamples.reduce((sum, s) => sum + s.y, 0) / filteredSamples.length
57   }
58 }
59
60
61 export default function Calibration({ onCalibrated }: { onCalibrated: () => void }) {
62   const [i, setI] = React.useState(0)
63   const [done, setDone] = React.useState(false)
64   const [sampling, setSampling] = React.useState(false)
65   const [pairs, setPairs] = React.useState<{ pred: { x: number; y: number }; true:
66     { x: number; y: number } }[]>([])
67   const [readyNow, setReadyNow] = React.useState<boolean>(isTrackerReady())
68   const boxRef = React.useRef<HTMLDivElement>(null)
69   const p = points[i]
70
71   React.useEffect(() => {
72     const id = setInterval(() => setReadyNow(isTrackerReady()), 150)
73     return () => clearInterval(id)
74   }, [])
75
76   async function handleClick() {
77     if (!readyNow || sampling) return
78
79     setSampling(true)
80     const el = boxRef.current
81     if (!el) return
82
83     const box = el.getBoundingClientRect()
84     const trueX = box.left + p.x * box.width
85     const trueY = box.top + p.y * box.height
86
87     const pred = await sampleAvg(2000)

```

```

87     setSampling(false)
88
89     if (!pred) {
90       alert('No gaze found. Look at the dot and try again.')
91       return
92     }
93
94     setPairs(prev => [...prev, { pred, true: { x: trueX, y: trueY } }])
95
96     if (i < points.length - 1) {
97       setI(i + 1)
98       return
99     }
100
101     // Fit calibration with all points
102     const src = pairs.map(pr => pr.pred).concat([pred])
103     const dst = pairs.map(pr => pr.true).concat([x: trueX, y: trueY ])
104
105     let T = fitAffine(src, dst)
106
107     // Fallback to simple translation if affine fails
108     if (!T) {
109       const avgSrcX = src.reduce((sum, p) => sum + p.x, 0) / src.length
110       const avgSrcY = src.reduce((sum, p) => sum + p.y, 0) / src.length
111       const avgDstX = dst.reduce((sum, p) => sum + p.x, 0) / dst.length
112       const avgDstY = dst.reduce((sum, p) => sum + p.y, 0) / dst.length
113
114       T = {
115         a: 1, b: 0, c: 0, d: 1,
116         e: avgDstX - avgSrcX,
117         f: avgDstY - avgSrcY
118       }
119     }
120
121     setCalibration(T)
122     setDone(true)
123     onCalibrated()
124   }
125
126   return (
127     <Card
128       title="Gaze Calibration"
129       status={
130         <div className={css({ display: 'flex', gap: 3, alignItems: 'center',
131           fontSize: 'xs' })}>
132           <StatusBadge status={readyNow ? 'success' : 'error'}>
133             tracker: readyNow ? 'ready' : 'not ready'
134           </StatusBadge>
135           <span className={css({ color: 'slate.600' })}>
136             {i + 1}/{points.length}
137           </span>
138           <StatusBadge status={done ? 'success' : sampling ? 'warning' : '
139             info'}>
140             {done ? 'Done' : sampling ? 'Sampling...' : 'Ready'}
141           </StatusBadge>
142         </div>
143       }
144     >
145       <p className={css({ fontSize: 'xs', color: 'slate.500', mb: 4 })}>
146         Look directly at each dot, then click. Keep your head still during
147         sampling.
148       </p>
149
150       <div
151         ref={boxRef}
152         className={css({
153           position: 'relative', flex: 1, bg: 'slate.100', rounded: 'xl',
154           cursor: readyNow && !sampling ? 'crosshair' : 'not-allowed',
155           overflow: 'hidden',
156           opacity: sampling ? 0.7 : 1
157         })}
158         onClick={handleClick}
159         aria-label="Calibration area"

```

```

157     >
158     {!done && i < points.length && (
159       <div
160         className={css({
161           position: 'absolute', w: '12', h: '12', rounded: 'full',
162           border: '3px solid white',
163           animation: sampling ? 'pulse 1s infinite' : undefined
164         })}
165         style={{
166           background: sampling ? 'var(--colors-orange-500)' : 'var(
167             --colors-red-500)',
168           left: `${p.x * 100}%`,
169           top: `${p.y * 100}%`,
170           transform: 'translate(-50%, -50%)'
171         }}
172       />
173     )}
174   </div>
175   <div className={css({ mt: 3, display: 'flex', gap: 3 })}>
176     <button
177       type="button"
178       onClick={() => {
179         setPairs([])
180         setCalibration(null)
181         setI(0)
182         setDone(false)
183         setSampling(false)
184         resetHeadPose()
185       }}
186       disabled={sampling}
187       className={css({
188         px: 3, py: 2, rounded: 'lg',
189         bg: 'slate.200',
190         _hover: { bg: 'slate.300' },
191         _disabled: { opacity: 0.5, cursor: 'not-allowed' }
192       })}
193     >
194       Reset calibration
195     </button>
196   </div>
197 </Card>
198 )
199 }

```

Listing 7: Calibration.tsx – 9-point gaze calibration UI

B.8 Video Player Component

```

1  import React from 'react'
2  import { css } from '../styled-system/css'
3  import Card from './Card'
4  import StatusBadge from './StatusBadge'
5
6  export default React.forwardRef<HTMLVideoElement, {
7    enabled: boolean
8    srcPath: string
9    onVideoClick?: () => void
10 }>((function VideoPlayer({ enabled, srcPath, onVideoClick }, ref) {
11   const videoRef = React.useRef<HTMLVideoElement>(null)
12   const [hasMeta, setHasMeta] = React.useState(false)
13
14   React.useImperativeHandle(ref, () => videoRef.current!, [])
15
16   React.useEffect(() => {
17     const v = videoRef.current
18     if (!v) return
19     const onMeta = () => setHasMeta(true)
20     const onEmptied = () => setHasMeta(false)
21     v.addEventListener('loadedmetadata', onMeta)

```

```

22     v.addEventListener('emptied', onEmptied)
23     return () => {
24         v.removeEventListener('loadedmetadata', onMeta)
25         v.removeEventListener('emptied', onEmptied)
26     }
27 }, [])
28
29 const handleVideoClick = () => {
30     if (!enabled) return
31     onVideoClick?.()
32 }
33
34 function handleStart() {
35     videoRef.current?.play()
36 }
37
38 return (
39     <Card
40         title="Video Player"
41         status={
42             <StatusBadge status={enabled ? 'success' : 'warning'}>
43                 {enabled ? 'Tracking active' : 'Calibration required'}
44             </StatusBadge>
45         }
46     >
47         <div className={css({ position: 'relative', flex: 1, rounded: 'xl',
48             overflow: 'hidden' })}>
49             <video
50                 ref={videoRef}
51                 className={css({ w: 'full', h: 'full', objectFit: 'contain' })}
52                 src={srcPath}
53                 preload="metadata"
54                 data-video-player="true"
55                 onClick={handleVideoClick}
56             />
57         </div>
58         <div className={css({ mt: 3, display: 'flex', gap: 3 })}>
59             <button
60                 type="button"
61                 onClick={handleStart}
62                 className={css({ px: 3, py: 2, rounded: 'lg', bg: 'slate.900',
63                     color: 'white', _hover: { bg: 'slate.800' } })}
64             >
65                 Start video
66             </button>
67         </div>
68     </Card>
69 )

```

Listing 8: VideoPlayer.tsx – Hazard clip playback

B.9 Card Component

```

1  import { css } from '../..//styled-system/css'
2
3  type CardProps = {
4      title: string
5      status?: React.ReactNode
6      children: React.ReactNode
7  }
8
9  export default function Card({ title, status, children }: CardProps) {
10     return (
11         <div className={css({
12             h: 'full', display: 'flex', flexDirection: 'column',
13             rounded: '2xl', borderWidth: '1px', p: 4,
14             bg: 'white', boxShadow: 'md'
15         })}>

```

```

16         <div className={css({ display: 'flex', justifyContent: 'space-between',
17             alignItems: 'center', mb: 4 })}>
18             <h2 className={css({ fontWeight: 'medium' })}>{title}</h2>
19             {status}
20         </div>
21         {children}
22     </div>
23 }

```

Listing 9: Card.tsx – Reusable card layout

B.10 Status Badge Component

```

1  import { css } from '../styled-system/css'
2
3  type StatusBadgeProps = {
4      status: 'success' | 'warning' | 'error' | 'info'
5      children: React.ReactNode
6  }
7
8  export default function StatusBadge({ status, children }: StatusBadgeProps) {
9      const colors = {
10         success: { bg: 'green.100', color: 'green.800' },
11         warning: { bg: 'amber.100', color: 'amber.900' },
12         error: { bg: 'red.100', color: 'red.800' },
13         info: { bg: 'blue.100', color: 'blue.800' }
14     }
15
16     const statusColor = colors[status] || colors.info
17
18     return (
19         <span className={css({
20             px: 2, py: 1, rounded: 'md', fontSize: 'xs',
21             bg: statusColor.bg,
22             color: statusColor.color
23         })}>
24             {children}
25         </span>
26     )
27 }

```

Listing 10: StatusBadge.tsx – Status indicator badge

B.11 Data Capture Manager

```

1  export interface TestSession {
2      sessionId: string
3      startTime: number
4      endTime?: number
5      testResults: ClipResult[]
6      metadata: SessionMetadata
7  }
8
9  export interface ClipResult {
10     clipId: number
11     clipName: string
12     startTime: number
13     endTime: number
14     clicks: ClickEvent[]
15     score: number
16     maxScore: number
17 }
18
19 export interface ClickEvent {
20     timestamp: number
21     videoTime: number

```

```

22     score: number
23     hitHazard: boolean
24     hazardId?: string
25 }
26
27 export interface SessionMetadata {
28     userAgent: string
29     screenWidth: number
30     screenHeight: number
31 }
32
33 class DataCaptureManager {
34     private session: TestSession | null = null
35
36     startSession(): TestSession {
37         this.session = {
38             sessionId: crypto.randomUUID(),
39             startTime: Date.now(),
40             testResults: [],
41             metadata: {
42                 userAgent: navigator.userAgent,
43                 screenWidth: window.screen.width,
44                 screenHeight: window.screen.height
45             }
46         }
47         return this.session
48     }
49
50     startClip(clipId: number, clipName: string) {
51         if (!this.session) return
52
53         this.session.testResults.push({
54             clipId,
55             clipName,
56             startTime: Date.now(),
57             endTime: 0,
58             clicks: [],
59             score: 0,
60             maxScore: 5
61         })
62     }
63
64     endClip(score: number) {
65         if (!this.session) return
66
67         const currentClip = this.session.testResults[this.session.testResults.length
68             - 1]
69         if (currentClip) {
70             currentClip.endTime = Date.now()
71             currentClip.score = score
72         }
73     }
74
75     endSession(): TestSession | null {
76         if (!this.session) return null
77         this.session.endTime = Date.now()
78         return this.session
79     }
80
81     getCurrentSession(): TestSession | null {
82         return this.session
83     }
84
85     exportAsJSON(): string | null {
86         if (!this.session) return null
87         return JSON.stringify(this.session, null, 2)
88     }
89
90     exportAsCSV(): string | null {
91         if (!this.session) return null
92
93         const lines = ['timestamp,clipId,videoTime,score,hitHazard,hazardId']

```

```

94     this.session.testResults.forEach(clip => {
95         clip.clicks.forEach(click => {
96             lines.push(`${click.timestamp},${clip.clipId},${click.videoTime},${
97                 click.score},${click.hitHazard},${click.hazardId || ''}`)
98         })
99     })
100     return lines.join('\n')
101 }
102 }
103
104 export const dataCapture = new DataCaptureManager()

```

Listing 11: dataCapture.ts – Session recording and CSV/JSON export

B.12 AoI Marker Tool

A standalone HTML page included in the project for marking Areas of Interest on hazard perception video clips. Used to generate the .aois.json files consumed by the test system.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Hazard Perception - AoI Marker</title>
5      <style>
6          body {
7              font-family: Arial, sans-serif;
8              padding: 20px;
9              background: #1e293b;
10             color: white;
11         }
12         video {
13             max-width: 900px;
14             border: 2px solid #475569;
15             border-radius: 8px;
16         }
17         .controls {
18             margin: 20px 0;
19             display: flex;
20             gap: 10px;
21             flex-wrap: wrap;
22         }
23         button {
24             padding: 10px 15px;
25             background: #3b82f6;
26             color: white;
27             border: none;
28             border-radius: 6px;
29             cursor: pointer;
30         }
31         button:hover { background: #2563eb; }
32         input[type="file"] {
33             padding: 8px;
34             background: #374151;
35             border: 1px solid #6b7280;
36             border-radius: 4px;
37             color: white;
38         }
39         textarea {
40             width: 100%;
41             height: 300px;
42             background: #374151;
43             color: white;
44             border: 1px solid #6b7280;
45             border-radius: 4px;
46             padding: 10px;
47             font-family: monospace;
48         }

```

```

49     .info {
50         background: #1f2937;
51         padding: 15px;
52         border-radius: 8px;
53         margin: 10px 0;
54     }
55     </style>
56 </head>
57 <body>
58 <h1>Hazard Perception - AoI Marker Tool</h1>
59
60 <div class="info">
61     <strong>Instructions:</strong>
62     1. Load a video file - 2. Pause at hazard moments - 3. Click "Mark Hazard" - 4.
        Export JSON
63 </div>
64
65 <video id="video" controls></video>
66
67 <!-- Add this time display -->
68 <div style="margin: 10px 0; padding: 10px; background: #374151; border-radius: 6px;">
69     <strong>Current Time: <span id="timeDisplay">0.00s</span></strong>
70 </div>
71
72 <div class="controls">
73     <input type="file" id="videoFile" accept="video/*">
74     <button onclick="markHazard()">Mark Hazard</button>
75     <button onclick="exportAoI()">Export JSON</button>
76     <button onclick="clearAll()">Clear All</button>
77 </div>
78
79 <div>
80     <label><strong>Current AoI Data:</strong></label>
81     <textarea id="output" placeholder="AoI data will appear here..."></textarea>
82 </div>
83
84 <script>
85     const video = document.getElementById('video')
86     const output = document.getElementById('output')
87     let aois = { videoWidth: 1280, videoHeight: 720, static: [], dynamic: [] }
88     let isSelecting = false
89     let startPos = null
90     let selectionBox = null
91
92     document.getElementById('videoFile').onchange = (e) => {
93         video.src = URL.createObjectURL(e.target.files[0])
94         video.onloadedmetadata = () => {
95             aois.videoWidth = video.videoWidth
96             aois.videoHeight = video.videoHeight
97             updateOutput()
98         }
99     }
100
101     video.addEventListener('mousedown', startSelection)
102     video.addEventListener('mousemove', updateSelection)
103     video.addEventListener('mouseup', endSelection)
104     video.addEventListener('timeupdate', updateTimeDisplay)
105
106     function updateTimeDisplay() {
107         const timeDisplay = document.getElementById('timeDisplay')
108         timeDisplay.textContent = video.currentTime.toFixed(2) + 's'
109     }
110
111     function startSelection(e) {
112         if (!isSelecting) return
113
114         const rect = video.getBoundingClientRect()
115         startPos = {
116             x: e.clientX - rect.left,
117             y: e.clientY - rect.top
118         }
119
120         if (!selectionBox) {

```

```

121     selectionBox = document.createElement('div')
122     selectionBox.style.position = 'absolute'
123     selectionBox.style.border = '2px solid #ff0000'
124     selectionBox.style.backgroundColor = 'rgba(255, 0, 0, 0.2)'
125     selectionBox.style.pointerEvents = 'none'
126     selectionBox.style.zIndex = '1000'
127     document.body.appendChild(selectionBox)
128 }
129
130     selectionBox.style.left = (rect.left + startPos.x) + 'px'
131     selectionBox.style.top = (rect.top + startPos.y) + 'px'
132     selectionBox.style.width = '0px'
133     selectionBox.style.height = '0px'
134     selectionBox.style.display = 'block'
135 }
136
137 function updateSelection(e) {
138     if (!isSelecting || !startPos || !selectionBox) return
139
140     const rect = video.getBoundingClientRect()
141     const currentPos = {
142         x: e.clientX - rect.left,
143         y: e.clientY - rect.top
144     }
145
146     const width = Math.abs(currentPos.x - startPos.x)
147     const height = Math.abs(currentPos.y - startPos.y)
148     const left = Math.min(startPos.x, currentPos.x)
149     const top = Math.min(startPos.y, currentPos.y)
150
151     selectionBox.style.left = (rect.left + left) + 'px'
152     selectionBox.style.top = (rect.top + top) + 'px'
153     selectionBox.style.width = width + 'px'
154     selectionBox.style.height = height + 'px'
155 }
156
157 function endSelection(e) {
158     if (!isSelecting || !startPos || !selectionBox) return
159
160     const rect = video.getBoundingClientRect()
161     const endPos = {
162         x: e.clientX - rect.left,
163         y: e.clientY - rect.top
164     }
165
166     const scaleX = video.videoWidth / video.clientWidth
167     const scaleY = video.videoHeight / video.clientHeight
168
169     const x = Math.min(startPos.x, endPos.x) * scaleX
170     const y = Math.min(startPos.y, endPos.y) * scaleY
171     const w = Math.abs(endPos.x - startPos.x) * scaleX
172     const h = Math.abs(endPos.y - startPos.y) * scaleY
173
174     selectionBox.style.display = 'none'
175
176     const id = prompt('Hazard ID:')
177     if (id) {
178         addHazardWithCoords(id, Math.round(x), Math.round(y), Math.round(w), Math
            .round(h))
179     }
180
181     isSelecting = false
182     startPos = null
183 }
184
185 function markHazard() {
186     alert('Click and drag on the video to select hazard area')
187     isSelecting = true
188 }
189
190 function addHazardWithCoords(id, x, y, w, h) {
191     const t = parseFloat(video.currentTime.toFixed(2))
192

```

```

193     let hazard = aois.dynamic.find(h => h.id === id)
194     if (!hazard) {
195         hazard = { id, frames: [] }
196         aois.dynamic.push(hazard)
197     }
198
199     hazard.frames.push({ t, x, y, w, h })
200     hazard.frames.sort((a, b) => a.t - b.t)
201
202     updateOutput()
203     alert('Added hazard "${id}" at ${t}s\nPosition: (${x}, ${y})\nSize: ${w}x${h}
204         px')
205 }
206
207 function updateOutput() {
208     output.value = JSON.stringify(aois, null, 2)
209 }
210
211 function exportAoI() {
212     const filename = prompt('Filename (e.g., "1.aois.json"):') || 'hazards.aois.
213         json'
214
215     const blob = new Blob([JSON.stringify(aois, null, 2)], { type: 'application/
216         json' })
217     const a = document.createElement('a')
218     a.href = URL.createObjectURL(blob)
219     a.download = filename
220     a.click()
221
222     alert('Exported ${filename}')
223 }
224
225 function clearAll() {
226     if (confirm('Clear all hazard data?')) {
227         aois = {
228             videoWidth: aois.videoWidth,
229             videoHeight: aois.videoHeight,
230             static: [],
231             dynamic: []
232         }
233         updateOutput()
234     }
235 }
236
237 updateOutput()
238 </script>
239 </body>
240 </html>

```

Listing 12: hazard-marker.html – AoI marking tool